

Princípio da multiplicação:

- Considere eventos E_1, E_2, \dots, E_k

Princípio da multiplicação:

- Considere eventos E_1, E_2, \dots, E_k
- Evento E_i pode ocorrer de n_i formas diferentes, independente dos outros eventos

Princípio da multiplicação:

- Considere eventos E_1, E_2, \dots, E_k
- Evento E_i pode ocorrer de n_i formas diferentes, independente dos outros eventos
- 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

Princípio da multiplicação:

- Considere eventos E_1, E_2, \dots, E_k
- Evento E_i pode ocorrer de n_i formas diferentes, independente dos outros eventos
- 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

Algoritmo para gerar todas as palavras com letras em um alfabeto

Princípio da multiplicação:

- Considere eventos E_1, E_2, \dots, E_k
- Evento E_i pode ocorrer de n_i formas diferentes, independente dos outros eventos
- 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

Algoritmo para gerar todas as palavras com letras em um alfabeto

Algoritmo para gerar todas as palavras sem repetição de letras

Permutações

Número de **sequências** de r objetos distintos que podem ser formadas a partir de um conjunto de n objetos distintos

Número de **sequências** de r objetos distintos que podem ser formadas a partir de um conjunto de n objetos distintos

Ordem importa

Permutações

Número de **sequências** de r objetos distintos que podem ser formadas a partir de um conjunto de n objetos distintos

Ordem importa

Geralmente vamos chamar essa quantidade de $P(n, r)$

Número de **sequências** de r objetos distintos que podem ser formadas a partir de um conjunto de n objetos distintos

Ordem importa

Geralmente vamos chamar essa quantidade de $P(n, r)$

Ja vimos problemas assim:

- Permutações de um conjunto ($n = p$) \equiv Traveling Salesman Problem
- Palavras sem repetição de letras

Exemplo

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

Evento E_1 = objeto na primeira posição

E_2 = objeto na segunda posição

Exemplo

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

Evento E_1 = objeto na primeira posição

E_2 = objeto na segunda posição

E_1 tem 4 possibilidades, E_2 tem 3 possibilidades

Exemplo

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

Evento E_1 = objeto na primeira posição

E_2 = objeto na segunda posição

E_1 tem 4 possibilidades, E_2 tem 3 possibilidades

Pelo princípio multiplicativo, temos $4 \cdot 3 = 12$ possibilidades

Em geral, aplicando o princípio multiplicativo, obtemos que

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

Exemplo

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

Exemplo (Variante TSP)

Existem 12 localidades $\{L_1, \dots, L_{12}\}$, sabemos o custo c_{ij} de ir de L_i a L_j . Um caminhão só pode entregar 4 mercadorias hoje. Se entregar a mercadoria na localidade L_i , ganha g_i

Desejamos escolher em quais 4 localidades entregar, e em que ordem, de modo a maximizar lucro (ganho - custo)

Exemplo (Variante TSP)

Existem 12 localidades $\{L_1, \dots, L_{12}\}$, sabemos o custo c_{ij} de ir de L_i a L_j . Um caminhão só pode entregar 4 mercadorias hoje. Se entregar a mercadoria na localidade L_i , ganha g_i

Desejamos escolher em quais 4 localidades entregar, e em que ordem, de modo a maximizar lucro (ganho - custo)

Q: Quantas possíveis soluções existem para esse problema?

Exemplo (Variante TSP)

Existem 12 localidades $\{L_1, \dots, L_{12}\}$, sabemos o custo c_{ij} de ir de L_i a L_j . Um caminhão só pode entregar 4 mercadorias hoje. Se entregar a mercadoria na localidade L_i , ganha g_i

Desejamos escolher em quais 4 localidades entregar, e em que ordem, de modo a maximizar lucro (ganho - custo)

Q: Quantas possíveis soluções existem para esse problema?

A: $P(12, 4) = 12 \cdot 11 \cdot 10 \cdot 9 = 11.880$

Permutações

Podemos ter **restrições** nas permutações

Permutações

Podemos ter **restrições** nas permutações

Temos que usar principio da multiplicação várias vezes

Permutações

Podemos ter **restrições** nas permutações

Temos que usar principio da multiplicação várias vezes

Exemplo

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

Podemos ter **restrições** nas permutações

Temos que usar principio da multiplicação várias vezes

Exemplo

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

Podemos tomar decisão para construir essas permutações da seguinte forma:

- Decidir ordem das consoantes BLGR (evento E_1)
- Decidir posição vogal U (evento E_2)
- Decidir posição vogal A (evento E_3)
- Decidir posição vogal O (evento E_4)

Permutações

Podemos ter **restrições** nas permutações

Temos que usar princípio da multiplicação várias vezes

Exemplo

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

Podemos tomar decisão para construir essas permutações da seguinte forma:

- Decidir ordem das consoantes BLGR (evento E_1) **4! poss**
- Decidir posição vogal U (evento E_2) **5 poss**
- Decidir posição vogal A (evento E_3) **4 poss**
- Decidir posição vogal O (evento E_4) **3 poss**

Total: **4! · 5 · 4 · 3**

Exemplo

Uma companhia de entrega entrega 3 tipos de produtos: eletrônicos, roupas, e comida. Um caminhão precisa fazer 4 entregas de eletrônicos, 7 de roupas, e 3 de comida.

Existe uma restrição de que todos os produtos do mesmo tipo tem que ser entregue de forma consecutiva (e.g., entregar todos os eletrônicos, depois todas as roupas, depois todas as comidas)

Exemplo

Uma companhia de entrega entrega 3 tipos de produtos: eletrônicos, roupas, e comida. Um caminhão precisa fazer 4 entregas de eletrônicos, 7 de roupas, e 3 de comida.

Existe uma restrição de que todos os produtos do mesmo tipo tem que ser entregue de forma consecutiva (e.g., entregar todos os eletrônicos, depois todas as roupas, depois todas as comidas)

Q: Quantos trajetos são possíveis? (Note que consideramos a ordem em que os produtos do mesmo tipo são entregues.)

Possibilidades para a **sequencia de tipos de produtos** (eletronicos, roupas, comida)

Possibilidades para a **sequencia de tipos de produtos** (eletronicos, roupas, comida)

- ERC, ECR, REC, RCE, CER, CRE: **3!** possibilidades (princípio da multiplicação)

Possibilidades para a **sequencia de tipos de produtos** (eletronicos, roupas, comida)

- ERC, ECR, REC, RCE, CER, CRE: **3! possibilidades**
(princípio da multiplicação)

Para cada uma destas possibilidades, devemos escolher a sequencia dos **eletronicos**, das **roupas**, e das comidas

- **4! poss. eletronicos**, **7! poss. roupas**, **3! poss. comidas**
(princípio da multiplicação)

Permutações

Possibilidades para a **sequencia de tipos de produtos** (eletronicos, roupas, comida)

- ERC, ECR, REC, RCE, CER, CRE: **3!** possibilidades (princípio da multiplicação)

Para cada uma destas possibilidades, devemos escolher a sequencia dos **eletronicos**, das **roupas**, e das comidas

- **4!** poss. eletronicos, **7!** poss. roupas, **3!** poss. comidas (princípio da multiplicação)

Total de possibilidades: **3!7!4!3!** (de novo, princípio da multiplicacao)

Geração: Sabemos gerar todas as sequências de r objetos distintos a partir de um conjunto de n objetos distintos:

- Equivalente a gerar todas a “palavras” com r letras de um alfabeto com n “letras” (exercício ultima aula)

Vetor P : vai guardando palavra parcialmente preenchida

```
PROCEDIMENTO GeraPalavrasSemRepeticao(i,P)  #P[i+1], P[i+2],..., P[n]
estão preenchida
  If  $i = 0$   #P está totalmente preenchido
    Imprima a palavra  $P$ 
  Else
    For  $j = 1$  to  $s$ 
      If letra  $P[i]$  não aparece em  $P[i+1], \dots, P[n]$   # não foi utilizada
         $P[i] \leftarrow$  letra  $Letra[j]$ 
        GeraPalavras( $i-1, P$ )
      End If
    End For
  End If

MAIN
   $Letra \leftarrow$  vetor com  $s$  posições contendo as letras possíveis
   $P \leftarrow$  vetor de  $n$  posições
  GeraPalavras( $n, P$ )
```

Exercício 1. Google recebeu uma busca por um produto, e deseja escolher os 3 primeiros resultados a serem mostrados. Google tem 10 possíveis resultados, sendo

- 3 da Empresa A,
- 2 da Empresa B, e
- 5 da Empresa C

Google quer mostrar 1 resultado de cada empresa. **Quantas possibilidades são possíveis?**

Exercício 2. Escreva um algoritmos que gere todas as permutações de $\{1, 2, 3, 4, 5\}$ onde 1 e 2 apareçam juntos (em qualquer ordem).

Quantas permutações seu algoritmo gera?