Árvores

Recapitulando:

Definição (Árvore)

*Uma **árvore** é um grafo conexo e sem ciclos (acíclico)*

Recapitulando:

Definição (Árvore)

*Uma **árvore** é um grafo conexo e sem ciclos (acíclico)*

Propriedade importante:

Teorema (   )

Sejam u e v dois vértices distintos de uma árvore T. Existe exatamente   caminho(s) entre u e v em T

Recapitulando:

Definição (Árvore)

*Uma **árvore** é um grafo conexo e sem ciclos (acíclico)*

Propriedade importante:

Teorema (Árvores tem caminho único)

*Sejam u e v dois vértices distintos de uma árvore T. Existe exatamente **1** caminho(s) entre u e v em T*

Recapitulando:

Definição (Árvore)

*Uma **árvore** é um grafo conexo e sem ciclos (acíclico)*

Propriedade importante:

Teorema (Árvores tem caminho único)

*Sejam u e v dois vértices distintos de uma árvore T. Existe exatamente **1** caminho(s) entre u e v em T*

Consequência:

Teorema

Se removermos qualquer aresta (u, v) de uma árvore, a quebramos em árvores (componentes conexos)

Recapitulando:

Definição (Árvore)

*Uma **árvore** é um grafo conexo e sem ciclos (acíclico)*

Propriedade importante:

Teorema (Árvores tem caminho único)

*Sejam u e v dois vértices distintos de uma árvore T. Existe exatamente **1** caminho(s) entre u e v em T*

Consequência:

Teorema

*Se removermos qualquer aresta (u, v) de uma árvore, a quebramos em **duas** árvores (componentes conexos)*

Teorema

Se T é uma árvore não vazia, então $\#arestas(T) = \#nós(T) - 1$

Exemplo:

Teorema

Se T é uma árvore não vazia, então $\#arestas(T) = \#nós(T) - 1$

Prova: Indução forte no

Teorema

Se T é uma árvore não vazia, então $\#arestas(T) = \#nós(T) - 1$

Prova: Indução forte no número de arestas

Teorema

Se T é uma árvore não vazia, então $\#arestas(T) = \#nós(T) - 1$

Prova: Indução forte no número de arestas

Caso base: 0 arestas. Então G tem exatamente nó(s),

Teorema

Se T é uma árvore não vazia, então $\#arestas(T) = \#nós(T) - 1$

Prova: Indução forte no número de arestas

Caso base: 0 arestas. Então G tem exatamente 1 nó(s),

Teorema

Se T é uma árvore não vazia, então $\#arestas(T) = \#nós(T) - 1$

Prova: Indução forte no número de arestas

Caso base: 0 arestas. Então G tem exatamente 1 nó(s), e temos $\#\text{arestas} = \#\text{nós} - 1$, OK!

Teorema

Se T é uma árvore não vazia, então $\#arestas(T) = \#nós(T) - 1$

Prova: Indução forte no número de arestas

Caso base: 0 arestas. Então G tem exatamente 1 nó(s), e temos #arestas = #nós −1, OK!

Passo indutivo: Suponha que o resultado valha para todas as árvores com $1, 2, \ldots, m$ arestas

Teorema

Se T é uma árvore não vazia, então $\#arestas(T) = \#nós(T) - 1$

Prova: Indução forte no número de arestas

Caso base: 0 arestas. Então G tem exatamente 1 nó(s), e temos #arestas = #nós −1, OK!

Passo indutivo: Suponha que o resultado valha para todas as árvores com $1, 2, \ldots, m$ arestas

Vamos provar pra árvore com $m + 1$ arestas

Teorema

Se T é uma árvore não vazia, então $\#arestas(T) = \#nós(T) - 1$

Seja T uma árvore com $m + 1$ arestas.

Teorema

Se T é uma árvore não vazia, então $\#arestas(T) = \#nós(T) - 1$

Seja T uma árvore com $m + 1$ arestas.

Q: Como relacionar com árvores com $\leq m$ arestas?

Teorema

Se T é uma árvore não vazia, então $\#arestas(T) = \#nós(T) - 1$

Seja T uma árvore com $m + 1$ arestas.

Q: Como relacionar com árvores com $\leq m$ arestas?

A: Remova uma aresta (u, v) qualquer. Isso quebra a árvore em duas árvores T_u e T_v (contendo u e v respectivamente)

Teorema

Se T é uma árvore não vazia, então #arestas(T) = #nós(T) -1

Seja T uma árvore com $m + 1$ arestas.

Q: Como relacionar com árvores com $\leq m$ arestas?

A: Remova uma aresta (u, v) qualquer. Isso quebra a árvore em duas árvores T_u e T_v (contendo u e v respectivamente)

Vendo o que queremos provar, temos que relacionar o #arestas e #nós de T com as novas árvores:

Teorema

Se T é uma árvore não vazia, então #arestas(T) = #nós(T) −1

Seja T uma árvore com $m + 1$ arestas.

Q: Como relacionar com árvores com $\leq m$ arestas?

A: Remova uma aresta (u, v) qualquer. Isso quebra a árvore em duas árvores T_u e T_v (contendo u e v respectivamente)

Vendo o que queremos provar, temos que relacionar o #arestas e #nós de T com as novas árvores:

#arestas(T)

$\square$

Teorema

Se T é uma árvore não vazia, então #arestas(T) = #nós(T) −1

Seja T uma árvore com $m+1$ arestas.

Q: Como relacionar com árvores com $\leq m$ arestas?

A: Remova uma aresta (u, v) qualquer. Isso quebra a árvore em duas árvores T_u e T_v (contendo u e v respectivamente)

Vendo o que queremos provar, temos que relacionar o #arestas e #nós de T com as novas árvores:

$$\#\text{arestas}(T) = \#\text{arestas}(T_u) + \#\text{arestas}(T_v) + 1$$

$\square$

Teorema

Se T é uma árvore não vazia, então #arestas(T) = #nós(T) −1

Seja T uma árvore com $m+1$ arestas.

Q: Como relacionar com árvores com $\leq m$ arestas?

A: Remova uma aresta (u, v) qualquer. Isso quebra a árvore em duas árvores T_u e T_v (contendo u e v respectivamente)

Vendo o que queremos provar, temos que relacionar o #arestas e #nós de T com as novas árvores:

$$\begin{aligned}\#\text{arestas}(T) &= \#\text{arestas}(T_u) + \#\text{arestas}(T_v) + 1 \\ &= \Big(\#\text{nós}(T_u) - 1\Big) + \Big(\#\text{nós}(T_v) - 1\Big) + 1 \qquad \text{(hip indut)}\end{aligned}$$

$\square$

Teorema

Se T é uma árvore não vazia, então #arestas(T) = #nós(T) −1

Seja T uma árvore com $m+1$ arestas.

Q: Como relacionar com árvores com $\leq m$ arestas?

A: Remova uma aresta (u, v) qualquer. Isso quebra a árvore em duas árvores T_u e T_v (contendo u e v respectivamente)

Vendo o que queremos provar, temos que relacionar o #arestas e #nós de T com as novas árvores:

$$\begin{aligned}\#\text{arestas}(T) &= \#\text{arestas}(T_u) + \#\text{arestas}(T_v) + 1 \\ &= \Big(\#\text{nós}(T_u) - 1\Big) + \Big(\#\text{nós}(T_v) - 1\Big) + 1 \qquad \text{(hip indut)} \\ &= \#\text{nós}(T) - 1 - 1 + 1\end{aligned}$$

$\square$

Teorema

Se T é uma árvore não vazia, então #arestas(T) = #nós(T) -1

Seja T uma árvore com $m+1$ arestas.

Q: Como relacionar com árvores com $\leq m$ arestas?

A: Remova uma aresta (u, v) qualquer. Isso quebra a árvore em duas árvores T_u e T_v (contendo u e v respectivamente)

Vendo o que queremos provar, temos que relacionar o #arestas e #nós de T com as novas árvores:

$$\begin{aligned}
\#\text{arestas}(T) &= \#\text{arestas}(T_u) + \#\text{arestas}(T_v) + 1 \\
&= \Big(\#\text{nós}(T_u) - 1\Big) + \Big(\#\text{nós}(T_v) - 1\Big) + 1 \qquad \text{(hip indut)} \\
&= \#\text{nós}(T) - 1 - 1 + 1 \\
&= \#\text{nós}(T) - 1 \qquad \square
\end{aligned}$$

Árvore Geradora Mínima

Árvore Geradora Mínima

Problema: Temos várias cidades, e queremos construir rodovias de forma a conectar todas elas (ou seja, dá pra ir de qualquer uma a qualquer outra)

Árvore Geradora Mínima

Problema: Temos várias cidades, e queremos construir rodovias de forma a conectar todas elas (ou seja, dá pra ir de qualquer uma a qualquer outra)

Temos custos diferentes de construir ruas entre cada duas cidades

Árvore Geradora Mínima

Problema: Temos várias cidades, e queremos construir rodovias de forma a conectar todas elas (ou seja, dá pra ir de qualquer uma a qualquer outra)

Temos custos diferentes de construir ruas entre cada duas cidades

Qual a forma mais barata de conectar essas cidades?

Diversas aplicações:

Dithering

Cluster analysis

Max bottleneck paths

Real-time face verification

LDPC codes for error correction

Image registration with Renyi entropy

Find road networks in satellite and aerial imagery

Reducing data storage in sequencing amino acids in a protein

Model locality of particle interactions in turbulent fluid flows

Autoconfig protocol for Ethernet bridging to avoid cycles in a network

Approximation algorithms for NP-hard problems (e.g., TSP, Steiner tree)

Network design (communication, electrical, hydraulic, computer, road)

Podemos representar esse problema em termos de grafos da seguinte forma

Podemos representar esse problema em termos de grafos da seguinte forma

Entrada. Um grafo conexo $G = (V, E)$, com um custo c_e para cada aresta e (modela as **possíveis** conexões e seus custos)

Árvore Geradora Mínima

Podemos representar esse problema em termos de grafos da seguinte forma

Entrada. Um grafo conexo $G = (V, E)$, com um custo c_e para cada aresta e (modela as **possíveis** conexões e seus custos)

Saída. Um subconjunto de arestas $E' \subseteq E$ que satisfaz:

Árvore Geradora Mínima

Podemos representar esse problema em termos de grafos da seguinte forma

Entrada. Um grafo conexo $G = (V, E)$, com um custo c_e para cada aresta e (modela as **possíveis** conexões e seus custos)

Saída. Um subconjunto de arestas $E' \subseteq E$ que satisfaz:

1) Essas arestas formam uma árvore conectando todos os nós de G, ou seja, o grafo $T = (V, E')$ é uma árvore

2) O custo total dessas arestas E' (ou seja, $\sum_{e \in E'} c_e$) é menor possível

Árvore Geradora Mínima

Podemos representar esse problema em termos de grafos da seguinte forma

Entrada. Um grafo conexo $G = (V, E)$, com um custo c_e para cada aresta e (modela as **possíveis** conexões e seus custos)

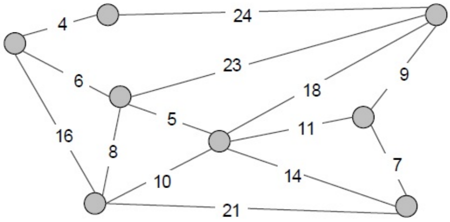
Saída. Um subconjunto de arestas $E' \subseteq E$ que satisfaz:

1) Essas arestas formam uma árvore conectando todos os nós de G, ou seja, o grafo $T = (V, E')$ é uma árvore

2) O custo total dessas arestas E' (ou seja, $\sum_{e \in E'} c_e$) é menor possível

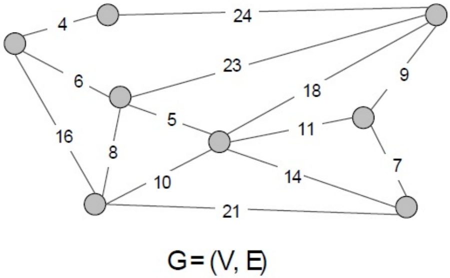
Chamada Árvore Geradora Mínima (AGM)

Exemplo:

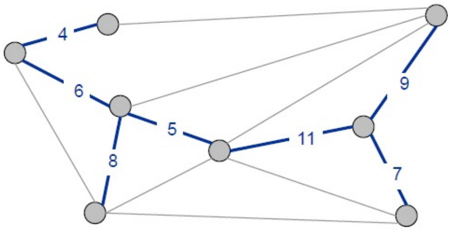


G = (V, E)

Exemplo:



Árvore Geradora Mínima:



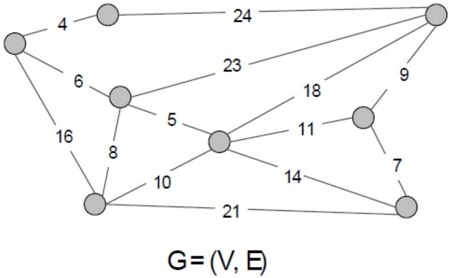
Árvore Geradora Mínima

Q: Estamos perdendo alguma coisa em pedir uma **árvore**, ou seja, **sem ciclos**?

Árvore Geradora Mínima

Q: Estamos perdendo alguma coisa em pedir uma **árvore**, ou seja, **sem ciclos**?

A: Não, se tivéssemos ciclos poderíamos deletar arestas e obter solução mais barata



Árvore Geradora Mínima

Q: Algoritmo para encontrar AGM?

Árvore Geradora Mínima

Q: Algoritmo para encontrar AGM?

Vamos assumir que as arestas tem custos distintos (não tem empate), pra simplificar

Árvore Geradora Mínima

Q: Algoritmo para encontrar AGM?

Vamos assumir que as arestas tem custos distintos (não tem empate), pra simplificar

Vamos provar uma propriedade que nos ajudará...

Proposição

Assuma custos distintos. Considere um ciclo C no grafo e seja (u, v) a aresta mais cara *no ciclo. Então a AGM do grafo essa aresta*

Exemplo:

Proposição

Assuma custos distintos. Considere um ciclo C no grafo e seja (u, v) a aresta mais cara no ciclo. Então a AGM do grafo não contém essa aresta

Exemplo:

Proposição

Assuma custos distintos. Considere um ciclo C no grafo e seja (u, v) a aresta mais no ciclo. Então a AGM do grafo não contém essa aresta

Prova: Por contradição assuma que a AGM T tem a aresta (u, v)

Proposição

Assuma custos distintos. Considere um ciclo C no grafo e seja (u, v) a aresta mais no ciclo. Então a AGM do grafo não contém essa aresta

Prova: Por contradição assuma que a AGM T tem a aresta (u, v)

Remova essa aresta (u, v), partindo T em duas árvores T_u e T_v

Proposição

Assuma custos distintos. Considere um ciclo C no grafo e seja (u, v) a aresta mais no ciclo. Então a AGM do grafo não contém essa aresta

Prova: Por contradição assuma que a AGM T tem a aresta (u, v)

Remova essa aresta (u, v), partindo T em duas árvores T_u e T_v

Q: O ciclo C tem **outra** aresta (u', v') com uma ponta em cada árvore T_u e T_v?

Proposição

Assuma custos distintos. Considere um ciclo C no grafo e seja (u, v) a aresta mais no ciclo. Então a AGM do grafo não contém essa aresta

Prova: Por contradição assuma que a AGM T tem a aresta (u, v)

Remova essa aresta (u, v), partindo T em duas árvores T_u e T_v

Q: O ciclo C tem **outra** aresta (u', v') com uma ponta em cada árvore T_u e T_v?

A: Sim! [verifique]

Proposição

Assuma custos distintos. Considere um ciclo C no grafo e seja (u, v) a aresta mais no ciclo. Então a AGM do grafo não contém essa aresta

Prova: Por contradição assuma que a AGM T tem a aresta (u, v)

Remova essa aresta (u, v), partindo T em duas árvores T_u e T_v

Q: O ciclo C tem **outra** aresta (u', v') com uma ponta em cada árvore T_u e T_v?

A: Sim! [verifique]

Como (u, v) é a mais cara do ciclo C, (u', v') é mais barata

Então pegue T e troque a aresta (u, v) pela aresta (u', v') (chame $T' = T - (u, v) + (u', v')$ esse novo grafo)

Então pegue T e troque a aresta (u, v) pela aresta (u', v') (chame $T' = T - (u, v) + (u', v')$ esse novo grafo)

O novo grafo T' é uma árvore [verifique] mais barata do que a Árvore Geradora Mínima T

$\Rightarrow$ **contradição!** $\square$

Q: Algoritmo? (baseado na proposição)

Q: Algoritmo? (baseado na proposição)

Algoritmo CalculaAGM(G)

If G tem ciclo

Encontre um ciclo C, e encontre sua aresta mais cara (u, v)

#roda recursivamente no grafo sem aresta cara

Return CalculaAGM($G - (u, v)$)

Else #G não tem ciclo

Q: Algoritmo? (baseado na proposição)

```
Algoritmo CalculaAGM(G)

If G tem ciclo
    Encontre um ciclo C, e encontre sua aresta mais cara (u, v)

    #roda recursivamente no grafo sem aresta cara
    Return CalculaAGM(G − (u, v))

Else #G não tem ciclo
    Retorne G
```

Q: Algoritmo? (baseado na proposição)

Algoritmo CalculaAGM(G)

If G tem ciclo
Encontre um ciclo C, e encontre sua aresta mais cara (u, v)

#roda recursivamente no grafo sem aresta cara
Return CalculaAGM($G - (u, v)$)

Else #G não tem ciclo
Retorne G

Obs: Na prática usamos algoritmos mais simples de implementar e mais eficientes: Kruskal, Prim, Boruvka

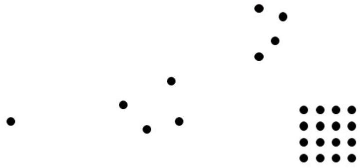
Aplicação de AGM: Clustering

Aplicação de AGM: Clustering

Clustering: Dados n objetos, queremos agrupá-los em grupos "coerentes"

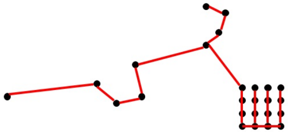
Distâncias: Temos valor especificando distancia/dissimilaridade de cada par de objetos

Geralmente especificamos o número de grupos desejados



Aplicação de AGM: Clustering

Q: Podemos usar AGM pra obter k grupos?

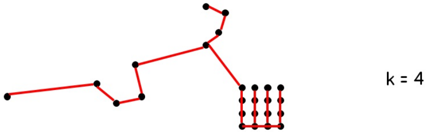


k = 4

Aplicação de AGM: Clustering

Q: Podemos usar AGM pra obter k grupos?

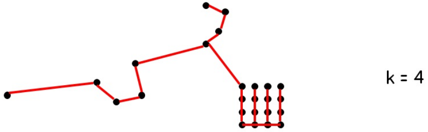
A: Removemos as **arestas mais caras** da AGM até obter k componentes conexos



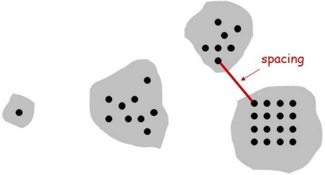
Aplicação de AGM: Clustering

Q: Podemos usar AGM pra obter k grupos?

A: Removemos as **arestas mais caras** da AGM até obter k componentes conexos

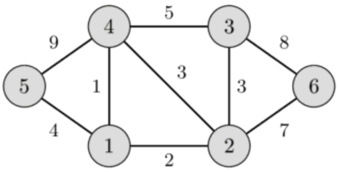


Esse procedimento é chamado single-linkage clustering. Maximiza a (menor) distância entre os grupos



Exercícios

Exercício 1: Compute a arvore geradora mínima para o grafo abaixo



Exercício 2: Responda:

a) Um grafo só tem uma árvore geradora mínima?

b) Considere um grafo conexo com n nós onde cada aresta tem custo 1. Podemos determinar o custo da sua AGM?

c) A aresta mais cara do grafo nunca está na sua AGM? Sempre está? Justifique

Exercícios

Exercício 3:* Prove por indução a corretude do algoritmo anterior: Se os custos das arestas são distinto e G é conexo, então o algoritmo **CalculaAGM**(G) retorna a arvore geradora mínima de G