

# Apostila de Estruturas Discretas Parte I

Eduardo Sany Laber

21 de setembro de 2016

# Sumário

<b>1</b>	<b>Técnicas de Demonstração</b>	<b>3</b>
1.1	O que é uma demonstração . . . . .	3
1.2	Conceitos Básicos . . . . .	4
1.2.1	Teorema . . . . .	4
1.2.2	Lema e Corolário . . . . .	4
1.2.3	Axiomas e Definições . . . . .	7
1.2.4	Quantificadores . . . . .	8
1.3	Tipos de Demonstração . . . . .	9
1.3.1	Exemplos e Contra-Exemplos . . . . .	9
1.3.2	Demonstração por Força Bruta . . . . .	11
1.3.3	Prova Direta . . . . .	13
1.3.4	Prova Construtiva . . . . .	14
1.3.5	Prova por Contradição . . . . .	16
1.4	Indução Matemática . . . . .	18
1.4.1	Definição . . . . .	18
1.4.2	Exemplos . . . . .	18
1.4.3	Indução Forte X Indução Fraca . . . . .	26
1.4.4	Corretude de Algoritmos . . . . .	32
1.4.5	Indução para Projeto de Algoritmo . . . . .	39
1.4.6	Erros Comuns . . . . .	46
<b>2</b>	<b>Princípios de Contagem e Enumeração Computacional</b>	<b>49</b>
2.1	Princípio da Multiplicação . . . . .	51
2.1.1	Gerando todas as palavras de uma alfabeto . . . . .	52
2.2	Permutações . . . . .	53
2.2.1	Gerando Permutações . . . . .	55
2.3	Combinações . . . . .	55
2.3.1	Gerando Subconjuntos . . . . .	59
2.4	Permutações com Repetições . . . . .	59
2.4.1	Enumeração . . . . .	61
2.5	Combinações com Repetições . . . . .	61

2.5.1	O Problema do Troco . . . . .	66
2.6	Princípio da Inclusão e Exclusão . . . . .	68
2.7	Princípio da Casa dos Pombos . . . . .	74
2.7.1	Princípio Generalizado . . . . .	75
<b>3</b>	<b>Teoria de Grafos</b>	<b>76</b>
3.1	Conceitos Básicos . . . . .	76
3.1.1	Operações em Grafos . . . . .	79
3.1.2	Isomorfismo de Grafos . . . . .	81
3.1.3	Subgrafos . . . . .	83
3.1.4	Passeios em Grafos . . . . .	84
3.1.5	Conexidade . . . . .	85
3.2	Alguns Grafos Importantes . . . . .	86
3.2.1	Caracterização de Grafos Bipartidos por Ciclos de Comprimento Ímpar . . . . .	88
3.3	Árvores . . . . .	89
3.3.1	Árvores Geradoras de Custo Mínimo . . . . .	91
3.4	Trajeto Euleriano . . . . .	94
3.5	Grafos Direcionados . . . . .	101
3.5.1	Passeios em Grafos Direcionados . . . . .	103
3.5.2	Grafos Direcionados Acíclicos . . . . .	104
3.6	Problema do Caminho mais curto . . . . .	107
3.7	Representações Computacionais . . . . .	111
3.7.1	Matriz de Adjacência . . . . .	111
3.7.2	Lista de Adjacência . . . . .	112
3.8	Grafos Hamiltonianos . . . . .	114
3.9	Emparelhamentos . . . . .	115
3.10	Coloração de Vértices . . . . .	116
3.11	Coloração de Arestas . . . . .	118

# Capítulo 1

## Técnicas de Demonstração

### 1.1 O que é uma demonstração

Demonstração é a maneira pela qual uma proposição é validada através de argumentos formais.

Na ciência da computação uma demonstração pode ser utilizada para se ter certeza de que um determinado algoritmo funciona de acordo com sua especificação.

Como exemplo de algoritmos que foram validados utilizando-se argumentos formais, podemos citar:

- Algoritmos de ordenação
- Algoritmos de busca em bases de dados
- Algoritmos de criptografia
- Algoritmos de compressão de dados

Nosso objetivo nesse capítulo é apresentar algumas técnicas de demonstração. Seguem agora alguns conceitos básicos envolvendo a noção de demonstração.

## 1.2 Conceitos Básicos

### 1.2.1 Teorema

O objeto básico de uma demonstração é a proposição que desejamos demonstrar. Essa proposição recebe o nome de *teorema*.

Um teorema é dividido em duas partes:

**1.Hipótese:** É onde se encontram as informações conhecidas sobre o teorema. Essas informações são tomadas como verdadeiras a fim de se tentar obter uma demonstração.

**2.Tese:** É a parte do teorema que desejamos validar. A partir da hipótese, utilizando-se uma sequência de argumentos formais, buscamos uma demonstração para a tese.

Dessa forma o teorema pode ser escrito como a seguinte implicação lógica:

$$\text{Hipótese} \Rightarrow \text{Tese}$$

### 1.2.2 Lema e Corolário

Em algumas circunstâncias os teoremas recebem nomes especiais. Quando um teorema é utilizado como parte da demonstração de um outro teorema, em geral mais complexo, ele recebe o nome de *lema*. Quando um teorema é uma consequência imediata de outro, ele recebe o nome de *corolário*. Por exemplo:

**Teorema 1** *A soma dos ângulos internos de um triângulo é 180 graus.*

**Corolário 1** *Cada ângulo de um triângulo equilátero vale 60 graus.*

Em geral a prova do corolário é bastante simples e por isso omitida. No caso acima basta saber que um triângulo equilátero possui todos os ângulos iguais.

Uma proposição pode ser falsa ou verdadeira. Caso seja encontrada uma prova, ela será verdadeira e se tornará um teorema. Caso seja encontrado um contra-exemplo, ela será considerada uma proposição falsa. Como exemplos de proposições, verdadeiras (teoremas) e falsas, podemos citar:

**Proposição 1** *Todo número par maior que 2 é soma de 2 primos.*

**Proposição 2** *Para  $n \geq 1$ ,  $n^2 + n + 5$  é um número primo.*

**Proposição 3** *Todo Malaio tem sangue A+.*

**Proposição 4**  *$2^{2^n} - 1$  é múltiplo de 3, para todo número natural  $n$ .*

**Proposição 5** *Todo número múltiplo de 6 é múltiplo de 3.*

**Proposição 6** *O produto de dois números pares é um número par.*

**Proposição 7** *Se  $n^2$  é ímpar, então  $n$  também é ímpar.*

**Proposição 8** *Todo inteiro menor que 10 é maior que 5.*

**Proposição 9**  *$\sqrt{2}$  não é um número racional.*

**Proposição 10** *Existem infinitas triplas de naturais  $(x, y, z)$  tais que  $x^2 + y^2 = z^2$ .*

**Proposição 11** *A série  $\sum_{i=1}^n 1/i$  diverge.*

**Proposição 12** *Não existe nenhuma quadra de inteiros  $(x, y, z, n)$  com  $x > 0$ ,  $y > 0$ ,  $z > 0$  e  $n \geq 3$  tal que*

$$x^n + y^n = z^n.$$

**Proposição 13** *O algoritmo Heapsort realiza no máximo  $5n \log n$  comparações para ordenar uma lista de  $n$  números.*

**Proposição 14** *Existem infinitos números primos.*

**Proposição 15** *A soma de três inteiros consecutivos é múltiplo de 3.*

Convidamos o leitor a tentar determinar a veracidade ou falsidade das proposições acima.

Se testarmos a validade da Proposição 1 para os números 4, 6 e 8, verificamos sua correção:

$$4 = 2 + 2, \text{ OK}$$

$$6 = 3 + 3, \text{ OK}$$

$$8 = 3 + 5, \text{ OK}$$

Infelizmente é impossível testar a proposição para os infinitos números pares. Logo, somente exibindo uma prova formal poderíamos chamá-la de teorema. Basta, no entanto, encontrar apenas um contra-exemplo, para ela ser considerada falsa.

Na Proposição 2 temos:

$$n = 1 \Rightarrow n^2 + n + 5 = 7 \text{ é primo. OK}$$

$$n = 2 \Rightarrow n^2 + n + 5 = 11 \text{ é primo. OK}$$

$$n = 3 \Rightarrow n^2 + n + 5 = 17 \text{ é primo. OK}$$

Mas se testarmos para  $n=4$ , temos:

$$n = 4 \Rightarrow n^2 + n + 5 = 25, \text{ contra-exemplo !!!}$$

Logo concluímos que a Proposição 2 é falsa.

### 1.2.3 Axiomas e Definições

Existem algumas proposições que não podem ser provadas. Elas podem ser *Axiomas* ou *Definições*.

*Definição* é uma enumeração das propriedades que um determinado objeto (matemático ou não) deve obrigatoriamente ter (ou não ter) para pertencer a uma determinada classe de objetos. Seguem dois exemplos

**Definição 1** *Um número primo é um número inteiro maior ou igual a 2 que só possui como divisores, a unidade e ele próprio.*

**Definição 2** *Um triângulo retângulo é aquele que possui um ângulo de 90 graus.*

*Axioma* é uma afirmação básica aceita por todos acerca de algo. Axiomas são normalmente informações óbvias, baseadas no senso comum. Como exemplo,

**Axioma 1** *Por dois pontos distintos no plano existe uma única reta.*

Um ponto importante aqui deve ser mencionado. Deve-se tomar extremo cuidado para não se considerar um axioma como uma verdade única e imutável. O próprio exemplo acima, embora aparentemente óbvio, pode ser tomado como falso sem que isso resulte em uma contradição. Na verdade sua negação serve de base para a construção de um tipo de geometria não-euclidiana com diversas aplicações práticas.



## 1.2.4 Quantificadores

Como último conceito básico vamos tratar dos quantificadores. Uma proposição qualquer (correta ou incorreta), pode tratar de um objeto específico. Por exemplo:

**Proposição 16** *O número 5 é primo.*

Não há problema formular proposições desse tipo (mesmo estando seu conteúdo incorreto). Entretanto, a sua utilidade é limitada. No caso acima, para demonstrá-la basta verificar se 5 é primo, o que é imediato.

Por isso é que normalmente encontramos os quantificadores **para todo** ( $\forall$ ) e **existe**  $\exists$  nos enunciados das proposições que gostaríamos de provar. O quantificador **para todo** ( $\forall$ ) indica que a proposição está se referindo a todos os elementos de um determinado conjunto, por exemplo:

**Proposição 17**  *$\forall n$  inteiro e múltiplo de 4,  $n$  também é múltiplo de 2.*

Significa que no conjunto dos inteiros, qualquer número (sem exceção) que seja múltiplo de 4 deve ser também múltiplo de 2. A negação do enunciado seria:

**Proposição 18**  *$\exists n$  inteiro com as seguintes propriedades*

- (i)  *$n$  é múltiplo de 4,*
- (ii)  *$n$  **não** é múltiplo de 2.*

É muito comum encontrar negações envolvendo os quantificadores em demonstrações por contradição, como veremos mais adiante. Como regra, um quantificador **para todo** negado, transforma-se em um quantificador **existe**.

Passamos agora a estudar algumas técnicas de demonstração. É importante conhecer os diversos tipos de técnicas para se aplicar aquela mais adequada ao teorema em questão. Muitas vezes a escolha de uma técnica determina a facilidade, ou até mesmo a viabilidade, de se demonstrar um teorema.

## 1.3 Tipos de Demonstração

A base para uma demonstração correta é que ela seja construída com proposições verdadeiras sob as mesmas condições da hipótese do teorema. Essas proposições poderão ser axiomas, definições, a própria hipótese e outros teoremas já devidamente provados e válidos sob as condições da hipótese.

Portanto, é importante observar que se a hipótese trata de um objeto genérico, a prova também deverá tratar. Por exemplo, se afirmamos que  $\forall n \in \mathbb{Z}$ ,  $2n^3 - 3n^2 + n$  é divisível por 6, não basta formularmos uma demonstração que utilize como condição que  $n$  seja par. Estaríamos particularizando o enunciado original do teorema. Devemos nesse caso exibir na demonstração, proposições que **não** dependam da condição de  $n$  ser um número par.

### 1.3.1 Exemplos e Contra-Exemplos

Quando uma proposição trata de um conjunto finito de elementos, ou quando ela afirma a existência de um número finito de valores que satisfaçam certas condições, é possível prová-la simplesmente mostrando sua validade para todos os elementos sob os quais ela é afirmada, ou exibindo o número pedido de exemplos que satisfaçam a afirmação. A facilidade de tal forma de demonstração, depende do tamanho do conjunto de elementos em questão. Por exemplo:

**Proposição 19**  $\forall n \in \{4, 6, 8\}$ ,  $n$  pode ser expresso como a soma de números primos.

Esse enunciado corresponde a uma particularização da Proposição 1 vista anteriormente. Da forma que ela está ela é um teorema, pois de fato o enunciado vale para esses três valores, como vimos anteriormente.

**Proposição 20**  $\exists n$  inteiro tal que  $n^2 + n + 5$  é um número primo.

*Prova:* Basta escolher  $n = 1$  ou  $n = 2$  ou  $n = 3$  para provar o teorema.

Como aplicação da utilização de um contra-exemplo para mostrar que uma proposição é falsa, podemos citar:

**Proposição 21** *Nenhum número primo é par.*

A proposição está incorreta pois 2 é par e é primo.

**Proposição 22** *Todo inteiro menor que 10, é maior que 5.*

Basta tomar  $n = 4$  como contra-exemplo, já que  $4 < 10$  e  $4 < 5$ .

### 1.3.2 Demonstração por Força Bruta

Os exemplos anteriores deixam implícito o fato de que para provar teoremas utilizando exemplos ou contra-exemplos, é necessário encontrá-los, o que muitas vezes pode ser uma árdua tarefa.

A técnica de provar com exemplos se torna extremamente difícil e muitas vezes inviável na prática. A resolução de vários problemas de interesse para a computação envolvem essa técnica de exemplos também conhecida como *enumeração completa, busca exaustiva ou força bruta*. Considere a tarefa de testar a primalidade de um número inteiro de 30 dígitos !!! Esta requer uma intensa busca exaustiva. Problemas dessa natureza, tem impulsionado os cientistas de computação a buscar mecanismos cada vez mais eficientes que forneçam uma solução dentro de um limite de tempo razoável. Muitas vezes, embora seja necessário teoricamente analisar todas as possibilidades, dependendo do tipo de problema podemos encontrar atalhos que diminuam o número de casos que devemos testar. Para exemplificar vamos aplicar essa técnica para demonstrar a seguinte proposição:

**Proposição 23** *Todo número par maior ou igual a 4 e menor que 20, pode ser escrito como a soma de dois primos.*

**Prova:** Basta resolver para todos os casos:

$$\begin{array}{lll} 4=2+2 & 10=7+3 & 16=11+5 \\ 6=3+3 & 12=7+5 & 18=13+5 \\ 8=5+3 & 14=7+7 & \end{array}$$

Se no entanto modificarmos a proposição anterior para:

**Proposição 24** *Todo número par maior ou igual a 4 e menor que 100.000 pode ser escrito como a soma de 2 primos.*

**Prova:** Utilizando-se um computador para gerar todas as possibilidades, procede-se de forma análoga à anterior.

Se, no entanto, quisermos provar a proposição geral:

**Proposição 25** *Todo número par maior ou igual a 4 pode ser escrito como a soma de 2 primos.*

Não conseguiríamos usar prova exaustiva pois o conjunto é infinito. No entanto o computador pode ser útil para buscar um contra-exemplo que invalide a proposição(até hoje não encontrado). Esse problema está em aberto e é conhecido como *Conjectura de Goldbach*.

### 1.3.3 Prova Direta

Prova direta é o tipo mais comum de prova encontrada para se demonstrar teoremas. Ela consiste em um encadeamento de argumentos lógicos a partir da hipótese, que podem ser: axiomas, definições ou outros teoremas (já provados), que resulta em uma implicação lógica da tese. Como exemplo temos a Proposição 15 :

**Teorema 2** *A soma de 3 números consecutivos é múltiplo de 3.*

**Prova** Sejam  $x, x + 1, x + 2$  os números em questão. Temos que a soma dos números é  $S = x + (x + 1) + (x + 2)$ . Logo,

$$\begin{aligned} S &= x + (x + 1) + (x + 2) \Rightarrow \\ S &= 3x + 3 \Rightarrow \\ S &= 3(x + 1) \Rightarrow \\ S &= 3m; m \in Z \Rightarrow S \text{ é múltiplo de } 3 \end{aligned}$$

Foram utilizados nessa prova argumentos algébricos básicos, como fatoração, por exemplo.

Podemos provar demonstrar as Proposição 4 e 5 utilizando prova direta:

**Teorema 3**  *$2^{2n} - 1$  é múltiplo de 3, para todo número natural  $n$ .*

**Teorema 4** *Todo múltiplo de 6 é múltiplo de 3.*

**Prova:** Seja  $n$  múltiplo de 6. Isso equivale a:

$$\begin{aligned} n &= 6k, \text{ para algum } k \\ n &= 3 \cdot (2k) \text{ fatoração} \\ n &= 3m, \text{ para algum inteiro } m \\ n &\text{ é múltiplo de } 3. \end{aligned}$$

Para concluir esta seção enunciamos um teorema conhecido que faz parte da teoria de Ramsey.

**Teorema 5** *Considere um hexágono regular cujos vértices são  $v_1, v_2, \dots, v_6$ . Mostre que toda maneira de colorir os segmentos de retas que unem dois vértices, utilizando as cores azul ou branca, produz **pelo menos** um triângulo cujos lados tem a mesma cor.*

**Prova:** Como existem cinco segmentos de reta partindo (chegando) em  $v_1$  então podemos concluir que **pelo menos** três deles tem a mesma cor. Sem perda de generalidade assumimos que esta cor é azul.

Sejam então  $A, B, C$  vértices que estão ligados a  $v_1$  com a cor azul.

caso 1:) **Pelo menos** um dos segmentos de reta  $\overline{AB}, \overline{AC}$  ou  $\overline{BC}$  é azul. Logo, formaremos um triângulo de cor azul. Por exemplo, se  $\overline{AB}$  for azul então o triângulo com vértices  $v_1, A, B$  é azul.

caso 2:)  $\overline{AB}, \overline{AC}$  e  $\overline{BC}$  são brancos. Neste caso, o triângulo com vértices  $A, B, C$  é branco. ■

É interessante notar que poderíamos provar este teorema utilizando força bruta. Para isso, "bastaria" testar se todas as  $2^{15}$  maneiras de colorir os segmentos determinados pelos vértices do hexágono produz um triângulo cujos lados tem a mesma cor.

### 1.3.4 Prova Construtiva

O objetivo da prova construtiva é apresentar um método pelo qual é possível construir o objeto do qual o teorema trata. É similar ao método de exemplos, com a diferença de que sua aplicação é mais geral. Nesse tipo de prova a definição do método permite construir um conjunto, possivelmente infinito, de exemplos, como no caso da Proposição 10.

**Teorema 6** *Existem infinitas triplas  $(x, y, z)$  de números inteiros tais que  $x^2 + y^2 = z^2$*

**Prova:** Podemos verificar facilmente que  $(3, 4, 5)$  satisfaz o teorema, já que  $3^2 + 4^2 = 5^2$ . Tentaremos agora verificar se as triplas da forma  $(3k, 4k, 5k)$ , com  $k \in \mathbb{Z}$ , também satisfazem. Temos que:

$$(3k)^2 + (4k)^2 = 4^2k^2 + 3^2k^2 = (3^2 + 4^2)k^2 = 5^2k^2 = (5k)^2$$

Logo as triplas da forma  $(3k, 4k, 5k)$  satisfazem o teorema. Como  $k \in \mathbb{Z}$  e existem infinitos números inteiros, provamos que o conjunto construído pela lei de formação acima é infinito. ■

Outro exemplo em que a técnica de prova por construção é adequada é a Proposição 11.

**Teorema 7** *A série harmônica*

$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

*é divergente.*

**Prova:** Dado um  $M$  genérico, devemos apresentar uma forma de construir  $n$ , como função de  $M$ , tal que:

$$\sum_{i=1}^n \frac{1}{i} > M$$

Dessa maneira, estaremos mostrando que a série é maior que qualquer valor apresentado.

Seja  $n = 2^{2M}$ , temos que:

$$\sum_{i=1}^n \frac{1}{i} = \sum_{i=1}^2 \frac{1}{i} + \sum_{i=3}^4 \frac{1}{i} + \sum_{i=5}^8 \frac{1}{i} + \dots + \sum_{i=n/2+1}^n \frac{1}{i} > \underbrace{\frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \dots + \frac{1}{2}}_{2M \text{ vezes}} = M$$

Na expressão acima utilizamos o fato de que:

$$\sum_{i=2^{k+1}}^{2^{k+1}} \frac{1}{i} > 2^k \left( \frac{1}{2^{k+1}} \right) = \frac{1}{2}$$

■



### 1.3.5 Prova por Contradição

A prova por contradição baseia-se na idéia de que provar um teorema verdadeiro é equivalente a provar que a sua negação é falsa. A importância desse método reside no fato de que para alguns tipos de teoremas é muito difícil provar diretamente, ou por construção. Nesses casos a demonstração por contradição pode ser a técnica mais indicada. Podemos provar a Proposição 14 utilizando esta técnica.

**Teorema 8** *Existe uma quantidade infinita de números primos*

**Prova** Vamos supor que o teorema seja falso, ou seja, que *existe uma quantidade finita de números primos*. Definindo  $P$  como o conjunto de todos os números primos, temos que:

$$P = \{p_1, p_2, \dots, p_n\}, \text{ onde } p_i \text{ indica o } i\text{-ésimo menor número primo.}$$

Se exibirmos um número primo que não esteja nesse conjunto, provaremos então que  $P$  não contém todos os números primos, o que contradiz a hipótese, implicando que o conjunto dos números primos não pode ser finito. Esse número pode ser obtido da seguintes forma:

$$q = p_1 \cdot p_2 \cdot \dots \cdot p_n + 1$$

Conseguimos dessa maneira um número que não é divisível por nenhum número primo do conjunto  $P$ , o que implica que ele não pode ser um número composto, já que todo número composto é produto de primos. Logo,  $q$  é primo e não está no conjunto  $P$ , uma vez que  $q$  é maior que  $p_n$ , o maior elemento de  $P$ . Chegamos então à contradição desejada o que prova o nosso teorema.

Segue outro exemplo.

**Teorema 9**  $\sqrt{2}$  não é um número racional, ou seja, não pode ser escrito como  $p/q$ , com  $p$  e  $q$  inteiros,  $q \neq 0$  e  $p, q$  primos entre si.

**Prova:** Assuma por contradição que  $\sqrt{2} = p/q$ , onde  $q \neq 0$  e  $p$  e  $q$  não apresentam fatores comuns. Como  $\frac{p^2}{q^2} = 2$ , concluímos que  $p$  é par. Portanto 4 é divisor de  $p^2$ . Logo  $2q^2 = 4k$ , para algum  $k$ . Segue então que  $q$  é múltiplo de 2, o que contradiz o fato de  $p$  e  $q$  não apresentarem fatores comuns. ■

## 1.4 Indução Matemática

### 1.4.1 Definição

O princípio da *Indução Matemática* é útil quando precisamos demonstrar uma proposição envolvendo números inteiros, a qual suspeitamos ser válida a partir de um número inteiro inicial  $n_0$ .

A indução matemática formaliza essa idéia da seguinte forma:

*Uma propriedade  $P$  qualquer é válida para  $\forall n \geq n_0, n, n_0 \in Z$ , se for possível provar que:*

1.  $P(n_0)$  é válida
2.  $\forall k \in Z$  tal que  $k \geq n_0, P(k) \Rightarrow P(k + 1)$

A primeira condição a ser satisfeita é chamada frequentemente de *Base* e a segunda é conhecida como *Passo Indutivo*.

Segue-se agora uma série de exemplos de aplicação da indução matemática para provar propriedades.

### 1.4.2 Exemplos

**Exemplo 1** *Suponha que o Sr Silva casou e teve 2 filhos. Suponha que cada filho do Sr Silva casou e tenha tido dois filhos. Assuma que este processo continue. Quantos membros terá a  $n$ -ésima geração considerando que a geração do Sr Silva é a geração 0.*

**Solução:**

**Hipótese:** o número de membros  $T(n)$  da geração  $n$  é  $2^n$

$$P(n) = \begin{cases} V, & \text{se } T(n) = 2^n \\ F, & \text{caso contrário} \end{cases}$$

**Base:**

$P(1)$  é verdadeiro já que o número de membros da geração 1 é  $2 = 2^1$

**Passo Indutivo:**  $P(k) \rightarrow P(k + 1), \forall k \geq 1$

**Hipótese Indutiva:**  $T(k) = 2^k \quad \forall k \geq 1$

**Prova do Passo.** Cada membro da geração  $k$  tem 2 filhos. Logo,

$$T(k + 1) = 2T(k) = 2 \cdot 2^k = 2^{k+1} \blacksquare$$

**Exemplo 2** *Provar utilizando indução que*

$$\sum_{i=1}^n (2i - 1) = n^2,$$

para todo  $n \geq 1$

**Solução:**

**Base:**

$P(1)$  é verdadeiro já que  $\sum_{i=1}^1 (2i - 1) = 1$

**Passo Indutivo.** Se

$$\sum_{i=1}^k (2i - 1) = k^2,$$

então

$$\sum_{i=1}^{k+1} (2i - 1) = (k + 1)^2 \quad \forall k \geq 1$$

**Hipótese Indutiva.**

$$\sum_{i=1}^k (2i - 1) = k^2, \quad \forall k \geq 1$$

**Prova do Passo:**

$$\sum_{i=1}^{k+1} (2i - 1) = \sum_{i=1}^k (2i - 1) + 2(k + 1) - 1$$

Como a hipótese indutiva garante que  $\sum_{i=1}^k (2i - 1) = k^2$ , temos que

$$\sum_{i=1}^{k+1} (2i - 1) = k^2 + 2(k + 1) - 1 = k^2 + 2k + 1 = (k + 1)^2 \blacksquare$$

**Exemplo 3** Prove por indução que  $\forall n \geq 0$ ,  $2^{2^n} - 1$  é múltiplo de 3.

**Solução:**

**Base:**

Para  $n = 0$ ,  $2^0 - 1 = 0$  que é múltiplo de 3.

**Passo Indutivo.** Se  $3|(2^{2^k} - 1)$ , então  $3|(2^{2^{k+1}} - 1) \forall k \geq 0$

**Hipótese Indutiva.**  $3|(2^{2^k} - 1) \forall k \geq 0$

**Prova do Passo:**

Devemos manipular algebricamente a expressão de modo a aparecer a hipótese indutiva.

$$2^{2^{k+1}} - 1 = 2^{2^k+2} - 1 = 4 \cdot 2^{2^k} - 1 = 3 \cdot 2^{2^k} + 2^{2^k} - 1$$

Como  $3|(3 \cdot 2^{2^k})$  e como a hipótese indutiva garante que  $3|(2^{2^k} - 1)$ , temos que

$$3|(2^{2^{k+1}} - 1) \blacksquare$$

**Exemplo 4** Prove por indução que  $3^n < n!$ , para  $n \geq 7$

**Solução:**

**Base:** Para  $n = 7$ ,  $3^7 = 2187 < 7! = 5040$ . OK!

**Passo Indutivo:** Se  $3^k < k!$ , então  $3^{k+1} < (k+1)! \forall k \geq 7$ .

**Hipótese Indutiva.**  $k \geq 7$ ,  $3^k < k!$ .

**Prova do Passo.**

$$3^{k+1} = 3 \cdot 3^k < 3 \cdot k! < (k+1)k! = (k+1)!$$

onde a desigualdade  $3 \cdot 3^k < 3 \cdot k!$  segue da hipótese indutiva e a desigualdade  $3 \cdot k! < (k+1)k!$  segue do fato que  $k \geq 7$ .  $\blacksquare$

**Exemplo 5** Prove que  $n^2 > 3n$  para  $n \geq 4$ .

**Prova:** Verificamos inicialmente que a proposição realmente não vale para  $n < 4$ :

$$n = 1 \Rightarrow 1 > 3 \text{ (falso)}$$

$$n = 2 \Rightarrow 4 > 6 \text{ (falso)}$$

$$n = 3 \Rightarrow 9 > 9 \text{ (falso)}$$

Agora passamos a prova por indução:

**Base:**

$$n = 4 \Rightarrow 16 > 12 \text{ OK !}$$

**Passo Indutivo:**

$$k^2 > 3k \Rightarrow (k + 1)^2 > 3(k + 1) \quad \forall k \geq 4$$

**Hipótese Indutiva:**

$$k^2 > 3k, \forall k \geq 4$$

**Prova do Passo:**

$$(k + 1)^2 = k^2 + 2k + 1 > 3k + 2k + 1, \text{ já que } k^2 > 3k \text{ pela hipótese indutiva.}$$

$$\underbrace{3k + 2k + 1 \geq 3k + 8 + 1}_{2k \geq 8, \text{ para } k \geq 4}, \text{ logo :}$$

$$(k + 1)^2 > 3k + 9 > 3k + 3 = 3(k + 1) \quad \blacksquare$$

**Exemplo 6** Prove que  $2^{n+1} < 3^n, \forall n > 1$

**Prova:**

**Base:**

$$n = 2 \Rightarrow 2^3 < 3^2 \text{ OK !}$$

**Passo Indutivo:**

$$2^{k+1} < 3^k \Rightarrow 2^{k+2} < 3^{k+1} \quad \forall k > 1$$

**Hipótese Indutiva:**

$$2^{k+1} < 3^k, \forall k > 1$$

**Prova do Passo:**

$$2^{k+2} = 2 \cdot 2^{k+1} < \underbrace{2 \cdot 3^k}_{\text{hip indutiva}} < 3 \cdot 3^k = 3^{k+1} \blacksquare$$

**Exemplo 7** Considere o seguinte trecho de código.

```
Prog1(k:inteiro)
  Se k=1
    Print('OI')
  Return
Senão
  Para i = 1, ..., k
    Print('OI')
  Fim Para
Prog1(k-1)
Fim Se
```

Seja  $T(n)$  o número de vezes que a palavra OI é impressa quando Prog1 é chamado com parâmetro  $n$ . Calcule  $T(1)$  e determine uma relação entre  $T(n)$  e  $T(n-1)$  para  $n > 1$ . Com base na relação encontrada, utilize indução para mostrar que  $T(n) \leq n^2$ , para todo  $n$  maior ou igual a 1.

**Solução:** Quando  $n = 1$ , Prog1 imprime OI apenas uma vez. Logo,  $T(1) = 1$ . Quando  $n$  é maior que 1, Prog1 imprime OI  $n$  vezes e depois chama recursivamente Prog1( $n-1$ ), que por definição imprime OI  $T(n-1)$  vezes. Portanto,

$$T(n) = n + T(n-1),$$

quando  $n > 1$ . Provaremos então que  $T(n) \leq n^2$ .

**Base:**  $n = 1$ .  $T(1) = 1 \leq 1^2$ . Ok !

**Hipótese Indutiva:**  $T(k) \leq k^2$ , para  $k > 1$

**Passo Indutivo:** Provar para  $k + 1$ .

**Prova do Passo:** A relação encontrada garante que  $T(k+1) = k+1 + T(k)$ . Como a hipótese garante que  $T(k) \leq k^2$ , concluímos que

$$T(k+1) \leq k+1 + k^2 < (k+1)^2$$



**Exemplo 8** Prove que a soma dos ângulos de um polígono convexo de  $n$  vértices é  $180(n - 2)$  para todo  $n \geq 3$ .

**Prova:**

**Base:**  $n = 3$ , já que  $n = 1$  ou  $n = 2$  não fazem sentido.

Nesse caso a figura é um triângulo e supomos já conhecido o teorema que demonstra que a soma dos ângulos internos de um triângulo é 180 graus.

**Hipótese Indutiva:**

A soma dos ângulos internos de um polígono com  $k$  lados é  $180.(k - 2)$

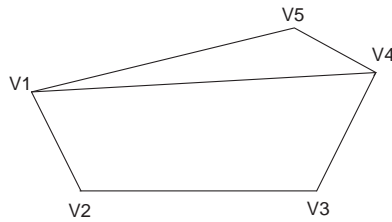
**Passo Indutivo**

Provar a propriedade para  $k + 1$ , ou seja, a soma vale  $180(k - 1)$  graus.

**Prova do Passo:**

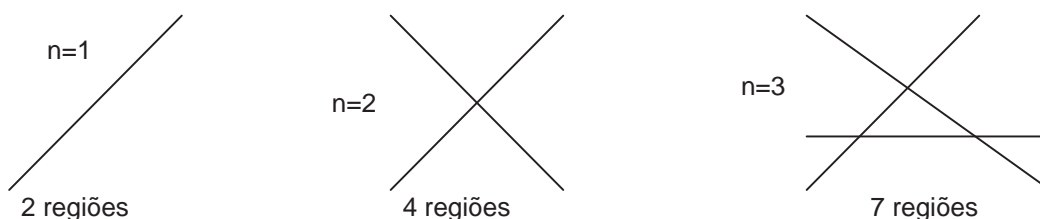
Considere um polígono de  $k + 1$  vértices  $V_1, V_2, \dots, V_{k+1}$ . Podemos traçar um segmento de reta unido os vértices  $V_1$  a  $V_k$ . Desta forma, obtemos um triângulo  $V_1 V_k V_{k+1}$  e um polígono de  $k$  vértices. A figura abaixo ilustra o caso em que  $k + 1 = 5$ .

Sabemos pela hipótese de indução que a soma dos ângulos de um polígono de  $k$  vértices é  $180(k - 2)$  graus. Logo a soma dos ângulos de um polígono com  $k + 1$  vértices é  $180(k - 2) + 180(\text{triângulo}) = 180(k - 1)$  ■.



**Exemplo 9** *Retas no plano: Um conjunto de  $n$  retas no plano estão em posição geral se e somente se não existem duas retas paralelas nem três retas se interceptando no mesmo ponto. Mostre que  $n$  retas em posição geral dividem o plano em  $\frac{n(n+1)}{2} + 1$  regiões*

**Solução:** Seja  $R_n$ , o número de regiões que  $n$  retas em posição geral dividem o plano. Devemos provar que  $R_n = \frac{n(n+1)}{2} + 1$ .



Os exemplos acima sugerem que a fórmula está correta. Tentemos prová-la por indução.

**Base.** Para  $n = 1$  a fórmula é válida.

**Passo Indutivo.** Se  $R_k = \frac{k(k+1)}{2} + 1$  então  $R_{k+1} = \frac{(k+1)(k+2)}{2} + 1$ , para  $k \geq 1$ .

**Prova do Passo:** Suponha um conjunto de  $k$  retas em posição geral. A reta  $k+1$  interceptará cada uma das  $k$  retas em um ponto distinto. Portanto  $k+1$  segmentos de retas são criados. Cada um dos segmentos divide a região que o contém em duas novas regiões. Portanto, temos:

$$R_{k+1} = R_k + k + 1$$

Como  $R_k = \frac{k(k+1)}{2} + 1$  (hipótese indutiva), segue que:

$$R_{k+1} = \frac{k(k+1)}{2} + 1 + k + 1 = \frac{(k+1)(k+2)}{2} + 1 \blacksquare$$

### 1.4.3 Indução Forte X Indução Fraca

Na definição vista anteriormente, nós utilizávamos na hipótese de indução o fato da propriedade  $P$  ser válida para um inteiro  $k$  qualquer e tentávamos mostrar que uma vez isso aceito,  $P$  também seria válida para  $k + 1$ . Essa definição na realidade se refere a **indução fraca**.

Pode ser o caso em que precisemos supor como hipótese de indução que a propriedade  $P$  seja válida não apenas para um inteiro  $k$  qualquer, mas sim, que seja válida para **todos** os inteiros entre 1 e  $k$  (inclusive). Dessa maneira obtem-se uma hipótese "mais forte" do que a anterior. A esse tipo de indução dá-se o nome de **indução forte**.

Formalmente temos:

#### Indução Fraca

*Uma propriedade  $P$  qualquer é válida para  $\forall n \geq n_0, n, n_0 \in Z$ , se for possível provar que:*

1.  $P(n_0)$  é válida
2.  $\forall k \in Z, k \geq n_0, P(k) \Rightarrow P(k + 1)$

#### Indução Forte

*Uma propriedade  $P$  qualquer é válida para  $\forall n \geq n_0, n, n_0 \in Z$ , se for possível provar que:*

1.  $P(n_0)$  é válida
2.  $\forall k \in Z, [P(r)$  válida para todo  $r \in \{n_0, n_0 + 1, \dots, k\}] \Rightarrow P(k + 1)$

**Exemplo 10** *Prove que para todo  $n \geq 2$ ,  $n$  é um número primo ou um produto de números primos.*

**Base.** A base é verdadeira já que 2 é primo.

**Passo Indutivo.** Se todos os números do conjunto  $\{2, \dots, k\}$  são primos ou produtos de primos então  $k+1$  é primo ou produto de primos, para  $k \geq 2$ .

**Prova do Passo.** Devemos considerar dois casos:

1.  $k+1$  é primo. Neste caso  $k+1$  tem a propriedade desejada
2.  $k+1$  não é primo. Se por hipótese  $k+1$  não é primo, ele deve dever ser composto, daí  $k+1 = ab$ , onde  $1 < a \leq k$  e  $1 < b \leq k$ . Pela hipótese de indução  $a$  e  $b$  são primos ou produto de primos. É nesse ponto que a hipótese de indução forte é importante, pois supomos que *qualquer* número entre 2 e  $k$  é primo ou produto de primos e não apenas  $k$  (indução fraca). É justamente esse fato que nos dá a garantia de afirmar que  $a$  e  $b$  são primos ou produto de primos já que  $a$  e  $b$  podem assumir qualquer valor entre 2 e  $k$ . Logo  $ab$  é produto de primos. ■

**Exemplo 11** Definimos a sequência de Fibonacci da seguinte forma:  $F(1) = F(2) = 1$  e  $F(n) = F(n-1) + F(n-2)$ ,  $\forall n \geq 3$ . Mostre que  $F(n+4) = 3F(n+2) - F(n)$ ,  $\forall n \geq 1$

**Base.** Neste caso, precisamos de uma base dupla. Por quê?

$$\begin{aligned} n = 1 &\Rightarrow 5 = 3 \cdot 2 - 1 \\ n = 2 &\Rightarrow 8 = 3 \cdot 3 - 1 \end{aligned}$$

**Passo Indutivo.** Se  $F(r+4) = 3F(r+2) - F(r)$ ,  $\forall r \in \{1, \dots, k\}$  então  $F((k+1)+4) = 3F((k+1)+2) - F((k+1))$ , para todo  $k \geq 2$ .

**Hipótese Indutiva(forte).**  $F(r+4) = 3F(r+2) - F(r)$ ,  $\forall r \in \{1, \dots, k\}$ , para todo  $k \geq 2$

**Prova do Passo.**

$$\underbrace{F(k+5) = F(k+4) + F(k+3)}_{\text{definição}} \quad (1.1)$$

$$\underbrace{F(k+4) = 3F(k+2) - F(k)}_{\text{hipótese indutiva em } k} \quad (1.2)$$

$$\underbrace{F(k+3) = 3F(k+1) - F(k-1)}_{\text{hipótese indutiva em } k-1} \quad (1.3)$$

Note que podemos aplicar a indução para  $k-1$  já que  $k-1 \in \{1, \dots, k\}$ . Somando (1.2) e (1.3) temos que,

$$F(k+4) + F(k+3) = 3 \underbrace{(F(k+2) + F(k+1))}_{F(k+3)} - \underbrace{(F(k) + F(k-1))}_{F(k+1)}$$

$$F(k+5) = F(k+4) + F(k+3) = 3F(k+3) - F(k+1) \blacksquare$$

**Exemplo 12** Prove que qualquer valor postal maior ou igual a 8 unidades, pode ser obtido utilizando apenas selos com valor 3 e 5.

**Prova:**

Alguns exemplos:  $14 = 3$  selos de 3 e 1 selo de 5;  $99 = 28$  selos de 3 e 5 selos de 5.

**Base**

$n = 8 \Rightarrow$  Verdadeiro, basta pegar um selo de 5 e um selo de 3

**Passo Indutivo.** Se é possível obter qualquer valor postal  $r \in \{8, \dots, k\}$  utilizando selos de valores 3 e 5 então é possível obter o valor  $k + 1$  utilizando selos de valores 3 e 5, para todo  $k \geq 10$ .

**Hipótese Indutiva.** Assuma que qualquer valor entre 8 e  $k$  pode ser obtido utilizando selos de valores 3 e 5

**Prova do Passo.** Como

$$(k + 1) = (k - 2) + 3,$$

podemos obter o valor  $k + 1$  utilizando a solução para  $k - 2$  obtida pela hipótese indutiva e mais um selo de 3.

A prova está correta? Para  $k = 9$  teríamos  $9 = 6 + 3$ , ou seja, um selo de 3 mais a solução de 6. O problema é que a solução de 6 não é prevista pela hipótese de indução (vale apenas para  $8 \leq r \leq k$ ).

Para resolver esse problema podemos incluir 9 na base já que 9 pode ser obtido utilizando 3 selos de valor 3.

O que acontece para  $k = 10$ ? Temos nesse caso  $10 = 7 + 3$ . Entretanto, a solução de 7 também não é prevista na hipótese de indução. Novamente colocamos 10 na base já que 10 pode ser obtido através de 2 selos de valor 5 !!!

Dessa forma, colocando 9 e 10 na base, a prova fica correta pois sempre poderemos alcançar a base. ■

**Exemplo 13** *Uma árvore binária enraizada é*

(i) *um conjunto vazio ou*

(ii) *um nó denominado raiz com uma subárvore a esquerda e outra a direita*

*Uma folha é um nó com 0 filhos enquanto que um nó interno é um nó com pelo menos um filho. Uma árvore é dita estritamente binária se todo nó possui 0 ou 2 filhos.*

*Prove por indução no número de nós que para toda árvore estritamente binária  $T$ , a relação  $l(T) - i(T) = 1$  é válida, onde  $l(T)$  e  $i(T)$  são, respectivamente, o número de folhas e o número de nós internos de  $T$ .*

**Solução.**

**Base.** A árvore tem apenas um nó. Este nó, além de ser raiz da árvore, é também uma folha já que não tem filhos. Portanto, a árvore tem 0 nós internos e uma folha.

**Passo Indutivo.** Se para toda árvore estritamente binária com até  $k$  nós a relação é válida então a relação também é válida para toda árvore estritamente binária com exatamente  $k + 1$  nós, para todo  $k \geq 1$

**Hipótese Indutiva.** A relação vale para toda árvore estritamente binária com no máximo  $k$  nós, para todo  $k \geq 1$

**Prova do Passo:** Seja  $T$  uma árvore estritamente binária com  $k + 1$  nós. Além disso, seja  $r$  a raiz de  $T$ ,  $T_E$  a árvore a esquerda e  $T_D$  a árvore a direita. Temos que

$$i(T) = 1 + i(T_E) + i(T_D)$$

e

$$l(T) = l(T_E) + l(T_D)$$

Diminuindo a primeira equação da segunda obtemos que

$$l(T) - i(T) = l(T_E) - i(T_E) + l(T_D) - i(T_D) - 1 \quad (1.4)$$

As árvores  $T_E$  e  $T_D$  são árvores estritamente binárias, caso contrário  $T$  não seria. Além disso, tanto  $T_E$  como  $T_D$  tem no máximo  $k$  nós. Portanto,

a hipótese indutiva vale para  $T_D$  e  $T_E$ , ou seja,  $l(T_E) - i(T_E) = 1$  e  $l(T_D) - i(T_D) = 1$ . Substituindo estas expressões em 1.4 obtemos que  $l(T) - i(T) = 1$ , provando o passo.

**Exemplo 14** *Mostre que a seguinte relação*

$$|\text{folhas}(T)| \leq 2^{\text{altura}(T)}$$

*vale para toda árvore estritamente binária  $T$ .*



### 1.4.4 Corretude de Algoritmos

Uma aplicação importante do princípio da Indução Matemática em computação é provar a corretude de algoritmos. É de fundamental importância existir um método formal que assegure que determinado algoritmo realmente funcione de acordo com sua especificação.

Apresentamos três exemplos da aplicação de indução para provar corretude de algoritmos.

#### Exemplo 15 *Calculando o fatorial*

Consider o código abaixo.

<b>Fatorial</b>
input: $n$ (inteiro positivo)
<b>Função</b> $Fat(n)$
Se $n \leq 1$
Return $n$
Senão
Return $n \cdot Fat(n - 1)$
<b>Fim Função.</b>

Tabela 1.1: Pseudo-código.

Podemos mostrar que a saída de  $Fat(n)$  é  $n!$ , para todo  $n \geq 0$ . De fato, para  $n \leq 1$ ,  $Fat(n) = 1 = 0! = 1!$ .

**Passo Indutivo.** Se  $Fat(k)$  devolve  $k!$  então  $Fat(k + 1)$  devolve  $(k + 1)!$ , para todo  $k \geq 1$ .

**Prova do Passo.** Pelo código temos que  $Fat(k + 1)$  devolve  $(k + 1) \cdot Fat(k)$ . Entretanto, por hipótese de indução  $Fat(k) = k!$ . Segue que  $Fat(k + 1)$  devolve  $(k + 1) \cdot (k)! = (k + 1)!$

#### Exemplo 16 *Conversor Decimal-Binário*

Considere o algoritmo abaixo que obtém a representação binária de um número natural  $n$ .

<b>Algoritmo do Conversor</b>
input: $n$ (inteiro positivo) output: $b$ (array de bits com a representação binária de $n$ ) <b>Função</b> Converte( $n$ ) $t \leftarrow n$ $k \leftarrow 0$ zere todos os bits de $b$ <b>Enquanto</b> $t > 0$ <b>faça</b> $k \leftarrow k + 1$ $b[k] \leftarrow t \bmod 2$ $t \leftarrow t \text{ div } 2$ <b>Fim Enquanto</b> ; <b>Fim Função.</b>

Tabela 1.2: Pseudo-código.

Como exemplo, considere a execução do algoritmo quando  $n = 57$  :

	t	b	k
Início	57	000000000	0
Loop 1	28	000000001	1
Loop 2	14	000000001	2
Loop 3	7	000000001	3
Loop 4	3	000001001	4
Loop 5	1	000011001	5
Loop 6	0	000111001	6

Tabela 1.3: Execução do Algoritmo para n=57.

Para provar que o algoritmo funciona, considere a seguinte proposição:

**Proposição:** Sejam  $m_k$  e  $t_k$ , respectivamente, o inteiro representado pelo vetor binário  $b$  e o valor de  $t$  ao final do  $k$ -ésimo loop. Então,  $n = t_k \cdot 2^k + m_k$ ,  $\forall k$

Por que esta proposição? Ela basta para estabelecer a corretude do algoritmo?

Se a proposição é verdadeira, então no final do último loop, digamos  $l$ , o inteiro  $m_l$  representado pelo vetor binário  $b$  é tal que

$$n = t_l \cdot 2^l + m_l$$

Como  $t_l = 0$  no final do último loop, então  $m_l = n$ .

Conclusão: A corretude da proposição implica na corretude do algoritmo.

Dificuldade: Encontrar a proposição adequada !.

Provaremos a proposição utilizando indução no número de laço  $k$ :

**Base:**

$$k = 0, t = n \Rightarrow m = 0$$

A idéia aqui é que sendo a iteração a de número 0, o algoritmo não executou nenhuma vez, logo o número representado no array até essa iteração ( $m$ ) é 0.

**Passo Indutivo.** Se a proposição vale ao término do laço  $k$  então ela também vale após o laço  $k + 1$ , para todo  $k \geq 0$

**Hipótese Indutiva.** A proposição vale ao término do laço  $k$ .

### Prova do Passo

Caso 1)  $t_k$  é par  $\Rightarrow$  No final do loop  $k + 1$ , temos  $t_{k+1} = \frac{t_k}{2}$  e  $m_{k+1} = m_k$ . Logo,

$$t_{k+1} \cdot 2^{k+1} + m_{k+1} = \underbrace{t_k \cdot 2^k + m_k}_{\text{hip indutiva}} = n$$

Caso 2)  $t_k$  é ímpar  $\Rightarrow$  O valor de  $t_{k+1}$  é  $\frac{t_k-1}{2}$  e  $m_{k+1} = m_k + 2^k$ , já que um bit 1 é acrescentado na posição  $k + 1$  de  $b$ .

$$t_{k+1} \cdot 2^{k+1} + m_{k+1} = \frac{t_k - 1}{2} \cdot 2^{k+1} + m_k + 2^k = \underbrace{t_k \cdot 2^k + m_k}_{\text{hip indutiva}} = n \blacksquare$$

**Exemplo 17** O Quicksort, cujo pseudo-código é mostrado abaixo um dos algoritmos mais rápidos conhecidos para ordenar um vetor. O procedimento ordena o subvetor de  $A$  que começa na posição  $p$  e termina na posição  $r$ . Para facilitar a nossa discussão assumimos que todos os elementos são distintos.

```

Quicksort( $A, p, r$ )
  Se  $p < r$  então
     $q \leftarrow \text{PARTITION}(A, p, r)$ 
    Quicksort( $A, p, q-1$ )
    Quicksort( $A, q+1, r$ )
  Fim Se

```

Considere que  $\text{PARTITION}(A, p, r)$  é uma rotina de 'pivotamento' que funciona da seguinte maneira: seja  $X = A[p]$  e seja  $j$  o número de elementos menores que  $x$  no subvetor  $A[p, r]$ .  $\text{PARTITION}$  coloca o elemento  $x$  na posição  $q = p + j$ , os elementos de  $A[p, r]$  menores que  $x$  nas posições  $A[p], \dots, A[q - 1]$  e os maiores a  $x$  nas posições  $A[q + 1], \dots, A[r]$ .

Assumindo que a rotina  $\text{PARTITION}$  funciona da maneira especificada, utilize indução para mostrar que  $\text{Quicksort}(A, 1, n)$  ordena o vetor  $A[1, n]$ .

**Solução.** Utilizamos indução no tamanho do vetor de entrada.

**Base:** Se o vetor tem tamanho 1 ou é vazio, ou seja,  $p \geq r$ , o Quicksort não faz nada já que todo vetor de tamanho 1 é ordenado.

**Hipótese Indutiva.** Assuma que o Quicksort funciona corretamente para todo vetor de tamanho menor ou igual a  $k$ .

**Passo Indutivo** Provar para  $k + 1$

**Prova do Passo** Assuma que  $r - p + 1 = k + 1$ . Primeiramente,  $\text{Partition}$  coloca o elemento  $x = A[p]$  na posição correta. Após, ele chama  $\text{Quicksort}(A, p, q-1)$  e  $\text{Quicksort}(A, q+1, r)$  para ordenar os subvetores  $A[p, q - 1]$  e  $A[q + 1, r]$ . O funcionamento de  $\text{PARTITION}$  garante que  $A[p, q - 1]$  contém os elementos de  $A[p, r]$  menores que  $x$  enquanto  $A[q + 1, r]$  contém os maiores ou iguais a  $x$ . Como os subvetores  $A[p, q - 1]$  e  $A[q + 1, r]$  tem menos do que  $k + 1$  elementos, a hipótese garante que o Quicksort ordena

eles corretamente. Portanto, ao término da chamada  $\text{Quicksort}(A,p,r)$ ,  $x$  é colocado na posição correta em  $A[p,r]$  e os subvetores  $A[p,q-1]$  e  $A[q+1,r]$  estão ordenados. Logo, podemos concluir que  $A[p,r]$  está ordenado, o que prova o passo.

**Exemplo 18** *Mostre que o procedimento a seguir devolve o elemento que aparece mais que  $|L|/2$  vezes em uma lista  $|L|$  se tal elemento existir e devolve vazio caso contrário.*

*Proc(L)*

**Se**  $L$  só tem um elemento

**Return** o único elemento de  $L$

*Senão*

*Divida*  $L$  em duas listas  $L_1$  e  $L_2$  de tamanhos  $\lceil |L|/2 \rceil$  e  $\lfloor |L|/2 \rfloor$

$a \leftarrow \text{Proc}(L_1)$

$b \leftarrow \text{Proc}(L_2)$

**Se**  $a \neq \emptyset$  *verifique se*  $a$  *aparece mais que*  $|L|/2$  *vezes em*  $L$ .

*Em caso positivo* **Return**  $a$

**Se**  $b \neq \emptyset$  *verifique se*  $b$  *aparece mais que*  $|L|/2$  *vezes em*  $L$ .

*Em caso positivo* **Return**  $b$

**Return**  $\emptyset$

*Fim Se*

Vamos mostrar que o algoritmo tem o comportamento enunciado. O lema abaixo será útil para nossa demonstração.

**Lema 1** *Se algum elemento ocorre mais que metade das vezes em  $L$  então este elemento ocorre mais que metade das vezes em  $L_1$  ou  $L_2$ .*

Podemos provar por contradição. Seja  $\text{freq}(x, L)$  a frequência que o elemento  $x$  aparece em uma lista  $L$ . Assuma que  $\text{freq}(x, L) > |L|/2$  e que  $\text{freq}(x, L_1) \leq |L_1|/2$  e  $\text{freq}(x, L_2) \leq |L_2|/2$ . Somando as duas últimas equações concluímos que

$$\text{freq}(x, L_1) + \text{freq}(x, L_2) \leq |L_1|/2 + |L_2|/2.$$

Como  $\text{freq}(x, L_1) + \text{freq}(x, L_2) = \text{freq}(x, L)$  e  $|L_1|/2 + |L_2|/2 = |L|/2$  obtemos que  $\text{freq}(x, L) \leq |L|/2$ , o que contradiz a nossa hipótese. ■

Para provar que o algoritmo tem o comportamento desejado, utilizamos indução no tamanho da lista  $L$ .

**Base.**  $|L| = 1$ . Nesse caso o algoritmo devolve o único elemento de  $L$  já que este ocorre 100% das vezes.

**Passo Indutivo** Se o algoritmo tem o comportamento desejado para toda lista de tamanho menor ou igual a  $k$ , então ele tem o comportamento desejado para toda lista de tamanho  $k + 1$ , para todo  $k \geq 1$ .

**Prova do Passo** Seja  $L$  uma lista de tamanho  $k + 1$ . Dividimos a prova em dois casos.

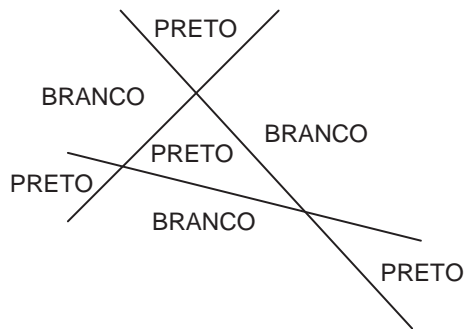
*Caso 1* Existe um elemento  $x$  que aparece mais que metade das vezes em  $L$ . Pelo lema anterior, uma das seguintes condições é satisfeita: (i)  $x$  aparece mais que  $|L_1|/2$  vezes em  $|L_1|$ ; (ii)  $x$  aparece mais que  $|L_2|/2$  vezes em  $|L_2|$ . Vamos considerar que a condição (i) é satisfeita. Neste caso, a hipótese indutiva garante que  $Proc(L_1)$  devolve  $x$ . Como  $x$  aparece em  $L$  mais que  $|L|/2$  vezes, o algoritmo devolve  $x$ . A prova da condição (ii) é análoga.

*Caso 2* Não existe elemento que aparece mais que metade das vezes em  $L$ . Neste caso, o algoritmo devolve vazio já que ele só devolve um elemento  $x$  se esse aparecer mais que metade das vezes em  $L$ .

### 1.4.5 Indução para Projeto de Algoritmo

Em muitas situações provas por indução auxiliam o projeto de algoritmos conforme ilustramos nesta seção.

**Exemplo 19** *Mostre que as regiões do plano determinadas por  $n$  retas em posição geral podem ser coloridas utilizando duas cores de modo que regiões adjacentes recebam cores diferentes. Escreva um algoritmo mostrando como pintar as regiões.*



**Prova:**

**Base:** Para  $n = 1$ , apenas 1 reta OK!



**Passo Indutivo.** Se o resultado é válido para quaisquer  $k$  retas em posição geral então o resultado também é válido para quaisquer  $k + 1$  retas em posição geral, para todo  $k \geq 1$

**Hipótese de Indução.** O resultado é válido para quaisquer  $k$  retas em posição geral, para todo  $k \geq 1$



**Prova do Passo.** Considere um conjunto de  $k$  retas. Pela hipótese indutiva existe uma coloração  $C_k$  para as regiões determinadas pelas  $k$  retas. Ao colocarmos a reta  $k + 1$ , as regiões em que a reta  $k + 1$  passa serão divididas em 2 regiões adjacentes que terão a mesma cor, segundo a coloração  $C_k$ (Figura 1.1).

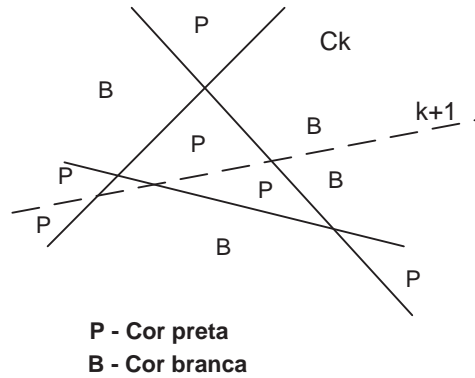


Figura 1.1: A reta  $k+1$  divide cada região, pela qual ela passa, em 2 novas regiões que terão a mesma cor.

Como contornar? Escolha um dos semi-planos determinados pela reta  $k + 1$  e inverta a cor de todas as regiões deste semi-plano. Dessa forma em  $C_{k+1}$ , regiões adjacentes tem cores diferentes(Figura 1.2).

Observamos 2 casos:

1. Se duas regiões adjacentes estão no mesmo semi-plano definida pela reta  $k + 1$ , então elas tinham cores diferentes em  $C_k$ , e portanto também terão em  $C_{k+1}$ .
2. Se duas regiões adjacentes, estão em semi-planos diferentes, então elas terão cores iguais em  $C_k$ , mas a cor de uma delas foi mudada em  $C_{k+1}$ . Portanto, terão cores diferentes. ■

Abaixo apresentamos o pseudo-código de um algoritmo para colorir as regiões de um plano geradas por  $n$  retas em posição geral.

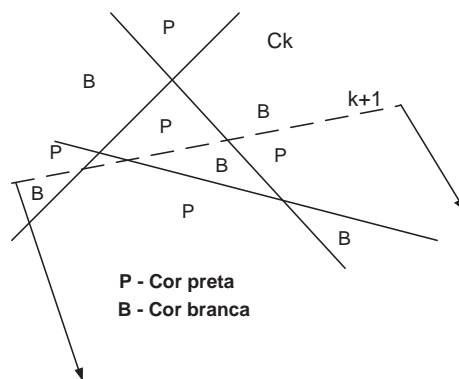


Figura 1.2: Trocando-se as cores nas regiões do semi-plano escolhido, resolve-se o problema.

**Algoritmo ColorePlano(n)**

**Se**  $n=1$

    Colore cada semi-plano de uma cor

**Senão**

    ColorePlano( $n-1$ )

    Inclua reta  $r_n$

    Inverta a cor de todas as regiões de um dos semi-planos determinados por  $r_n$

**Fim Se**

**Exemplo 20** *Mostre que qualquer tabuleiro  $2^n \times 2^n$ ,  $n \geq 1$ , que não possui uma casa no canto pode ser coberto por peças no formato  $L$ . Escreva um algoritmo mostrando como obter tal cobertura.*

A figura abaixo ilustra uma solução para um tabuleiro  $2^2 \times 2^2$ . O X representa a casa retirada do tabuleiro e a mesma peça é representado por letras iguais.

D	D	E	X
D	C	E	E
A	C	C	B
A	A	B	B

**Solução:**

Seja a proposição  $P(n)$  definida como

$$P(n) = \begin{cases} V, & \text{se um tabuleiro } 2^n \times 2^n \text{ sem uma peça do canto pode ser coberto.} \\ F, & \text{caso contrário} \end{cases}$$

**Base.** Para  $n = 1$  o resultado é válido como ilustra a figura abaixo.

A	X
A	A

**Passo Indutivo.**  $P(k) \Rightarrow P(k + 1), \forall k \geq 1$

**Hipótese Indutiva.**  $P(k)$  é válido, ou seja, um tabuleiro  $2^k \times 2^k$  sem a peça do canto pode ser coberto por peças em  $L$ .

**Prova do Passo.** Considere um tabuleiro com dimensões  $2^{k+1} \times 2^{k+1}$  sem uma casa no canto. Colocando uma peça no centro, obtemos 4 tabuleiros  $2^k \times 2^k$  sem uma casa no canto ( casa coberta). Portanto, segue da hipótese indutiva que estes podem ser cobertos ( A Figura 1.3 ilustra a construção). ■

A Figura 1.4.5 apresenta o pseudo-código de um algoritmo para cobrir um tabuleiro com peças em formato  $L$ .

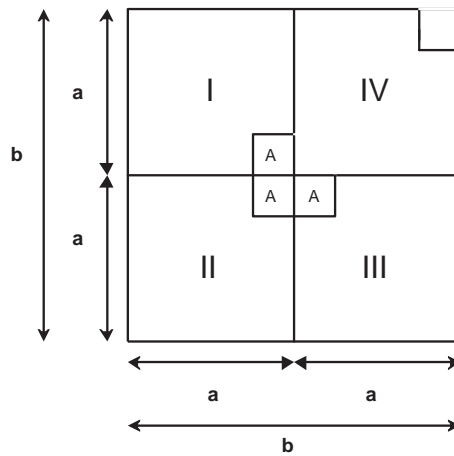


Figura 1.3:  $a = 2^k$  e  $b = 2^{k+1}$ .

**Algoritmo CobreTabuleiro( $T, k$ )**

**Se**  $k=1$

Utilize uma peça para cobrir  $T$

**Senão**

Insira uma peça no centro de  $T$  de modo a obter  
quatro tabuleiros  $T_1, T_2, T_3$  e  $T_4$  de lado  $2^{k-1}$  sem uma peça no canto.

CobreTabuleiro( $T_1, k - 1$ )

CobreTabuleiro( $T_2, k - 1$ )

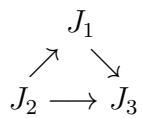
CobreTabuleiro( $T_3, k - 1$ )

CobreTabuleiro( $T_4, k - 1$ )

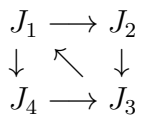
**Fim Se**

**Exemplo 21** *Mostre que em qualquer torneio de t nis, onde todos os jogadores jogam entre si,   poss vel dispor os jogadores em uma fila tal que o jogador na  $(i+1)$ - sima posi o tenha ganho do jogador na  $i$ - sima posi o, para  $i = 1, \dots, n - 1$ . Escreva um algoritmo mostrando como obter tal fila.*

Abaixo mostramos 2 torneios, um com tr s jogadores e outro com quatro. Uma seta do jogador  $a$  para o  $b$  indica que  $b$  ganhou de  $a$ .



Fila:  $J_2 J_1 J_3$



Fila:  $J_4 J_3 J_1 J_2$

Primeiramente, mostramos que a seguinte proposi o   verdadeira.

**Proposi o 26** *Dado um torneio com  $n \geq 2$  jogadores  $J_1, \dots, J_n$ , existe uma seq ncia  $J_{i_1} J_{i_2} \dots J_{i_n}$  tal que o jogador  $J_{i_{\ell+1}}$  ganha do jogador  $J_{i_\ell}$ , para  $\ell = 1, 2, \dots, n - 1$ .*

**Prova.** Utilizamos indu o no n mero  $n$  de jogadores.

**Base.** Para  $n = 2$ :

Caso1) Se o jogador 1 ganhou de 2  $\implies$  Seq ncia  $J_2 J_1$

Caso2) Se o jogador 2 ganhou de 1  $\implies$  Seq ncia  $J_1 J_2$

**Passo Indutivo.** Se para todo torneio com  $k$  jogadores existe uma seq ncia que satisfaz a condi o desejada ent o para todo torneio com  $k + 1$  jogadores tamb m existe tal seq ncia, para todo  $k \geq 2$ .

**Prova do Passo.** Seja um torneio com  $k + 1$  jogadores. Isolamos o jogador  $k + 1$ . Pela hip tese indutiva existe uma seq ncia para os jogadores

$J_1, \dots, J_k$  que satisfaz a condição desejada. Seja  $J_{l_1} J_{l_2} \dots J_{l_k}$  tal sequência.

Como construir a sequência para  $k + 1$  jogadores? Como inserir  $J_{k+1}$  na sequência obtida por indução? Por exemplo: se a hipótese indutiva nos fornece  $J_4 J_2 J_3 J_1$ , onde colocar  $J_5$ ?

Seja  $p$  o menor índice tal que  $J_{i_p}$  ganhou de  $J_{k+1}$ . Insira  $J_{k+1}$  antes de  $J_{i_p}$  na sequência obtida por indução. Se  $p$  não existe, então insira  $J_{k+1}$  no final da sequência.

A seguir mostramos que a sequência obtida utilizando a regra acima é válida.

Caso 1)  $p$  existe.: Neste caso,  $J_{i_p}$  ganhou de  $J_{k+1}$  devido a definição de  $p$ . Como  $p$  é o menor índice possível de um jogador que ganhou de  $J_{k+1}$ , então  $J_{k+1}$  ganhou de  $J_{i_{p-1}}$ .

Caso 2)  $p$  não existe. Neste caso,  $J_{k+1}$  ganhou de todos os outros e logo pode ser colocado no final da sequência. ■

Abaixo apresentamos o pseudo-código de um algoritmo para encontrar a sequência desejada.

<b>Algoritmo Torneio(n)</b>
<b>Se</b> $n=2$ <b>então</b>
<b>Se</b> $J_1$ Ganhou de $J_2$
<b>retorne</b> $J_2 J_1$
<b>Senão</b>
<b>retorne</b> $J_1 J_2$
<b>Fim Se</b>
<b>Senão</b>
$seq \leftarrow$ torneio( $n-1$ )
$p \leftarrow$ primeiro jogador de $seq$ que ganhou de $J_n$
<b>Se</b> $p$ existe
<b>Insira</b> $J_n$ antes de $p$
<b>Senão</b>
<b>Insira</b> $J_n$ no final da $seq$ .
<b>Fim Se</b>

### 1.4.6 Erros Comuns

**Proposição 27** *Em um conjunto de  $n \geq 2$  retas não paralelas e não coincidentes, todas interceptam-se no mesmo ponto.*

**Prova:**

**Base:**  $n = 2$ . Duas retas se cruzando, OK !

**Hipótese de Indução:** O resultado vale, para qualquer conjunto de  $k$  retas.

**Passo Indutivo:** Provar para  $k + 1$ .

Considere um conjunto de retas  $r_1, r_2, \dots, r_{k+1}$ . Segue da hipótese de indução que as retas  $r_1, r_2, \dots, r_k$  se interceptam em um ponto. Pelo mesmo motivo, as retas  $r_2, r_3, \dots, r_{k+1}$  também se interceptam em um único ponto (visto que esse último conjunto também possui  $k$  elementos). Como há retas que pertencem ao primeiro conjunto e ao segundo, o ponto tem de ser o mesmo. Logo *todas* as retas se interceptam em um único ponto ■

Onde está o furo?

Argumentação: Para garantir que as  $k + 1$  retas se interceptam em um único ponto, seria necessário que a interseção dos conjuntos  $\{r_1, \dots, r_k\}$  e  $\{r_2, \dots, r_{k+1}\}$  tivessem pelo menos duas retas, pois essas determinariam o ponto. Entretanto quando  $k + 1 = 3$ , a interseção entre  $r_1, r_2$  e  $r_2, r_3$  é apenas a reta  $r_2$ . Logo, não é possível passar de  $k = 2$  para  $k = 3$ , o que fura a demonstração.

**Proposição 28** *Em toda festa com  $n \geq 3$  pessoas em que cada pessoa conhece mais da metade das demais pessoas ( $n - 1/2$ ) é possível dispor as pessoas em uma roda de modo que cada pessoa conhece seus dois vizinhos.*

Para demonstrar a proposição utilizamos o seguinte lema.

**Lema 2** *Seja  $C$  uma roda com  $n \geq 3$  pessoas. Se existe uma pessoa  $p_{n+1}$  que não está em  $C$  e conhece mais do que  $n/2$  pessoas em  $C$  então é possível colocar a pessoa na roda  $C$  de modo que ela conheça seus dois vizinhos.*

Se não existe lugar para colocar  $p_{n+1}$  então entre dois conhecidos de  $p_{n+1}$  sempre há pelo menos um desconhecido de  $p_{n+1}$ . Portanto, o número de desconhecidos de  $p_{n+1}$  é maior ou igual ao número de conhecidos de  $p_{n+1}$ , o que implica que o total de pessoas na roda é maior que  $n$ , o que é uma contradição. ■

Focamos agora no resultado principal

**Prova.** Indução no número de pessoas.

**Base.**  $n = 3$ . O resultado é verdadeiro já que todo mundo se conhece.

**Passo Indutivo.** Se toda festa com  $k$  pessoas, em que cada pessoa conhece mais que  $(k - 1)/2$  pessoas, tem a propriedade desejada então toda festa com  $k + 1$  pessoas, em que cada pessoa conhece mais de  $(k)/2$  pessoas, tem a propriedade desejada,  $\forall k \geq 3$ .

**Prova do Passo** Seja uma festa  $F_k$  com  $k$  pessoas em que cada pessoa conhece mais que  $k/2$  pessoas. Considere uma festa  $F_{k+1}$  com  $k + 1$  pessoas que obtemos convidando para  $F_k$  uma pessoa que conhece pelo menos  $k/2$  convidados de  $F_k$ .

Por hipótese de indução a festa  $F_k$  admite uma roda em que cada pessoa conhece seus vizinhos. Como a pessoa  $(k + 1)$  conhece mais que  $k/2$  pessoas da festa  $F_k$ , então, pelo lema anterior, existe um lugar na roda  $C$  aonde podemos colocar a pessoa  $k + 1$  de modo que ela conheça seus dois vizinhos. Portanto, obtemos uma nova roda com a propriedade desejada, concluindo o passo indutivo. ■

**A prova acima está errada! Por que?**

O problema é que estamos particularizando. Para obter a festa  $F_{k+1}$  partimos de uma festa  $F_k$  em que cada pessoa conhece pelo menos  $k/2$  pessoas. Este processo não permite obter toda festa  $F_{k+1}$  que tem a propriedade desejada. Por exemplo, a festa  $\{AB, BC, CD, DA\}$  não pode ser obtida desta forma.

Devemos notar que apesar da demonstração acima estar errada, a proposição sobre a festa é verdadeira já que existe uma demonstração para ela.

A demonstração utiliza terminologia de teoria de grafos. Dizemos que uma festa é 'boa' se todo convidado conhece mais da metade dos outros convidados. Seja  $F$  uma festa 'boa' onde não é possível colocar os convidados em roda e seja  $n$  o número de convidados de  $F$ . Seja  $H$  a festa 'boa' com maior número de relações de conhecimento (arestas) dentre as festas boas com  $n$  convidados. Sejam  $x$  e  $y$  duas pessoas em  $H$  que não se conhecem.



Seja  $H'$  uma festa igual a  $H$  exceto pelo fato que  $x$  e  $y$  se conhecem. Note que a maximalidade de  $H$  implica que é possível colocar os convidados de  $H'$  em uma roda em que  $x$  e  $y$  são vizinhos. Portanto, é possível colocar os convidados de  $H$  em uma fila começando em  $x$  e terminando em  $y$  de modo que todos conheçam seus vizinhos. Seja  $x = v_1v_2 \cdots v_{n-1}v_n = y$  tal fila. Note que na festa  $H$ , para  $i = 3, \dots, n-1$ , ou  $x$  não conhece  $v_i$  ou  $y$  não conhece  $v_{i-1}$ , caso contrário teríamos uma roda  $xv_2 \cdots v_{i-1}yv_{n-1} \cdots v_ix$  na festa  $H$ . Portanto, segue que a soma dos vizinhos de  $x$  e  $y$  é no máximo  $n-1$ , o que contradiz o fato que tanto  $x$  quanto  $y$  conhecem mais da metade dos convidados.

## Capítulo 2

# Princípios de Contagem e Enumeração Computacional

### Objetivo

- Contar/listar o número de elementos de conjuntos finitos

### Aplicações

- Determinar o número de operações realizadas por um algoritmo para determinar sua eficiência
- Ferramenta utilizada para resolver problemas estatísticos
- Desenvolver o raciocínio dos alunos de ED

Por que estudar problemas do tipo

P1) De quantas maneiras é possível dispor os alunos de uma turma em fila, sem que alunos com nomes consecutivos na ordem alfabética fiquem juntos?

P2) Quantas permutações distintas tem a palavra MISSISSIPI?

Duas justificativas são bastante razoáveis

1. Estes problemas são fáceis de formular;

2. 2 As técnicas envolvidas na solução destes problemas podem ser aplicadas na resolução de problemas práticos

Em algumas situações estamos interessados em calcular o número de elementos de um conjunto, enquanto que em outras desejamos listar todos os elementos de um conjunto.

**Exemplo 22** *Seja  $S$  o conjunto de todos os números de três algarismos que satisfazem as seguintes propriedades*

(i) *todo algarismo pertence ao conjunto  $\{2, 3, 6\}$*

(ii) *nenhum algarismo pode repetir no mesmo número*

P1) *Quantos elementos possui  $S$ ?*

P2) *Quem são os elementos de  $S$ ?*

Para responder a pergunta P1, devemos calcular o número de elementos em  $S$ , enquanto que para responder P2 devemos listar (enumerar) todos os elementos.

Para resolver P1, basta observar que existem 3 possibilidades para o algarismo mais a esquerda, 2 para o algarismo do centro e uma possibilidade para o mais a direita. Portanto,  $|S| = 3 \times 2 \times 1 = 6$

Para resolver P2, devemos listar todos os elementos de  $S$ . Podemos coloca-los em ordem crescente por exemplo: 236,263,326,362,623,632.

**Exemplo 23** *Considere um caminhão que necessita entregar mercadorias em 12 localidades  $\{L_1, \dots, L_{12}\}$  ao longo de um dia. Sabe-se que o consumo médio para ir da localidade  $L_i$  para localidade  $L_j$  é  $c_{ij}$ , e que o caminhão deve partir de sua garagem e retornar para mesma. Considere que  $L_0$  identifica a garagem.*

P1) *Quantos trajetos distintos o caminhão pode percorrer?*

P2) *Qual o trajeto que minimiza o consumo do caminhão ?*

A questão P1 envolve contagem. Existem 12 possibilidades para a primeira localidade, 11 para segunda, 10 para terceira. Em geral,  $13-i$  possibilidades para  $i$ -ésima localidade. Portanto, o número de trajetos é  $12! = 479001600$ .

Uma forma de resolver a segunda questão é listar todos os percursos possíveis, calculando o consumo de cada um, e selecionar aquele(s) de menor consumo. Esta tarefa é bastante árdua de realizar já que o número de percursos é extremamente grande. Entretanto, com o auxílio de um computador isto pode ser feito. Para isto, necessitamos de um método sistemático para listar todos os percursos possíveis, ou seja, um algoritmo de enumeração.

Uma observação interessante é que não é conhecido nenhum algoritmo eficiente (polinomial no tamanho da entrada) para resolução deste problema. Inclusive, existe um site na Internet (<http://www.claymath.org/prizeproblems/pvsnp.htm>) que oferece U\$1.000.000 para alguém que desenvolva tal algoritmo ou que prove que tal não pode existir.

## 2.1 Princípio da Multiplicação

Uma das técnicas mais elementares de contagem é conhecida como princípio da multiplicação.

Sejam os eventos  $E_1, E_2, \dots, E_k$ . Se o evento  $E_i$ ,  $i = 1, \dots, k$ , pode ocorrer de  $n_i$  formas diferentes e a forma com que o evento  $E_i$  ocorre não influencia no número de formas que o evento  $E_j$  pode ocorrer,  $\forall i \neq j$ , então a sequência de eventos  $E_1 E_2 E_3 \dots E_k$  pode ocorrer de  $n_1 \times n_2 \times n_3 \times \dots \times n_k$  formas diferentes.

**Exemplo 24** *Em uma placa de carro, as três primeiras posições são letras e as quatro restantes são dígitos. Quantas placas distintas são possíveis?*

O evento  $E_i$ ,  $i = 1, 2, 3$ , consiste em atribuir uma letra a  $i$ -ésima posição, enquanto que os eventos  $E_4, E_5, E_6, E_7$  consistem em atribuir dígitos as posições 4, 5, 6 e 7 respectivamente. Logo temos,  $|E_i| = 26$ , para  $i = 1, 2, 3$  e  $|E_i| = 10$ , para  $i = 4, \dots, 7$ . Segue do princípio da multiplicação que o número de placas é  $10^4 \times 26^3$ .

**Exercício 1** *Quantas placas são possíveis de modo que nenhuma letra e nenhum dígito apareçam repetidos?*

**Exemplo 25** Seja  $S$  o conjunto dos números de telefone com as seguintes propriedades:

- (i) o número é formado de 8 dígitos
- (ii) o primeiro dígito pertence a  $\{2, 3, 5, 7\}$
- (iii) deve haver pelo menos um número repetido dentre os quatro últimos dígitos.

Quantos elementos possui  $S$ ?

**Exemplo 26** Considere o trecho de código abaixo

```
Para  $i=1, \dots, l$   
  Para  $j=1, \dots, m$   
    Para  $k=1, \dots, n$   
      PRINT('OI')
```

Determine em função de  $m, n$  e  $l$  o número de vezes que o trecho de código imprime *OI*

### 2.1.1 Gerando todas as palavras de uma alfabeto

Nesta seção consideramos o problema de gerar todas as palavras de  $n$  letras a partir de um alfabeto com  $s$  letras distintas. O número de palavras possíveis é  $s^n$  já que temos  $s$  possibilidades para primeira letra,  $s$  para segunda e assim por diante. Por exemplo, seja o alfabeto  $A = \{a, b, c, d\}$  e  $n = 3$ . Temos as seguintes sequências: aaa, aab, aac, aad, aba, abb, abc, abd, aka, acb, acc, acd, ada, adb, adc, add, baa, bab, bac, bad, bba, bbb, bbc, bbd, bca, bcb, bcc, bcd, bda, bdb, bdc, bdd, caa, cab, cac, cad, cba, cbb, cbc, cbd, cca, ccb, ccc, ccd, cda, cdb, cdc, cdd, daa, dab, dac, dad, dba, dbb, dbc, dbd, dca, dcab, dcc, dcd, dda, ddb, ddc, ddd,

Para listar todas as palavras de comprimento  $n$  utilizamos uma abordagem indutiva. O caso base,  $n = 1$ , consiste em gerar todas palavras com uma única letra, o que é trivial. Vamos assumir, como hipótese de indução, que sabemos gerar todas as palavras de comprimento  $k$ . Como isto ajuda a gerar todas as palavras de comprimento  $k + 1$ , ou seja, dar o passo?

Note que para gerar o conjunto de todas as palavras de comprimento  $k + 1$ , devemos escolher uma letra do alfabeto para iniciar a palavra e depois

concatenar a letra escolhida com cada uma das palavras possíveis de comprimento  $k$  (que sabemos gerar por hipótese). Esta rotina deve ser repetida para cada letra do alfabeto.

Esta abordagem induz o procedimento **GeraPalavras** apresentado na Figura 2.1. Ao ser executado com parâmetro  $n$ , o procedimento imprime todas as palavras de comprimento  $i$  que podem ser obtidas utilizando o alfabeto  $A$  de  $s$  letras. O vetor  $A$  de  $s$  posições armazena as letras do alfabeto e a variável  $P$  é utilizada para armazenar a palavra que está sendo construída.

```

PROCEDIMENTO GeraPalavras(i)
  Se  $i = 0$ 
    Imprima a palavra  $P$ 
  Senão
    Para  $j = 1$  até  $s$ 
      Insira a letra  $A[j]$  na posição  $n + 1 - i$  de  $P$ 
      GeraPalavras(i-1)
    Fim Para
  Fim Se

MAIN
  Leia  $n, s, A$ 
   $P \leftarrow$  vetor global de  $n$  posições
  GeraPalavras( $n$ )

```

Figura 2.1: Algoritmo de geração de palavras

## 2.2 Permutações

Utilizamos  $P(n, r)$  para denotar o número de seqüências <sup>1</sup> de  $r$  objetos distintos que podem ser formadas a partir de um conjunto de  $n$  objetos distintos.

**Exemplo 27** *Sejam  $n = 4$  objetos  $\{o_1, o_2, o_3, o_4\}$  e  $r = 2$ . Quantas são as seqüências possíveis de 2 objetos distintos?*

<sup>1</sup>o fato de utilizarmos o termo seqüência faz com que a ordem de aparição dos objetos seja levada em conta

As seqüências são

$$o_1o_2, o_1o_3, o_1o_4, o_2o_1, o_2o_3, o_2o_4, o_3o_1, o_3o_2, o_3o_4, o_4o_1, o_4o_2, o_4o_3.$$

Portanto,  $P(4, 2) = 12$ . Em geral, aplicando o princípio multiplicativo, obtemos que

$$P(n, r) = n \times (n - 1) \times \dots \times (n - r + 1) = \frac{n!}{(n - r)!}$$

A idéia é que temos  $n$  possibilidades para o primeiro elemento da seqüência,  $n - 1$  para o segundo, e assim sucessivamente, até  $n - r$  para o  $r$ -ésimo elemento.

**Exemplo 28** *Quantas palavras de 3 letras podem ser formadas com as letras da palavra compilar, se não for permitido repetir letras?*

Devemos escolher seqüências de três letra distintas de um conjunto de 8 letras. Portanto, o total é  $P(8, 3) = 8 \cdot 7 \cdot 6 = 336$ .

**Exercício 2** *De quantas maneiras podem ser escolhidos o presidente e o vice-presidente de uma empresa, a partir de um grupo de 20 funcionários?*

**Exercício 3** *Uma biblioteca possui 4 livros sobre (S)ociologia, 7 livros sobre (P)rogramação e 3 sobre (E)xoterismo. De quantas maneiras podemos dispor este livros em uma prateleira de modo que livros do mesmo assunto fiquem juntos?*

Devemos levar em conta a ordem dos assuntos na prateleira e as ordem dos livros de um mesmo assunto. Observe que existem  $3!$  possibilidade de dispor os assuntos: SPE,SEP,PSE,PES,ESP,EPS. Para cada uma destas possibilidades, podemos permutar os livros do mesmo assunto de  $7!4!3!$  formas. Logo, o total de possibilidades é  $3!7!4!3!$ .

**Exercício 4** *Quantas são as permutações da palavra BULGARO que não possuem duas vogais em posições consecutivas?*

Fixemos primeiramente uma ordem entre as consoantes. Por exemplo, BLGR. Observe que existem 5 posições para colocar a vogal U: antes de B, entre B e L, entre L e G, etc. Após restam 4 possibilidades para a vogal A, já que vogais não podem ficar juntas. Finalmente, sobram 3 possibilidades para a vogal O. Portanto, temos  $5 \times 4 \times 3$  possibilidades de dispor as vogais dado que a ordem das consoantes foi fixada. Como existem  $4!$  possibilidades de fixar a ordem entre as consoantes, temos que o total de possibilidades é  $5 \times 4 \times 3 \times 4!$ .

### 2.2.1 Gerando Permutações

Nesta seção consideramos o problema de listar todas as  $n!$  permutações de um conjunto de  $n$  elementos.

Utilizamos uma abordagem indutiva. Obter o conjunto de permutações de um conjunto com 1 elemento é trivial. Vamos assumir, como hipótese de indução, que sabemos gerar todas as permutações de um conjunto de  $k$  elementos para  $k \geq 1$ . Como isto ajuda a gerar todas as permutações de um conjunto de  $k + 1$  elementos, ou seja, dar o passo?

Note que para gerar todas as permutações de um conjunto  $L$  de  $k + 1$  elementos, podemos fixar algum elemento  $s$  para ser o primeiro da permutação e depois concatenar  $s$  com cada uma das permutações do conjunto  $L - \{s\}$  (que sabemos gerar por hipótese). Esta rotina deve ser repetida para cada uma das escolhas possíveis de  $s$ .

A abordagem descrita induz o procedimento PERM apresentado na Figura 2.2. Ao ser executado com parâmetro  $L$ , o procedimento imprime todas as permutações do conjunto  $L$ . A variável  $P$  é utilizada para armazenar a permutação que está sendo construída. Note que existem várias maneiras de implementar a linha (\*) do procedimento. Uma possibilidade é no loop  $j$  escolher o  $j$ -ésimo menor elemento da lista  $L$ . Outra é utilizar uma estrutura auxiliar para anotar os elementos que já foram escolhidos.

**Exercício 5** *Em que ordem o procedimento PERM( $L$ ) imprime as permutações se  $L = \{a, b, c, d\}$ . Considere as possibilidades em que o procedimento escolhe o  $j$ -ésimo menor elemento de  $L$  na linha (\*) e a possibilidade que o  $j$ -ésimo maior elemento de  $L$  é escolhido na linha (\*).*

**Exercício 6** *Como podemos modificar o procedimento PERM para que ele receba um inteiro  $k$  e uma lista  $L$  de elementos distintos e imprima todas as sequências de comprimento  $k$ , sem repetição de elementos, que podem ser obtidas utilizando os elementos da lista  $L$ ?*

## 2.3 Combinações

De quantas maneiras podemos selecionar um subconjunto de  $r$  objetos de um conjunto de  $n$  objetos? Utilizamos  $\binom{n}{r}$  para denotar tal quantidade.



```

PROCEDIMENTO PERM( $L$ )
  Se a lista  $L$  é vazia
    imprima a permutação  $P$ 
  Senão
    Para  $j = 1$  até  $|L|$ 
       $s \leftarrow$  elemento de  $L$  que ainda não foi escolhido no loop (*)
      Insira  $s$  na posição  $n + 1 - |L|$  de  $P$ .
      PERM( $L - s$ )
    Fim Para
  Fim Se

MAIN
  Leia  $L$ 
   $P \leftarrow$  vetor global com  $|L|$  posições.
  PERM( $L$ )

```

Figura 2.2: Algoritmo de Geração de Permutações

**Exemplo 29** *Seja o conjunto  $\{o_1, o_2, o_3, o_4\}$  de  $n = 4$  objetos e  $r = 2$ . Então os subconjuntos possíveis são:  $\{o_1, o_2\}, \{o_1, o_3\}, \{o_1, o_4\}, \{o_2, o_3\}, \{o_2, o_4\}, \{o_3, o_4\}$ . Portanto,  $\binom{4}{2} = 6$ .*

Podemos calcular  $\binom{n}{r}$  utilizando a seguinte idéia: a cada subconjunto de  $r$  objetos associamos  $r!$  seqüências, cada uma delas contendo os  $r$  objetos do subconjunto em alguma ordem. Considerando o exemplo anterior temos

subconjunto	sequencias
$\{o_1, o_2\}$	$o_1o_2, o_2o_1$
$\{o_1, o_3\}$	$o_1o_3, o_3o_1$
$\{o_1, o_4\}$	$o_1o_4, o_4o_1$
$\{o_2, o_3\}$	$o_2o_3, o_3o_2$
$\{o_2, o_4\}$	$o_2o_4, o_4o_2$
$\{o_3, o_4\}$	$o_3o_4, o_4o_3$

Como cada um dos  $\binom{n}{r}$  subconjuntos esta associado a  $r!$  seqüências distintas e o total de seqüências de  $r$  objetos distintos é  $P(n, r)$ , segue que

$$r! \binom{n}{r} = P(n, r) \rightarrow \binom{n}{r} = \frac{P(n, r)}{r!} = \frac{n!}{(n-r)!r!}$$

**Exemplo 30 (Jogo da Sena)** *De quantas maneiras podemos escolher seis números de um total de 60 números? Qual é a probabilidade de ganhar o prêmio máximo da Sena jogando um único cartão com seis números?*

O total de possibilidades é  $\binom{60}{6}$ . Portanto a probabilidade é de  $1/\binom{60}{6}$

**Exemplo 31** *Qual a probabilidade de ganharmos na Sena jogando um cartão com 8 números?*

Jogando um cartão com 8 números cobrimos  $\binom{8}{6}$  subconjuntos de 6 números. Portanto, a probabilidade de ganhar é  $\binom{8}{6}/\binom{60}{6}$ . Observe que de fato o preço de um cartão com 8 números é  $\binom{8}{6} = 28$  vezes maior que o de um cartão simples.

**Exercício 7** *Um comitê de 8 estudantes deve ser selecionado de uma turma de 19 calouros e 34 veteranos.*

- a) *De quantas formas podem ser selecionados 3 calouros e 5 veteranos?*
- b) *De quantas formas podemos selecionar comitês com exatamente um calouro?*
- c) *De quantas maneiras podemos selecionar comitês com no máximo um calouro?*
- d) *De quantas maneiras podemos selecionar comitês com pelo menos um calouro?*

Resolveremos somente o item (d). Observe que qualquer comitê é válido, exceto aqueles que não tem nenhum calouro. Portanto, podemos calcular o total de comitês e diminuir do número de comitês que não contem calouros. O total de comitês é  $\binom{53}{8}$ . Por outro lado, o número de comitês sem calouros é igual ao número de comitês só com veteranos,  $\binom{34}{8}$ . Logo, a resposta é  $\binom{53}{8} - \binom{34}{8}$ .

**Exercício 8** Considere o trecho de código abaixo

```
Para i=3,...n
  Para j=2,...i-1
    Para k=1,...j-1
      PRINT('OI')
```

Determine em função de  $n$  o número de vezes que o trecho de código imprime OI.

**Exercício 9** Mostre que para todo natural  $r$  menor ou igual a  $n$ ,

$$\binom{n}{r} = \binom{n}{n-r}$$

Algebricamente é bastante simples mostrar a relação proposta.

Apresentamos uma forma alternativa de mostrar que relação é verdadeira. Sejam  $S$  e  $T$ , respectivamente, o número de subconjuntos de  $\{1, \dots, n\}$  com  $r$  elementos e  $n-r$  elementos. Sabemos que  $|S| = \binom{n}{r}$  e  $|T| = \binom{n}{n-r}$ . Portanto, para mostrar que  $\binom{n}{r} = \binom{n}{n-r}$  basta mostrar que  $|S| = |T|$ .

Uma forma de demonstrar que dois conjuntos tem o mesmo número de elementos é apresentar uma bijeção (correspondência um para um) entre eles. Neste caso, a bijeção é obtida associando cada subconjunto  $s \in S$  o subconjunto  $\{1, \dots, n\} - s \in T$ . Como exemplo, se  $n = 5$  e  $r = 3$ , associamos o conjunto  $\{1, 2, 4\}$  ao conjunto  $\{3, 5\}$  e o conjunto  $\{3, 4, 5\}$  ao conjunto  $\{1, 2\}$ .

**Exercício 10** Prove a relação de Stiffel

$$\binom{n}{r} + \binom{n}{r+1} = \binom{n+1}{r+1}$$

**Exercício 11** Mostre que

$$\sum_{r=0}^n \binom{n}{r} = 2^n$$

**Exercício 12** Considere um tabuleiro de damas  $8 \times 8$ . De quantas maneiras podemos levar uma peça da casa inferior esquerda para casa superior direita se os únicos movimentos permitidos são mover a peça uma casa para direita ou uma casa para cima?

### 2.3.1 Gerando Subconjuntos

Nesta seção mostramos como escrever um procedimento para gerar todos os subconjuntos de  $k$  elementos de um conjunto  $L$ . Como vimos na Seção 2.3, o total destes subconjuntos é  $\binom{|L|}{k}$ .

Novamente, utilizamos uma abordagem indutiva. Se  $k = 0$ , o problema é trivial já que não existe subconjunto.

Vamos assumir, como hipótese de indução, que para todo conjunto  $L$  com no máximo  $n$  elementos e todo inteiro  $k \leq |L|$  sabemos gerar todos os subconjuntos de tamanho  $k$  de  $L$ . Como isto ajuda a gerar todos os subconjuntos de tamanho de  $k'$  de um conjunto  $L'$  com  $n + 1$  elementos onde  $k' \leq n + 1$ , ou seja, dar o passo? Note que para gerar todos os subconjuntos de tamanho  $k'$  a partir de um conjunto  $L'$  podemos escolher um elemento  $s$  e depois unir o elemento  $s$  com cada um dos subconjuntos de  $k' - 1$  elementos obtidos a partir do conjunto  $L' - \{s\}$  (que sabemos gerar por hipótese já que este tem no máximo  $n$  elementos). Essa rotina deve ser repetida para todas as escolhas possíveis de  $s$ .

Note, entretanto, que devemos tomar certos cuidados para evitar gerar o mesmo subconjunto mais de uma vez. Uma forma de garantir isso é fixar uma ordem entre os elementos de  $L$  e gerar os elementos em ordem crescente.

A abordagem descrita induz o procedimento `GERA_CONJUNTOS` apresentado na Figura 2.3. Ao ser executado com parâmetros  $L$  e  $k$ , o procedimento imprime todos os subconjuntos de  $L$  de tamanho  $k$ . A variável  $S$  é utilizada para armazenar o subconjunto que está sendo construído. Como estamos escolhendo os elementos que vão compor  $S$  em ordem crescente, sempre que escolhemos um elemento  $s$ , removemos de  $L$  o conjunto  $T$  com os elementos menores ou iguais a  $s$  para evitar que estes sejam escolhidos mais adiante pelo procedimento.

**Exercício 13** *Qual a ordem de impressão dos subconjuntos quando o procedimento é executado com  $L = \{a, b, c, f, g\}$  e  $k = 3$ .*

## 2.4 Permutações com Repetições

Considere uma coleção de  $n$  objetos, onde  $n_i$  objetos são do tipo  $t_i$ , para  $i = 1, \dots, k$ . Duas sequências de  $n$  objetos,  $\sigma_1$  e  $\sigma_2$ , são ditas equivalentes se, para toda posição  $i$ , os objetos nas  $i$ -ésimas posições de  $\sigma_1$  e  $\sigma_2$  tem o mesmo tipo. Quantas sequências não equivalentes destes  $n$  objetos existem?

```

PROCEDIMENTO GERA_CONJUNTOS( $L, k$ )
    Se  $k = 0$ 
        Imprima os elementos do conjunto  $S$ 
    Senão
        Para  $j = 1$  até  $|L| - (k - 1)$ 
             $s \leftarrow j$ -ésimo menor elemento de  $L$ 
            Insira  $s$  no conjunto  $S$ .
             $T \leftarrow$  elementos de  $L$  menores ou iguais a  $s$ .
            GERA_CONJUNTOS( $L - T, k - 1$ )
            Remova  $s$  do conjunto  $S$ .
        Fim Para
    Fim Se

MAIN
    Leia  $L, k$ 
     $S \leftarrow$  conjunto vazio.
    GERA_CONJUNTOS( $L, k$ )

```

Figura 2.3: Algoritmo de Geração de todos os subconjuntos de  $k$  elementos de um conjunto  $L$

**Exemplo 32** *Considere um conjunto de 5 objetos, onde 3 são do tipo  $t_1$ , 1 é do tipo  $t_2$  e 1 é do tipo  $t_3$ . Quais são as seqüências distintas não equivalentes que podem ser formadas com estes objetos?*

Uma seqüência pode ser definida unicamente indicando o tipo que ocupa cada posição. Portanto, temos as seguintes seqüências

$t_1 t_1 t_1 t_2 t_3$	$t_1 t_1 t_1 t_3 t_2$	$t_1 t_1 t_2 t_1 t_3$	$t_1 t_1 t_2 t_3 t_1$	$t_1 t_1 t_3 t_1 t_2$
$t_1 t_1 t_3 t_2 t_1$	$t_1 t_2 t_1 t_1 t_3$	$t_1 t_2 t_1 t_3 t_1$	$t_1 t_2 t_3 t_1 t_1$	$t_1 t_3 t_1 t_1 t_2$
$t_1 t_3 t_1 t_2 t_1$	$t_1 t_3 t_2 t_1 t_1$	$t_2 t_1 t_1 t_1 t_3$	$t_2 t_1 t_1 t_3 t_1$	$t_2 t_1 t_3 t_1 t_1$
$t_2 t_3 t_1 t_1 t_1$	$t_3 t_1 t_1 t_1 t_2$	$t_3 t_1 t_1 t_2 t_1$	$t_3 t_1 t_2 t_1 t_1$	$t_3 t_2 t_1 t_1 t_1$

**Exemplo 33** *Quantas permutações distintas tem a palavra MISSISSIPI?*

O problema consiste em determinar o número de sequências de 10 letras em que a letra  $M$  aparece uma vez, a letra  $I$  4 vezes, a letra  $S$  4 vezes e a letra  $P$  uma vez. Podemos pensar em  $M$  como o tipo  $t_1$ ,  $I$  como o tipo  $t_2$ , etc. Para resolver o problema, observamos que existem  $\binom{10}{1} = 10$  posições para colocar a letra  $M$ . Dado que a letra  $M$  já foi colocada, existem  $\binom{9}{4} = 126$  possibilidades para dispor as letras  $I$ . Dado que as letras  $M$  e  $I$  já foram colocadas, existem  $\binom{5}{4} = 5$  possibilidades para dispor as letras  $S$ . Finalmente, sobra uma posição para letra  $P$ . Utilizando o princípio da multiplicação, temos um total de  $10 \times 126 \times 5 \times 1 = 6300$  possibilidades.

O raciocínio utilizado para resolver o problema anterior pode ser empregado para resolver a questão colocada no início da seção. De fato, o número de sequências distintas de  $n$  objetos, onde  $n_i$  objetos são do tipo  $t_i$ , para  $i = 1, \dots, k$ , é dado pela seguinte fórmula

$$\prod_{i=1}^k \binom{n - \sum_{j=1}^{i-1} n_j}{n_i} = \frac{n!}{n_1! n_2! \dots n_k!} \quad (2.1)$$

### 2.4.1 Enumeração

Podemos adaptar o código apresentado na Seção ?? para gerar todas as permutações quando existem símbolos repetidos. A idéia chave é modificar o procedimento para permitir que apenas tipos distintos sejam colocados em uma dada posição da variável  $P$ .

## 2.5 Combinações com Repetições

Nesta seção mostramos como contar o número de soluções inteiras não negativas da equação

$$\sum_{i=1}^k x_i = r \quad (2.2)$$

A técnica desenvolvida nesta seção pode ser aplicada em diversos problemas de contagem.

**Exemplo 34** *Quantas soluções inteiras não negativas tem a equação*

$$n_1 + n_2 + n_3 + n_4 = 3$$

Podemos representar cada solução por uma quadra  $(n_1, n_2, n_3, n_4)$ . A tabela a seguir lista todas as possibilidades.

```

PROCEDIMENTO PERM-REP( $L$ )
    Se a lista  $L$  é vazia
        Imprima a permutação  $P$ 
    Senão
         $t \leftarrow$  número de tipos distintos em  $L$ 
        Para  $j = 1$  até  $t$ 
             $s \leftarrow$  algum elemento de um tipo que ainda não foi escolhido neste
loop    (*)
            Insira  $s$  na posição  $n + 1 - |L|$  de  $P$ .
            PERM-REP( $L - s$ )
        Fim Para
    Fim Se

MAIN
    Leia  $L$ 
     $P \leftarrow$  vetor global de  $|L|$  posições.
    PERM-REP( $L$ )

```

Figura 2.4: Algoritmo de Geração de Permutações quando há repetições de símbolos

(3,0,0,0)	(2,1,0,0)	(2,0,1,0)	(2,0,0,1)
(0,3,0,0)	(1,2,0,0)	(0,2,1,0)	(0,2,0,1)
(0,0,3,0)	(1,0,2,0)	(0,1,2,0)	(0,0,2,1)
(0,0,0,3)	(1,0,0,2)	(0,1,0,2)	(0,0,1,2)
(1,1,1,0)	(1,1,0,1)	(1,0,1,1)	(0,1,1,1)

Fica então a seguinte pergunta,

*Como calcular o número de soluções sem precisar listar todas elas?*

**Técnica de palitos e asteriscos**

Como exemplo, considere a equação

$$x_1 + x_2 + x_3 = 5$$

Para codificar uma solução desta equação, utilizamos os símbolos \* e |. Os \* são utilizados para representar a soma total que deve ser obtida, enquanto que os |'s são utilizados para representar o valor das variáveis  $x_i$ .

Para a equação considerada utilizamos 5 \* e 2 |. O número de \* à esquerda do primeiro palito corresponde ao valor de  $x_1$ , o número de \*'s entre o primeiro e o segundo palito corresponde a  $x_2$ , enquanto que o número de \* à direita do segundo palito corresponde a  $x_3$ . Considere as codificações abaixo:

$$**|*|**,$$

$$|***|**$$

A primeira codificação corresponde a solução (2,1,2), enquanto que a segunda corresponde a solução (0,3,2). Podemos observar que cada permutação distinta dos \* e | corresponde a uma solução distinta da equação. Portanto, o número de soluções da equação é igual ao número de permutações de 5 \* e 2 |. Utilizando a fórmula (2.1) temos que o número de soluções é  $\frac{7!}{(5!2!)} = 21$ .

A técnica ilustrada acima permite obter o número de soluções da equação (2.2). Para isso calculamos o número de permutações de um conjunto com  $r$  \*'s e  $k - 1$  |'s.

**Teorema 10** *O número de soluções inteiras não negativas de  $\sum_{i=1}^k x_i = r$  é*

$$\frac{(r+k-1)!}{r!(k-1)!}$$

**Exemplo 35** *Um caminhão deve transportar 1000kg de alimentos, entre açúcar, arroz, feijão e sal. Sabendo que cada alimento está armazenado em sacos de 5kg, quantas são as possibilidades de completar os 1000kg?*

Utilizamos  $x_1, x_2, x_3, x_4$ , respectivamente, para denotar o número de sacos de açúcar, arroz, feijão e sal. O número de possibilidades é igual ao número de soluções inteiras não negativas da equação

$$5x_1 + 5x_2 + 5x_3 + 5x_4 = 1000,$$

que é igual ao número de soluções inteiras não negativas da equação

$$x_1 + x_2 + x_3 + x_4 = 200,$$



Utilizando o Teorema 10, obtemos o valor  $\frac{203!}{200!3!}$ .

Consideremos agora o problema de contar o número de soluções **positivas** de uma equação.

**Exemplo 36** *Quantas soluções inteiras positivas tem a equação*

$$x_1 + x_2 + x_3 + x_4 = 7$$

Para resolver este problema podemos fazer as seguintes substituições de variáveis:  $x_1 = 1 + z_1$ ,  $x_2 = 1 + z_2$ ,  $x_3 = 1 + z_3$  e  $x_4 = 1 + z_4$ , onde os  $z_i$ 's são inteiros não negativos. A idéia desta substituição é garantir que cada variável  $x$  tenha obrigatoriamente valor maior ou igual a 1.

Portanto, o número de soluções da equação do exemplo é igual ao número de soluções inteiras **não negativas** da equação

$$(z_1 + 1) + (z_2 + 1) + (z_3 + 1) + (z_4 + 1) = 7,$$

que é igual ao número de soluções inteiras **não negativas** da equação

$$z_1 + z_2 + z_3 + z_4 = 3$$

Utilizando a técnica dos  $|$  e  $*$ , obtemos  $6!/(3!3!) = 20$ .

**Exemplo 37** *Quantas soluções inteiras não negativas tem a equação*

$$x_1 + x_2 + x_3 + x_4 \leq 6 \tag{2.3}$$

Este problema pode ser resolvido de duas formas. A maneira mais imediata, dado o que foi estudado até aqui, é somar o número de soluções não negativas das equações:  $\sum_{i=1}^4 x_i = 0$ ,  $\sum_{i=1}^4 x_i = 1$ ,  $\sum_{i=1}^4 x_i = 2$ ,  $\sum_{i=1}^4 x_i = 3$ ,  $\sum_{i=1}^4 x_i = 4$ ,  $\sum_{i=1}^4 x_i = 5$ ,  $\sum_{i=1}^4 x_i = 6$ . Portanto, o total de soluções é dado por

$$\sum_{r=0}^6 \frac{(r+3)!}{r!3!} = 210$$

A segunda maneira de resolver este problema é observar que existe uma bijeção (correspondência um para um) entre o conjunto de soluções da equação (2.3) e o conjunto de soluções inteiras não negativas de

$$x_1 + x_2 + x_3 + x_4 + z = 6 \tag{2.4}$$

De fato, podemos associar a solução  $(x_1, x_2, x_3, x_4)$  da equação (2.3) a solução  $(x_1, x_2, x_3, x_4, 6 - x_1 + x_2 + x_3 + x_4)$  da equação (2.4). Como existe a correspondência um para um, o número de soluções inteiras não negativas de (2.3) é igual ao número de soluções inteiras não negativas de (2.4). Utilizando a técnica de  $*$  e  $|$ , obtemos um total de

$$\frac{10!}{6!4!} = 210$$

soluções para equação (2.4).

**Exercício 14** *Quantos números entre 1 e 1.000.000 têm a soma dos seus algarismos igual a 9?*

Primeiramente, observe que a soma dos dígitos de 1000000 não é 9. Portanto, podemos desconsiderar este número. Podemos representar todo número entre 1 e 999999 da seguinte forma  $d_1d_2d_3d_4d_5d_6$ , onde  $0 \leq d_i \leq 9$ , para  $i = 1, \dots, 6$ . Por exemplo, 23 é representado por  $d_1 = 0, d_2 = 0, d_3 = 0, d_4 = 0, d_5 = 2$  e  $d_6 = 3$ . Logo, o total de números cuja soma dos dígitos é 9 é igual ao número de soluções da equação

$$\sum_{i=1}^6 d_i = 9$$

Aplicando o Teorema 10, temos  $\binom{14}{5}$  números.

**Exercício 15** *Quantos números entre 1 e 1.000.000 têm a soma dos seus algarismos igual a 13?*

A primeira idéia que vem em mente, uma vez resolvido o exercício anterior, é calcular o número de soluções inteiras não negativas da equação

$$\sum_{i=1}^6 d_i = 13 \tag{2.5}$$

O problema é que certas soluções não correspondem a números entre 1 e 999999. Por exemplo,  $(1,0,0,0,12,0)$  é uma solução da equação, mas não corresponde a nenhum número do intervalo.

De fato, devemos calcular o número de soluções inteiras não negativas da equação (2.5) sujeito as restrições  $0 \leq d_i \leq 9$ , para  $i = 1, \dots, 6$ . Este número é igual a  $X - Y$ , onde  $X$  o número de soluções inteiras não negativas da

equação (2.5) e  $Y$  é o número de soluções inteiras não negativas da equação (2.5) que contém pelo menos um  $d_i$  com valor maior que 9.

Aplicando o Teorema 10, obtemos  $X = \binom{18}{5}$ . Para obter o valor de  $Y$ , devemos considerar as soluções da equação (2.5) em que pelo menos um das seguintes condições ocorrem:  $d_1 \geq 10$ ,  $d_2 \geq 10$ ,  $\dots$ ,  $d_6 \geq 10$ . Observe que estas condições são mutuamente exclusivas, já que não existe solução da equação com 2 variáveis maiores que 10.

Consideremos o caso em que  $d_1 \geq 10$ . Para contar o número de soluções satisfazendo tal condição façamos  $d_1 = 10 + z_1$ , com  $z_1 \geq 0$ . Logo, o número de soluções satisfazendo  $d_1 \geq 10$  é igual ao número de soluções de

$$z_1 + \sum_{i=2}^6 d_i = 3.$$

Portanto, temos  $\binom{8}{5}$  soluções em que  $d_1 \geq 10$ . Aplicando o mesmo raciocínio para os demais casos, obtemos que o total de soluções  $Y$  em que algum dígito é maior ou igual a 10 é  $6\binom{8}{5}$ . Portanto, a resposta do problema é  $\binom{18}{5} - 6\binom{8}{5}$ .

### 2.5.1 O Problema do Troco

Nesta seção abordamos o seguinte problema: dado um valor total  $T$  e um conjunto de moedas de valores positivos  $v_1, v_2, \dots, v_n$ , queremos determinar de quantas maneiras diferentes podemos somar  $T$  utilizando moedas com estes valores. Vamos assumir a existência de um número suficientemente grande de moedas de cada valor.

Por exemplo, sejam  $T = 13$ ,  $v_1 = 5$ ,  $v_2 = 3$  e  $v_3 = 1$ . Existem 10 maneiras de somar 13:  $13 = 2v_1 + v_2$ ,  $13 = 2v_1 + 3v_3$ ,  $13 = v_1 + 2v_2 + 2v_3$ ,  $13 = v_1 + v_2 + 5v_3$ ,  $13 = v_1 + 8v_3$ ,  $13 = 4v_2 + v_3$ ,  $13 = 3v_2 + 4v_3$ ,  $13 = 2v_2 + 7v_3$ ,  $13 = v_2 + 10v_3$  e  $13 = 13v_3$ .

Observe que estamos interessado em determinar o número de soluções inteiras não-negativas da equação

$$5x_1 + 3x_2 + x_3 = 13$$

No problema geral, buscamos determinar o número de soluções inteiras não-negativas da equação

$$\sum_{i=1}^n v_i x_i = T \tag{2.6}$$

Antes de mostrar como resolver o problema, abrimos um parêntese para seguinte definição:

**Definição 3** *Dado um conjunto de moedas de valores positivos  $v_1, v_2, \dots, v_n$ , defina  $S(k, A)$  como o número de maneiras de somar o valor  $A$  utilizando moedas de valores  $v_k, v_{k+1}, \dots, v_n$ .*

Em nosso exemplo, onde  $v_1 = 5$ ,  $v_2 = 3$  e  $v_3 = 1$ , temos  $S(1, 13) = 10$  e  $S(2, 8) = 3$ . **Confira !**

Observe que o problema descrito no começo da seção é o de determinar  $S(1, T)$ . Para encontrar o número de soluções da equação (2.6) (determinar  $S(1, T)$ ), primeiramente consideramos quais são as possibilidades para utilizar as moedas de valor  $v_1$ . Podemos utilizar de 0 a  $\lfloor T/v_1 \rfloor$  moedas de valor  $v_1$ , já que ao utilizar mais do que  $\lfloor T/v_1 \rfloor$ , estaremos somando mais do que  $T$ , o que não é permitido. Se utilizarmos exatamente  $p$  moedas de valor  $v_1$ , devemos então somar  $T - pv_1$  utilizando moedas cujos valores são  $v_2, v_3, \dots, v_n$ . Portanto, podemos afirmar que o número de soluções da equação (2.6) que utiliza exatamente  $p$  moedas de valor  $v_1$  é igual ao número de soluções da equação

$$\sum_{i=2}^n v_i x_i = T - pv_1, \quad (2.7)$$

ou seja,  $S(2, T - pv_1)$ .

Considerando as possibilidades de utilização de moedas de valor  $v_1$ , chegamos a seguinte equação de recorrência

$$S(1, T) = \sum_{j=0}^{\lfloor T/v_1 \rfloor} S(2, T - j \times v_1)$$

De forma análoga, temos que

$$S(k, A) = \sum_{j=0}^{\lfloor A/v_k \rfloor} S(k+1, A - j \times v_k) \quad (2.8)$$

Dada esta definição podemos escrever um programa recursivo para calcular  $S(1, T)$ , **Ok?** Entretanto, devemos nos preocupar com a parada da recursão. Com este propósito, consideremos  $k = n$ . Neste caso, devemos encontrar o número de maneiras de somar  $A$  utilizando moedas de valor  $v_n$ . Observe que se  $A$  é múltiplo de  $v_n$  temos uma possibilidade, caso contrário temos 0 possibilidades.

## Implementação

O procedimento para calcular  $S(k, A)$  é apresentado na Figura 2.5. Utilizamos um vetor global  $v$  para armazenar os valores das moedas. A variável Total serve para acumular a somatório da equação (2.8).

```
PROCEDIMENTO S(k,A)
  Se  $k = n$ 
    Se  $A$  é múltiplo de  $v[n]$  retorne 1
    Senão retorne 0
  Senão
    Total  $\leftarrow 0$ 
    Para  $j = 0$  até  $\lfloor A/v[k] \rfloor$ 
      Total  $\leftarrow$  Total +  $S(k + 1, A - jv[k])$ 
    Fim Para
    Retorne Total
  Fim Se

MAIN
  Leia  $n, v, T$ 
  IMPRIMA  $S(1, T)$ 
```

Figura 2.5: Algoritmo para o Problema do Troco

## 2.6 Princípio da Inclusão e Exclusão

Sejam os conjuntos finitos  $A_1, A_2, \dots, A_k$ . Nesta seção consideramos o problema de calcular

$$|A_1 \cup A_2 \cup \dots \cup A_k|$$

Se todos os conjuntos são disjuntos entre si, ou seja, não tem elementos em comum, temos

$$|A_1 \cup A_2 \cup \dots \cup A_k| = |A_1| + |A_2| + \dots + |A_k|$$

Entretanto, quando os conjuntos não são disjuntos, o problema fica mais complicado. Utilizando o diagrama de Venn para dois conjuntos  $A_1$  e  $A_2$ , obtemos a relação

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$$

Para três conjuntos obtemos a seguinte relação

$$|A_1 \cup A_2 \cup A_3| = |A_1| + |A_2| + |A_3| - |A_1 \cap A_2| - |A_1 \cap A_3| - |A_2 \cap A_3| + |A_1 \cap A_2 \cap A_3| \quad (2.9)$$

**Exemplo 38** *Uma quitanda vende apenas brocolis, cenoura e quiabo. Em um determinado dia a quitanda atende a 208 pessoas. Se 114 compram brocolis, 152 compram cenoura, 17 compram apenas quiabos, 64 compram brocolis e cenoura, 12 compram cenouras e quiabos e 9 compram os três. Quantas compram brocolis e quiabo.*

Sejam  $Q = \{ \text{pessoas que compram quiabo} \}$ ,  $B = \{ \text{pessoas que compram brócolis} \}$  e  $C = \{ \text{pessoas que compram cenoura} \}$ . Aplicando a fórmula (2.9) temos

$$108 = 114 + 152 + 17 - 64 - 12 - |B \cup Q| + 9$$

Resolvendo, obtemos que  $|B \cup Q| = 8$ .

**Exemplo 39** *No elevador de um edifício entram 6 pessoas. De quantas maneiras essas pessoas podem saltar nos andares 2, 3 e 4, de modo que pelo menos uma pessoa salte em cada andar?*

Seja  $S$  o conjunto de todas as possibilidades das pessoas saltarem nos andares 2,3 e 4. Seja  $T$  o conjunto das possibilidades das pessoas saltarem em no máximo dois andares entre os andares 2,3 e 4. O problema pede para calcular  $|S - T| = |S| - |T|$ . Temos que  $|S| = 3^6$  já que cada pessoa tem 3 possibilidades de sair do elevador. Falta, portanto, calcular  $|T|$ .

Sejam,

$$T_2 = \{ \text{possibilidades em que nenhuma pessoa salta no andar 2} \}$$

$$T_3 = \{ \text{possibilidades em que nenhuma pessoa salta no andar 3} \}$$

$T_4 = \{ \text{possibilidades em que nenhuma pessoa salta no andar 4} \}$

Temos que  $T = T_2 \cup T_3 \cup T_4$ . Entretanto,  $|T_2| = |T_3| = |T_4| = 2^6$  e  $|T_2 \cap T_3| = |T_2 \cap T_4| = |T_3 \cap T_4| = 1$  e  $|T_2 \cap T_3 \cap T_4| = 0$ . Portanto, utilizando a fórmula obtemos que  $|T| = 3 \times 2^6 - 3$  e logo,  $|S| = 3^6 - 3 \times 2^6 + 3$ .

A equação (2.9) pode ser generalizada para  $n$  conjuntos.

**Teorema 11** *Dados  $n$  conjuntos  $A_1, \dots, A_n$ , temos*

$$\left| \bigcup_{i=1}^n A_i \right| = \sum_{I \subseteq \{1, \dots, n\}} (-1)^{|I|+1} \left| \bigcap_{j \in I} A_j \right| = \sum_{i=1}^n |A_i| - \sum_{1 \leq i_1 < i_2 \leq n} |A_{i_1} \cap A_{i_2}| + \dots$$

$$+ (-1)^{k+1} \sum_{1 \leq i_1 < \dots < i_k \leq n} |A_{i_1} \cap \dots \cap A_{i_k}| + \dots + (-1)^{n+1} |A_1 \cap A_2 \cap \dots \cap A_n|$$

**Exercício 16** *Prove o Teorema 11 utilizando indução.*

**Exemplo 40** *Um número  $n$  entre 1 e 210 é dito bom se ele satisfaz pelo menos umas das condições abaixo:*

- (i)  $n$  é múltiplo de 2
- (ii)  $n$  é múltiplo de 3
- (iii)  $n$  é múltiplo de 5
- (iv)  $n$  é múltiplo de 7

*Quantos números entre 1 e 210 são bons?*

Sejam os conjuntos:

$$A_1 = \{ \text{múltiplos de 2 entre 1 e 210} \}$$

$$A_2 = \{ \text{múltiplos de 3 entre 1 e 210} \}$$

$$A_3 = \{ \text{múltiplos de 5 entre 1 e 210} \}$$

$$A_4 = \{ \text{múltiplos de 7 entre 1 e 210} \}$$

Devemos calcular  $|A_1 \cup A_2 \cup A_3 \cup A_4|$ . Utilizando o Teorema 11, obtemos

$$|A_1 \cup A_2 \cup A_3 \cup A_4| = \sum_{i=1}^4 |A_i| - \sum_{1 \leq i < j \leq 4} |A_i \cap A_j| +$$

$$\sum_{1 \leq i < j < l \leq 4} |A_i \cap A_j \cap A_l| - |A_1 \cap A_2 \cap A_3 \cap A_4|$$

Para o cálculo das interseções 2 a 2, 3 a 3 e 4 a 4, utilizamos o fato de que um dado número  $c$  é múltiplo de dois números  $a$  e  $b$  sss é múltiplo do mínimo múltiplo comum de  $a$  e  $b$  ( $MMC(a, b)$ ). Como exemplo, a quantidade de números que são múltiplos de 2, 5 e 7 simultaneamente entre 1 e 210 ( $|A_1 \cap A_3 \cap A_4|$ ) é igual ao número de múltiplos de  $2 \times 5 \times 7$  entre 1 e 210<sup>2</sup>.

Portanto, temos que  $|A_1| = 210/2 = 105$ ,  $|A_2| = 210/3 = 70$ ,  $|A_3| = 210/5 = 42$  e  $|A_4| = 210/7 = 30$ . Além disso,  $|A_1 \cap A_2| = 210/6 = 35$ ,  $|A_1 \cap A_3| = 210/10 = 21$ ,  $|A_1 \cap A_4| = 210/14 = 15$ ,  $|A_2 \cap A_3| = 210/15 = 14$ ,  $|A_2 \cap A_4| = 210/21 = 10$  e  $|A_3 \cap A_4| = 210/35 = 6$ . Temos também que  $|A_1 \cap A_2 \cap A_3| = 210/30 = 7$ ,  $|A_1 \cap A_2 \cap A_4| = 210/42 = 5$ ,  $|A_1 \cap A_3 \cap A_4| = 210/70 = 3$  e  $|A_2 \cap A_3 \cap A_4| = 210/105 = 2$ . Finalmente,  $|A_1 \cap A_2 \cap A_3 \cap A_4| = 210/210 = 1$ . Substituindo estes valores obtemos um total de 162 múltiplos.

**Exemplo 41** *De quantas maneiras podemos distribuir 3 moedas de 25 centavos, 5 moedas de 50 centavos e 4 moedas de um real entre dois meninos de modo que cada um dos meninos ganhe pelo menos uma moeda?*

**Exemplo 42** *De quantas maneiras podemos distribuir 8 uvas, 10 peras e 7 laranjas entre 4 caixas se cada caixa deve receber pelo menos uma fruta?*

Seja  $S = \{$  possibilidades de distribuir as frutas entre as caixas permitindo que caixas fiquem vazias $\}$ . Note que o número de maneiras de distribuir as uvas entre as 4 caixas é igual ao número de soluções inteiras não-negativas da equação  $x_1 + x_2 + x_3 + x_4 = 8$ . Portanto, o número de maneiras de distribuir as uvas é  $\binom{11}{3}$ . Utilizando o mesmo raciocínio concluímos que o número de maneiras de dispor as 10 peras é  $\binom{13}{3}$  e o número de maneiras de dispor as laranjas é  $\binom{10}{3}$ . Portanto,  $|S| = \binom{11}{3} \binom{13}{3} \binom{10}{3}$ .

Seja  $T = \{$  possibilidades de distribuir as frutas entre as caixas de modo que alguma caixa fique vazia $\}$ . Para resolver o problema devemos calcular  $|T|$  e diminuir de  $|S|$ .

Para  $i = 1, 2, 3, 4$ , seja  $T_i = \{$  possibilidades de distribuir as frutas entre as caixas permitindo que a caixa  $i$  fique vazia $\}$ . Temos que  $T = T_1 \cup T_2 \cup T_3 \cup T_4$ . Portanto, podemos utilizar a fórmula dada pelo Teorema 11.

Note que  $|T_1|$  é o número de maneiras de colocar as frutas nas caixas 2, 3 e 4. Utilizando o mesmo raciocínio que foi feito para calcular  $|S|$  concluímos que  $|T_1| = \binom{10}{2} \binom{12}{2} \binom{9}{2}$ . Além disso, por simetria, temos que  $|T_1| = |T_2| = |T_3| = |T_4|$ .

<sup>2</sup>Observe que nem sempre o MMC é o produto dos números



Note que  $|T_1 \cap T_2|$  é o número de maneiras de colocar as frutas nas caixas 3 e 4. Utilizando o mesmo raciocínio que foi feito para calcular  $|S|$  concluímos que  $|T_1 \cap T_2| = \binom{9}{1} \binom{11}{1} \binom{10}{1}$ . Além disso, temos que  $|T_1 \cap T_2| = |T_i \cap T_j|$  para  $1 \leq i < j \leq 4$ .

Finalmente, temos que  $|T_1 \cap T_2 \cap T_3| = |T_1 \cap T_2 \cap T_4| = |T_1 \cap T_3 \cap T_4| = |T_2 \cap T_3 \cap T_4| = 1$  e  $|T_1 \cap T_2 \cap T_3 \cap T_4| = 0$ .

Portanto, obtemos que  $|T| = 4 \binom{10}{2} \binom{12}{2} \binom{9}{2} - 6 \binom{9}{1} \binom{11}{1} \binom{10}{1} + 4$ , e a resposta desejada é

$$|S| - |T| = \binom{11}{3} \binom{13}{3} \binom{10}{3} - 4 \binom{10}{2} \binom{12}{2} \binom{9}{2} + 6 \binom{9}{1} \binom{11}{1} \binom{10}{1} - 4$$

**Exemplo 43** *Considere um amigo oculto com 6 pessoas. Qual a probabilidade de pelo menos uma pessoa ter como amigo oculto ela própria?*

Podemos numerar os participantes de 1 a 6. Queremos determinar a probabilidade de que pelo menos um participante  $i$  tenha como amigo oculto ele próprio. Cada possibilidade de sorteio corresponde a um permutação dos números de 1 a 6.

Por exemplo, a permutação 243165 corresponde a possibilidade em que o participante 1 tem como amigo oculto o participante 2, o participante 2 tem como amigo oculto o participante 4, o participante 3 tem como amigo oculto ele próprio, etc ...

Queremos então determinar o número de permutações em que o elemento  $i$  aparece na posição  $i$  para pelo menos um valor de  $i$ . Definimos então os seguintes conjuntos

$$\begin{aligned} A_1 &= \{ \text{permutações em que o elemento 1 aparece na posição 1} \} \\ A_2 &= \{ \text{permutações em que o elemento 2 aparece na posição 2} \} \\ A_3 &= \{ \text{permutações em que o elemento 3 aparece na posição 3} \} \\ A_4 &= \{ \text{permutações em que o elemento 4 aparece na posição 4} \} \\ A_5 &= \{ \text{permutações em que o elemento 5 aparece na posição 5} \} \\ A_6 &= \{ \text{permutações em que o elemento 6 aparece na posição 6} \} \end{aligned}$$

Queremos calcular

$$|A_1 \cup A_2 \cup A_3 \cup A_4 \cup A_5 \cup A_6|.$$

Para aplicar o Teorema 11 devemos conhecer o número de elementos da interseção entre  $k$  conjuntos, para  $k = 1, \dots, 6$ . Como exemplo, sejam os conjuntos  $A_1, A_2, \dots, A_k$ . Para uma permutação pertencer a interseção destes conjuntos devemos ter o elemento 1 na posição 1, 2 na posição 2,

...,  $k$  na posição  $k$ , ficando os demais elementos,  $k + 1, k + 2, \dots, n$ , livres para serem permutados. Portanto, o número de elementos da interseção de  $A_1 \cap A_2 \cap \dots \cap A_k$  é  $(n - k)!$ . Este raciocínio vale para quaisquer  $k$  conjuntos. Logo,

$$|A_i| = 5!,$$

para  $1 \leq i \leq 6$ ,

$$|A_i \cap A_j| = 4!,$$

para  $1 \leq i < j \leq 6$ .

$$|A_i \cap A_j \cap A_k| = 3!,$$

para  $1 \leq i < j < k \leq 6$ ,

$$|A_i \cap A_j \cap A_k \cap A_l| = 2!,$$

para  $1 \leq i < j < k < l \leq 6$ ,

$$|A_i \cap A_j \cap A_k \cap A_l \cap A_m| = 1!,$$

para  $1 \leq i < j < k < l < m \leq 6$  e

$$|A_1 \cap A_2 \cap A_3 \cap A_4 \cap A_5 \cap A_6| = 1$$

Portanto, temos

$$|A_1 \cup A_2 \cup A_3 \cup A_4 \cup A_5 \cup A_6| = \sum_{i=1}^6 \binom{6}{i} (n-i)! = \binom{6}{1} 5! - \binom{6}{2} 4! + \binom{6}{3} 3! - \binom{6}{4} 2! + \binom{6}{5} 1! - \binom{6}{6} 0! = 455$$

Na expressão acima, o termo  $\binom{6}{i}$ , para  $i = 1, \dots, 6$ , conta o número de possibilidades de escolher  $i$  conjuntos. Portanto, a probabilidade desejada é  $455/6! = 91/124$

**Exercício 17** *No caso de  $n$  pessoas, como fica o problema do amigo oculto? O que acontece com a probabilidade quando  $n \rightarrow \infty$ ?*

**Exercício 18** *Quantos números entre 1 e 1.000.000 têm a soma dos seus algarismos igual a 23?*

## 2.7 Princípio da Casa dos Pombos

Nesta seção apresentamos o princípio da casa dos pombos, uma idéia bastante simples que permite a resolução de problemas combinatórios bastante complexos. O princípio na sua forma mais simples diz que

*Se existem  $k + 1$  pombos e  $k$  casas de pombos então pelo menos dois pombos devem ocupar a mesma casa.*

**Exemplo 44** *Quantas pessoas devem estar na mesma festa para garantirmos que duas pessoas tem nomes que começam com a mesma letra.*

Podemos pensar que cada uma das 26 letras do alfabeto corresponde a uma casa de pombo e que uma pessoa ocupa uma casa se e somente se a letra inicial do seu nome corresponde a casa. Portanto se houverem 27 pessoas podemos garantir que duas delas estarão associadas a mesma casa e como consequência seus nomes começarão com a mesma letra.

**Exercício 19** *Mostre que em uma festa com  $n$  pessoas, existem pelo menos duas que conhecem o mesmo número de pessoas (não necessariamente as mesmas pessoas). Assuma que*

- (i) *a relação de conhecer é simétrica;*
- (ii) *Não há penetras na festa, ou seja, toda pessoa na festa conhece pelo menos outra pessoa.*

Considere  $n - 1$  casas:  $1, 2, \dots, n - 1$ . Uma pessoa é associada a casa  $i$  se e somente se conhece exatamente  $i$  pessoas. Observe que cada pessoa da festa conhece pelo menos uma pessoa (item (ii)) e no máximo  $n - 1$  pessoas. Portanto, cada pessoa esta associada a uma das  $n - 1$  casas. Como existem  $n$  pessoas e  $n - 1$  casas, segue do princípio da casa dos pombos que pelo menos duas pessoas estão associadas a mesma casa, ou seja, conhecem o mesmo número de pessoas.

**Exercício 20** *O resultado anterior continua válido sem assumir (ii)? Prove ou dê um contra-exemplo*

**Exercício 21** *Prove que se 4 números são escolhidos do conjunto  $\{1, 2, 3, 4, 5, 6\}$ , então a soma de 2 deles é 7.*

Considere as duplas  $\{1, 6\}$ ,  $\{2, 5\}$ ,  $\{3, 4\}$ . Note que a soma dos dois elementos de uma mesma dupla é 7. Entretanto, o princípio da casa dos pombos garante que ao escolhermos 4 elementos, dois deles estarão na mesma dupla. Portanto, estes somarão 7.

**Exercício 22** *Mostre que sempre que colocamos 5 pontos dentro de um quadrado de lado 1cm, pelo menos dois pontos distam menos do que 0.8 cm.*

### 2.7.1 Princípio Generalizado

Apresentamos agora o Princípio da casa dos pombos na sua versão generalizada.

*Se existem  $n$  pombos e  $k$  casas de pombos, então existe uma casa com pelo menos  $\lceil n/k \rceil$  pombos.*

**Exemplo 45** *Qual é o número mínimo de pessoas que devemos ter para garantir que quatro delas nasceram no mesmo dia da semana*

Temos 7 casas, uma para cada dia da semana. Portanto, queremos encontrar o menor  $n$  tal que  $\lceil n/7 \rceil = 4$ . Resolvendo a equação encontramos  $n = 22$ .

**Exercício 23** *Considere 6 pontos no plano 3 a 3 não colineares. Mostre que se pintarmos cada segmento de reta utilizando branco ou azul, então sempre existirá um triângulo monocromático.*

**Exercício 24** *Considere o seguinte jogo. João deve escolher 10 inteiros entre 1 e 40. O objetivo de Maria é encontrar 2 conjuntos distintos (não necessariamente disjuntos) de três inteiros, dentre os 10 escolhidos por João, tais que a soma dos elementos do primeiro conjunto é igual a soma dos elementos do segundo conjunto. Maria sempre pode alcançar seu objetivo?*

Seja  $S$  o conjunto de 10 números escolhidos por João. Sejam  $x, y, z \in S$ . Temos que,  $6 = 1 + 2 + 3 \leq x + y + z \leq 40 + 39 + 38 = 117$ . A conclusão é que a soma de três números quaisquer de  $S$  só pode assumir  $117 - 6 + 1 = 112$  valores distintos. Como existem  $\binom{10}{3} = 120$  triplas distintas em  $S$ , concluímos que duas delas devem somar o mesmo valor. Portanto, Maria sempre pode alcançar seu objetivo.

# Capítulo 3

## Teoria de Grafos

Nesta parte do curso estudaremos uma estrutura matemática denominada grafos. Os grafos são utilizados na modelagem de diversos processos e problemas que aparecem em ciência da computação.

### 3.1 Conceitos Básicos

Inicialmente, estudaremos os grafos não direcionados. Um grafo é uma tripla ordenada  $(V(G), E(G), \psi_G)$ , onde

- $V(G)$  é um conjunto não vazio de vértices
- $E(G)$  é um conjunto de arestas (possivelmente vazio)
- $\psi_G$  é uma função que associa cada aresta de  $G$  a um par não ordenado de vértices de  $G$

**Exemplo 46** *Represente graficamente o grafo  $(V(G), E(G), \psi_G)$ , onde  $V(G) = \{a, b, c, d\}$ ,  $E(G) = \{e_1, e_2, e_3, e_4, e_5, e_6\}$  e  $\psi_G(e_1) = ab$ ,  $\psi_G(e_2) = ab$ ,  $\psi_G(e_3) = bb$ ,  $\psi_G(e_4) = bc$ ,  $\psi_G(e_5) = ac$ ,  $\psi_G(e_6) = cd$ ,*

É importante ressaltar que existem inúmeras possibilidades de representar um grafo no plano.

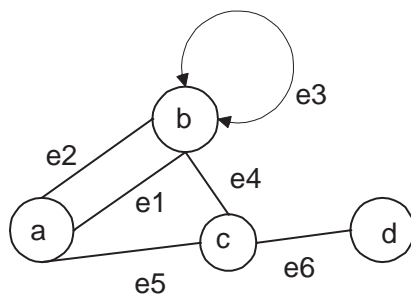


Figura 3.1:

**Definição 4 (extremidades)** *Se  $e$  é uma aresta de  $G$  e  $\psi_G(e) = uv$ , então  $u$  e  $v$  são extremidades de  $e$ .*

No grafo da figura 3.1  $a$  e  $b$  são as extremidades de  $e_1$  e  $e_2$ .

**Exemplo 47** *Podemos representar os estados do Sudeste através de um grafo  $G = (V, E)$ , onde  $V = \{ES, MG, SP, RJ\}$  e  $E = \{(RJ, SP), (RJ, MG), (RJ, ES), (MG, ES), (MG, SP)\}$ . Existe uma aresta entre dois vértices se e somente se os estados correspondentes tem fronteira geográfica.*

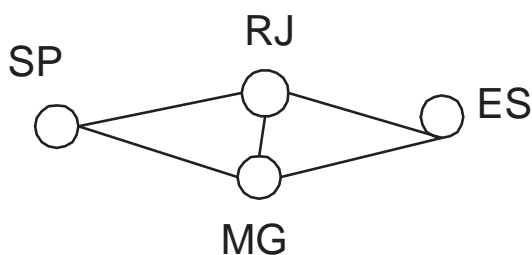


Figura 3.2: Estados do Sudeste

**Definição 5 (vértices adjacentes)** *Se dois vértices são extremidades de uma mesma aresta, eles são ditos adjacentes.<sup>1</sup>*

<sup>1</sup>Em algumas referências utiliza-se *vizinho*

Na figura 3.1  $c$  e  $b$  são adjacentes, enquanto que  $a$  e  $d$  não são.

**Definição 6 (laço)** *aresta cujas duas extremidades são idênticas*

Na figura 3.1,  $e_3$  é um laço.

**Definição 7 (arestas paralelas)** *duas arestas distintas com extremidades idênticas.*

Na figura 3.1  $e_1$  e  $e_2$  são arestas paralelas.

**Definição 8 (grafo simples)** *Um grafo é dito simples se não contém arestas paralelas nem laços.*

O grafo da figura 3.1 não é simples. O grafo mostrado abaixo na figura 3.3 é simples. Como o vértice  $e$  não é vizinho a nenhum outro do grafo, chamamos  $e$  de *vértice isolado*.

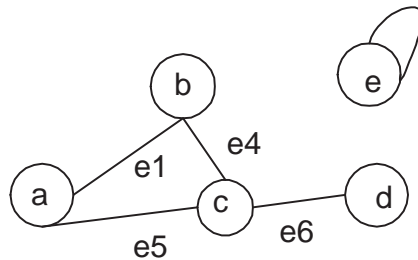


Figura 3.3: Grafo simples com um vértice isolado

Muito dos resultados e dos algoritmos apresentados ao longo do curso dizem respeito a grafos simples. Em se tratando de grafos simples, uma aresta fica identificada de forma única dado dois vértices.

**Definição 9 (Grau de um vértice)** *O grau de um vértice  $v$  é o número de vezes que  $v$  é extremidade de uma aresta.*

Usualmente, utilizamos  $d(v)$  para denotar o grau de um vértice  $v$ . No grafo da figura 3.3,  $d(a) = 2$ ,  $d(c) = 3$  e  $d(e) = 2$  (nesse vértice existe um laço).

### 3.1.1 Operações em Grafos

**Definição 10 (Remoção de Vértices)** Quando se remove um vértice  $v$ , remove-se **todas** as arestas que tem  $v$  como extremidade.

Usualmente utilizamos  $G - \{v\}$  para denotar o grafo obtido a partir do grafo  $G$  pela remoção de um vértice  $v \in V(G)$ .

**Definição 11 (Remoção de Arestas)** Quando se remove uma aresta  $e$  de  $G = (V, E)$ , obtém-se o grafo  $G' = (V, E')$ , onde  $E' = E - \{e\}$ .

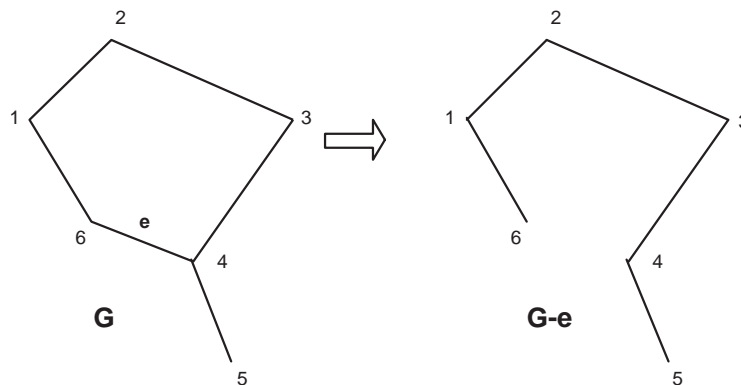


Figura 3.4: O grafo à direita foi obtido a partir do grafo à esquerda pela remoção da aresta  $e$

O teorema a seguir relaciona a soma dos graus dos vértices em um grafo com o número de arestas.

**Teorema 12** Seja  $G$  um grafo. Então,

$$\sum_{v \in V(G)} d(v) = 2|E(G)|$$

Utilizamos indução no número de arestas do grafo.

Hipótese:) O resultado vale para todo grafo com  $n$  arestas.

Base:)  $k = 0$ . Em um grafo com 0 arestas, todo vértice tem grau 0. Portanto, a hipótese é válida.



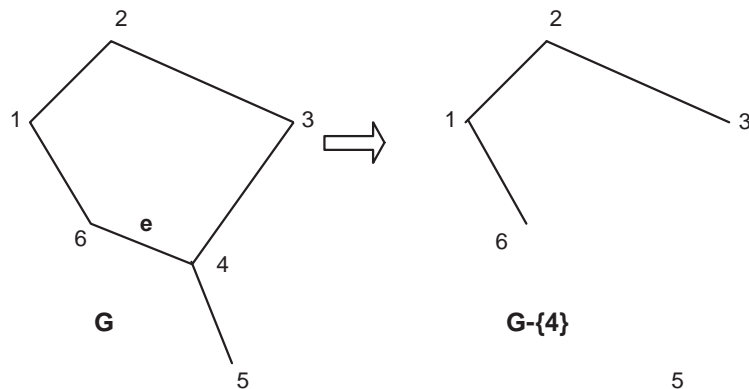


Figura 3.5: O grafo à direita foi obtido a partir do grafo da esquerda pela remoção do vértice 4

Passo Indutivo:) Seja um grafo  $G$  com  $k+1$  arestas. Retirando uma aresta deste grafo, obtemos um grafo  $G'$  com  $k$  arestas. Pela hipótese indutiva,

$$\sum_{v \in V(G')} d(v) = 2|E(G')| \tag{3.1}$$

Como uma aresta foi retirada, temos que

$$|E(G)| = |E(G')| + 1, \tag{3.2}$$

Além disso,

$$\sum_{v \in V(G)} d(v) = 2 + \sum_{v \in V(G')} d(v), \tag{3.3}$$

já que a aresta retirada contribui em 2 para soma dos graus dos vértices de  $G$ . Combinando (3.1), (3.2) e (3.3) temos

$$\sum_{v \in V(G)} d(v) = 2 + \sum_{v \in V(G')} d(v) = 2(1 + |E(G')|) = 2|E(G)|,$$

o que completa a prova ■

Uma consequência imediata do resultado anterior é o seguinte corolário

**Corolário 2** *Em todo grafo, o número de vértices com grau ímpar é par.*

Seja  $G$  um grafo. Particionamos os vértices de  $G$  da seguinte forma.

$V_1$  : vértices de  $G$  com grau ímpar

$V_2$  : vértices de  $G$  com grau par

Temos que

$$\sum_{v \in V(G)} d(v) = \sum_{v \in V_1} d(v) + \sum_{v \in V_2} d(v)$$

O teorema anterior garante que

$$\sum_{v \in V(G)} d(v)$$

é um número par. Além disso,

$$\sum_{v \in V_2} d(v)$$

é par, já que todos os vértices de  $V_2$  tem grau par. Logo, concluímos que  $\sum_{v \in V_1} d(v)$  é par.

Como todo vértice em  $V_1$  tem grau ímpar, logo o número de vértices em  $V_1$  deve ser par, caso contrário  $\sum_{v \in V_1} d(v)$  não seria par ■

### 3.1.2 Isomorfismo de Grafos

**Definição 12 (Grafos Idênticos)** *Dois grafos  $G$  e  $H$  são idênticos se e somente se  $V(G) = V(H)$ ,  $E(G) = E(H)$  e  $\psi_G = \psi_H$*

**Definição 13 (Grafos Isomorfos)** *Dois grafos  $G$  e  $H$  são isomorfos se e somente se existem bijeções  $\theta : V(G) \rightarrow V(H)$  e  $\phi : E(G) \rightarrow E(H)$  tal que  $\psi_G(e) = uv \iff \psi_H(\phi(e)) = \theta(u)\theta(v)$ . O par  $(\theta, \phi)$  é chamado de isomorfismo ente  $G$  e  $H$*

A idéia desta definição é que grafos isomorfos são idênticos, salvo seus rótulos. Ou seja, se  $G$  e  $H$  são isomorfos, é possível obter um grafo  $H'$  idêntico a  $G$  renomeando os vértices e as arestas de  $H$  de maneira apropriada.

Como consequência da definição de grafos isomorfos, podemos afirmar que dois grafos simples são isomorfos caso exista uma bijeção  $\theta : V(G) \rightarrow V(H)$  tal que  $u$  e  $v$  são adjacentes em  $G$  se e somente se  $\theta(u)$  e  $\theta(v)$  são adjacentes em  $H$ .

**Exercício 25** *Mostrar que se  $G$  e  $H$  são isomorfos então  $d(v) = d(\theta(v))$ , para todo  $V \in v(G)$ .*

**Exemplo 48** *Encontre um isomorfismo entre os grafos da figura 3.6 e entre os grafos da figura 3.7.*

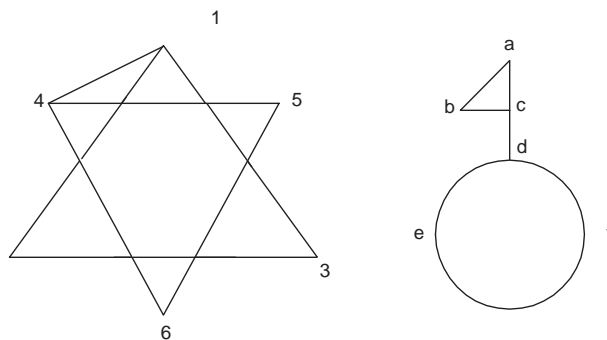


Figura 3.6:

Para a figura 3.6 temos:  $\theta(1) = c$ ,  $\theta(4) = d$ ,  $\theta(2) = a$ ,  $\theta(3) = b$ ,  $\theta(5) = e$ ,  $\theta(6) = f$ . Observe que como o Grafo é simples,  $\phi$  está automaticamente definida

Para a figura 3.7 temos  $\theta(1) = a$ ,  $\theta(2) = d$ ,  $\theta(3) = b$ ,  $\theta(4) = e$ ,  $\theta(5) = f$ .

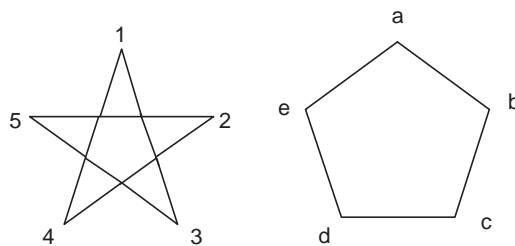


Figura 3.7:

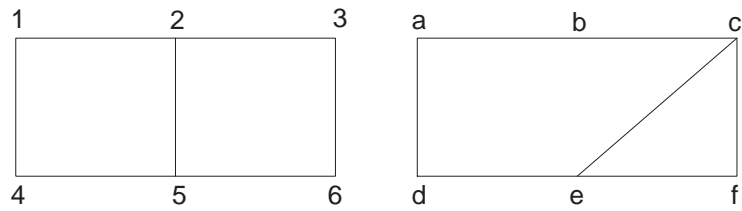


Figura 3.8:

**Exemplo 49** *Mostre que os grafos da figura 3.8 não são isomorfos.*

Para ser isomorfo, temos duas possibilidades:

- Caso 1  $\theta(2) = c$  ;  $\theta(5) = e$
- Caso 2  $\theta(2) = e$  ;  $\theta(5) = c$

Consideremos o caso 1. Seja  $x$  tal que  $\theta(x) = f$ . Logo,  $x$  tem que ser adjacente a 2 e 5, o que é impossível ! O outro caso é análogo.

### 3.1.3 Subgrafos

**Definição 14 (Subgrafo)** *Um grafo  $H$  é um subgrafo de  $G$  ( $H \subseteq G$ ) se  $V(H) \subseteq V(G)$ ,  $E(H) \subseteq E(G)$  e  $\psi_H$  é a restrição de  $\psi_G$  a  $E(H)$  (aplica-se  $\psi_G$  somente as arestas de  $H$ )*

**Definição 15 (Supergrafo)** *Se  $H$  é um subgrafo de  $G$ , então  $G$  é um supergrafo de  $H$ .*

**Definição 16 (Subgrafo Gerador)**  *$H$  é dito um subgrafo gerador de  $G$  se e somente se  $H \subset G$  e  $V(H) = V(G)$*

Seja  $G$  o grafo da figura 3.9. O grafo  $G' = (V', E')$ , onde  $V = \{a, b, c, d, e\}$  e  $E' = \{e_1, e_2, e_7\}$ , é gerador de  $G$ .

**Exercício 26** *Quantos subgrafos geradores  $G$  possui?*

**Definição 17 (Subgrafo Induzido)** *Seja um grafo  $G$  e seja  $V' \subseteq V(G)$ . O subgrafo de  $G$  com conjunto de vértices  $V'$  e conjunto de arestas  $E'$ , onde  $E' = \{uv \mid u, v \in V' \text{ e } uv \in E(G)\}$  é denominado de subgrafo de  $G$  induzido por  $V'$  ( $G[V']$ ).*

**Exemplo 50** *Seja  $G = (V, E)$ , onde  $V = \{a, b, c, d, e, f\}$  e  $E = \{ac, ae, bd, ce, cf\}$ . Sendo  $V' = \{a, c, e, d\}$ , temos que as aresta de  $G[V']$  são  $\{ac, ae, ce\}$ .*

### 3.1.4 Passeios em Grafos

Em várias aplicações modeladas por grafos, é necessário decidir se é possível alcançar um vértice  $u$  partindo de um vértice  $v$ , ou determinar o menor caminho entre dois vértices do grafo, ou ainda determinar se existe um ciclo que passe por todos os vértices do grafo. Um exemplo típico onde estas questões aparecem é o problema de rotear pacotes na Internet. Nesta seção, definimos conceitos que são fundamentais na resolução de tais problemas.

**Definição 18 (Passeio)** *Um passeio (walk) em um grafo  $G$  é uma sequência não nula  $W = v_0e_1v_1e_2 \dots e_kv_k$ , tal que para  $i = 1, \dots, k$ , as extremidades de  $e_i$  são  $v_{i-1}$  e  $v_i$ . O inteiro  $k$  é o comprimento do passeio  $W$ .*

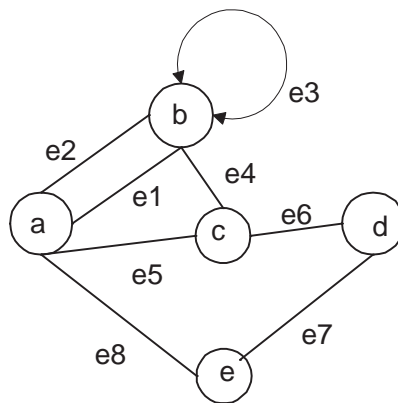


Figura 3.9:

Em um grafo simples, não é necessário indicar as arestas de  $W$ , já que um par de vértices identifica de forma única uma aresta.

**Definição 19 (Trajeto)** *Um passeio onde todas as arestas são distintas.*

**Definição 20 (Caminho)** *Um passeio onde todos os vértices são distintos.*

Na Figura 3.9  $be_3be_1ae_5ce_4be_2a$  é um trajeto, mas não é um caminho, já que o vértice  $b$  aparece mais de uma vez. Por outro lado,  $ce_4be_2a$  é um caminho de comprimento 2.

**Definição 21 (Passeio Fechado)** *Um passeio com comprimento  $k > 0$ , onde  $v_0 = v_k$ .*

Na Figura 3.9  $ae_5ce_6de_7ee_8a$  é um passeio fechado.

**Definição 22 (Trajeto Fechado)** *Passeio fechado que é um trajeto*

**Definição 23 (Ciclo)** *Trajeto fechado em que os únicos vértices idênticos são o primeiro e o último.*

Na Figura 3.9  $ae_5ce_6de_7ee_8a$  é um ciclo.

### 3.1.5 Conexidade

**Definição 24 (Vértices Conexos)** *Dois vértices  $u$  e  $v$  são conexos em  $G$  se e somente se existe um caminho entre  $u$  e  $v$  em  $G$ .*

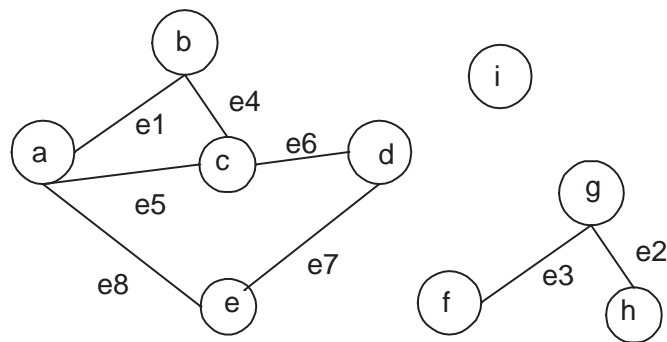


Figura 3.10:

Na Figura 3.10,  $e$  e  $b$  são conexos enquanto que  $g$  e  $a$  não são.

**Definição 25 (Conjuntos Conexos)** *Um conjunto  $S \subset V$  é dito conexo se e somente se  $\forall u, v \in S$ ,  $u$  e  $v$  são conexos*

Na Figura 3.10, o conjunto  $S = \{f, g, h\}$  é conexo enquanto que o conjunto  $S' = \{f, g, h, a\}$  não é.

**Definição 26 (Grafo Conexos)** *Um grafo  $G$  é conexo se e somente se  $V(G)$  é conexo.*

Muitos dos algoritmos desenvolvidos para resolver problemas em grafos assumem como entrada um grafo conexo. Portanto, quando o grafo não é conexo, é comum pré-processar o grafo de modo a obter seus “maiores” conjuntos conexos, ou de modo mais formal, suas componentes conexas.

**Definição 27 (Componentes Conexas)** *Seja  $S$  um subconjunto dos vértices de um grafo  $G$ . Dizemos que  $S$  é uma componente conexa de  $G$  se e somente se*

- (i) *o conjunto  $S$  é conexo*
- (ii)  *$\forall S'$ , com  $S \subseteq S'$ ,  $S'$  não é conexo.*

O item (ii) força que cada componente conexa seja a maior possível, no sentido de que se um vértice for adicionado a componente, então o novo conjunto obtido não será mais conexo.

O grafo da Figura 3.10 tem três componentes conexas. Quais são?

**Definição 28 (Distância)** *A distância entre dois vértices  $u$  e  $v$  ( $d(u, v)$ ) em um grafo  $G$  é o comprimento do menor caminho entre  $u$  e  $v$  em  $G$ .*

A distância entre  $a$  e  $d$  na Figura 3.10 é 2.

## 3.2 Alguns Grafos Importantes

Consideramos agora alguns grafos com propriedades especiais.

**Definição 29 (Grafo Completo)** *Um grafo simples em que todo par de vértices está ligado por uma aresta é dito grafo completo.*

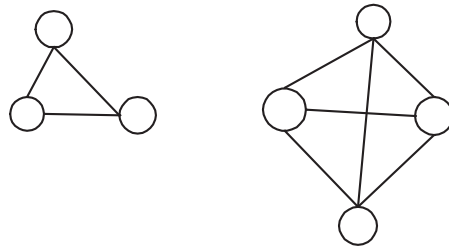


Figura 3.11: Grafos completos  $K_3$  e  $K_4$

Usualmente, utilizamos  $K_n$  para denotar um grafo completo com  $n$  vértices.

**Exemplo 51** *Quantas arestas tem  $K_n$ ?*

*Para cada par de vértices deve existir uma aresta. Como o total de vértices é  $n$ , o total de pares é  $\binom{n}{2} = n(n-1)/2$*

Uma outra classe importante de grafos é a dos grafos bipartidos. Grafos desta classe são muito comuns em aplicações que envolvem alocação de recursos.

**Definição 30 (Grafo Bipartido)** *Um grafo  $G$  é dito bipartido se somente se seus vértices podem ser particionados em dois conjuntos  $X$  e  $Y$  tal que*

- (i)  $X \cap Y = \emptyset$
- (ii)  $X \cup Y = V(G)$
- (iii) *toda aresta  $e \in E(G)$  tem uma extremidade em  $X$  e outra em  $Y$*

**Exemplo 52** *O grafo  $G = (V, E)$ , onde  $V = \{a, b, c, d\}$  e  $E = \{(a, b), (a, c), (c, d), (b, d)\}$  é bipartido já que podemos particionar  $V$  em  $(X, Y)$ , onde  $X = \{a, d\}$  e  $Y = \{b, c\}$ .*

**Exemplo 53** *O grafo  $G = (V, E)$ , com  $V = \{a, b, c\}$  e  $E = \{(a, b), (a, c), (b, c)\}$  não é bipartido. Assuma sem perda de generalidade que  $a \in X$ . Neste caso,  $b \in Y$ . Logo, não podemos colocar  $c$  em nenhum dos conjuntos  $X$  e  $Y$ , o que mostra que  $G$  não é bipartido.*



**Definição 31 (Grafo Bipartido Completo)** *Um grafo simples  $G$  com bipartição  $(X, Y)$  é completo se e somente se para todo par de vértices  $x \in X$  e  $y \in Y$ ,  $xy \in E(G)$*

Utilizamos  $K_{m,n}$  para denotar o grafo bipartido completo onde  $|X| = n$  e  $|Y| = m$ .

**Exercício 27** *Quantas arestas tem  $K_{m,n}$ ?*

### 3.2.1 Caracterização de Grafos Bipartidos por Ciclos de Comprimento Ímpar

O teorema a seguir mostra que para decidir se um grafo simples  $G$  é bipartido ou não, basta verificar se este contém algum ciclo de comprimento ímpar.

**Teorema 13** *Um grafo simples  $G$  é bipartido se e somente se não contém nenhum ciclo de comprimento ímpar.*

→ Seja  $G$  bipartido com bipartição  $(A, B)$ . Além disso, seja  $C = v_0v_1, \dots, v_k, v_0$  um ciclo em  $G$ . Assuma que  $v_0 \in A$ . Neste caso,  $v_1 \in B$ ,  $v_2 \in A$ , etc ... De forma geral  $v_{2i} \in A$  e  $v_{2i+1} \in B$ . Como  $v_k \in B$ , caso contrário  $G$  não é bipartido, temos que  $k$  é um número ímpar e portanto o comprimento de  $C$  é par.

← Assuma que  $G$  não tem ciclos de comprimento ímpar. Sendo  $v$  um vértice de  $G$ , defina os seguintes conjuntos.

$$X = \{x \in V \mid d(v, x) \text{ é par} \}$$

$$Y = \{y \in V \mid d(v, y) \text{ é ímpar} \}$$

Mostramos que  $(X, Y)$  é uma bipartição para  $G$ . Seja  $x_i, x_j$  um par de vértices de  $X$ . Vamos mostrar que a aresta  $x_i x_j$  não pode existir.

Sejam  $P_i$  e  $P_j$ , respectivamente, os caminhos mais curtos entre  $v$  e  $x_i$  e  $v$  e  $x_j$ . Além disso, seja  $z$  o vértice comum a  $P_i$  e  $P_j$  mais distante de  $v$ .

**Fato:** O subcaminho de  $P_i$  que começa em  $v$  e termina em  $z$  tem comprimento igual ao subcaminho de  $P_j$  que começa em  $v$  e termina em  $z$ .

Convidamos o leitor a tentar provar este fato.

Sejam  $P_i^z$  e  $P_j^z$  os subcaminhos de  $P_i$  e  $P_j$  que começam em  $z$  e terminam em  $x_i$  e  $x_j$ , respectivamente. Utilizando  $|P|$  para denotar o comprimento de um caminho  $P$  e  $k$  para denotar o comprimento do subcaminho de  $P_i(P_j)$  que começa em  $v$  e termina em  $z$ , temos que

$$|P_i| = k + |P_i^z|$$

$$|P_j| = k + |P_j^z|$$

Como os comprimentos de  $P_i$  e  $P_j$ , por definição, tem a mesma paridade, segue que  $|P_i^z|$  e  $|P_j^z|$  tem a mesma paridade. Portanto, a aresta  $x_i x_j$  não pode existir, caso contrário o grafo teria o ciclo de comprimento ímpar  $P_i^z x_i x_j P_j^z$ .

O mesmo argumento permite mostrar que para qualquer par  $y_i, y_j$  de vértices de  $Y$ , a aresta  $y_i y_j \notin E$ . Portanto,  $(X, Y)$  é uma bipartição. ■

### 3.3 Árvores

Nesta seção estudamos as árvores, uma classe de grafos que ocorre em várias aplicações. Inicialmente, definimos o que vem a ser um grafo acíclico.

**Definição 32 (Grafo Acíclico)** *Um grafo acíclico é um grafo que não contém ciclos.*

**Definição 33 (Árvore)** *Uma árvore é um grafo conexo e acíclico (figura 3.12)*

**Definição 34 (Floresta)** *Uma floresta é um grafo  $G$  em que todas as componentes conexas são árvores.*

Para poder trabalhar com árvores, a seguinte observação é importante.

**Proposição 29** *Seja  $T$  uma árvore e seja  $e = uv$  uma aresta de  $T$ . O grafo  $T - e$ , obtido devido através da remoção de  $e$ , é uma floresta com duas componentes conexas, uma contendo o nó  $v$  e outra contendo o nó  $u$*

**Teorema 14** *Sejam  $u$  e  $v$  dois vértices distintos de uma árvore  $T$ . Então só existe um único caminho entre  $u$  e  $v$  em  $T$ .*

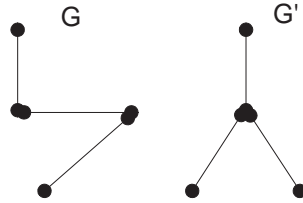


Figura 3.12:

**Prova:** Assuma a existência de dois caminhos distintos entre  $u$  e  $v$ :  $P_1 = x_0x_1 \dots x_p$  e  $P_2 = y_0y_1 \dots y_q$ , onde  $x_0 = u$ ,  $x_p = v$ ,  $y_0 = u$  e  $y_q = v$ .

Seja o primeiro  $i$  tal que  $x_i \neq y_i$  (observe que este  $i$  tem que existir já que  $P_1$  e  $P_2$  são distintos). Além disso, considere o menor  $j$ ,  $j \in \{i + 1, \dots, p\}$ , tal que  $x_j = y_k$ , para algum  $k > i$  (este  $j$  tem que existir já que  $x_p = y_q$ ). Basicamente, ao escolher estes índices, estamos capturando os pontos em que os caminhos  $P_1$  e  $P_2$  se “separam” e se “juntam” pela primeira vez.

Logo,  $x_{i-1}x_i \dots x_j y_{k-1} \dots y_{i-1}$  é um ciclo, o que contradiz o fato de  $T$  ser uma árvore. ■

**Teorema 15** Se  $T = (V, E)$  é uma árvore, então  $|E| = |V| - 1$ .

**Prova:** Indução no número de arestas.

**Base.**  $|E(G)| = 0 \Rightarrow |V(G)| = 1$ , OK !

**Passo Indutivo.** Se o resultado válido para todas as árvores com  $k$  ou menos arestas então vale para todas árvores com  $k + 1$  arestas.

**Prova do Passo.** Seja  $T = (V, E)$  uma árvore com  $k + 1$  arestas. Nosso objetivo é provar que  $T$  tem  $k + 2$  vértices.

Seja  $e = uv$  uma aresta de  $T$  e sejam  $T_u = (V_u, E_u)$  e  $T_v = (V_v, E_v)$  as componentes conexas de  $T - e$  que contém  $u$  e  $v$ , respectivamente. Temos que  $|E| = |E_u| + |E_v| + 1 = k + 1$  e  $|V| = |V_u| + |V_v|$ .

Como  $T_u$  e  $T_v$  tem no máximo  $k$  arestas, segue da hipótese de indução que  $|V_u| = 1 + |E_u|$  e  $|V_v| = 1 + |E_v|$ . Logo,

$$|V| = |V_u| + |V_v| = 1 + |E_u| + 1 + |E_v| = k + 2.$$

■

É possível mostrar também que se  $G$  é um grafo conexo com  $n$  vértices e  $n - 1$  arestas então  $G$  é uma árvore. Essa prova fica a cargo do leitor.

**Exercício 28** *Mostre que se  $G$  é um grafo conexo com  $n$  vértices e  $n - 1$  arestas então  $G$  é uma árvore.*

**Exercício 29** *Mostre que se uma floresta  $G = (V, E)$  tem  $k$  componentes conexas então  $|E| = |V| - k$ .*

### 3.3.1 Árvores Geradoras de Custo Mínimo

Seja  $G = (V, E)$  um grafo e seja  $T = (V, E')$  um subgrafo gerador de  $G$  (ver Definição 16). Se  $T$  é também uma árvore, dizemos que  $T$  é uma subárvore geradora para  $G$ .

Uma aplicação interessante envolvendo árvores geradoras ocorre quando desejamos conectar vértices de um grafo ao menor custo possível. Imagine que temos um conjunto de cidades  $V = \{v_1, \dots, v_n\}$  e sabemos que o custo da construção de uma estrada entre  $v_i$  e  $v_j$  é um número positivo  $c_{ij}$ . O problema da árvore geradora de custo mínimo (AGM) consiste em determinar quais estradas devemos construir (arestas devemos escolher) de modo a conectar as cidades de  $V$  gastando o mínimo possível.

Considere o grafo no topo da Figura 3.13. O valor associado a uma aresta é o custo de escolhê-la para compor a solução. O conjunto de arestas de menor custo que permite conectar os vértices do grafo é apresentado em azul (reforçado) no grafo mais abaixo. O custo desta solução é  $50 = 4 + 6 + 8 + 5 + 11 + 9 + 7$ . Uma outra forma possível de conectar os vértices utilizaria as arestas de custos 24, 6, 8, 5, 11, 9, 14. Essa solução não seria a melhor possível já que teria custo maior do que a apresentada em azul.

Utilizamos a notação  $\text{custo}(S)$  para indicar a soma dos custos das arestas de um conjunto de arestas  $S$ . O problema da árvore geradora de custo mínimo é formalizado da seguinte maneira:

*Entrada* Um grafo conexo  $G = (V, E)$  e uma função de custo que associa cada aresta a um número positivo.

*Saida* Um conjunto de arestas  $E'$  que satisfaz simultaneamente

i  $E' \subseteq E$ ;

ii  $E'$  conecta o grafo, ou seja, o grafo  $G' = (V, E')$  é conexo;

- iii  $\text{custo}(E') \leq \text{custo}(E'')$  para todo conjunto de arestas  $E''$  que satisfaz as condições (i) e (ii).

Note que o conjunto de arestas que é solução do exemplo da Figura 3.13 forma uma árvore geradora para o grafo. Surge então a seguinte pergunta:

**Será que a solução de custo mínimo é sempre uma árvore geradora?**

O próximo lema mostra que sim.

**Lema 3** *Seja um grafo  $G = (V, E)$  e seja  $E^*$  a solução ótima do problema da árvore geradora mínima para  $G$ . Então  $T = (V, E^*)$  é uma árvore geradora para  $G$*

Para mostrar que  $T$  é uma árvore devemos argumentar que  $T$  é conexo e que não contém ciclos. A condição (ii) do problema garante que  $T$  é conexo. Além disso,  $T$  não tem ciclos porque, caso tivesse, poderíamos obter uma outra solução do problema com custo menor que  $E^*$  removendo alguma aresta do ciclo. Note que a condição dos pesos serem positivos é utilizada nessa última afirmação. ■

O lema anterior justifica porque o problema é conhecido como o problema da árvore geradora de custo mínimo. Focamos agora na construção de um algoritmo para obter a AGM. O lema a seguir é útil para para tal propósito.

**Lema 4** *Seja  $C$  um ciclo no grafo  $G$  e seja  $e$  uma aresta de custo máximo no ciclo. Então existe uma árvore geradora de custo mínimo que não utiliza a aresta  $e$ .*

**Prova.** Seja  $T$  uma AGM para o grafo  $G$ .

caso 1)  $e \notin T$ . Neste caso, o lema está provado.

caso 2)  $e \in T$ . Sejam  $u$  e  $v$  as extremidades de  $e$  e seja  $P$  o caminho entre  $u$  e  $v$  obtido ao remover a aresta  $e$  do ciclo  $C$ . Seja  $f$  uma aresta em  $P$  que liga um nó da componente de  $T - e$  que contém  $u$  a um nó da componente  $T - e$  que contém  $v$ . Note que  $f$  não pertence a  $T$ . Como o custo de  $f$  é menor ou igual ao custo de  $e$ , temos que a árvore  $T - \{e\} \cup f$  é uma AGM que não contém  $e$ . ■

Podemos aplicar o resultado do lema anterior para concluir que existe uma árvore geradora mínima que não utiliza as arestas de custo 24 e 14 na

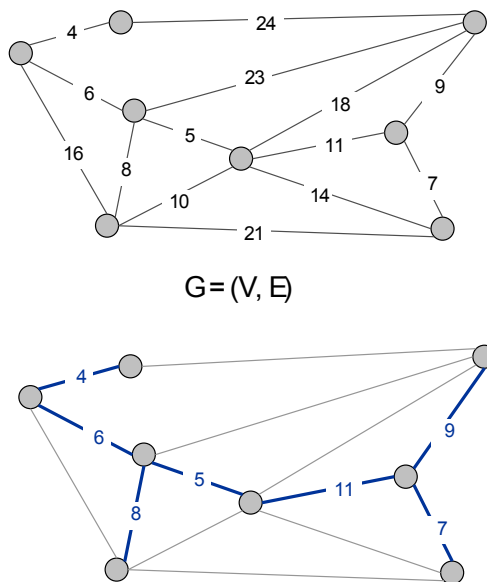


Figura 3.13:

Figura 3.13. De fato, a aresta de custo 24 é a mais cara do ciclo que contém as arestas de custos 4,6,24 e 23. Já a aresta de custo 14 é a de maior custo do ciclo que contém as arestas de custo 7,11 e 14.

Podemos então obter a árvore geradora mínima para um grafo utilizando sucessivamente o lema anterior para remover arestas. O algoritmo da Figura 3.14 obtém uma AGM para  $G$  procedendo desta forma. O teorema a seguir mostra que o algoritmo está correto.

**Teorema 16** *Para todo grafo conexo  $G$ , o algoritmo  $\text{CalculaAGM}(G)$  obtém uma árvore geradora de custo mínimo para  $G$ .*

Vamos provar por indução no número de arestas que o algoritmo funciona corretamente para todo grafo  $G$  com  $n$  vértices.

Procedimento  $\text{CalculaAGM}(G)$

$T \leftarrow G$ .

**Enquanto** existir um ciclo em  $T$

Escolha um ciclo em  $T$  e remova de  $T$  a aresta de maior custo do ciclo

**Fim Enquanto**

Return  $T$

Figura 3.14: Procedimento AGM Iterativo

**Caso Base.**  $G$  tem  $n - 1$  arestas. Como  $G$  é conexo segue do resultado do Exercício 28 que  $G$  é uma árvore. Como a única AGM para  $G$  é o próprio  $G$ , o algoritmo age corretamente.

**Passo Indutivo.** Se o procedimento  $\text{CalculaAGM}$  obtém a AGM para todo grafo conexo  $G$  com  $k$  arestas então  $\text{CalculaAGM}$  também devolve a AGM para todo grafo conexo  $G'$  com  $k + 1$  arestas, para todo  $k \geq n - 1$ .

**Prova do Passo.** Seja  $G'$  um grafo conexo com  $k + 1$  arestas e seja  $e$  a aresta removida na primeira iteração de  $\text{CalculaAGM}(G')$ . Devemos observar que o algoritmo  $\text{CalculaAGM}(G')$  devolve a mesma solução que  $\text{CalculaAGM}(G' - e)$  já que ele executa os mesmos passos depois de remover a aresta  $e$ . Por hipótese de indução,  $\text{CalculaAGM}(G' - e)$  devolve a AGM para  $G' - e$ . Como  $e$  é aresta mais cara de um ciclo de  $G'$  segue do lema anterior que a AGM para  $G' - e$  é também uma AGM para  $G'$ . Portanto,  $\text{CalculaAGM}(G')$  devolve uma AGM para  $G'$ . ■

## 3.4 Trajeto Euleriano

Um trajeto Euleriano em um grafo  $G$  é um trajeto que utiliza todas as arestas do grafo.

**Definição 35** *Um grafo  $G$  é Euleriano se e somente se possui um trajeto Euleriano.*

Os Grafo  $G$  e  $G'$  na figura 3.15 são Eulerianos. Em  $G$ ,  $1a2c3b1$  é um trajeto Euleriano.

**Exercício 30** *Determine um trajeto Euleriano no grafo  $G'$  da figura 3.15.*

O grafo  $H$  não contém um trajeto Euleriano. Informalmente, isto quer dizer que não é possível desenhar  $H$  sem tirar o lápis do papel e sem passar duas vezes pela mesma aresta. Você já tentou fazer isto alguma vez?

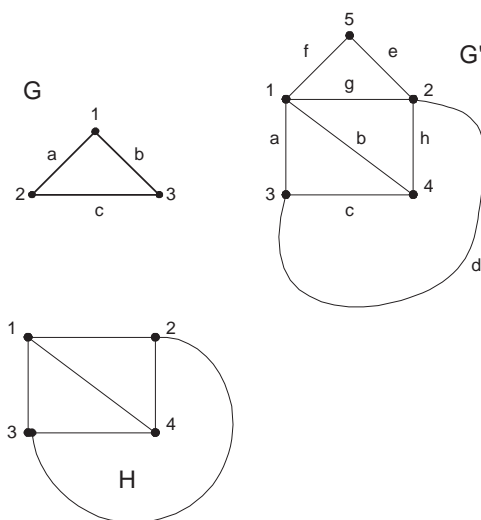


Figura 3.15:

Surge então a seguinte questão:

**Como determinar se um grafo é Euleriano?**

Este tipo de problema ocorre na seguinte situação:

**Exemplo 54** (*Problema do Carteiro Chinês*) *Um carteiro tem que percorrer uma série de ruas para entregar correspondências. Qual a rota mais curta que o carteiro pode utilizar?*

*Podemos modelar este problema utilizando um grafo, onde cada rua corresponde a uma aresta e cada encontro de duas arestas a um vértice. Note*



que se existir um trajeto Euleriano, este corresponderá a rota mais curta, já que todas as ruas serão percorridas exatamente uma vez.

De volta a nossa questão de determinar se um dado grafo  $G$  admite um trajeto Euleriano, vamos tentar entender que propriedades  $G$  **precisa ter** para ser Euleriano. A seguinte condição é fundamental em nossa investigação.

**Lema 5** *Seja  $G$  um grafo Euleriano. Se  $G$  possui um vértice  $v$  com grau ímpar, então ou o trajeto Euleriano termina em  $v$  ou começa em  $v$ .*

**Prova:** Existem dois casos:

1. O trajeto começa em  $v$ . OK !!!
2. O trajeto não começa em  $v$ . Logo, toda vez que chegamos em  $v$  por uma aresta, devemos sair por outra diferente, o que consome 2 unidades do grau. Como, o grau de  $v$  é ímpar e um trajeto Euleriano passa por todas as arestas com extremidade em  $v$ , devemos terminar em  $v$  (figura 3.16).

■

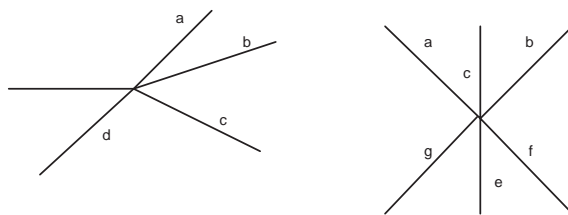


Figura 3.16:

A seguinte propriedade pode então ser obtida. Convidamos o leitor a tentar deduzi-la antes de examinar a prova.

**Lema 6** *Se um grafo  $G$  possui um trajeto Euleriano, então o número de vértices com grau ímpar em  $G$  é 0 ou 2.*

**Prova:** Pelo lema anterior  $G$  pode possuir no máximo 2 vértices com grau ímpar, caso contrário existiria um vértice de grau ímpar que não começa nem termina o trajeto Euleriano. Como todo grafo possui um número par de vértices com grau ímpar (vide Corolário 2), logo  $G$  possui 0 ou 2 vértices com grau ímpar. ■

**Exemplo 55** Podemos utilizar o lema anterior para mostrar que o grafo  $H$  da figura 3.15 não é de fato Euleriano. Com efeito,  $H$  tem 4 vértices com grau ímpar. Como todo grafo Euleriano tem 0 ou 2 vértices com grau ímpar (Lema 6), segue que  $H$  não é Euleriano.

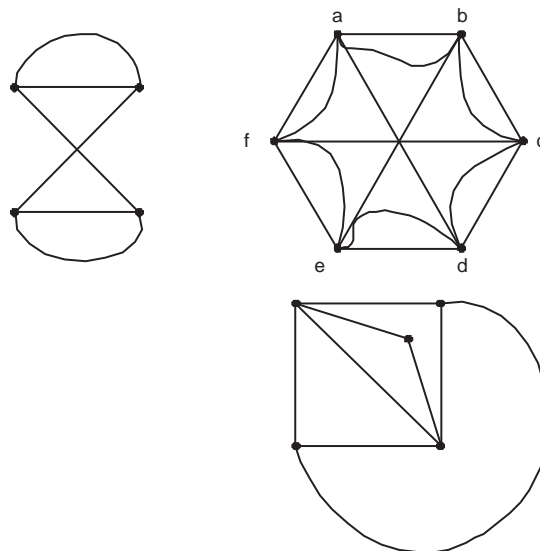


Figura 3.17:

**Exercício 31** Determine quais dos três grafos da figura 3.17 são Eulerianos.

O Lema 6 provê uma condição necessária para um grafo ser Euleriano. A partir desta condição, podemos provar que certos grafos não são Eulerianos, tal qual no Exemplo 55. Entretanto, como podemos garantir que um grafo

é Euleriano? Uma possibilidade é ficar testando trajetos até encontrar um Euleriano. O problema é que isto pode ser uma tarefa muito árdua, ou seja, tomar tempo demais.

No decorrer desta seção vamos provar que se um grafo  $G$  tem 0 ou 2 vértices de grau ímpar, então ele é Euleriano. Para começar precisamos de um resultado auxiliar.

**Lema 7** *Todo grafo conexo (não trivial <sup>2</sup>) com 0 vértices de grau ímpar possui um ciclo.*

**Prova:** Se  $G$  só possui um vértice então  $G$  consiste em um conjunto de laços, caso contrário  $G$  seria trivial. Vamos assumir então que  $G$  possui pelo menos dois vértices. Seja  $P = v_0e_0v_1e_1, \dots, e_{k-1}v_k$  o caminho mais longo entre dois vértices de  $G$ . Como todo vértice de  $G$  tem grau par, então  $v_k$  é extremidade de alguma aresta  $e$  diferente de  $e_{k-1}$ . Além disso, a outra extremidade de  $e$  tem que ser um vértice  $v_i$ , com  $i \in \{0, 1, \dots, k\}$ , caso contrário teríamos um caminho maior que  $P$ . Portanto,  $v_i e_i v_{i+1} e_{i+1} \dots e_{k-1} v_k e v_i$  é um ciclo. ■

O seguinte lema mostra que se  $G$  tem 0 vértices de grau ímpar, então  $G$  é Euleriano.

**Lema 8** *Seja  $G$  um grafo conexo com 0 vértices de grau ímpar. Então,  $G$  contém um trajeto Euleriano fechado.*

**Prova:** Utilizamos indução no número de arestas de  $G$ .

Base:  $G$  tem somente uma aresta. Como  $G$  é conexo e não tem vértices de grau ímpar, então  $G$  consiste em um vértice  $v$  com um laço  $e$ . Portanto  $vev$  é um trajeto Euleriano fechado em  $G$ .

Hipótese Indutiva: Todo grafo conexo, sem vértices de grau ímpar, e com no **máximo**  $k$  arestas tem um trajeto Euleriano fechado.

Passo Indutivo: Se todo grafo conexo, sem vértices de grau ímpar, e com no **máximo**  $k$  arestas tem um trajeto Euleriano fechado, então todo grafo

---

<sup>2</sup>um grafo  $G = (V, E)$  é trivial se e somente se  $|V| = 1$  e  $E = \emptyset$

conexo, sem vértices de grau ímpar e com  $k + 1$  arestas, então  $G$  tem um trajeto Euleriano fechado, para todo  $k \geq 1$ .

Prova do Passo: Como  $G$  tem 0 vértices de grau ímpar, segue do Lema 7 que  $G$  possui um ciclo  $C$ . Seja  $G' = (V', E')$  o grafo obtido a partir de  $G$  removendo todas as arestas de  $C$ . Observe que  $V' = V$  e  $E' = E - \{\text{arestas em } C\}$ . Seja  $v$  um vértice de  $V$  e sejam  $d_G(v)$  e  $d_{G'}(v)$ , os graus de  $v$  em  $G$  e  $G'$ , respectivamente. Se  $v$  está no ciclo  $C$ , então  $d_{G'}(v) = d_G - 2$ . Caso contrário,  $d_{G'}(v) = d_G(v)$ . A conclusão desta análise é que todo vértice de  $G'$  tem grau par.

Sejam  $G'_1, \dots, G'_l$  as componentes conexas (Definição 43) de  $G'$  que tem pelo menos uma aresta.

Temos que

- a) O numero de arestas de  $G'_i$  é menor ou igual a  $k$ , para  $i = 1, \dots, l$
- b) Todo vértice de  $G'_i$  tem grau par, para  $i = 1, \dots, l$ .

Segue da hipótese indutiva que  $G'_i$ ,  $i = 1, \dots, l$ , tem um trajeto Euleriano fechado. Seja  $T_i$  o trajeto de  $G'_i$ .

Como exemplo, no canto superior esquerdo da figura 3.18 temos o grafo  $G$  e no canto direito um ciclo para  $G$ . Os dois grafos na parte inferior da figura 3.18 são as componentes de  $G'$  com pelo menos uma aresta.

Para obter um trajeto Euleriano para  $G$  percorremos o ciclo  $C$ . Cada vez que visitarmos um vértice  $v$  de  $C$  que pertence a uma componente  $G'_i$  cujo trajeto  $T_i$  ainda não foi percorrido, interrompemos momentaneamente a nossa caminhada por  $C$  para percorremos o trajeto  $T_i$  (obtido pela hipótese). Após, continuamos a percorrer o ciclo  $C$  a partir de  $v$ . Este processo é encerrado quando todas as arestas de  $C$  tiverem sido visitadas. ■

Na figura 3.18, o trajeto Euleriano fechado  $T$  para  $G$  é obtido da seguinte forma: começamos visitando  $Z$ . Como  $Z$  pertence a  $G'_1$ , percorremos  $T_1$  (começando a partir de  $Z$ ). Após, voltamos ao ciclo  $C$  e percorremos a aresta 11; visitamos o vértice  $H$ ; percorremos a aresta 13 e visitamos o vértice  $I$ . Como  $I \in G'_2$ , percorremos  $T_2$  (começando a partir de  $I$ ). Após, voltamos

ao ciclo  $C$  e percorremos as aresta 12 e visitamos o vértice  $F$ . Embora,  $F \in G'_1$ , não percorremos novamente  $T_1$ , uma vez que este já foi percorrido. Finalmente, passamos pela aresta 9 para chegar no vértice  $Z$ , completando o ciclo  $C$ . Portanto, o seguinte trajeto foi obtido

$$\underbrace{Z8D4C2A1B3C5E6F10Z}_{T_1} 11H13 \underbrace{I15L20K18L16J17K19J14I}_{T_2} 12F9Z$$

O Lema 8 garante que se um grafo conexo tem 0 vértices de grau ímpar então ele é Euleriano. E se um grafo tem dois vértice de grau ímpar, o que acontece? O próximo Lema resolve esta questão

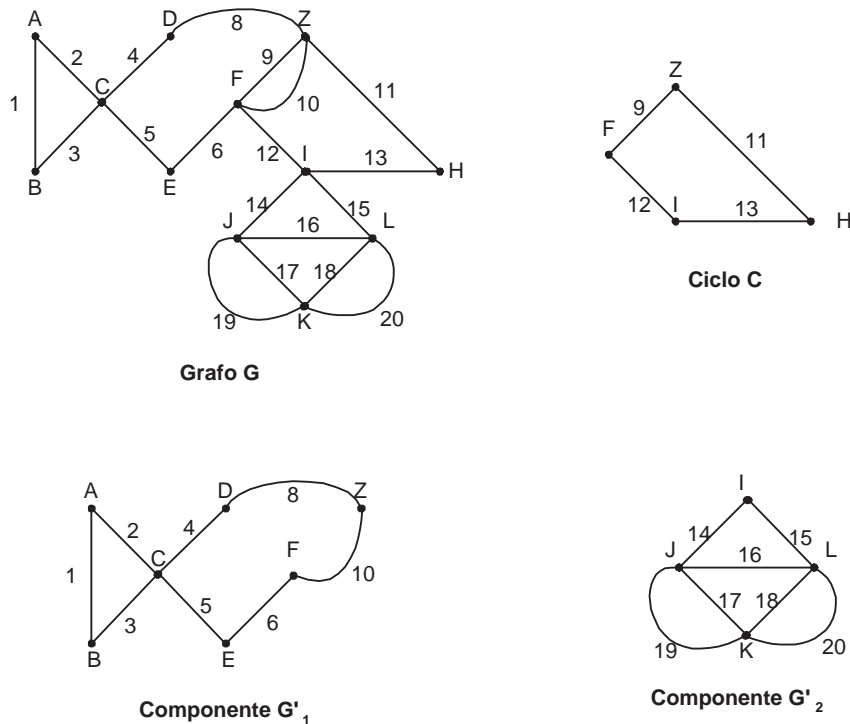


Figura 3.18: Ilustração do Lema 8

**Lema 9** *Seja  $G$  um grafo conexo com 2 vértices de grau ímpar,  $u$  e  $v$ . Então,  $G$  contém um trajeto Euleriano que começa em  $u$  e termina em  $v$ .*

**Prova:** Seja  $G = (V, E)$  e seja  $G'$  o grafo obtido a partir de  $T$  adicionando uma aresta  $e'$  que une  $u$  a  $v$ , os dois vértices de grau ímpar em  $G$ . Observe que isto pode ser feito independentemente se já existe uma aresta ligando  $u$  a  $v$  em  $G$  ou não.

Observe que  $G'$  é conexo e que todos os vértices de  $G'$  tem grau par, já que os únicos dois que tinham grau ímpar em  $G$  ganharam uma unidade a mais em  $G'$ . Portanto, o Lema 8 garante que  $G'$  tem um trajeto Euleriano fechado  $T$ .

Retirando a aresta  $e'$  de  $T$ , obtemos um trajeto Euleriano para  $G$  que começa em  $u$  e termina em  $v$ . ■

Os Lemas 8 e 9 implicam no seguinte teorema, que constitui o resultado principal desta seção.

**Teorema 17** *Um grafo conexo  $G$  é Euleriano se e somente se possui 0 ou 2 vértices de grau ímpar.*

*Se  $G$  tem 0 vértices com grau ímpar, então  $G$  possui um trajeto Euleriano fechado. Por outro lado, se  $G$  tem 2 vértices com grau ímpar,  $u$  e  $v$ , então ele contém um trajeto Euleriano que começa em  $u$  e termina em  $v$ .*

## 3.5 Grafos Direcionados

Em redes de computadores, as vezes podemos mandar um pacote de um computador  $i$  para um computador  $j$  através de uma conexão  $e$  mas não podemos mandar um pacote de  $j$  para  $i$  através da mesma conexão. Para modelar estas situações em que apenas um dos sentidos das arestas é permitido, utilizamos o conceito de grafos direcionados.

Um grafo direcionado<sup>3</sup>  $D$  é uma tripla ordenada  $(V(D), E(D), \psi_D)$ , onde

- $V(D)$  é um conjunto não vazio de vértices

---

<sup>3</sup>é muito comum encontrar o termo digrafo em vez de grafos direcionados em textos de língua portuguesa

- $E(D)$  é um conjunto de arcos (possivelmente vazio)
- $\psi_D$  é uma função que associa cada aresta de  $D$  a um par **ordenado** de vértices de  $D$

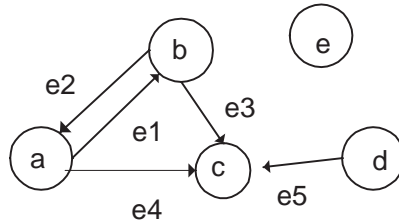


Figura 3.19: Grafo Direcionado

Dado um arco  $e = (u, v)$ , dizemos que  $u$  é a cauda de  $e$  e  $v$  é a cabeça de  $v$ . Na Figura 3.19 os nós  $b$  e  $a$  são, respectivamente, a cauda e a cabeça do arco  $e_2$ .

**Definição 36 (Grau de Entrada)** *O grau de entrada  $d^-(v)$  de um vértice  $v$  é o número de arcos que tem  $v$  como cabeça.*

**Definição 37 (Grau de Saída)** *O grau de saída  $d^+(v)$  de um vértice  $v$  é o número de arcos que tem  $v$  como cauda.*

Na Figura 3.19,  $d^-(a) = 1, d^+(a) = 2, d^-(c) = 3$  e  $d^+(c) = 0$ .

Para grafos não direcionados mostramos que a soma dos graus dos vértice é igual ao dobro do número de arestas. No caso dos grafos direcionados vale a seguinte relação

$$\sum_{v \in V(G)} d^+(v) = \sum_{v \in V(G)} d^-(v) = |E(G)|$$

### 3.5.1 Passeios em Grafos Direcionados

Assim como nos grafos não direcionados, podemos definir passeios, caminhos e trajetos nos digrafos.

**Definição 38 (Passeio Direcionado)** *Um passeio (walk) em um grafo  $G$  é uma sequência não nula  $W = v_0e_1v_1e_2 \dots e_kv_k$ , tal que para  $i = 1, \dots, k$ ,  $v_{i-1}$  é a cauda de  $e_i$  e  $v_i$  é a cabeça de  $e_i$ . O inteiro  $k$  é o comprimento do passeio  $W$ .*

**Definição 39 (Trajeto Direcionado)** *Um passeio direcionado, onde não há repetição de arestas.*

**Definição 40 (Caminho Direcionado)** *Um passeio direcionado, onde não há repetição de vértices*

Na figura 3.19,  $be_2ae_4c$  é um caminho que começa em  $b$  e termina em  $c$ . Observe, todavia, que não existe nenhum caminho que comece em  $c$  e termine em  $b$ .

**Definição 41 (Alcançabilidade)** *Um nó  $u$  é alcançável a partir de  $v$  se e somente se existe um caminho direcionado que começa em  $v$  e termina em  $u$ .*

**Definição 42 (Grafo Fortemente Conexo)** *Um grafo direcionado  $D = (V, E)$ , é fortemente conexo se e somente se para todo para  $u, v \in V$ ,  $u$  é alcançável a partir de  $v$  e vice-versa.*

O grafo da figura 3.20 não é fortemente conexo.

O conceito de componentes conexas pode ser estendido para grafos direcionados. Primeiramente, definimos um conjunto de vértices  $S$  de um digrafo  $G = (V, E)$  como fortemente conexo se  $S$  tem apenas um vértice ou se existe caminho entre  $u$  e  $v$  e  $v$  e  $u$  para todo par de vértices  $u, v \in S$ . A partir de então, definimos o conceito de componentes fortemente conexas.



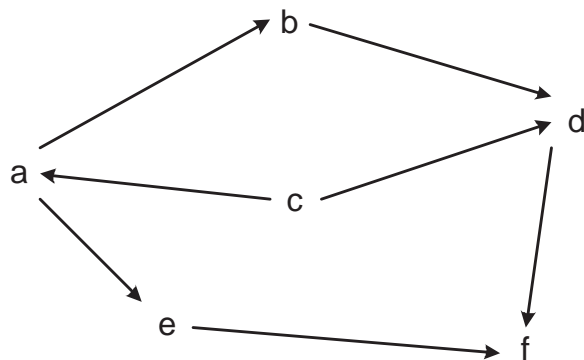


Figura 3.20:  $f$  é alcançável a partir de  $b$ , mas  $b$  não é alcançável a partir de  $f$

**Definição 43 (Componentes Fortemente Conexas)** *Seja  $V'$  um subconjunto dos vértices de um grafo  $G$ . Dizemos que  $V'$  é uma componente fortemente conexa de  $G$  se e somente se*

- (i) o conjunto  $V'$  é fortemente conexo
- (ii)  $\forall S$  tal que  $V' \subset S \subseteq V$ ,  $S$  não é um conjunto fortemente conexo.

Na Figura 3.20 temos quatro componentes fortemente conexas:  $\{a, b\}$ ,  $\{c\}$ ,  $\{d\}$ ,  $\{e\}$ .

Uma classe de grafos direcionados particularmente importante é a classe dos grafos direcionados acíclicos.

### 3.5.2 Grafos Direcionados Acíclicos

Um grafo direcionado acíclico (DAG) é um grafo que não contém ciclos. O grafo da Figura 3.20 é um DAG enquanto que o grafo  $G = (V, E)$  com  $V = \{a, b, c, d\}$  e  $E = \{(a, b), (b, c), (c, a), (c, d)\}$  não é um DAG já que  $(a, b), (b, c), (c, a)$  formam um ciclo

Os DAG's permitem modelar algumas situações de interesse como relações de precedência entre tarefas de um projeto. Considere que cada nó da Figura 3.20 corresponde a uma tarefa e que um arco indica que uma tarefa tem que

ser realizada antes da outra. De acordo com o grafo a tarefa  $c$  tem que ser realizada antes da tarefa  $a$  e a tarefa  $a$  antes da tarefa  $e$ .

Em situações como a descrita podemos ter que encontrar uma ordem para realizar as tarefas do projeto. Em nosso exemplo uma ordem possível seria  $c - a - b - e - d - f$ . Outra ordem seria  $c - a - e - b - d - f$ . Essas ordens são chamadas de *ordenações topológicas* para o grafo.

Formalmente, uma ordenação topológica para um grafo direcionado  $G = (V, E)$  é uma função que associa cada vértice  $v$  do grafo a um número inteiro  $f(v)$  no conjunto  $\{1, \dots, n\}$  e que satisfaz as seguintes condições:

- (i)  $f(u) \neq f(v)$  para  $u \neq v$
- (ii) Se  $(u, v) \in E$  então  $f(u) < f(v)$

A condição (i) garante que vértices diferentes recebem valores diferentes. A condição (ii) garante que se existe uma aresta ligando  $u$  a  $v$  então  $u$  recebe valor menor que  $v$ .

Para o grafo da Figura 3.20, a função  $f(c) = 1, f(a) = 2, f(b) = 3, f(e) = 4, f(d) = 5, f(f) = 6$  é uma ordenação topológica.

Por outro lado, o grafo  $G = (V, E)$  com  $V = \{a, b, c, d\}$  e  $E = \{(a, b), (b, c), (c, a), (c, d)\}$  não admite uma ordenação topológica. Essa situação motiva a seguinte pergunta.

### Quais grafos direcionados admitem uma ordem topológica?

Vamos mostrar que somente os DAG's admitem uma ordenação topológica. O seguinte lema é útil para chegar esta conclusão.

**Lema 10** *Em um DAG existe um vértice com grau de entrada 0.*

Podem existir vários caminhos em um DAG. Dentre todos estes caminhos, seja  $P = v_1 \dots v_k$  um caminho no DAG tal que nenhum outro é mais longo que ele. Vamos mostrar que  $v_1$ , o primeiro vértice deste caminho, tem grau de entrada 0. Seja  $v$  um vértice qualquer do grafo. Devemos mostrar que  $v$  não aponta para  $v_1$ . Para isso, dividimos a argumentação em dois casos:

Caso 1.)  $v$  não está em  $P$ . Então  $v$  não aponta para  $v_1$ , caso contrário o caminho  $v \rightarrow P$  seria mais longo que  $P$ , o que não é possível.

Caso 2).  $v \in P$ . Então,  $v$  não pode apontar para  $v_1$ , caso contrário teríamos um ciclo no grafo. ■

**Teorema 18** *Um grafo direcionado admite uma ordenação topológica se e somente se ele é acíclico*

Primeiro vamos mostrar que se um grafo  $G$  tem um ciclo então ele não admite uma ordem topológica. Vamos assumir que  $G$  tem simultaneamente um ciclo  $C$  e uma ordenação topológica  $f$  e, concluir então, que tal hipótese leva a uma contradição. Seja  $v$  o vértice com menor valor de  $f$  no ciclo. Além disso, seja  $u$  o predecessor de  $v$  no ciclo. Temos que  $(u, v) \in E$  e  $f(u) > f(v)$ , o que contradiz o fato de  $f$  ser uma ordenação topológica.

Por outro lado, podemos mostrar que se  $G$  não tem ciclos então  $G$  admite uma ordem topológica. A prova utiliza indução no número de vértices.

**Base.**  $G$  tem apenas um vértice  $v$ . Neste caso  $f(v) = 1$  é uma ordenação topológica para  $G$ .

**Passo Indutivo.** Se todo grafo acíclico  $G$  com  $k$  vértices admite uma ordenação topológica então todo DAG com  $k + 1$  vértices admite uma ordenação topológica.

**Prova do Passo.** Seja  $G$  um grafo com  $k + 1$  vértices e seja  $v$  um vértice com grau de entrada 0 cuja existência é garantida pelo lema anterior. Note que o grafo  $G - v$  é acíclico e tem  $k$  vértices. Logo, por hipótese,  $G - v$  admite uma ordem topológica, que vamos chamar de  $f'$ . Portanto, a função  $f$  definida como  $f(u) = 1 + f'(u)$  para todo  $u \in V - v$  e  $f(v) = 1$  é uma ordem topológica para  $G$  ■

A partir da prova do teorema anterior podemos extrair um algoritmo recursivo, Figura 3.21, para obter uma ordenação topológica para um DAG  $G$ . O procedimento utiliza um vetor global  $f$  com  $n$  posições que será preenchido com os valores da ordenação topológica. Este vetor pode ser iniciado com valores 0.

Procedimento  $\text{OrdemTopologica}(G)$

**Se**  $G$  tem apenas um vértice  $v$

$$f(v) = 1$$

**Senão**

Seja  $v$  um vértice de grau de entrada 0 em  $G$

$\text{OrdemTopologica}(G - v)$

**Para** todo vértice  $u$  de  $G - v$

$$f(u) \leftarrow f(u) + 1$$

**Fim Para**

$$f(v) \leftarrow 1$$

**Fim Se**

Figura 3.21: Procedimento para encontrar uma ordem topológica para um DAG

## 3.6 Problema do Caminho mais curto

Seja  $G$  um grafo direcionado com pesos **positivos** nas arestas. Um problema importante, que surge em diversas aplicações, é o de determinar o caminho mais curto entre dois vértices do grafo. Este problema aparece em aplicações de roteamento de pacotes na Internet, distribuição de produtos através de malhas rodoviárias e em diversas outras situações práticas. Podemos definir o problema da seguinte forma

*Entrada.* Um grafo direcionado  $G = (V, E)$ , uma origem  $s$ , um destino  $t$  e uma função  $w$  que associa cada aresta de  $E$  a um peso (valor positivo).

*Saída.* O caminho de menor peso entre  $s$  e  $t$ , onde o peso de um caminho é a soma dos pesos das arestas do caminho.

No grafo da Figura 3.22 as arestas dos caminhos de menor peso entre  $s$  e os demais vértices do grafo estão em azul. O caminho de menor peso entre  $s$  e  $t$  é  $(s, 1, 3, 5, 4, t)$  e tem peso 50.

Se tentarmos resolver este problema utilizando força bruta teremos que gerar todos os caminhos entre  $s$  e  $t$  e selecionar o de menor peso. Esta abordagem é muito cara computacionalmente já que o número de caminhos

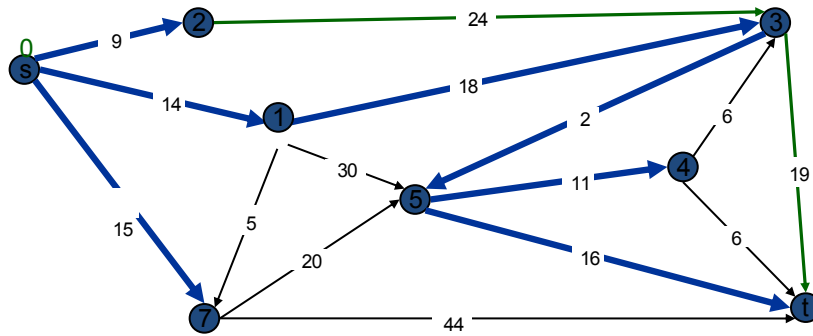


Figura 3.22:

pode ser exponencial no número de vértices do grafo. Vamos apresentar então uma alternativa mais barata. Antes, porém, apresentamos uma observação simples sobre a estrutura dos caminhos de peso mínimo.

**Lema 11** *Seja  $P$  o caminho de menor peso entre dois vértice  $u$  e  $v$  e seja  $w$  um vértice de  $P$ . Portanto, o subcaminho de  $P$  que começa em  $u$  e termina em  $w$  é o caminho mais curto entre  $u$  e  $w$ .*

Se isso não fosse verdade poderíamos obter um caminho de peso menor que  $P$  substituindo o subcaminho de  $P$  que começa em  $u$  e termina em  $w$  por um caminho de menor peso. ■

Vamos definir a distância de  $s$  a um vértice  $v$  como o peso do caminho de menor peso entre  $s$  e  $v$ . Dizemos que um vértice  $u$  é mais próximo a  $s$  do que  $v$  se a distância entre  $s$  e  $u$  é menor que a distância entre  $s$  e  $v$ .

Em linhas gerais a nossa abordagem será encontrar o caminho de menor peso até o vértice mais próximo de  $s$  depois o caminho de menor peso até o segundo vértice mais próximo a  $s$  e, assim por diante, até encontrar o caminho de menor peso de  $s$  até  $t$ . No grafo da Figura 3.22 a abordagem encontra primeiro o caminho de peso mínimo até o vértice 2, depois até o vértice 1, depois ao vértice 7.

Seja  $S_k$  o conjunto dos  $k$  vértices mais próximos de  $s$ . Em nosso exemplo,  $S_1 = \{2\}$ ,  $S_2 = \{2, 1\}$ ,  $S_3 = \{2, 1, 7\}$ . Assuma que já conhecemos o caminho mais curto de  $s$  até cada um dos vértices em  $S_{k-1}$ . Vamos tentar encontrar então  $P_{sk}$ , o caminho mais curto de  $s$  até  $v_k$ , onde  $v_k$  é o  $k$ -ésimo vértice mais próximo de  $s$ . Apesar de não conhecermos quem é  $v_k$ , tampouco  $P_{sk}$ , podemos tirar algumas conclusões sobre  $P_{sk}$  que nos ajudarão a determinar  $v_k$  assim como  $P_{sk}$ . A primeira delas diz respeito ao predecessor de  $v_k$  em  $P_{sk}$ . Podemos afirmar que esse predecessor pertence ao conjunto  $S_{k-1}$  como mostra o lema a seguir.

**Lema 12** *Seja  $u$  o predecessor de  $v_k$  no caminho de menor peso entre  $s$  e  $v_k$ . Temos que  $u \in S_{k-1}$*

Note que o peso do caminho  $P_{su}$  é menor que o peso de do caminho  $P_{sv}$ . Portanto, se  $u$  não pertencesse a  $S_{k-1}$ ,  $v_k$  não poderia ser o  $k$ -ésimo vértice mais próximo a  $s$ , o que contradiz a definição  $v_k$ . Logo,  $u \in S_{k-1}$ . ■

Uma outra conclusão é que o subcaminho de  $P_{sk}$  que começa em  $s$  e termina em  $u$  é o caminho de menor peso entre  $s$  e  $u$ . Esta é uma consequência imediata do Lema 11.

A partir destas conclusões deduzimos que  $P_{sk}$  é composto de um caminho de peso mínimo entre  $s$  e algum vértice  $u$  pertencente a  $S_{k-1}$  e de uma aresta deste vértice  $u$  até  $v_k$ . Portanto, para encontrar  $P_{sk}$  podemos considerar todos os caminhos que contém a estrutura descrita e ficar com aquele de

```

EncontraCaminhoPesoMínimo( $s, t$ )
   $S \leftarrow \{s\}$ ;
  Enquanto  $t \notin S$ 
    Esvazie a lista  $\mathcal{L}$  dos caminhos candidatos
    Para todo vértice  $u \in S$ 
      Para todo vértice  $v$  adjacente a  $u$  tal que  $v \notin S$ 
        Inclua o caminho  $P_{su} \rightarrow v$  na lista dos caminhos candidatos.
      Fim Para
    Fim Para
     $P \leftarrow$  o caminho de menor peso na lista  $\mathcal{L}$  dos caminhos candidatos
     $v \leftarrow$  último vértice do caminho  $P$ ;  $P_{sv} \leftarrow P$ 
    Adicione  $v$  a  $S$ 
  Fim Enquanto

```

Figura 3.23: Procedimento para encontrar o caminho de peso mínimo entre uma origem e um destino

menor peso. É importante notar que o fato de já conhecermos os caminhos de menor peso entre  $s$  e os vértices de  $S_{k-1}$  facilita bastante a busca.

Para exemplificar a abordagem vamos assumir que já conhecemos  $S_4 = \{2, 1, 7, 3\}$  e vamos encontrar o caminho de menor peso de  $s$  até  $v_5$ , o quinto vértice mais próximo de  $s$ . Apesar de não conhecermos  $v_5$  sabemos que o caminho de menor peso de  $s$  a  $v_5$  é composto do caminho de menor peso de  $s$  até um vértice de  $S_4$  e depois de uma aresta deste vértice até  $v_5$ . Isso permite concluir que os caminhos candidatos são  $(s, 1, 3, 5)$ ,  $(s, 7, 5)$  e  $(s, 1, 5)$ ,  $(s, 7, t)$  e  $(s, 1, 3, t)$ . O caminho de menor peso dentre os candidatos é  $(s, 1, 3, 5)$  cujo peso é 34. Portanto, descobrimos que  $v_5 = 5$  e que o caminho de peso mínimo entre  $s$  e 5 é  $(s, 1, 3, 5)$ . Poderíamos então continuar da mesma maneira para encontrar  $S_6$  e então  $S_7$ .

A partir da discussão acima podemos obter o procedimento da Figura 3.23 para encontrar o caminho mais curto entre  $s$  e  $t$ . Devemos observar que para uma implementação eficiente do algoritmo devemos escolher estruturas de dados adequadas mas esta discussão está fora de nosso escopo.

## 3.7 Representações Computacionais

Para poder utilizar os grafos na modelagem e resolução de problemas computacionais, é necessário utilizar estruturas de dados que permitam armazená-los eficientemente em meios digitais. Nesta seção examinamos brevemente duas estruturas, a matriz de adjacências e a lista de adjacências.

### 3.7.1 Matriz de Adjacência

#### Grafos Não Direcionados

Seja  $G = (V, E)$  um grafo, onde  $V = \{1, \dots, n\}$ . A entrada  $(i, j)$  de uma matriz de adjacência indica o número de arestas que tem como extremidades  $i$  e  $j$ .

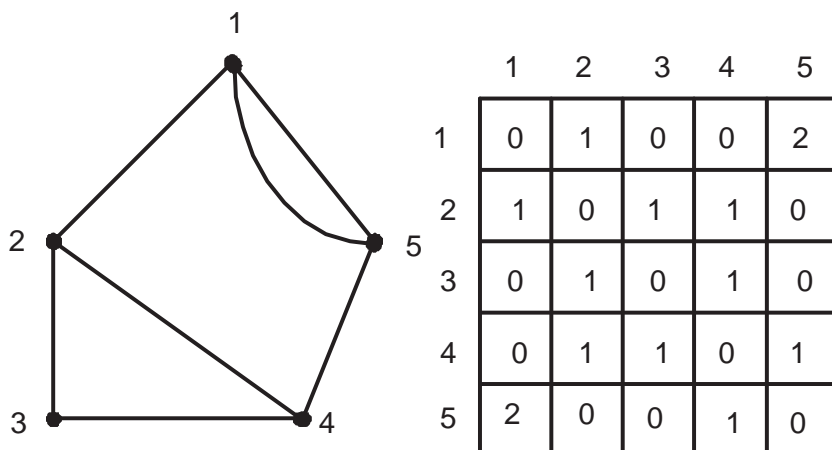


Figura 3.24: Um grafo Não direcionado e sua matriz de adjacências.

A matriz é simétrica, como podemos observar na Figura 3.24. Esta simetria permite guardar somente os elementos da diagonal principal e os elementos abaixo (ou acima) dela. Dessa forma economiza-se espaço no armazenamento da estrutura.



## Grafos Direcionados

Seja  $G = (V, E)$  um dígrafo, onde  $V = \{1, \dots, n\}$ . A entrada  $(i, j)$  de uma matriz de adjacência indica o número de arestas que tem  $i$  como cauda e  $j$  como cabeça. Neste caso, a matriz não é necessariamente simétrica, conforme a Figura 3.25.

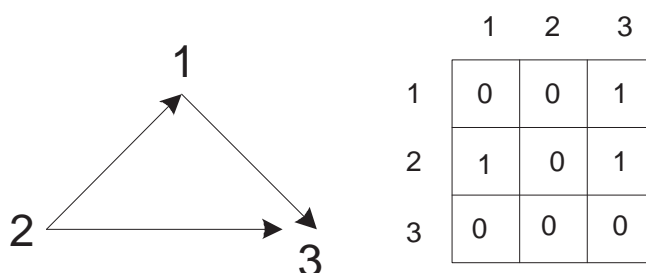


Figura 3.25: Um grafo direcionado e sua matriz de adjacências

Dentre as propriedades das matrizes de adjacências, destacamos:

- Necessita cerca de  $n^2$  posições de memória.
- A existência de uma aresta pode ser testada com uma única operação.
- Para listar todos os vértices e arestas do grafo precisamos de cerca de  $n^2$  operações.

### 3.7.2 Lista de Adjacência

Em muitos casos, lidamos com grafos esparsos, ou seja, com “poucas” arestas. Nesse caso é um desperdício utilizar  $n^2$  posições de memória. A figura 3.26 mostra um grafo com 5 vértices e 4 arestas, e sua matriz de adjacências. Observe que a maioria das entradas são nulas.

Uma alternativa para evitar este desperdício, é utilizar um vetor de listas encadeadas, onde a lista correspondente a  $i$ -ésima posição guarda os elementos adjacentes ao vértice  $i$ . A figura 3.27 exemplifica esta estrutura.

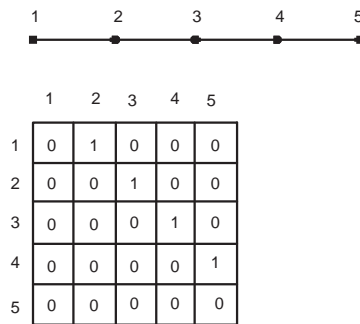


Figura 3.26: Um grafo e sua lista de adjacências

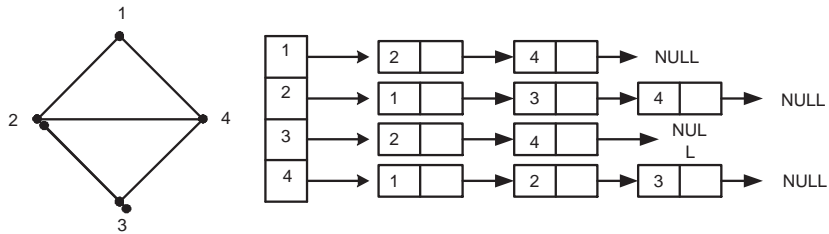


Figura 3.27:

Para representar arestas paralelas em listas de adjacências, utilizamos um campo extra para guardar a multiplicidade da aresta, conforme a Figura 3.28. Dentre as características das listas de adjacências destacamos:

- cerca de  $n + |E|$  posições de memória são necessárias.
- Para descobrir se uma aresta pertence ao grafo, pode ser necessário percorrer uma lista encadeada inteira.
- O grafo pode ser percorrido em um tempo proporcional ao número de arestas.

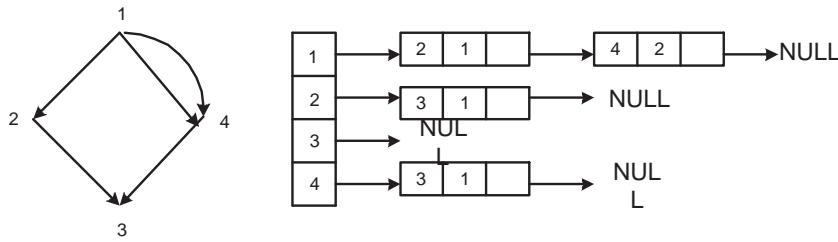


Figura 3.28:

### 3.8 Grafos Hamiltonianos

Nesta seção estudamos os grafos Hamiltonianos. Para definir um grafo Hamiltoniano é necessário definir o que vem a ser um ciclo Hamiltoniano.

**Definição 44 (Ciclo Hamiltoniano)** *Um ciclo Hamiltoniano em um grafo  $G$  é um ciclo que contém todos os vértices de  $G$*

Dizemos que um grafo  $G$  é Hamiltoniano quando ele contém um ciclo Hamiltoniano.

**Exercício 32** *Quantos ciclos Hamiltonianos tem  $K_n$ ? E  $K_{n,n}$*

Apesar da proximidade entre as definições de grafos Eulerianos e Hamiltonianos, é muito mais complicado caracterizar quando um grafo é Hamiltoniano do que quando é Euleriano. De fato, diferentemente dos grafos Eulerianos, não se conhece nenhuma caracterização para os grafos Hamiltonianos que permita reconhecer eficientemente (baixo custo computacional) se um grafo é Hamiltoniano ou não.

A seguir exibimos uma condição que é necessária e outra que é suficiente para que um grafo seja Hamiltoniano.

**Teorema 19** *Se  $G$  é Hamiltoniano então o grafo  $G - S$  tem no máximo  $|S|$  componentes conexas, para todo subconjunto  $S \subseteq V(G)$ .*

**Teorema 20** *Se  $G$  tem pelo menos três vértices e o grau mínimo de  $G$  é maior que  $|V(G)|/2$ , então  $G$  é Hamiltoniano.*

**Exercício 33** *Existe algum grafo bipartido com 11 vértices que é Hamiltoniano? Por que?*

### 3.9 Emparelhamentos

Seja  $M$  um subconjunto de arestas de um grafo  $G = (V, E)$ . Dizemos que  $M$  é um emparelhamento se e somente se não existem arestas em  $M$  que compartilhem vértices.

**Definição 45** *Um emparelhamento  $M$  em  $G$  é máximo se e somente se  $|M'| \leq |M|$  para todo emparelhamento  $M'$  em  $G$ .*

Dizemos que um emparelhamento  $M$  satura um vértice  $v$  se  $v$  é extremidade de alguma aresta de  $M$ . Um *caminho  $M$ -alternante* em  $G$  é um caminho que alterna arestas de  $G$  que estão em  $M$  com arestas que não estão em  $M$ ; Um *caminho  $M$ -aumentante* em  $G$  é um caminho  $M$ -alternante em  $G$  que começa e termina em um vértice não saturado por  $M$ .

**Exercício 34** *Seja  $G$  um grafo e seja  $M$  um emparelhamento de  $G$ . Mostre que se  $M$  é máximo então  $G$  não contém caminhos  $M$ -aumentantes.*

A volta do resultado acima também é válida.

**Teorema 21 (Teorema de Berge)** *Seja  $G$  um grafo e seja  $M$  um emparelhamento em  $G$ .  $M$  é máximo se e somente se  $G$  não contém caminhos  $M$ -aumentantes.*

Existem emparelhamentos máximos que saturam todos os vértices do grafo e outros que não. Isso motiva a seguinte definição.

**Definição 46** *Um emparelhamento  $M$  em  $G$  é perfeito se e somente se todo vértice de  $G$  incide em alguma aresta de  $M$ .*

Uma questão interessante e importante é determinar se um grafo possui um emparelhamento perfeito ou não. A seguir discutimos como caracterizar grafos bipartidos que possuem emparelhamentos perfeitos.

**Exercício 35** *Seja  $G$  um grafo bipartido com bipartição  $(X, Y)$ . Mostre que se  $G$  possui um emparelhamento perfeito então  $|N(S)| \geq |S|$  para todo subconjunto  $S \subseteq X$ , onde  $N(S)$  é o conjunto de nós que são adjacentes aos nós de  $S$ .*

Observe que a condição acima é necessária para que um grafo bipartido tenha um emparelhamento perfeito. De fato, pode-se mostrar que ela é suficiente também.

**Teorema 22 (Teorema de Hall)** *Seja  $G$  um grafo bipartido com bipartição  $(X, Y)$ .  $G$  possui um emparelhamento que satura todos os vértices de  $X$  se e somente se  $|N(S)| \geq |S|$  para todo subconjunto  $S \subseteq X$ .*

Dizemos que um grafo é  $k$ -regular se todo vértice do grafo tem grau  $k$ .

**Exercício 36** *Mostre que todo grafo bipartido  $k$ -regular possui um emparelhamento perfeito.*

### 3.10 Coloração de Vértices

Considere que temos que armazenar 5 compostos químicos, A, B, C, D e E em recipientes. Entretanto, para evitar contaminação os seguintes pares de compostos não podem ser armazenados no mesmo recipiente: A-B, A-C, D-E, C-D. Neste caso, qual o número mínimo de recipientes que precisamos?

Esta situação pode ser modelada através de um grafo  $G = (V, E)$ , com  $V = \{A, B, C, D, E\}$  e  $E = \{AB, AC, DE, CD\}$ . Devemos então associar cada vértice a um recipiente de modo que vértices vizinhos estejam associados a recipientes diferentes. Para o nosso exemplo, o número mínimo de recipientes é 2 já que podemos colocar A e D no recipiente 1 e os demais compostos no recipiente 2.

Situações como essa motivam a seguinte definição

**Definição 47** Uma  $k$ -coloração de vértices para um grafo simples  $G$  é uma atribuição de  $k$  cores para os vértices de  $G$  de modo que vértices adjacentes recebam cores diferentes. O número cromático  $x(G)$  de um grafo  $G$  é o menor número  $k$  tal que  $G$  admite uma  $k$ -coloração.

Em nosso exemplo cada recipiente corresponde a uma cor e o número mínimo de recipientes corresponde ao número cromático do grafo.

**Exercício 37** Determine o número cromático dos seguintes grafos.

- a)  $G = (V, E)$ , onde  $V = \{a, b, c, d, e\}$  e  $E = \{ab, bc, cd, ce\}$ .
- b)  $G = (V, E)$ , onde  $V = \{a, b, c\}$  e  $E = \{ab, bc, ac\}$ .
- c)  $G = (V, E)$ , onde  $V = \{a, b, c, d, e\}$  e  $E = \{ac, ae, bc, be, dc, de\}$ .
- d)  $G = K_4$ .

Seja  $\Delta(G)$  o grau do vértice de maior grau em  $G$ . Será que existe alguma relação entre  $x(G)$  e  $\Delta(G)$ ? O resultado a seguir mostra que sim.

**Teorema 23** Em um grafo simples  $G$ , temos  $x(G) \leq \Delta(G) + 1$ .

**Prova:** Indução no número de vértices.

Base. Se  $G$  tem apenas um vértice, então  $\Delta(G) = 0$  e  $x(G) = 1$ .

Hipótese. O resultado vale para todo grafo de  $p$  vértices.

Passo Indutivo. Seja  $G$  um grafo de  $p + 1$  vértices:  $v_1, v_2, \dots, v_{p+1}$ . Como o grafo  $G - v_{p+1}$  possui  $p$  vértices e  $\Delta(G - v_{p+1}) \leq \Delta(G)$ , segue da hipótese que o número cromático de  $G - v_{p+1}$  é no máximo  $\Delta(G) + 1$ . Temos dois casos.

Caso 1)  $x(G - v_{p+1}) < \Delta(G) + 1$ . Podemos atribuir uma cor para  $v_{p+1}$  diferente de todas as outras já utilizadas obtendo uma coloração para  $G$  com não mais que  $\Delta(G) + 1$  cores.

Caso 2)  $x(G - v_{p+1}) = \Delta(G) + 1$ . Como o número de vizinhos de  $v_{p+1}$  é no máximo  $\Delta(G)$  e o número de cores disponíveis é  $\Delta(G) + 1$ , atribuímos uma cor para  $v_{p+1}$  diferente da cor de todos os seus vizinhos.

**Exercício 38** *O teorema anterior mostra que  $x(G) \leq \Delta(G) + 1$ . É possível mostrar que  $x(G) \leq \Delta(G)$ ? Por que?*

**Exercício 39** *Qual o número cromático de um grafo bipartido?*

**Exercício 40** *Qual o número cromático de um grafo completo?*

### 3.11 Coloração de Arestas

Considere um campeonato de futebol com 10 times em que todos os times jogam entre si. Todos os jogos ocorrem no fim de semana e cada time pode jogar no máximo uma vez por fim de semana. Qual é o número mínimo de semanas necessárias para concluir a liga?

Esta situação pode ser modelada através de um grafo  $G = (V, E)$ , com  $V = \{\text{times}\}$  e  $E = \{\text{jogos}\}$ . Devemos então associar cada aresta a uma semana de modo que arestas que incidem no mesmo vértice estejam associadas a semanas distintas. Para o nosso exemplo, é possível mostrar que o número mínimo de semanas é 9.

Situações como essa motivam a seguinte definição

**Definição 48** *Uma  $k$ -coloração de arestas para um grafo simples  $G$  é uma atribuição de  $k$  cores para as arestas de  $G$  de modo que arestas com extremidades em comum recebam cores diferentes. O índice cromático  $x'(G)$  de um grafo  $G$  é o menor número  $k$  tal que  $G$  admite uma  $k$ -coloração de arestas.*

Em nosso exemplo cada semana corresponde a uma cor e o número mínimo de semanas corresponde ao índice cromático do grafo.

**Exercício 41** *Determine o índice cromático dos seguintes grafos.*

- a)  $G = (V, E)$ , onde  $V = \{a, b, c, d, e\}$  e  $E = \{ab, bc, cd, ce\}$ .
- b)  $G = (V, E)$ , onde  $V = \{a, b, c\}$  e  $E = \{ab, bc, ac\}$ .
- c)  $G = (V, E)$ , onde  $V = \{a, b, c, d, e\}$  e  $E = \{ac, ae, bc, be, dc, de\}$ .
- d)  $G = K_4$ .

O teorema a seguir caracteriza o índice cromático para a classe dos grafos bipartidos.

**Teorema 24** *Se  $G$  é bipartido então  $\chi'(G) = \Delta(G)$ .*

**Prova:** Indução no número de arestas.

Base. Se  $G$  tem apenas uma aresta, então  $\Delta(G) = 1$  e  $\chi'(G) = 1$

Hipótese. Se um grafo bipartido simples tem  $p$  arestas, então seu grau máximo é igual ao seu índice cromático.

Passo Indutivo. Seja  $G$  um grafo bipartido de  $p + 1$  arestas e seja  $e$  uma aresta de  $G$ . Logo,  $G - e$  é bipartido e possui  $p$  arestas. Portanto, segue da hipótese que  $G - e$  admite uma coloração com  $\Delta(G - e)$  cores. Falta, entretanto, atribuir uma cor a aresta  $e = (v, w)$ .

Caso 1)  $\Delta(G - e) = \Delta(G) - 1$ . Neste caso podemos atribuir à aresta  $e$  uma cor que não foi utilizada na coloração para  $G - e$ .

Caso 2)  $\Delta(G - e) = \Delta(G)$ . Dividimos em dois subcasos.

Subcaso 2.1) Existe uma cor, dentre as  $\Delta(G)$  cores utilizadas na coloração, que não está atribuída a nenhuma aresta que incide em  $v$  ou  $w$ . Neste caso, basta atribuir esta cor a  $e$  que não haverá conflito.

Subcaso 2.2) Toda cor utilizada para colorir  $G - e$  gera conflito. Seja  $A$  uma cor que não é atribuída a nenhuma aresta incidente em  $v$  na coloração do grafo  $G - e$ . Note que tal cor tem que existir, caso contrário o grau de  $v$  em  $G - e$  seria  $\Delta(G)$  e como consequência o grau de  $v$  em  $G$  seria  $\Delta(G) + 1$ , o que não é possível. De forma análoga, seja  $B$  uma cor que não é atribuída a nenhuma aresta incidente em  $w$  na coloração do grafo  $G - e$ . Note que  $A \neq B$ , caso contrário a cor  $A$  não geraria conflito e poderia ser atribuída a  $e$ .

Considere o conjunto de vértices que podem ser alcançados a partir de  $v$  por um caminho que alterna arestas de cores  $B$  e  $A$  em  $G - e$ . O nó  $w$  não pertence a este conjunto, caso contrário o grafo  $G$  apresentaria um ciclo de comprimento ímpar. Logo, trocando as cores  $B$  por  $A$  ao longo deste caminho



obtemos uma nova coloração para  $G - e$  com no máximo  $\Delta(G)$  cores <sup>4</sup>. Nesta coloração, entretanto, podemos atribuir a cor  $B$  a  $e$  sem gerar conflitos. ■

O seguinte teorema, enunciado sem a sua prova, diz respeito ao índice cromático de qualquer grafo simples.

**Teorema 25 (Vizing64)** *Se  $G$  é simples então  $\Delta(G) \leq x'(G) \leq \Delta(G) + 1$ .*

Para grafos completos temos o seguinte resultado.

**Teorema 26**  $x'(K_{2n}) = 2n - 1$  e  $x'(K_{2n+1}) = 2n + 1$

**Prova:** Primeiro consideramos  $K_{2n+1}$ . Vamos assumir que  $x'(K_{2n+1}) = 2n$  e mostrar que isto leva a uma contradição. Como  $\Delta(K_{2n+1}) = 2n$ , podemos ter no máximo  $n$  arestas da mesma cor, caso contrário haveriam duas arestas de uma mesma cor incidindo em um mesmo vértice. Logo, com  $2n$  cores podemos colorir no máximo  $2n \times n = 2n^2$  arestas. Como o total de arestas em  $K_{2n+1}$  é  $\binom{2n+1}{2} = 2n^2 + n$ , algumas arestas ficariam sem cor, o que contradiz a nossa hipótese. Como  $x'(K_{2n+1}) \neq 2n$ , segue do Teorema 25 que  $x'(K_{2n+1}) = 2n + 1$ .

De fato, podemos explicitar uma  $2n+1$ -coloração para  $K_{2n+1}$ . Atribuímos um número de 1 a  $2n+1$  a cada vértice de  $K_{2n+1}$  de modo que vértices distintos recebam números distintos. A seguir atribuímos a cor  $(i+j) \bmod (2n+1)$  a aresta que liga o vértice de número  $i$  ao de número  $j$ . Convidamos o leitor a provar que esta coloração é válida.

Consideramos agora  $K_{2n}$ . Sejam  $v_1, \dots, v_{2n}$  os vértices de  $K_{2n}$ . Para colorir  $K_{2n}$  com  $2n - 1$  cores, primeiro colorimos as arestas de  $K_{2n} - v_{2n}$  atribuindo a cor  $(i+j) \bmod (2n-1)$  a aresta que liga o vértice  $v_i$  ao vértice  $v_j$ . Falta colorir então as arestas que incidem no vértice  $v_{2n}$ . Atribuímos a cor  $2i \bmod (2n-1)$  para colorir a aresta que liga  $v_i$  a  $v_{2n}$ . Convidamos o leitor, novamente, a provar que esta coloração é válida. ■

---

<sup>4</sup>convidamos o leitor a checar este fato

**Exercício 42** *Considere uma liga de futebol com 11 times em que cada fim de semana cada time pode jogar no máximo uma vez. A liga termina quando todos os times já tiverem jogado entre si. Qual é o número mínimo de semanas necessárias para concluir a liga?*

**Exercício 43** *Podemos afirmar que a desigualdade  $x(G) \leq x'(G)$  é sempre válida? Por que?*