Teoria dos Grafos

Valeriano A. de Oliveira, Socorro Rangel, Silvio A. de Araujo

Departamento de Matemática Aplicada

Capítulo 13: Árvores

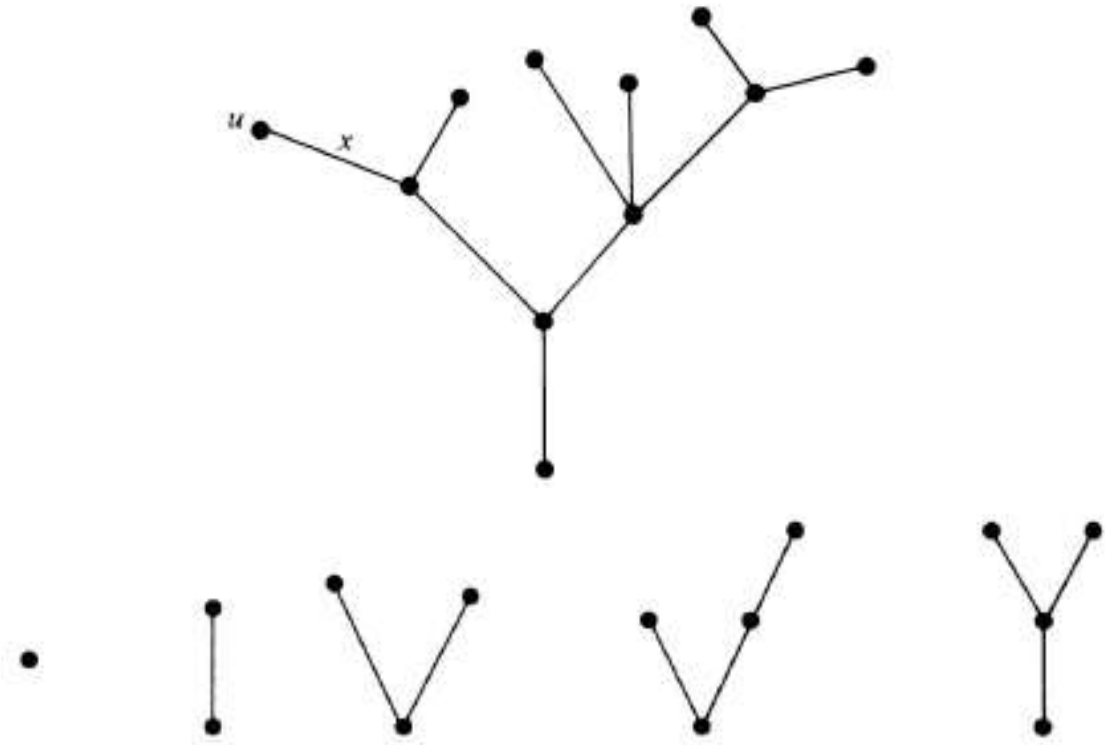
Preparado a partir do texto:
Rangel, Socorro. Teoria do Grafos, Notas de aula, IBILCE, Unesp, 2002-2013.

Outline

Definições

Definição

*Uma **árvore** é um grafo conexo que não possui circuitos.*



Definições

Definição

*Uma **árvore orientada** é um digrafo conexo que não possui circuitos ou semi-circuitos.*

Aplicações e Exemplos

- Aplicações:

  Construção de rodovias, instalação de redes em geral.

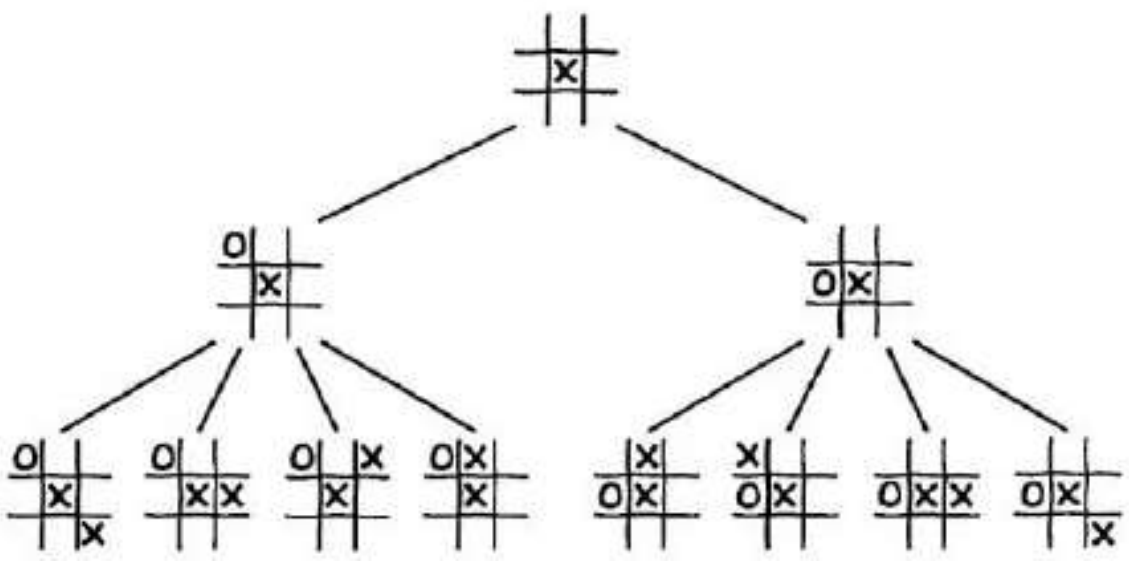
  Em alguns casos, para se mostrar um resultado para grafos é interessante começar mostrando para árvores.

- Exemplos:

  Vamos representar as situações a seguir através de grafos.

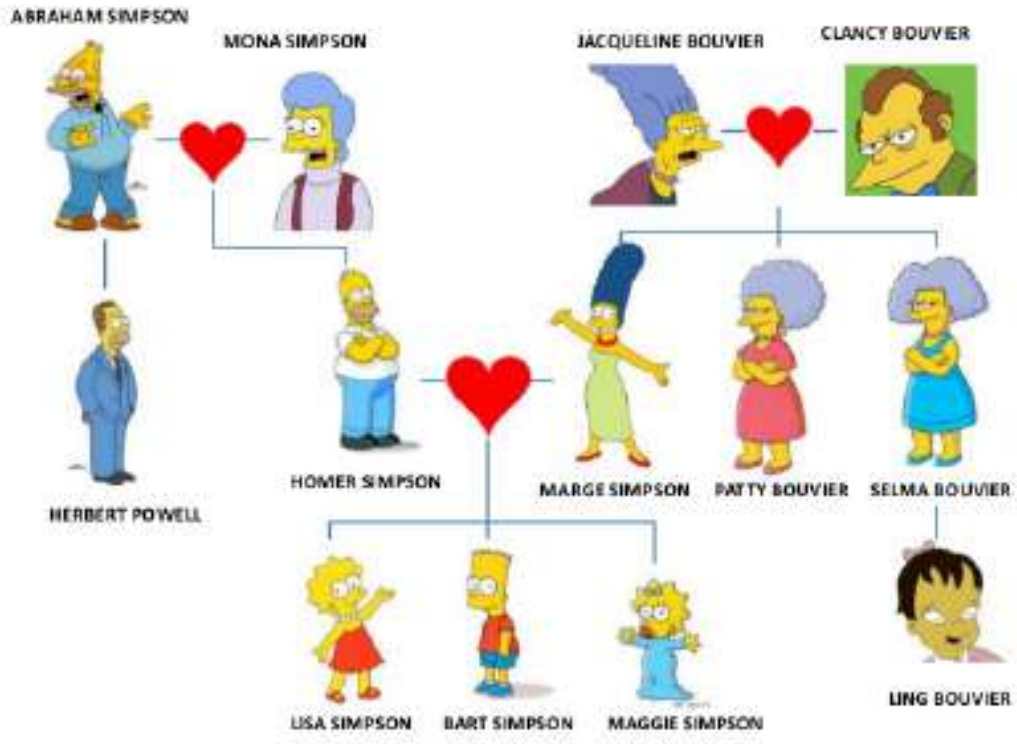
Jogo da velha

Vértices: os estados do jogo.
Arestas: existe uma aresta entre um estado do jogo e um estado que poder ser obtido através deste.



Árvore Genealógica

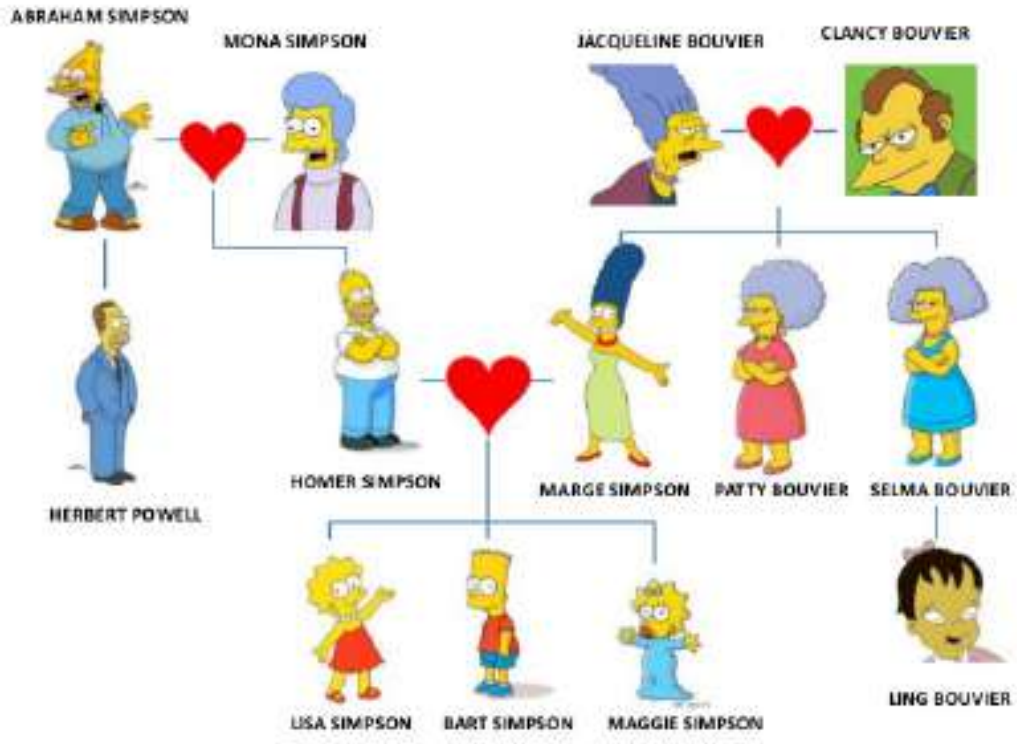
Vértices: pessoas.
Arestas: relação de parentesco em primeiro grau (mãe (pai) – filho(a)).



O que os grafos da situações acima possuem em comum?

Árvore Genealógica

Vértices: pessoas.
Arestas: relação de parentesco em primeiro grau (mãe (pai) – filho(a)).



O que os grafos da situações acima possuem em comum?

Propriedades de árvores

Teorema

Um grafo G é uma árvore se, e somente se, existir um e apenas um caminho entre cada par de vértices.

Demonstração. [$\Rightarrow$]

Se G é uma árvore, então, por definição, G é conexo e sem circuitos. Como G é conexo, então existe um caminho entre cada par de vértices.

Precisamos mostrar que este caminho é único. Vamos supor que existam dois caminhos distintos entre um par de vértices. Ora, se existem dois caminhos distintos entre um par de vértices então a união destes caminhos contém um circuito. Mas por hipótese, o grafo não possui circuitos, portanto existe apenas um caminho entre cada par de vértices.

[⇐]

Vamos mostrar que se existe um, e apenas um, caminho entre cada par de vértices, então G é uma árvore.

Como existe um caminho entre cada par de vértices, temos que G é conexo.

Vamos supor que G contenha um circuito. A existência de um circuito no grafo implica que existe pelo menos um par de vértices a, b tais que existem dois caminhos distintos entre a e b. Mas por hipótese existe um e apenas um caminho entre cada par de vértices e portanto o grafo não tem circuitos. Por definição um grafo conexo e sem circuitos é uma árvore. □

Propriedades de árvores

Teorema

Seja $G(V, A)$ um grafo com n vértices. As seguintes afirmações são equivalentes:

a) *G é uma árvore.*

b) *G é conexo e possui $n - 1$ arestas.*

c) *G possui $n - 1$ arestas e não possui circuitos.*

d) *Existe exatamente um caminho entre cada par de vértices.*

e) *G não contém circuitos, e para todo $v, w \in V$, a adição da aresta (v, w) produz no grafo exatamente um circuito.*

Observação

Qualquer uma destas afirmativas pode ser usada como definição de uma árvore.

Demonstração.

Para mostrar a equivalência das afirmativas temos que mostrar que a) $\Rightarrow$ b), b) $\Rightarrow$ c), etc.

Vamos mostrar apenas que a) $\Rightarrow$ b): Se G é uma árvore, então G é conexo e possui $n-1$ arestas.

Como por hipótese G é uma árvore, temos que G é conexo. Precisamos mostrar apenas que G possui $n-1$ arestas. Vamos mostrar usando indução matemática sobre n.

Vamos verificar o resultado para um valor particular de n. Por exemplo para $n=1$ e $n=2$.

Para $n=1$, temos 0 arestas.

Para $n=2$ temos 1 aresta.

Vamos supor agora que o resultado vale para um grafo G' com $k-1$ vértices. Isto é, se G' é uma árvore então G' é conexo e possui $k-2$ arestas.

Vamos acrescentar uma nova aresta (v, w) a este grafo. Para manter o grafo conexo e sem circuitos um e apenas um dos vértices em (v, w) pode pertencer a G'. Assim ao acrescentar a aresta (v, w) a G', precisamos acrescentar também um vértice. Assim teremos um novo grafo G'' com k vértices e $k-1$ arestas.

A forma como G'' foi construído garante que é conexo e sem circuitos. Portanto temos que G'' é uma árvore.

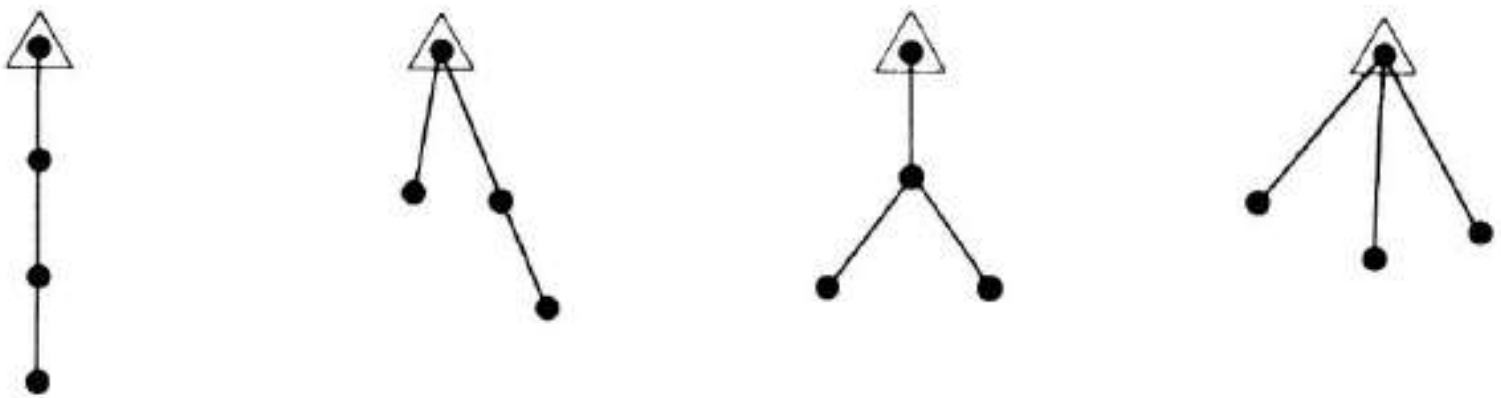
Mostramos assim que se G é uma árvore então G é conexo com $n-1$ arestas. □

Raízes e Árvores Binárias

Definição

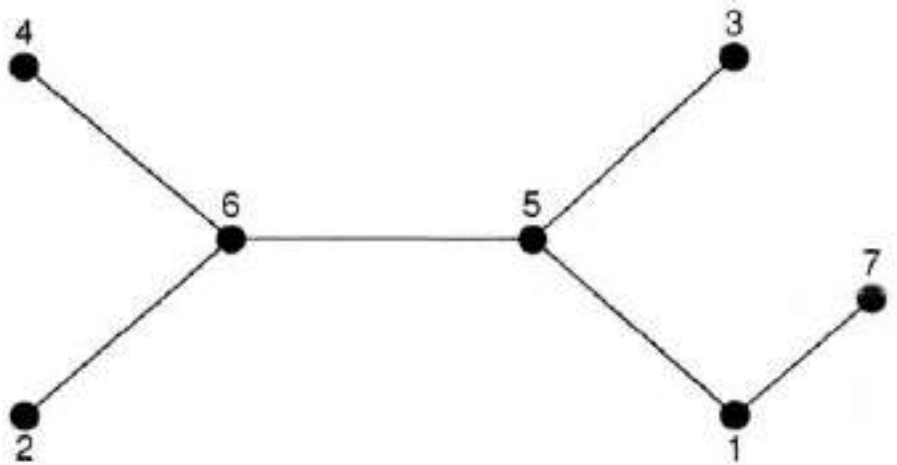
Uma árvore na qual podemos distinguir um determinado vértice, denominado ***vértice raiz****, é chamada de* ***árvore enraizada****.*

Por exemplo as árvores de 4 vértices abaixo são enraizadas.



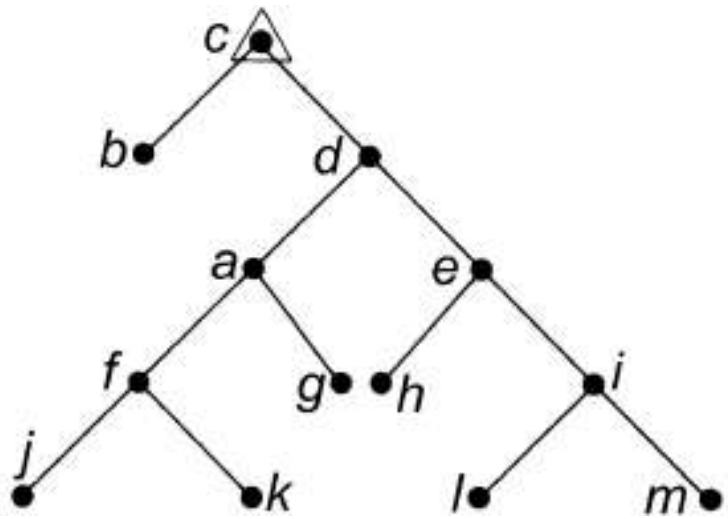
Em geral, o vértice raiz aparece naturalmente com a aplicação que o grafo representa.

Uma árvore não enraizada é chamada de árvore livre.



Raízes e Árvores Binárias

Se representarmos uma árvore enraizada com o vértice raiz posicionado na parte superior da figura, podemos definir níveis na árvore. Considere, por exemplo, a seguinte árvore enraizada:



Dizemos que o vértice raiz, c, está no nível zero; os vértices b e d no nível 1, os vértices a, e e e no nível 2, os vértices f, g, h e i no nível 3 e j, k, l e m no nível 4.

Definições

Definição

*A **distância** entre dois vértices v e w em um grafo G, denotada por $d(v, w)$, é igual ao comprimento do menor caminho entre v e w.*

Definição

*O **nível** de um vértice x em uma árvore enraizada é igual à distância entre o vértice raiz e o vértice x.*
*A **altura** de uma árvore enraizada é o comprimento do maior caminho existente na árvore a partir do vértice raiz.*

Definições

Definição

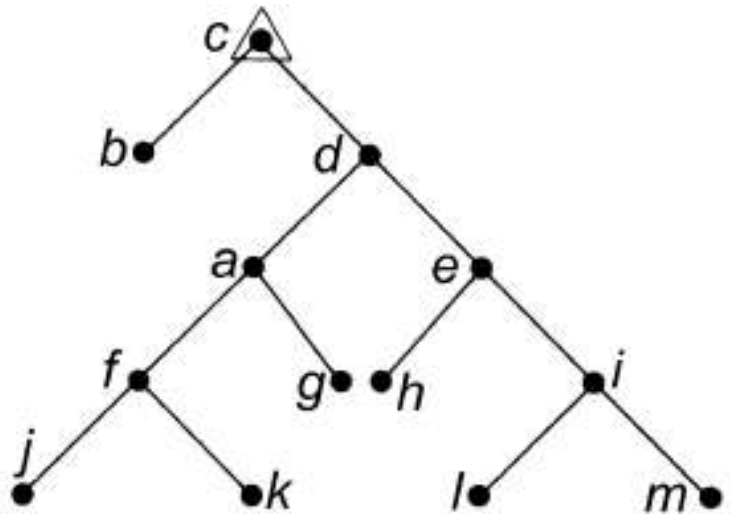
*Uma árvore **binária completa** é uma árvore enraizada tal que existe exatamente um vértice de grau dois e cada um dos vértices restantes tem grau 1 ou 3.*
Naturalmente o vértice de grau 2 é o vértice raiz da árvore.

Definição

*Um vértice não pendente em uma árvore é chamado de **vértice interno**.*

Exemplo

Abaixo temos um exemplo de árvore binária completa.



Os vértices d, a, e, f, i são vértices internos.

Propriedades de árvores binárias

Proposição

O número de vértices em uma árvore binária completa (com três ou mais vértices) é sempre ímpar.

Demonstração.

Existe exatamente um vértice de grau par. Os $n-1$ vértices restantes tem grau ímpar. Mas sabemos que o número de vértice com grau ímpar é par. Portanto, se $n-1$ é par, n é ímpar. $\square$

Propriedades de árvores binárias

Proposição

Quantos vértices pendentes existem em uma árvore binária completa com n vértices?

Demonstração.

Seja p o número de vértices pendentes. Então, existem $n - p - 1$ vértices de grau 3. Além disso,

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

onde usamos uma propriedade de árvores na última igualdade. Somando os graus dos vértices pendentes, internos e raiz tem-se $p + 3(n - p - 1) + 2 = 2(n - 1)$. Logo $p = (n + 1)/2$. □

Exemplo

Quantos jogos são necessários em um torneio de tênis com 56 inscritos?

Se representarmos a competição através de uma árvore binária, teremos que os vértices pendentes são os inscritos e os vértices internos mais a raiz os jogos.

Assim, queremos calcular o número de vértices internos mais a raiz em uma árvore binária com 56 vértices pendentes.

Propriedades de árvores binárias

Proposição

Seja T uma árvore binária completa de altura h e p vértices pendentes. Então:

$$p \leq 2^h, \tag{1}$$

$$h \geq \lceil \log_2 p \rceil = \lceil \log_2 ((n+1)/2) \rceil = \lceil \log_2 (n+1) - 1 \rceil. \tag{2}$$

Demonstração.

Este resultado pode ser provado da seguinte forma:

(1) princípio de indução;

(2) aplica-se logaritmo a ambos os lados da expressão em (1).

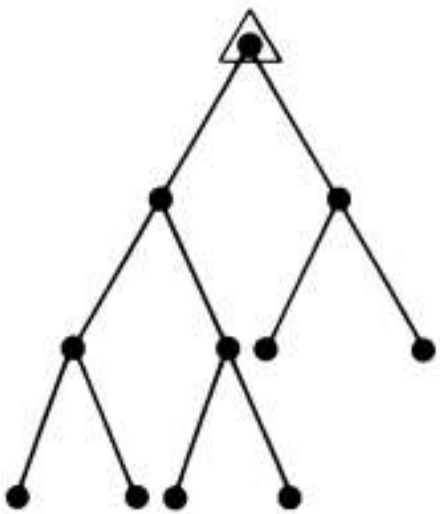
Exercício

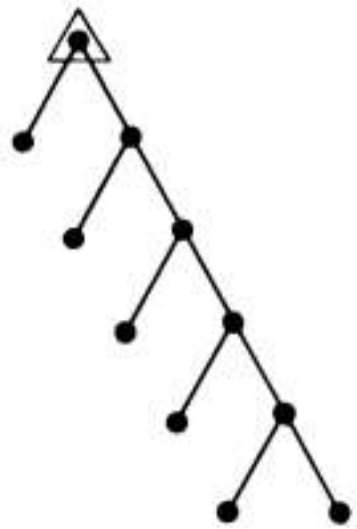
Qual é o nível máximo de uma árvore binária com n vértices?

Observando que a maior altura da árvore será obtida com o menor número possível de vértices em cada nível, temos que em uma árvore com n vértices,

$$h \leq (n-1)/2.$$

Abaixo temos um exemplo de níveis mínimo e máximo em uma árvore binária completa com 11 vértices.





Procedimentos de busca em árvores

Árvore binárias são muito utilizadas em procedimentos de busca.

Considere que cada vértice da árvore representa um teste com duas respostas possíveis.

Iniciando o teste no vértice raiz, a resposta ao teste nos leva a um dos dois vértices do próximo nível onde novos testes são efetuados.

Quando atingimos um vértice pendente (o objetivo da busca) o procedimento de busca se encerra.

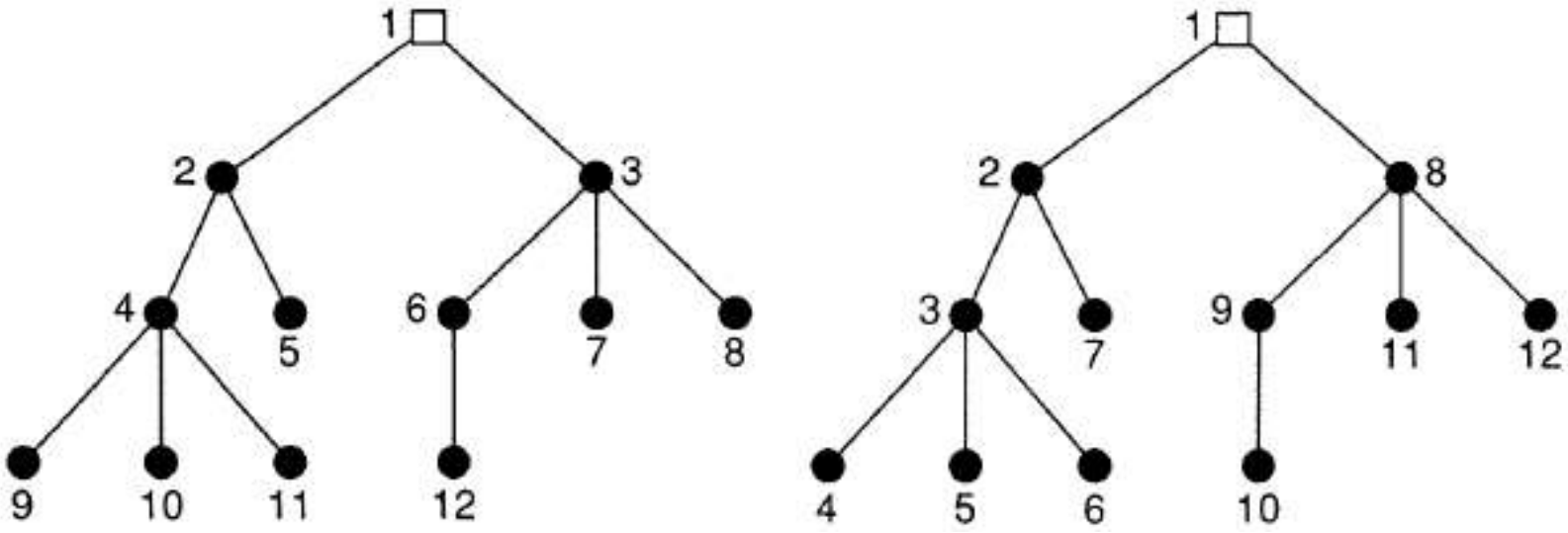
Em algumas aplicações é importante construir árvores tais que os vértices pendentes estejam o mais próximo possível do vértice raiz.

Procedimentos de busca em árvores

Há dois procedimentos de busca bem conhecidos:

- Busca em profundidade (depth first);
- Busca em largura (breadth first).

Exemplo:



Centro de um grafo

Definição

Definimos a ***excentricidade de um vértice*** v *em um grafo* $G(V, A)$*, denotada por* $E(v)$*, como sendo o valor da distância máxima entre* v *e os outros vértices do grafo:*

$$E(v) = \max_{w \in V} d(v, w).$$

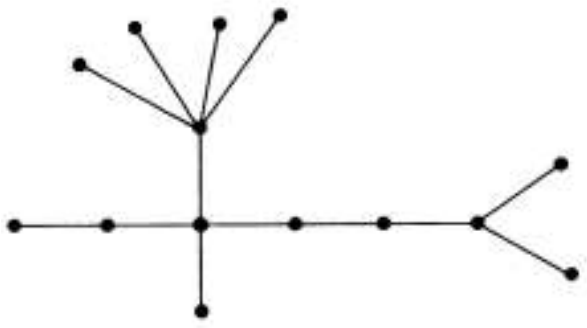
O centro de um grafo é igual ao subconjunto de vértices com excentricidade mínima.

Exemplo 1

Considere um grupo de 14 pessoas. Suponha que a comunicação entre as pessoas deste grupo esteja representada através do grafo, onde os vértices representam as pessoas e as arestas representam a possibilidade de comunicação entre duas pessoas. Supondo que todos os membros podem ser alcançados diretamente ou através de outros membros do grupo, temos que o grafo é conexo. Se usado o critério de facilidade de acesso às pessoas quem deverá ser escolhido como líder do grupo?

A resposta a esta questão envolve o conceito de centro de um grafo, pois a melhor escolha para líder do grupo seria a pessoa que tivesse acesso mais fácil às outras pessoas do grupo, direta ou indiretamente.

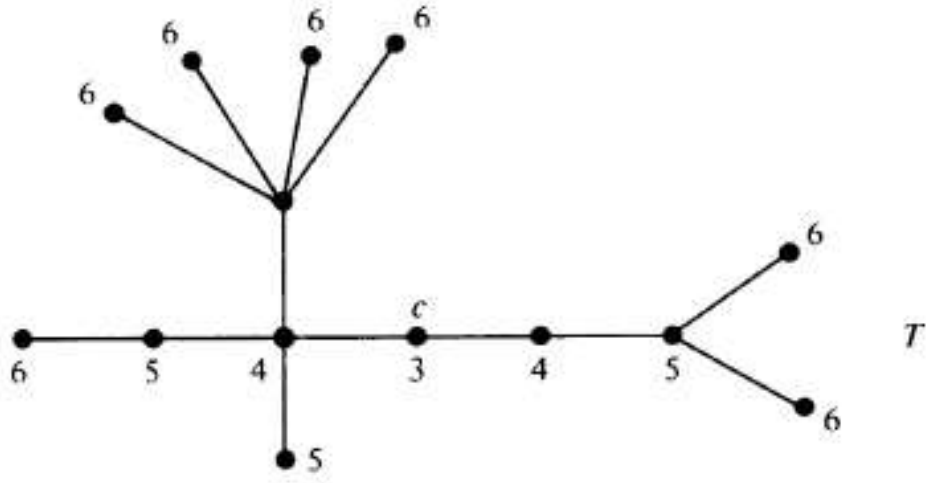
No caso de árvore este conceito é simplificado pois existe apenas um caminho entre cada par de vértices. Como definir o centro de uma árvore?



Uma maneira de determinar o centro de uma árvore é eliminando progressivamente os vértices pendentes e as arestas incidentes a eles até que reste um vértice isolado (o centro) ou dois vértices ligados por uma aresta (o bicentro).

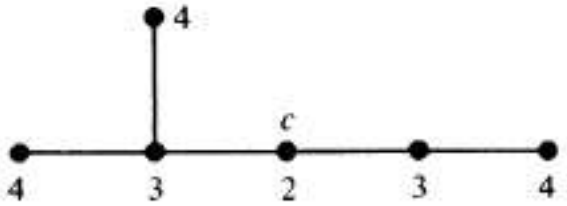
Observe que ao retirarmos um vértice de um grafo, retiramos também uma aresta. Assim, o grau e a excentricidade do vértice que permanece no grafo diminuem de valor.

Exemplo 2



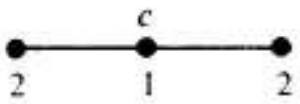
(a)

T



(b)

T'



(c)

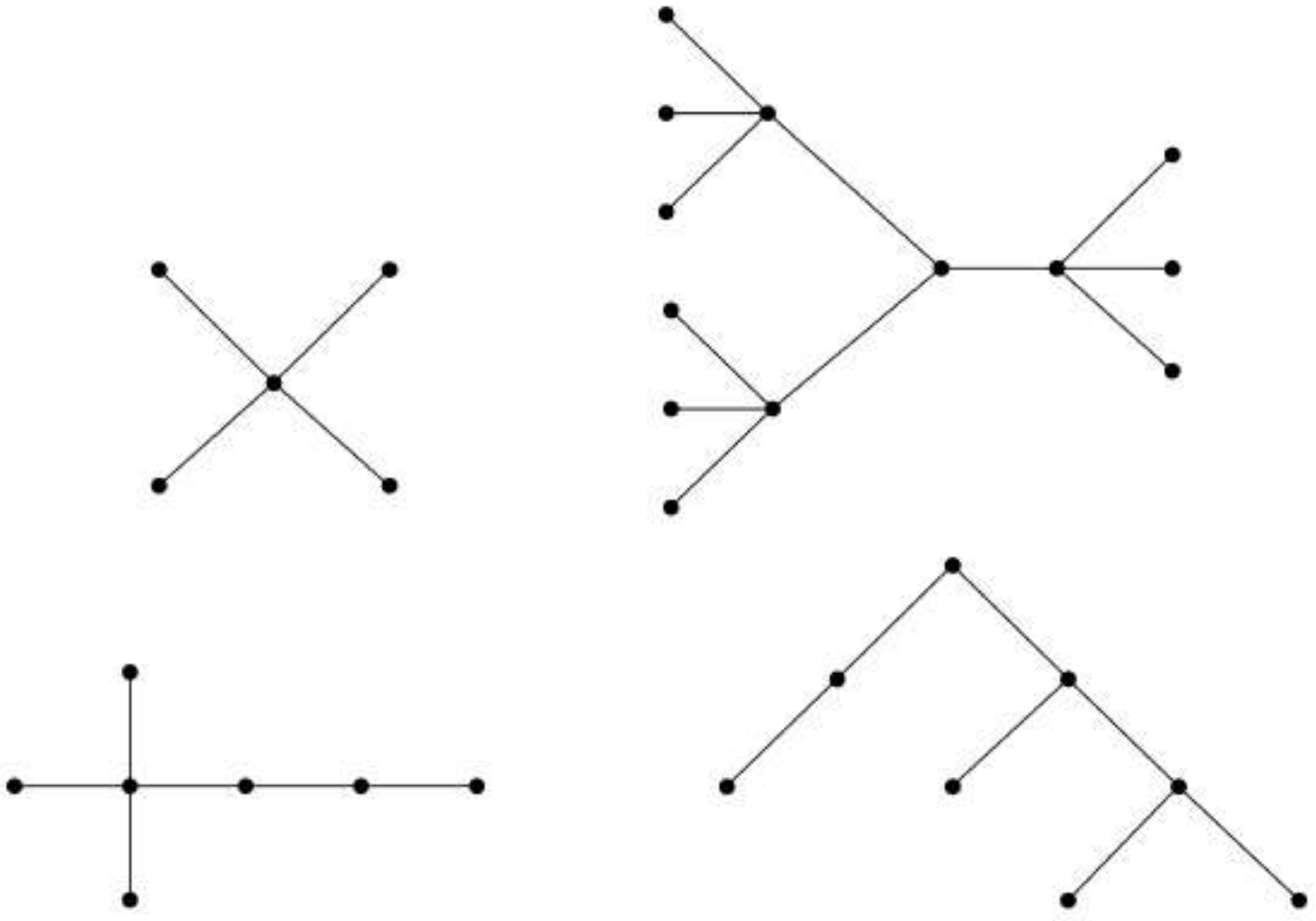
T''

Center

(d)

Exercícios

Encontre o centro das seguintes árvores:



Centro de um grafo

Proposição

Seja T uma árvore com pelo menos 3 vértices. Seja T' a árvore obtida de T pela exclusão dos vértices pendentes. Então T e T' possuem o mesmo centro.

Proposição

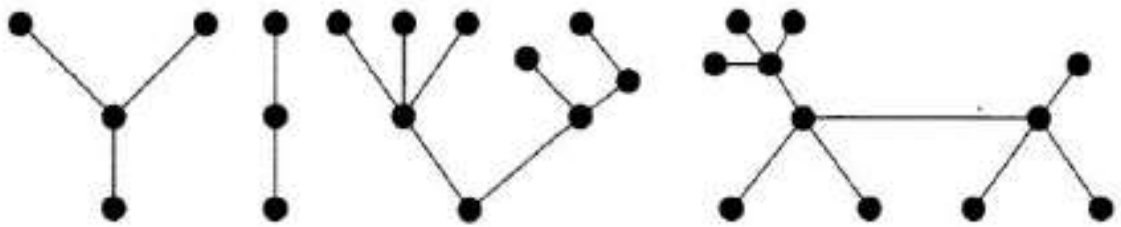
O centro de uma árvore possui um ou dois vértices.

Demonstração.

Ver [N. Deo, Graph Theory with Applications to Engineering and Computer Science, Prenticce-Hall, Inc., New Jersey, 1974]. □

Exercícios

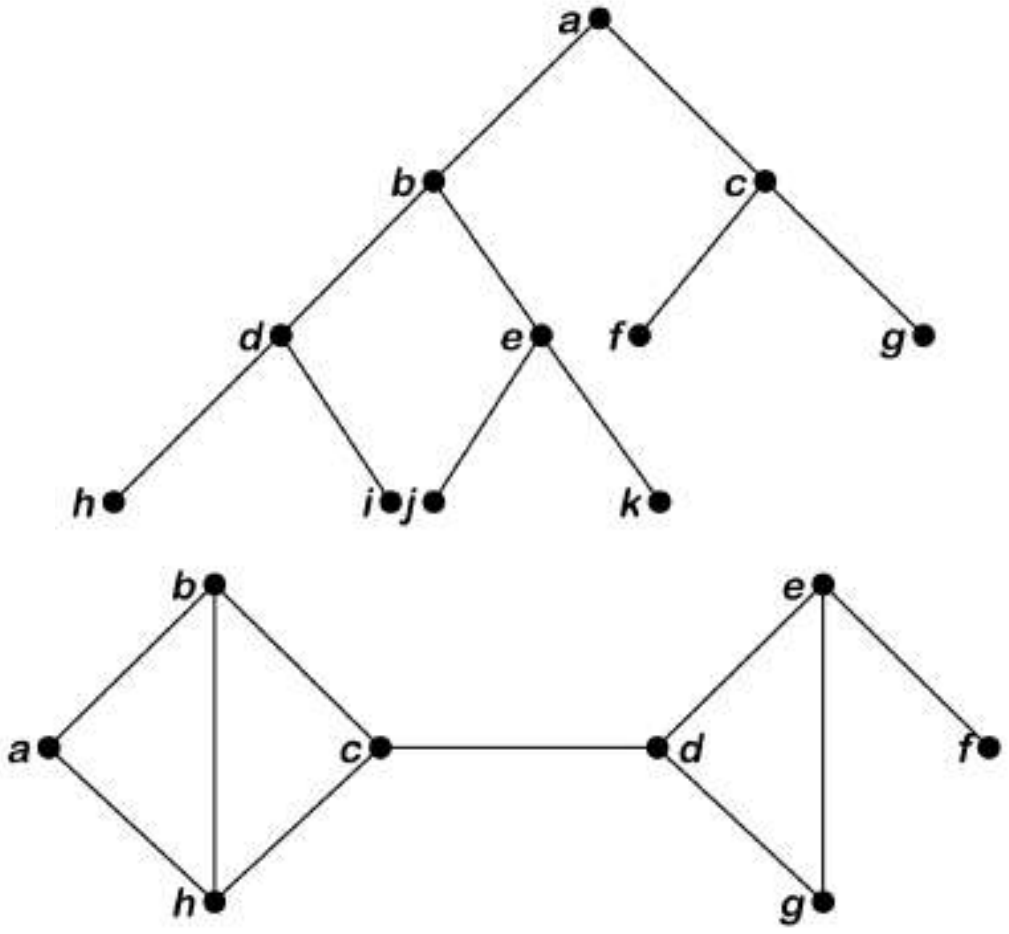
1. Dê exemplos de 5 situações que podem ser representadas através de árvores.
2. Uma floresta é um grafo desconexo composto pela união disjunta de árvores. Se G é um grafo com n vértices e t árvores, quantas arestas possui?



3. Demonstre o teorema que descreve as propriedades equivalentes de árvores.

Exercícios

4. Faça uma busca em largura e uma busca em profundidade nos grafos a seguir:



Outline

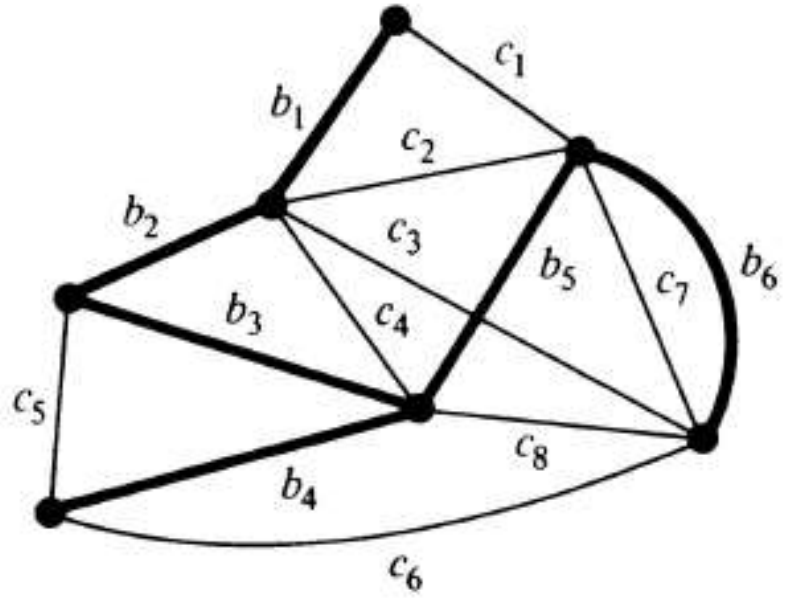
Definição

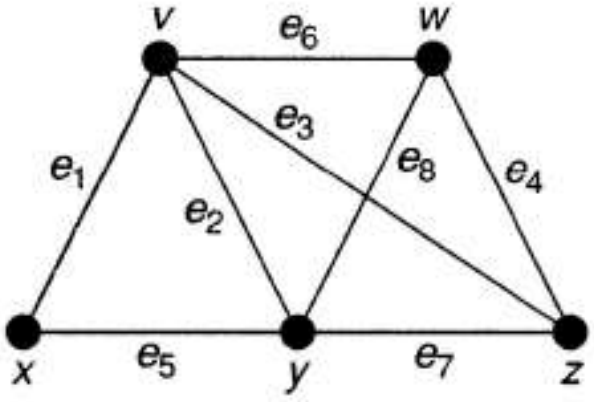
Em muitas aplicações, estamos interessados em subgrafos especiais de um determinado grafo.

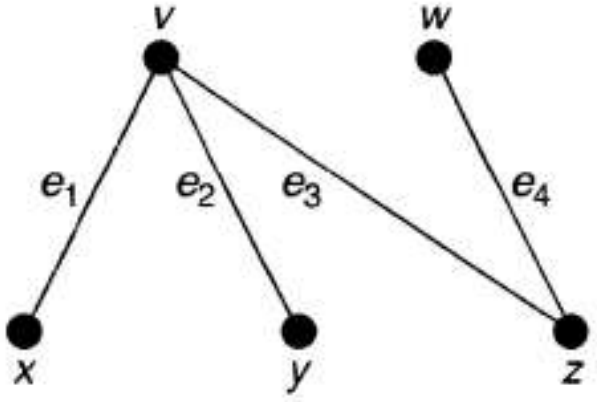
Definição

Uma árvore T é chamada de ***árvore geradora*** *de um grafo conexo G se T é um subgrafo de G que possui todos os vértices de G.*

Exemplos


c_1
b_1
c_2
b_2
c_3
b_5
c_7
b_6
b_3
c_4
c_5
c_8
b_4
c_6


v
e_6
w
e_3
e_1
e_8
e_4
e_2
x
e_5
y
e_7
z


v
w
e_1
e_2
e_3
e_4
x
y
z

Como obter uma árvore geradora de um grafo dado G?

Procedimento 1:

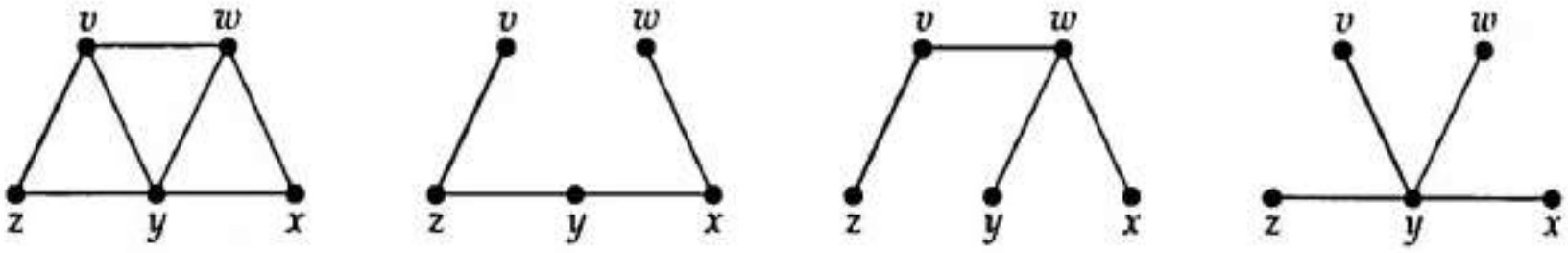
1. Se G não possui circuitos, G é sua própria árvore geradora.
2. Se G possui circuitos, retire uma aresta do circuito. O subgrafo resultante é conexo.
3. Se existirem mais circuitos, repita a operação até retirar uma aresta do último circuito do grafo.
4. O subgrafo resultante é conexo, sem circuitos e possui todos os vértices de G. Portanto é uma árvore geradora de G.

Teorema

Todo grafo conexo contém pelo menos uma árvore geradora.

Exemplo

Um grafo e três de suas árvores geradoras:



Vamos aplicar o Procedimento 1:

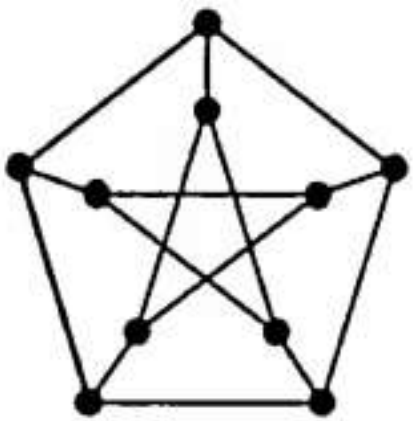
- Remover (v, y), destruindo o circuito v, w, y, v;
- Remover (y, z), destruindo o circuito v, w, y, z, v;
- Remover (x, y), destruindo o circuito w, x, y, w.

Obtivemos a segunda das árvores acima.

Observação

O número de árvores geradoras de um grafo pode ser muito grande. Por exemplo, o grafo de Petersen possui 2000 árvores geradoras.

Exercício: Encontre três árvores geradoras no grafo de Petersen aplicando o Procedimento 1.



Elos e ramos

Definição

*Seja $G(V, A)$ um grafo conexo e $T(V, E)$ uma árvore geradora de G. Uma aresta de G que não pertence à árvore geradora T é chamada de **elo** de G em relação a T. As arestas que compõem uma árvore geradora são chamadas de **ramos**.*

Exemplo

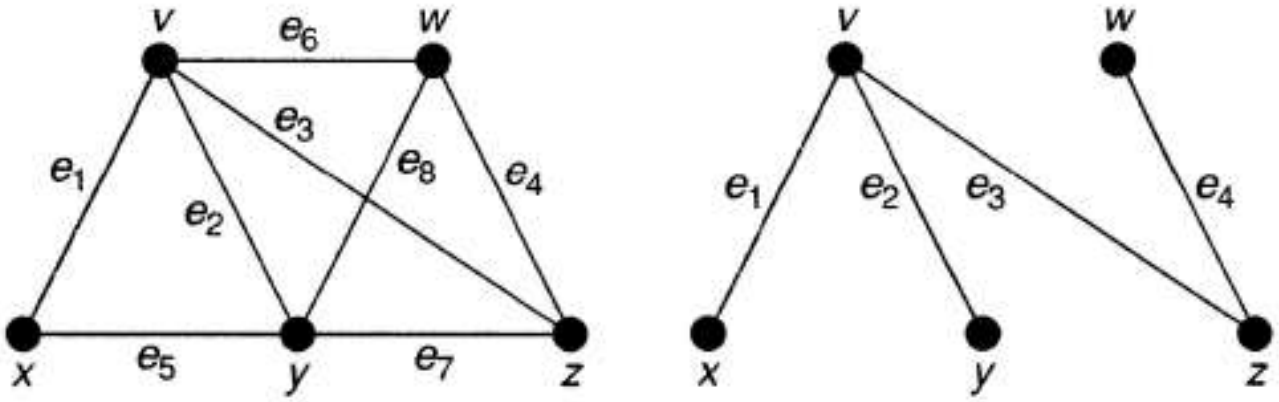


Figura: Grafo G à esquerda e sua árvore geradora T à direita

Os elos de G relativos a T são e_5, e_6, e_7, e_8.

Observação

Uma aresta que pertence a T pode ser um elo de G em relação a uma outra árvore geradora de G. No entanto, o número de elos de um grafo é fixo. Quantos são?

Exemplo

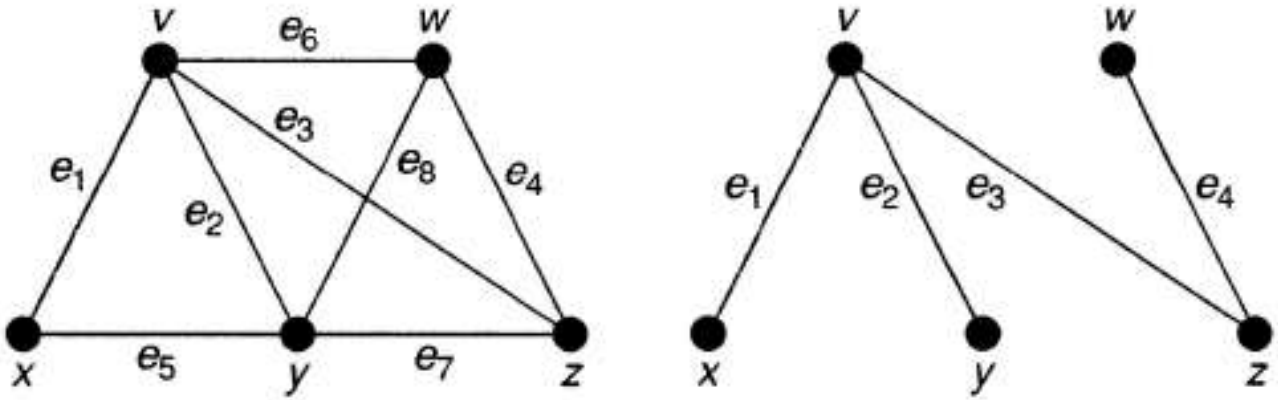


Figura: Grafo G à esquerda e sua árvore geradora T à direita

Os elos de G relativos a T são e_5, e_6, e_7, e_8.

Observação

Uma aresta que pertence a T pode ser um elo de G em relação a uma outra árvore geradora de G. No entanto, o número de elos de um grafo é fixo. Quantos são?

Exemplo

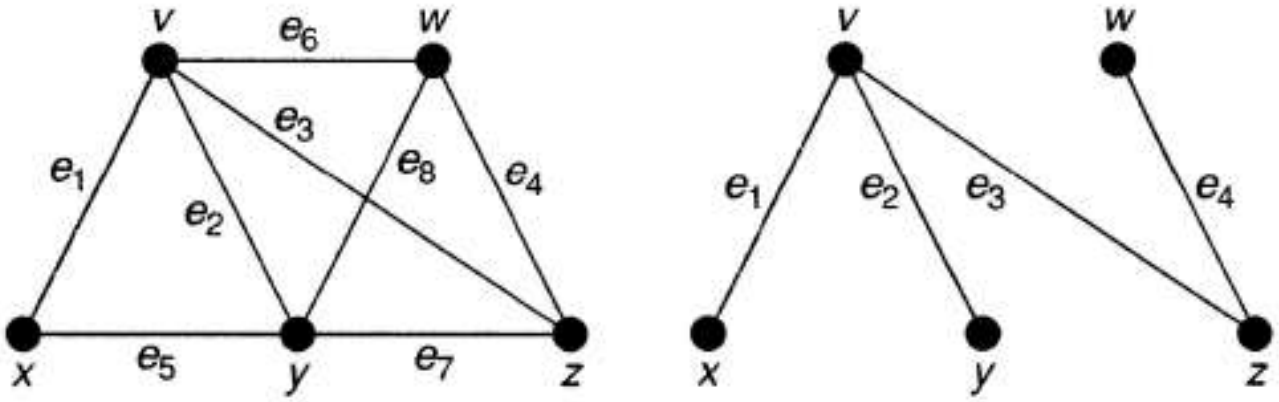


Figura: Grafo G à esquerda e sua árvore geradora T à direita

Os elos de G relativos a T são e_5, e_6, e_7, e_8.

Observação

Uma aresta que pertence a T pode ser um elo de G em relação a uma outra árvore geradora de G. No entanto, o número de elos de um grafo é fixo. Quantos são?

Teorema

Um grafo conexo com n vértices e m arestas possui $(m - n + 1)$ elos.

Definição

Se adicionarmos um elo de G à uma árvore geradora T, um único circuito será formado. Este circuito é chamado de ***circuito fundamental*** *de G.*

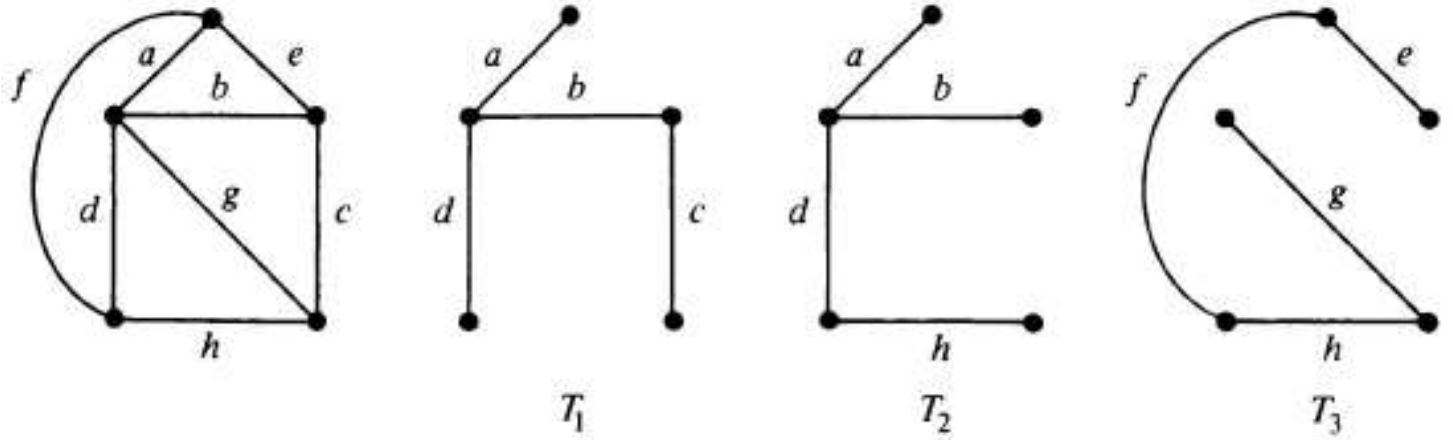
Quantos circuitos fundamentais um grafo possui?

Como obter todas as árvores geradoras de um grafo dado G?

Procedimento 2:

1. Utilize o Procedimento 1 para obter uma árvore geradora inicial T_1.
2. Determine os elos de G relativos a esta árvore. Acrescentando um elo de G a T_1, um circuito é formado.
3. Retire, uma a uma, as arestas do circuito fundamental formado. Desta forma são geradas as árvores geradoras associadas às arestas deste circuito, ou seja, tem-se $(k-1)$ árvores geradoras, onde k é o número de arestas no circuito fundamental. Esta operação é chamada de transformação elementar (troca cíclica, *cyclic exchange*).
4. Repita a transformação elementar considerando os outros elos do grafo.

Exemplo



- Em T_1, adicione o elo h e considere o circuito formado b, c, h, d;
- Remova a aresta c do circuito fundamental b, c, h, d e obtenha a árvore T_2;
- Repita a operação elementar removendo d e depois b;
- Adicione outro elo (e, f ou g) e repita o processo.

Perguntas

A análise do Procedimento 2 esboçado acima permite a formulação de uma série de perguntas:

1. Partindo de qualquer árvore e fazendo um certo número de transformações elementares, é possível obter uma determinada árvore geradora?
2. Usando transformações elementares é possível obter todas as árvores geradoras? Quantas transformações elementares serão necessárias?
3. A eficiência do algoritmo depende da árvore geradora inicial?

Para responder algumas dessas perguntas precisamos definir alguns novos conceitos.

Perguntas

A análise do Procedimento 2 esboçado acima permite a formulação de uma série de perguntas:

1. Partindo de qualquer árvore e fazendo um certo número de transformações elementares, é possível obter uma determinada árvore geradora?
2. Usando transformações elementares é possível obter todas as árvores geradoras? Quantas transformações elementares serão necessárias?
3. A eficiência do algoritmo depende da árvore geradora inicial?

Para responder algumas dessas perguntas precisamos definir alguns novos conceitos.

Definições

Definição

Dados dois conjuntos, S_1 e S_2, a ***soma direta*** *de S_1 e S_2, representada por $S_1 \oplus S_2$, é dada por*

$$S_1 \oplus S_2 = (S_1 \cup S_2) \setminus (S_1 \cap S_2).$$

Exemplo

Sejam $S_1 = \{v_1, v_2, v_3\}$ e $S_2 = \{v_1, v_4, v_5, v_6\}$. Então

$$S_1 \oplus S_2 = \{v_2, v_3, v_4, v_5, v_6\}.$$

Definições

Definição

*A **distância** entre duas árvores geradoras de um grafo G, T_i e T_j, é igual ao número de arestas que estão presentes em T_i e que não pertencem a T_j. Denotamos por $d(T_i, T_j)$.*

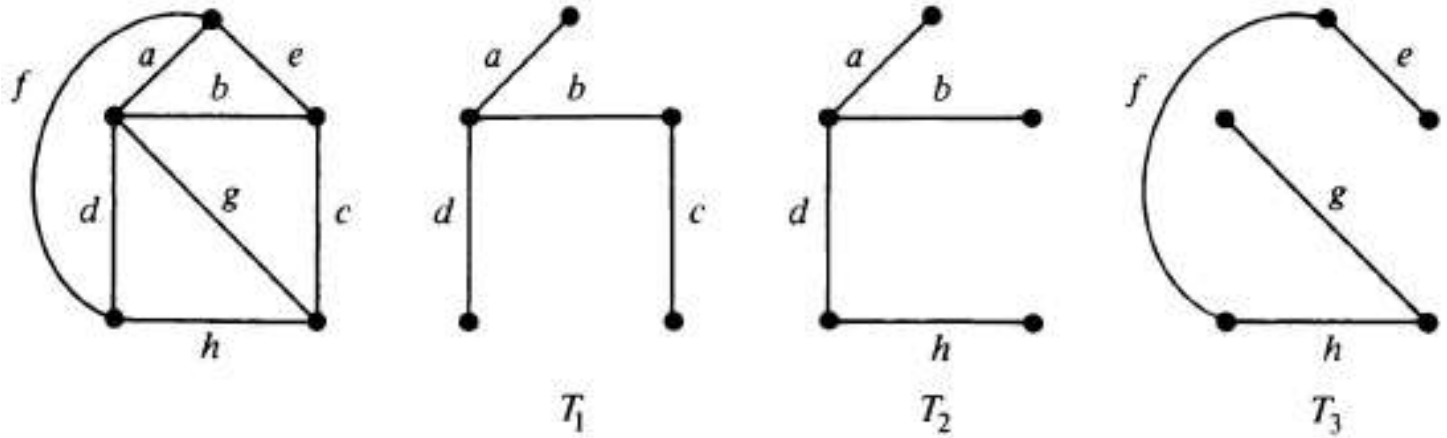
Observação

Podemos definir a distância entre duas árvores geradoras de G como sendo o número mínimo de transformações elementares necessárias para obter T_j a partir de T_i. Isto é,

$$d(T_i, T_j) = \frac{1}{2}|A_{T_i \oplus T_j}|,$$

onde $A_{T_i \oplus T_j}$ é o conjunto de arestas obtido pela soma direta do conjunto de arestas de T_i e T_j.

Exemplo



Temos:

- $d(T_1, T_2) = 1$;
- $d(T_1, T_3) = 4$;
- $d(T_2, T_3) = 3$.

Definições

Definição

Para uma árvore geradora T_0 de um grafo G, seja $\max_i d(T_0, T_i)$ a distância máxima de T_0 a qualquer outra árvore geradora T_i de G. Então T_0 é chamada de ***árvore central*** *de G se*

$$\max_i d(T_0, T_i) \leq \max_j d(T, T_j)$$

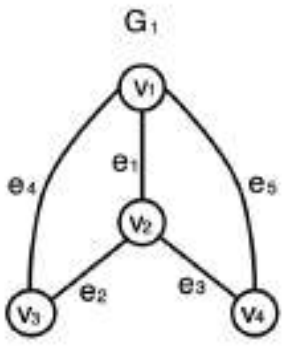
para qualquer árvore geradora T de G.

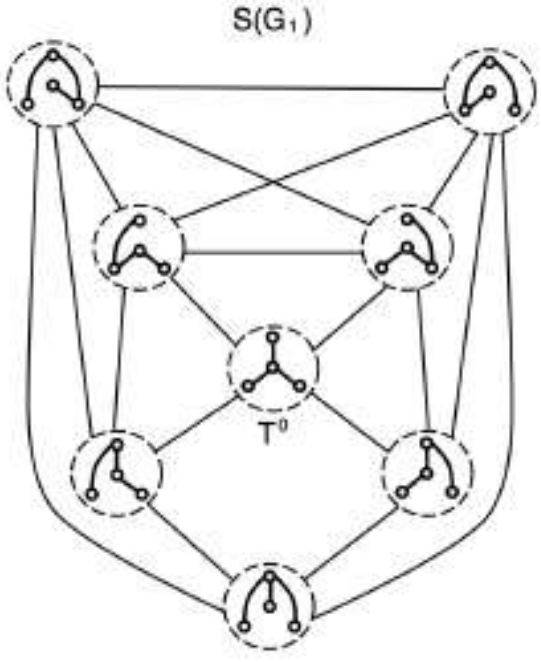
Definição

O ***grafo árvore*** *de G, $S(G)$, é definido como sendo o grafo em que cada vértice representa uma árvore geradora de G e existe uma aresta entre dois pares de vértices se a distância entre as árvores geradoras associadas for igual a um.*

Observação

Estes conceitos são usados em [2] para encontrar todas as árvores geradoras de um grafo dado G. A Figura a seguir mostra um grafo G_1 e o grafo árvore, $S(G_1)$ associado. Um outro algoritmo para listar todas as árvores geradoras de G é proposto em [3].





Teorema

É possível gerar todas as árvores geradoras de um grafo dado G começando de uma árvore geradora qualquer e executando sucessivas transformações elementares.

Demonstração.

Exercício. □

Observação

Segue do teorema acima que o grafo $S(G)$ é sempre conexo.

Procedimento 3 – Determinar uma árvore geradora

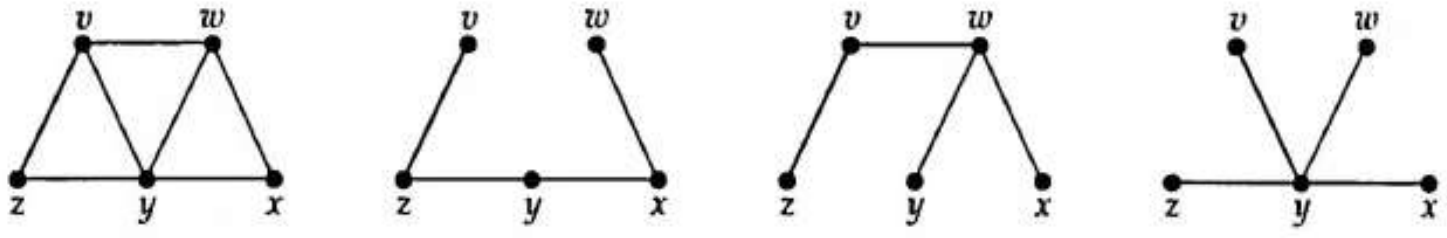
O Procedimento 1 constrói uma árvore geradora de G através da exclusão de arestas que fazem parte de um circuito em G.

O Procedimento 3 a seguir constrói uma árvore geradora de G incluindo arestas e evitando a formação de circuitos.

Procedimento 3: Considere um grafo simples com n vértices e m arestas. Ideia:

- Inicie a árvore T com uma aresta qualquer de G. A cada iteração, inclua uma nova aresta em T de maneira que nenhum circuito é formado, até que $n-1$ arestas tenham sido incluídas.

Exemplo



- Adicione as arestas (v, z), (z, y), (y, x) e (x, w);
- Nenhum ciclo é formado.

Obtemos a primeira das árvores acima.

Procedimento 3 – Determinar uma árvore geradora

1. O que acontece se o grafo não for conexo?

   Iremos obter várias árvores geradoras, isto é, uma floresta geradora.

2. Como garantir que, ao inserir uma aresta, nenhum circuito é formado?

   Verificar se as extremidades da aresta já foram incluídas. Assim ao tentarmos acrescentar a aresta (v_k, w_k) à árvore, as seguintes situações podem ocorrer:

Procedimento 3 – Determinar uma árvore geradora

i. Nem o vértice v_k, nem o vértice w_k pertencem a alguma árvore T_i já construída. Neste caso crie uma nova árvore a partir destes vértices e desta aresta. Considere que exista mais de uma componente no grafo, faça $cp = cp + 1$. Associe o rótulo cp aos vértices v_k e w_k.

ii. O vértice v_k pertence à árvore T_i e o vértice w_k pertence à árvore T_j, $i \neq j$. Neste caso, a aresta (v_k, w_k) é usada para unir as duas árvores. Faça os vértices de T_j receberem o mesmo rótulo cp dos vertices de T_i. Faça $cp = cp - 1$.

iii. Os dois vértices pertencem a árvore T_i. Neste caso a aresta é descartada pois sua inclusão criaria um circuito.

iv. Apenas um dos dois vértices v_k (ou w_k) pertence a alguma árvore T_i já construída. Neste caso acrescente a aresta e o vértice w_k (ou v_k) à árvore. O vértice w_k (ou v_k) recebe o mesmo rótulo cp que os vértices já pertencentes a T_i.

Procedimento 3 – Determinar uma árvore geradora

Como fazer para implementar as ideias acima?

- A eficiência do algoritmo depende da rapidez com que verificamos se as extremidades da aresta que estamos considerando pertence ou não a alguma árvore já criada.
- Para facilitar esta busca, criamos um vetor n-dimensional VERTEX que armazena esta informação. Quando uma aresta (i, j) é inserida em alguma árvore com rótulo cp, as posições i e j do vetor recebem o valor cp. Assim, para verificar se a aresta (v_k, w_k) já foi incluída em alguma árvore, verificamos se as posições correspondentes de VERTEX são diferentes de zero. Se para algum vértice q, VERTEX$(q) = 0$, o vértice q não está incluído em nenhuma árvore. Ao final do algoritmo, o vetor VERTEX identifica os vértices em cada componentes do grafo.

É suficiente?

Procedimento 3 – Determinar uma árvore geradora

Precisamos ainda identificar as arestas que compõe cada árvore do grafo.

- Criamos o vetor m-dimensional ARESTA, e inicializamos com 0. Assim se a k-ésima aresta foi incluída na árvore c, faça ARESTA$(k) = cp$.
- Ao final do algoritmo, as posições do vetor com ARESTA$(i) = 0$ identificam os elos de G.

Exercício

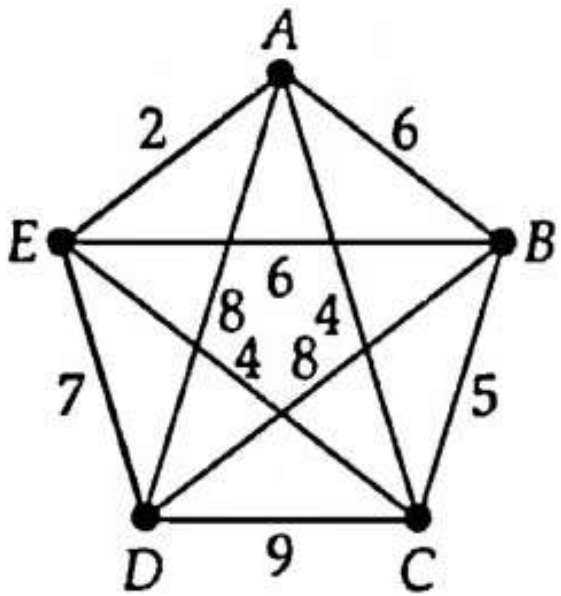
Considere o grafo G com 9 vértices e 12 arestas, representado através da lista de arestas abaixo (dois vetores m-dimensionais F e H, de tal forma que as extremidades da aresta k é armazenada nas posições f_k e h_k dos vetores F e H, respectivamente). Aplicar o Procedimento 3 e as ideias nele contidas, para construir uma árvore geradora para G.

$$
\begin{aligned}
F &= \begin{bmatrix} a & e & i & i & b & b & c & b & f & c & f & a \end{bmatrix} \\
H &= \begin{bmatrix} b & h & b & c & c & e & g & f & g & e & d & c \end{bmatrix}.
\end{aligned}
$$

Árvore Geradora Mínima

Considere uma rede e o problema de encontrar a árvore geradora mínima associada.

O **valor de árvore** é a soma dos pesos associados às arestas contidas na árvore.



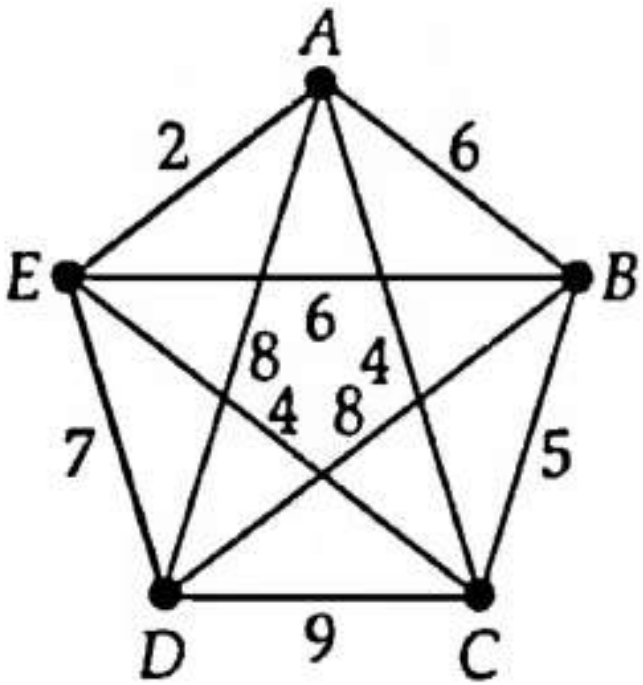
Algoritmo de Kruskal

Determinar uma árvore geradora mínima em um grafo qualquer.

- **Passo 1:** Ordene as arestas do grafo em ordem não-decrescente de peso.
- **Passo 2:** Aplique o Procedimento 3 para encontrar a árvore geradora, considerando que as arestas serão selecionadas de acordo com a ordem estabelecida no Passo 1.

Exemplo

Aplicar o Algoritmo de Kruskal para encontrar a árvore geradora mínima na rede abaixo:



Teorema

A árvore geradora T obtida pelo Algoritmo de Kruskal é uma árvore geradora mínima de G.

Prova.

Sejam $e_1, e_2, ..., e_{n-1}$ as arestas de T na ordem em que foram consideradas no Algoritmo de Kruskal. Isto é, $p(e_1) \leq p(e_2) \leq ... \leq p(e_{n-1})$.

Vamos supor que T não é uma árvore geradora mínima de G.

Dentre as árvores geradoras de valor mínimo, seja T_{min} a árvore geradora que contém as arestas $e_1, e_2, ..., e_j$ de T, tal que j seja o maior índice possível. Temos que $j < n - 1$, pois em caso contrário, e_j estaria em T.

Prova cont.

Considere que a aresta e_{j+1} é adicionada a T_{min}. Um circuito é então criado.

Este circuito contém uma aresta x que não pertence a T (se todas as arestas do circuito estivessem em T, T não seria uma árvore, pois também teria um circuito).

Pela ordem em que as arestas foram consideradas na construção de T, temos que e_{j+1} foi adicionada a T, mas x não foi incluída.

Portanto $p(e_{j+1}) \leq p(x)$ (caso contrário x teria sido incluída em T sem a formação de um circuito).

Prova cont.

Vamos então construir uma nova árvore:
$T_{nova} = T_{min} - \{x\} + \{e_{j+1}\}$.

Se $p(e_{j+1}) < p(x)$, então $p(T_{nova}) < p(T_{min})$, o que contraria a hipótese que T_{min} é mínima.

Se $p(e_{j+1}) = p(x)$, então $p(T_{nova}) = p(T_{min})$, $T_{nova} \neq T_{min}$, T_{nova} é mínima e contém as arestas $e_1, e_2, \ldots, e_j, e_{j+1}$, o que contradiz que T_{min} é aquela que possui a maior quantidade de arestas de T.

Portanto temos uma contradição quando dizemos que $p(e_{j+1}) \leq p(x)$, e neste caso a suposição de que T não é mínima é falsa. Assim mostramos que T é mínima. $\square$

Algoritmo de Prim

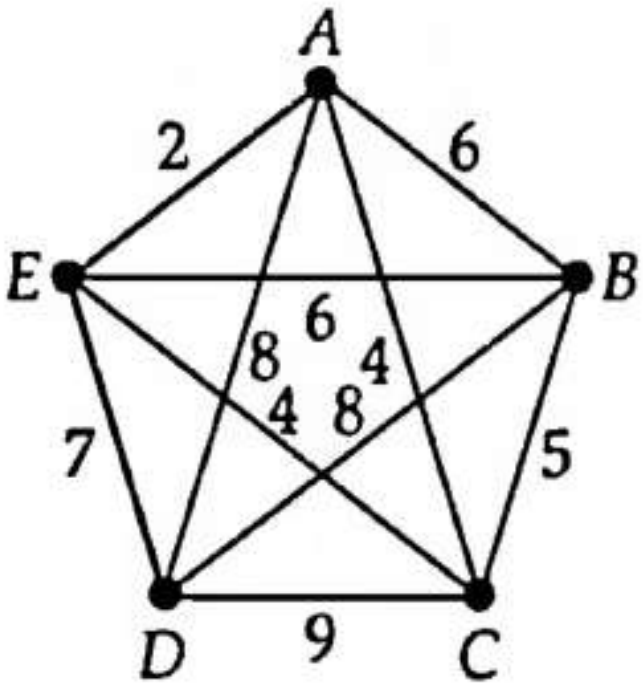
Determinar uma árvore geradora mínima em um grafo qualquer.

- **Passo 1:** Selecione um vértice v_k de G e inclua em T;
- **Passo 2:** Repita este passo até que todos os vértices de G pertençam a T.

  Selecione a aresta de menor peso (v_j, w_j) tal que v_j pertença a T e w_j não pertença a T.

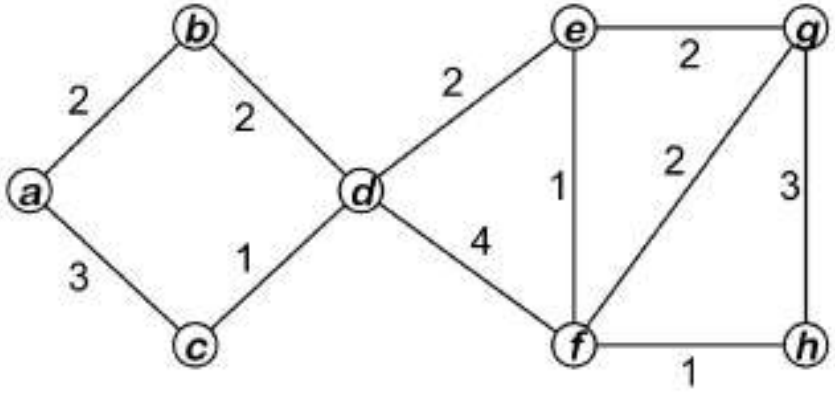
Exemplo

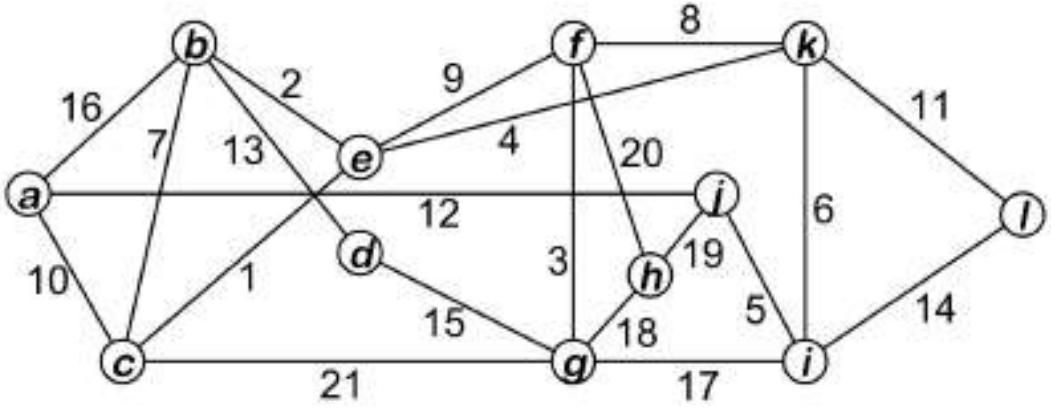
Aplicar o Algoritmo de Prim para encontrar a árvore geradora mínima na rede abaixo:



Exercício

1. Utilize os algoritmos de Kruskal e de Prim para identificar uma árvore geradora mínima em cada um dos grafos abaixo. Qual é o melhor?





Exercício

2. Verificar que uma submatriz $(n-1)$ x $(n-1)$ da matriz de incidência de um grafo G é não-singular se, e somente se, as arestas associadas às $n-1$ colunas desta submatriz constituem uma árvore geradora de G.

   Obs.: O *posto* de um grafo com n vértices é igual a $n-1$.

Referências

[1] Michel Gagnon, Notas de aula do curso: CI065 Algoritmos e teoria dos grafos. UFPR, 2002.

[2] A. Shioura, A. Tamura and T. Uno, An optimal algorithm for scanning all spanning trees of undirected graphs. SIAM Journal on Computing, 26(3), 678–692, 1997.

[3] G. Minty, A Simple Algorithm for Listing All the Trees of a Graph. Circuits and Systems, IEEE Transactions on Circuit Theory, Volume 12(1), 120–120, 1965.