MC558 — Análise de Algoritmos II

Cid C. de Souza Cândida N. da Silva Orlando Lee

15 de março de 2023

Antes de mais nada...

- Uma versão anterior deste conjunto de slides foi preparada por Cid Carvalho de Souza e Cândida Nunes da Silva para uma instância anterior desta disciplina.
- O que vocês tem em mãos é uma versão modificada preparada para atender a meus gostos.
- Nunca é demais enfatizar que o material é apenas um guia e não deve ser usado como única fonte de estudo. Para isso consultem a bibliografia (em especial o CLR ou CLRS).

Orlando Lee

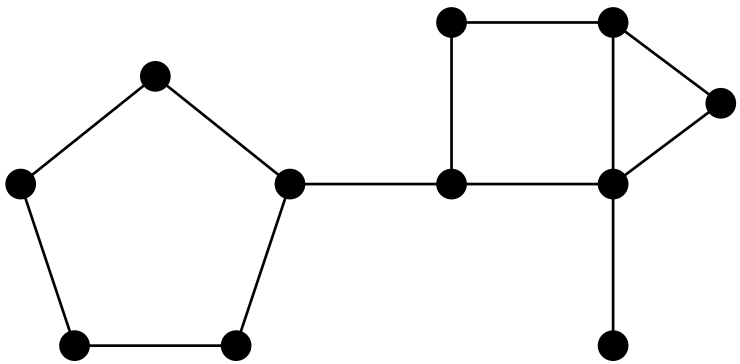
Agradecimentos (Cid e Cândida)

- Várias pessoas contribuíram direta ou indiretamente com a preparação deste material.
- Algumas destas pessoas cederam gentilmente seus arquivos digitais enquanto outras cederam gentilmente o seu tempo fazendo correções e dando sugestões.
- Uma lista destes "colaboradores" (em ordem alfabética) é dada abaixo:
  - Célia Picinin de Mello
  - José Coelho de Pina
  - Orlando Lee
  - Paulo Feofiloff
  - Pedro Rezende
  - Ricardo Dahab
  - Zanoni Dias

Arestas-de-corte e ciclos

Denote por $c(G)$ o número de componentes de um grafo G.

Proposição. Seja G um grafo e seja $e \in E(G)$. Então $c(G) \leq c(G - e) \leq c(G) + 1$.



Arestas-de-corte e ciclos

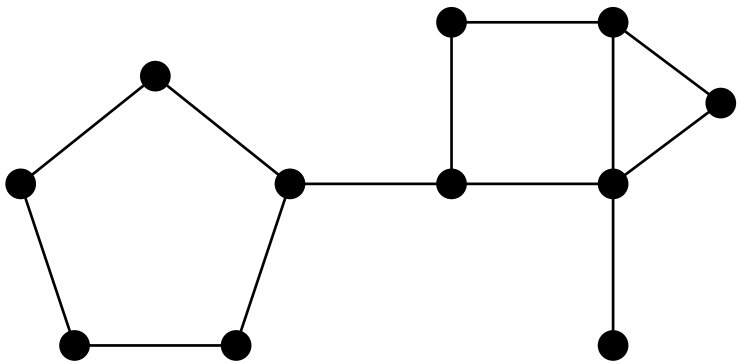
Denote por $c(G)$ o número de componentes de um grafo G.

Proposição. Proposição. Seja G um grafo e seja $e \in E(G)$. Então $c(G) \leq c(G - e) \leq c(G) + 1$.

Prova. Considere o grafo $G - e$ obtido pela remoção de e. Se $c(G - e) = c(G)$, então o resultado segue.
Senão, necessariamente temos que $c(G - e) > c(G)$. Quando devolvemos a aresta e a $G - e$, temos que reduzir o número de componentes de volta ao valor $c(G)$. Assim, existem dois componentes C_1 e C_2 de $G - e$, cada um contendo um extremo de e e $c(G - e) = c(G) + 1$. ■

Arestas-de-corte e ciclos

Sejam G um grafo e $e \in E(G)$. Se $c(G - e) = c(G) + 1$, então dizemos que e é uma aresta-de-corte de G. No caso em que G é conexo, dizemos que a remoção de e desconecta G.



Como seria uma condição necessária e suficiente para e ser uma aresta-de-corte de um grafo G?

Arestas-de-corte e ciclos

Proposição. Seja G um grafo e seja $e \in E(G)$. Então e é uma aresta-de-corte de G se, e somente se, e não pertence a nenhum ciclo de G.

Prova. Podemos supor que G é conexo. (Por quê?)

$\Rightarrow$: suponha que $e = uv$ seja uma aresta-de-corte. Então u e v pertencem a componentes distintos de $G - e$. Se e pertencesse a um ciclo C de G, então $C - e$ seria um caminho de u a v em $G - e$, uma contradição.

$\Leftarrow$: pela contrapositiva, suponha que $e = uv$ não é uma aresta-de-corte. Então $G - e$ é conexo e existe um caminho P de u a v em $G - e$. Então $P + e$ é um ciclo em G. ■

Vértice-de-corte

Seja G um grafo e seja $v \in V(G)$. Se $c(G - v) > c(G)$, então dizemos que v é um vértice-de-corte de G.
Se G for conexo, então dizemos que a remoção de v desconecta G.

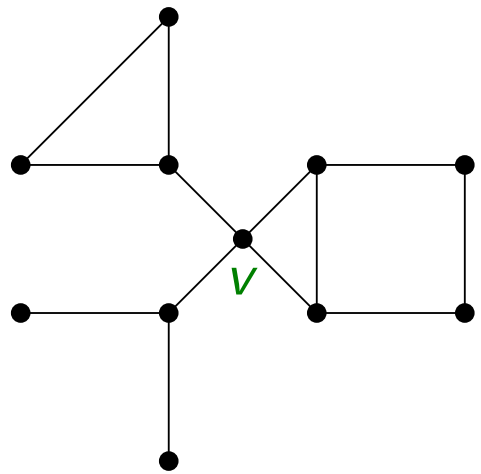


Figura: Vértice-de-corte v em um grafo.

Vértice-de-corte

Proposição. Sejam G um grafo e seja $v \in V(G)$. Então $c(G) \leq c(G - v) \leq c(G) + d_G(v) - 1$.

Exercício. Seja G um grafo e seja $e \in E$. Se $e = uv$ é uma aresta-de-corte, então u ou v é um vértice-de-corte de G, a menos que $\{u, v\}$ induza um componente de G.

Exercício. Seja G um grafo e seja $v \in V(G)$. Então v é um vértice-de-corte de G se, e somente se, existem dois vértices x e y distintos de v pertencentes a um mesmo componente de G tais que todo caminho de x a y em G passa por v.

Proposição. Todo grafo conexo G possui pelo menos $n(G) - 1$ arestas.

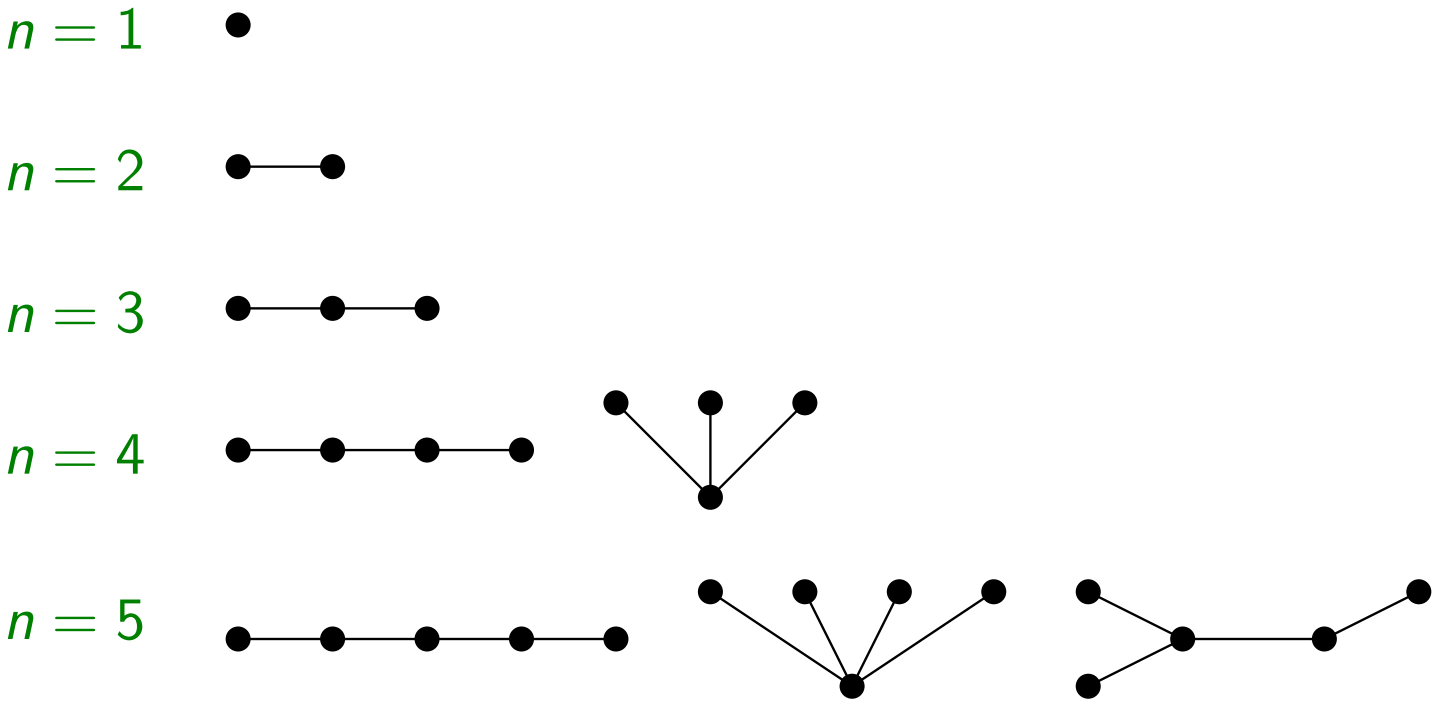
Prova. Seja $G = (V, E)$ e considere o seguinte algoritmo.

1. $H \leftarrow (V, \emptyset)$
2. seja $e_1, \ldots, e_m$ uma ordenação das arestas de G
3. **para** $i \leftarrow 1$ **até** m **faça**
4. $\quad H \leftarrow H + e_i$

Note que na linha 1, temos que $c(H) = n := n(G)$ e após a última execução da linha 4, temos que $H = G$ e portanto, $c(H) = 1$. Na iteração i, a aresta e_i só pode conectar dois componentes distintos de H, reduzindo o número de componentes de no máximo uma unidade. Portanto, devemos ter $m \geq n - 1$. ■

Árvores

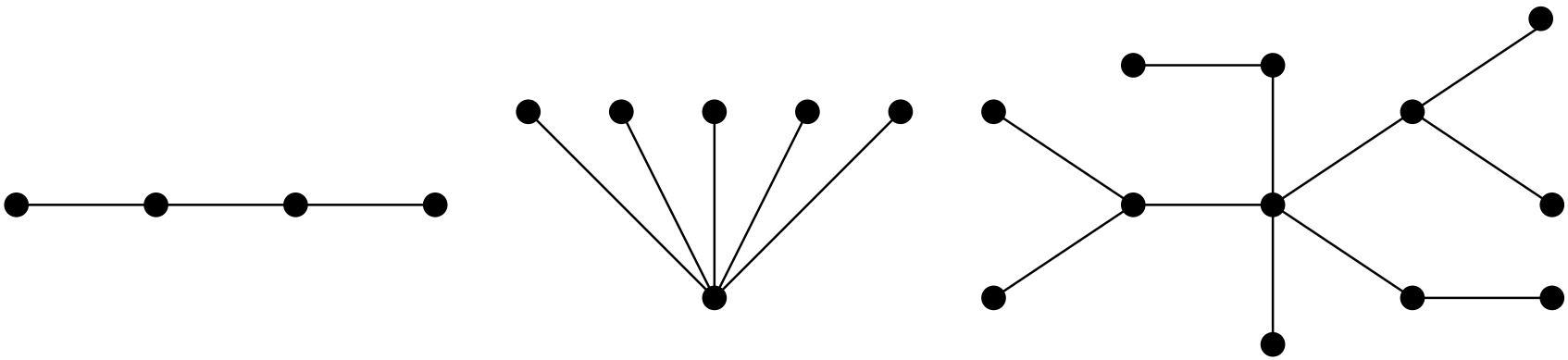
O argumento anterior mostra que se G é conexo e tem exatamente $n-1$ arestas, então toda aresta é uma aresta-de-corte (e portanto, não pertence a um ciclo).



Árvores

Um grafo G é acíclico se não contém ciclos. Também dizemos que G é uma floresta.
Uma árvore é um grafo conexo e acíclico.

- Um caminho é uma árvore com grau máximo dois.
- Uma estrela é uma árvore isomorfa a $K_{1,r}$ para algum $r \geq 1$.
- Cada componente de uma floresta é uma árvore.

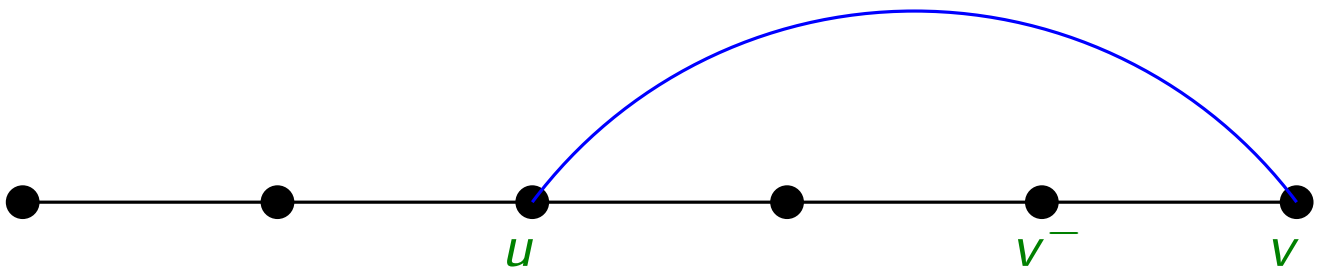


Folhas

Uma folha em um grafo G é um vértice de grau um.

Lema. Toda árvore G com pelo menos dois vértices possui pelo menos duas folhas.

Prova. Seja P um caminho maximal em G. Seja v o término de P e seja v^- o predecessor de v em P. Pela maximalidade de P, v não pode ter nenhum vizinho em $V - V(P)$. Como G é acíclico, v não pode ter nenhum vizinho u distinto de v^- em P. Portanto, v é uma folha. De modo análogo, o início de P também é uma folha. ■



Folhas

Lema. Se v é uma folha de uma árvore G, então $G - v$ é uma árvore.

Prova. Seja v uma folha de G. Claramente $G - v$ é conexo e acíclico. Portanto, $G - v$ é uma árvore. ■

O lema acima fornece uma ferramenta conveniente para se fazer provas por indução (em $n(G)$ ou $m(G)$) sobre uma árvore G.

Propriedades de árvores

Considere as seguintes propriedades relativas a um grafo G:

- (A) G é conexo
- (B) G é acíclico
- (C) $m(G) = n(G) - 1$

Por definição G é uma árvore se satisfaz (A) e (B). Mostraremos que quaisquer duas condições em $\{A, B, C\}$ implicam na outra e caracterizam árvores.

Proposição. Se G é um grafo conexo e acíclico (i.e., uma árvore), então $m(G) = n(G) - 1$.

Prova. Provaremos o resultado por indução em $n(G)$.

Base: $n(G) = 1$. O resultado é óbvio.

Hipótese de indução: para toda árvore G' de ordem $n' < n$ temos que $m(G') = n(G') - 1$.

Seja G uma árvore de ordem $n \geq 2$. Pelo lema, G possui uma folha v. Assim, $G' := G - v$ é uma árvore. Pela HI, temos que $m(G') = n(G') - 1$. Como $m(G') = m(G) - 1$ e $n(G') = n(G) - 1$, segue que $m(G) = n(G) - 1$. ■

Proposição. Se G é um grafo conexo tal que $m(G) = n(G) - 1$, então G é acíclico.

Prova. Considere o seguinte algoritmo.

1. $H \leftarrow G$
2. **enquanto** H contiver um ciclo **faça**
3. seja e uma aresta pertencente a um ciclo de H
4. $H \leftarrow H - e$

Ao final do algoritmo, H é uma árvore. De fato, H é obviamente acíclico e também é conexo, pois o algoritmo nunca remove uma aresta-de-corte. Pela Proposição 1, $m(H) = n(H) - 1$. Como $n(G) = n(H)$, segue que $m(G) = m(H)$ e portanto, $H = G$. Logo, G é acíclico. ■

Proposição. Se G é um grafo acíclico tal que $m(G) = n(G) - 1$, então G é conexo.

Prova. Sejam $G_1, \ldots, G_k$ os componentes de G. Mostraremos que $k = 1$.

Cada componente de G é uma árvore. Pela Proposição 1, $m(G_i) = n(G_i) - 1$ para $i \in \{1, \ldots, k\}$. Somando para todo i, temos que

$$m(G) = \sum_{i=1}^{k} m(G_i) = \sum_{i=1}^{k} [n(G_i) - 1] = n(G) - k.$$

Por hipótese, $m(G) = n(G) - 1$ e portanto, $k = 1$ e G é conexo. ■

Caracterizações de árvores

Teorema. As seguintes afirmações são equivalentes:

(a) G é uma árvore (i.e., G é conexo e acíclico),

(b) G é conexo e $m(G) = n(G) - 1$,

(c) G é acíclico e $m(G) = n(G) - 1$,

(d) G é conexo e toda aresta é uma aresta-de-corte,

(e) G não tem laços e para todo par $u, v \in V(G)$, existe um **único** caminho de u a v em G, e

(f) G é acíclico e para todo par de vértices não-adjacentes $u, v \in V(G)$, o grafo $G + uv$ contém exatamente um ciclo.

Prova. Já vimos que (a), (b) e (c) são equivalentes. Além disso, (a) e (d) são equivalentes, pois uma aresta é uma aresta-de-corte se, e somente se, não pertence a nenhum ciclo.

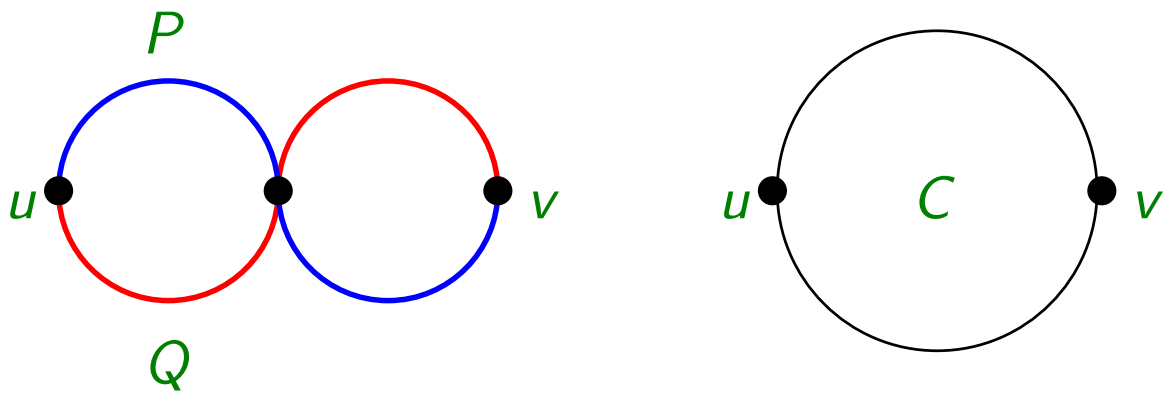
Provaremos agora que (a) $\Rightarrow$ (e) $\Rightarrow$ (f) $\Rightarrow$ (a).

Caracterizações de árvores

(a) G é uma árvore

(e) G não tem laços e para todo par $u, v \in V(G)$, existe um **único** caminho de u a v em G

(a) $\Rightarrow$ (e) Seja G uma árvore. Então G não tem laços. Suponha por contradição que existem vértices em G que são ligados por dois caminhos distintos. Entre todos estes pares escolha um par u, v com caminhos distintos P e Q de u a v tal que $|P| + |Q|$ seja o menor possível. Pela escolha de u e v, P e Q não tem vértices internos em comum. Isto implica que $C := P \bullet Q^{-1}$ é um ciclo, uma contradição.



Caracterizações de árvores

(e) G não tem laços e para todo par $u, v \in V(G)$, existe um **único** caminho de u a v em G

(f) G é acíclico e para todo par de vértices não-adjacentes $u, v \in V(G)$, o grafo $G + uv$ contém exatamente um ciclo

(e) $\Rightarrow$ (f) Seja G um grafo sem laços tal que para todo par $u, v \in V(G)$, existe um **único** caminho de u a v em G. Claramente G é acíclico. Seja u, v um par de vértices não-adjacentes e seja P o único caminho de u a v em G. Então $G + uv$ contém o ciclo $P \bullet (v, u)$. Note que todo ciclo C de $G + uv$ deve conter uv. Assim, $C - uv$ é um caminho de u a v em G. Por hipótese, segue que $P = C - uv$ e portanto, $G + uv$ contém um único ciclo.

Caracterizações de árvores

(a) G é uma árvore

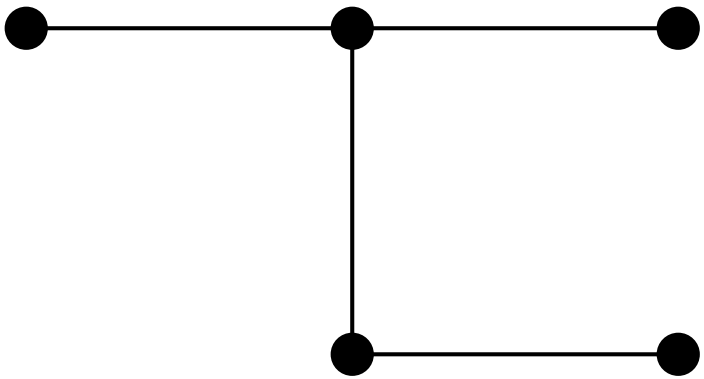
(f) G é acíclico e para todo par de vértices não-adjacentes $u, v \in V(G)$, o grafo $G + uv$ contém exatamente um ciclo

(f) $\Rightarrow$ (a) Seja G um grafo acíclico tal que para todo par de vértices não-adjacentes $u, v \in V(G)$, o grafo $G + uv$ contém exatamente um ciclo. Assim G é acíclico. Resta mostrar que G é conexo. Suponha por contradição que G não seja conexo. Sejam u e v vértices em componentes distintos de G. Considere o grafo $G + uv$. Claramente uv não pertence a nenhum ciclo de $G + uv$, uma contradição. Portanto, G é conexo.

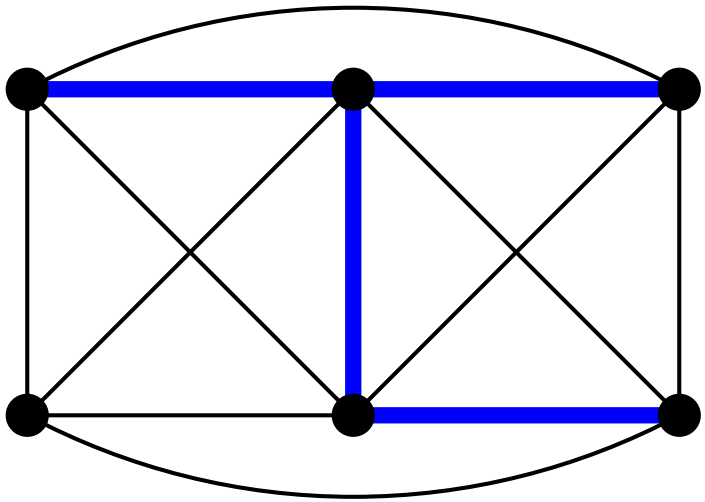
Isto termina a prova do teorema. ■

Exemplo

Proposição. Todo grafo simples G com $\delta(G) \geq k$ contém uma cópia de qualquer árvore T com k arestas (i.e., $|V(T)| = k + 1$).



T



G

Exemplo

Proposição. Todo grafo simples G com $\delta(G) \geq k$ contém uma cópia de qualquer árvore T com k arestas (i.e., $|V(T)| = k + 1$).

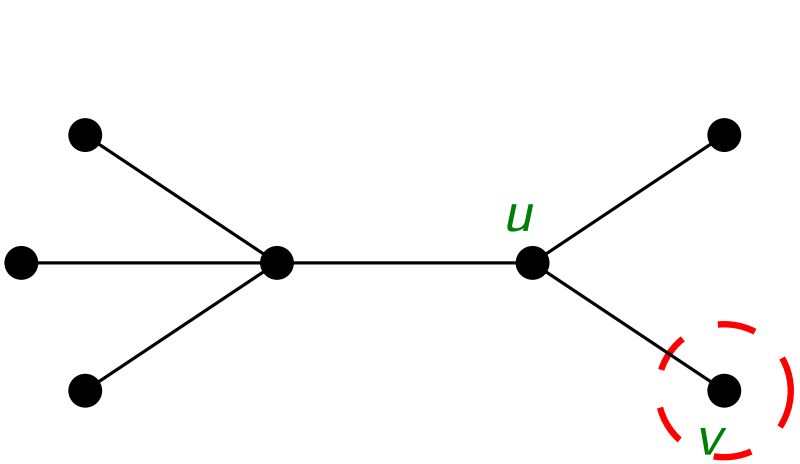
Prova. A prova é por indução em k.

Base: $k = 0$. Todo grafo simples contém uma cópia de $T = K_1$ e o resultado é óbvio.

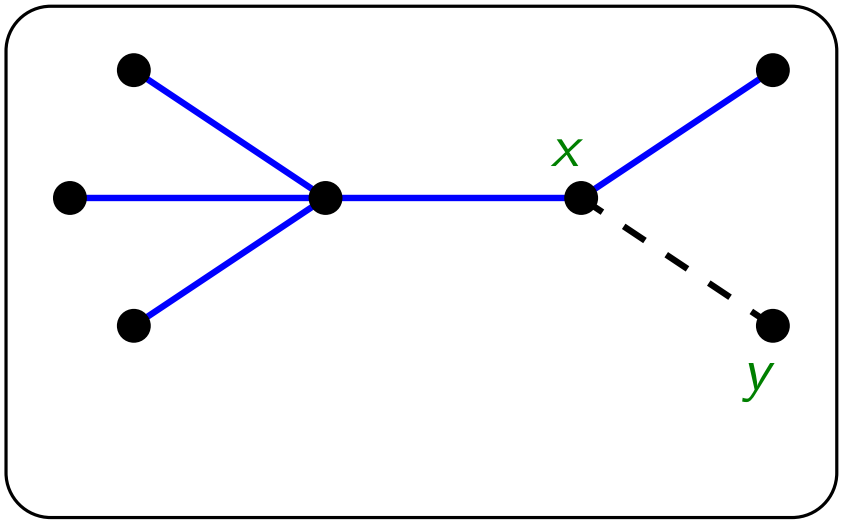
Hipótese de indução: suponha que $k > 0$ e que todo grafo simples com $\delta(G) \geq k - 1$ contém uma cópia de qualquer árvore com $k - 1$ arestas.

Como $k > 0$, então T possui uma folha, digamos v; seja u o único vizinho de v em T. Seja $T' := T - v$. Pela HI, G contém uma cópia de T' já que $\delta(G) \geq k > k - 1$.

Exemplo



T



G

Seja x o vértice da cópia de T' correspondente a u. Como $V(T')$ tem apenas $k-1$ vértices distintos de u e $d_G(x) \geq k$, segue que x tem algum vizinho y que não pertence à cópia de T'. Então podemos adicionar xy a esta cópia para obter uma cópia de T (com y fazendo o papel de v). ■

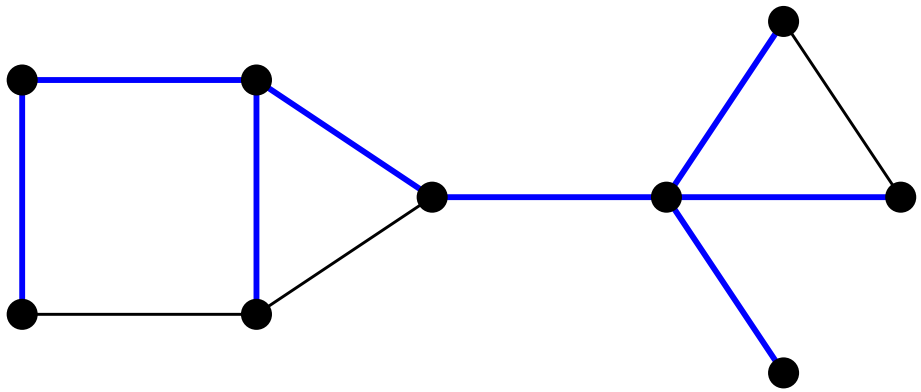
Conjectura de Erdős-Sós

- A condição $\delta(G) \geq k$ é a melhor possível; o grafo K_k tem grau mínimo $k-1$, mas não contém nenhuma árvore com k arestas.
- Uma consequência da proposição é que todo grafo simples com mais do que $n(k-1)$ arestas contém uma cópia de qualquer árvore com k arestas (Exercício!).
- Erdős e Sós (1964) conjecturaram o seguinte: um grafo simples com mais do que $n(k-1)/2$ arestas contém uma cópia de qualquer árvore T com k arestas. Esta conjectura está quase provada no sentido de que é verdadeira para n suficientemente grande.

Árvores geradoras

Dizemos que um subgrafo H de um grafo G é um subgrafo gerador se $V(H) = V(G)$.

Se H for uma árvore, então dizemos que H é uma árvore geradora de G.



Observação. Note que um subgrafo gerador não precisa ser conexo. Por exemplo, o subgrafo gerador vazio $H = (V, \emptyset)$ de um grafo $G = (V, E)$ com $n(G) \geq 2$ não é conexo.

Árvores geradoras

Proposição. Todo grafo conexo contém uma árvore geradora.

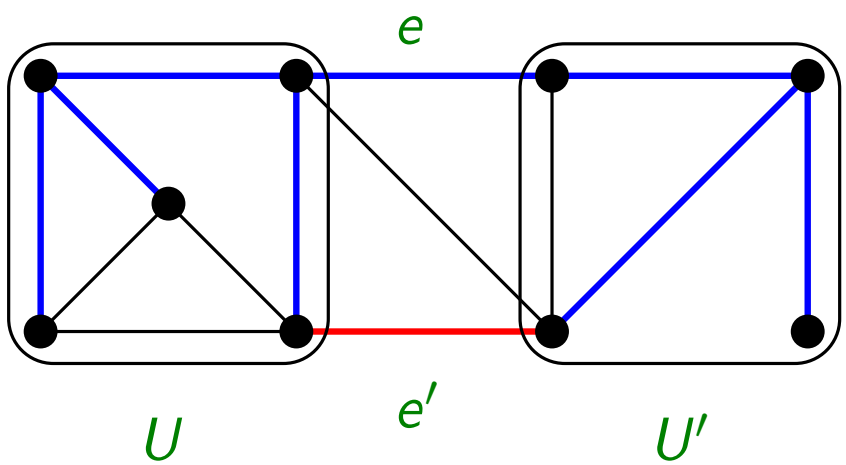
Prova. Isto segue da prova da proposição $A + C \Rightarrow B$, mas apresentamos aqui outra prova. Seja T um subgrafo gerador conexo minimal de G. Claramente T é acíclico, pois senão poderíamos remover uma aresta em um ciclo. Logo, T é uma árvore geradora de G. ■

A seguir apresentamos dois resultados que descrevem operações de trocas de arestas entre árvores geradoras T e T'.

Árvores geradoras

Proposição. Sejam T, T' árvores geradoras de um grafo G e seja $e \in E(T) - E(T')$. Então existe $e' \in E(T') - E(T)$ tal que $T - e + e'$ é uma árvore geradora de G.

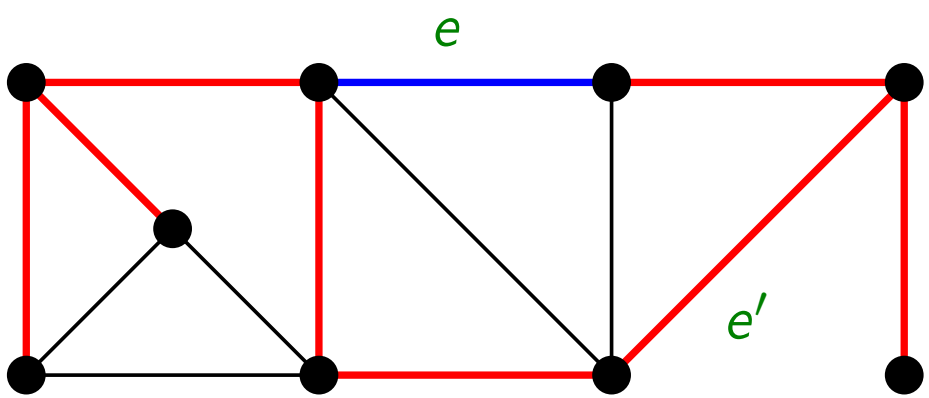
Prova. Sejam U e U' os (conjuntos de vértices dos) componentes de $T - e$. Como T' é conexo, existe alguma aresta $e' \in E(T') - E(T)$ com extremos em U e U'. Assim, $T - e + e'$ é conexo e tem exatamente $n(G) - 1$ arestas. Portanto, $T - e + e'$ é uma árvora geradora de G. ■



Árvores geradoras

Proposição. Sejam T, T' árvores geradoras de um grafo G e seja $e \in E(T) - E(T')$. Então existe $e' \in E(T') - E(T)$ tal que $T' - e' + e$ é uma árvore geradora de G.

Prova. Seja C o único ciclo de $T' + e$. Como T é acíclico, alguma aresta e' de C pertence a $E(T') - E(T)$. Assim, $T' - e' + e$ é conexo e tem exatamente $n(G) - 1$. Portanto, $T' - e' + e$ é uma árvore geradora de G. ■



Grafos bipartidos de novo...

A ideia do seguinte exercício é obter uma nova prova do Teorema de König: *um grafo G é bipartido, se e somente se, G não contém um ciclo ímpar.*

Exercício 1A. Prove que toda árvore é um grafo bipartido por indução.

Exercício 1B. Seja G um grafo e seja T uma árvore geradora de G. Usando o resultado anterior prove que G é bipartido se, e somente se, para toda aresta $e \in E(G) - E(T)$, o (único) ciclo de $T + e$ é par.

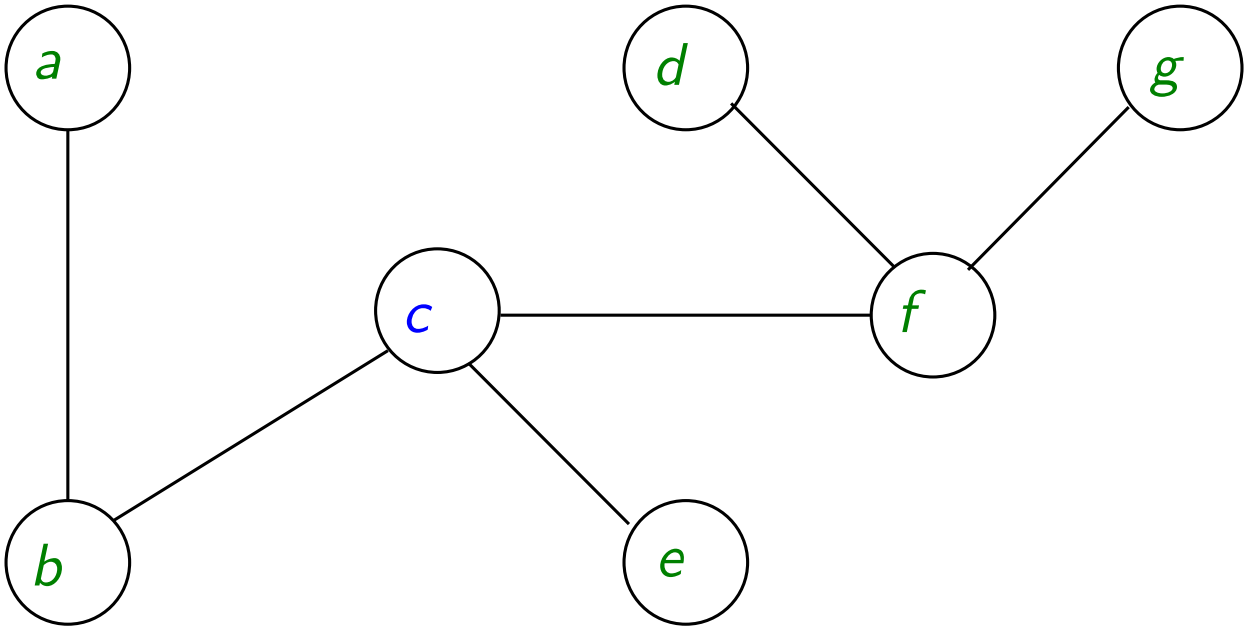
Árvore geradora mínima

Problema da Árvore Geradora de Peso Mínimo: dado um grafo conexo $G = (V, E)$ e uma função peso $\omega : E \mapsto \mathbb{R}$, encontrar uma árvore geradora T cujo peso $\omega(T) := \sum_{e \in E(T)} \omega(e)$ seja mínimo.

Este é um problema clássico de Otimização Combinatória. Veremos depois no curso como resolver eficientemente este problema.

Representação de árvores

Uma *árvore enraizada* é uma árvore com um vértice especial chamado *raiz*.



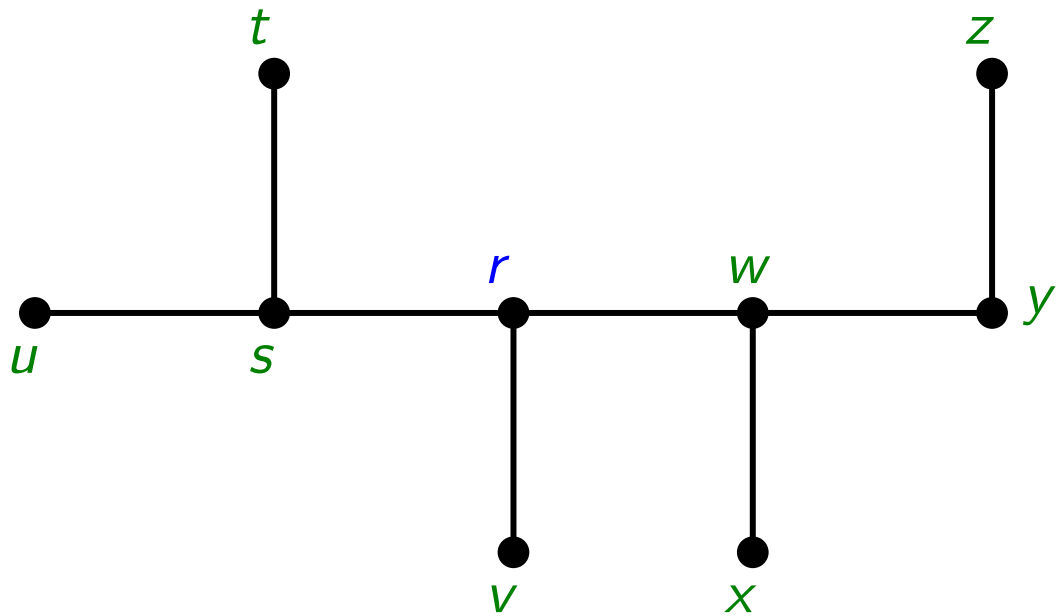
raiz *c*

Representação de árvores

Vetor π ($\pi[v]$ é pai de v):

v	r	s	t	u	v	w	x	y	z
$\pi[v]$	N	r	s	s	r	r	w	w	y

N é um símbolo usado para indicar não existência.

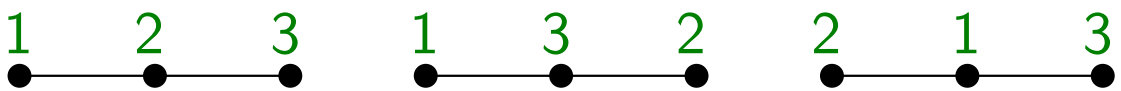


raiz r

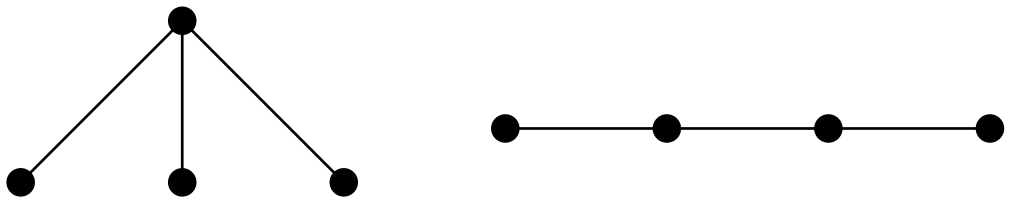
Enumeração de árvores

Seja $V := \{1, 2, \ldots, n\}$ ($n \geq 2$). Quantas árvores com conjunto de vértices igual a V existem?

- $n = 2$. Existe $1 = 2^0$ árvore.
- $n = 3$. Existem $3 = 3^1$ árvores.



- $n = 4$. Existem $16 = 4^2$ árvores (4 estrelas e 12 caminhos).



- $n = 5$. Existem $125 = 5^3$ árvores (por verificação).
- n arbitrário. Existem n^{n-2} árvores?

Fórmula de Cayley

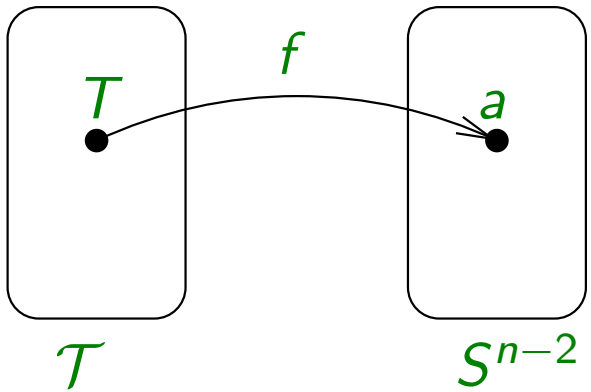
Teorema (Fórmula de Cayley). Existem n^{n-2} árvores com conjunto de vértices $\{1, 2, \ldots, n\}$ ($n \geq 2$).

Várias pesquisadores encontraram (diferentes) provas deste resultado. Apresentamos aqui a prova devida a Prüfer (1918).

Código de Prüfer

Seja S um conjunto qualquer e seja n um inteiro positivo. Denotamos por S^n o conjunto das n-uplas formadas por elementos de S.

Seja $S \subseteq \mathbb{N}$ de tamanho n. Nosso objetivo é exibir uma bijeção (código de Prüfer) f entre o conjunto $\mathcal{T}$ de todas as árvores T com $V(T) = S$ e o conjunto S^{n-2}. Como $|S^{n-2}| = n^{n-2}$, isto implica a Fórmula de Cayley.



$$f(T) = a := (a_1, a_2, \ldots, a_{n-2}) \in S^{n-2}$$

Código de Prüfer

Código de Prüfer

Entrada: árvore T com $V(T) = S \subseteq \mathbb{N}$ e $|S| = n \geq 2$.

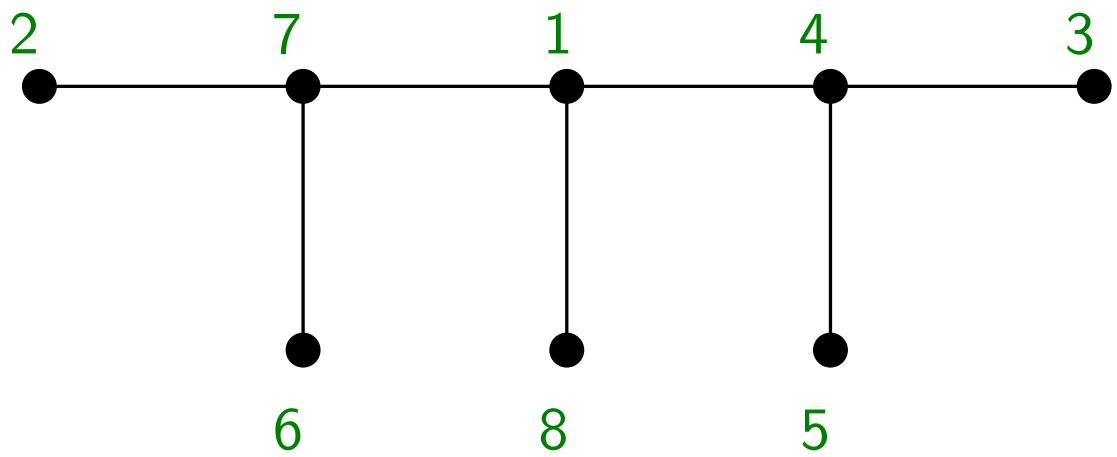
Saída: uma $(n-2)$-upla $f(T) = (a_1, a_2, \ldots, a_{n-2})$.

1. **para** $i \leftarrow 1$ **até** $n-2$ **faça**
2. seja v a menor folha de T
3. seja a_i o **único vizinho** de v em T
3. $T \leftarrow T - v$
4. **devolva** $(a_1, a_2, \ldots, a_{n-2})$

Dizemos que $f(T) = (a_1, \ldots, a_{n-2})$ é o código de Prüfer da árvore T.

Exemplo do Código de Prüfer

$f(T) = ?$

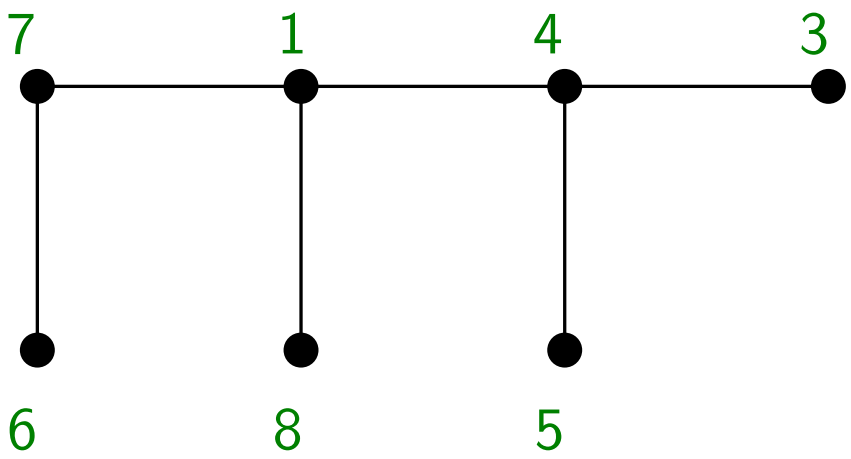


$a_1 = 7$ (vizinho da menor folha 2)

$f(T) = (7,$

Exemplo do Código de Prüfer

$f(T) = (7,$

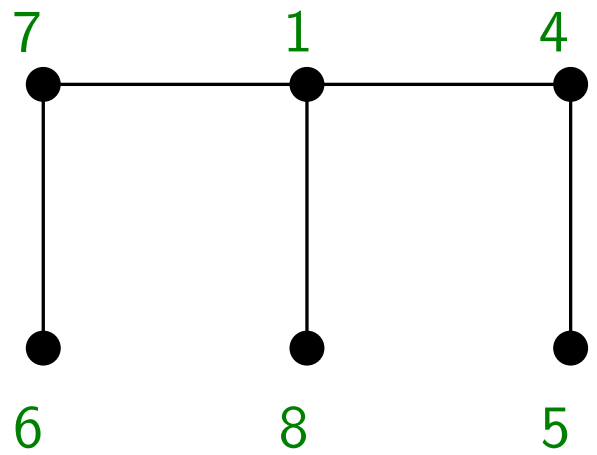


$a_2 = 4$ (vizinho da menor folha 3)

$f(T) := (7, 4$

Exemplo do Código de Prüfer

$f(T) = (7, 4$

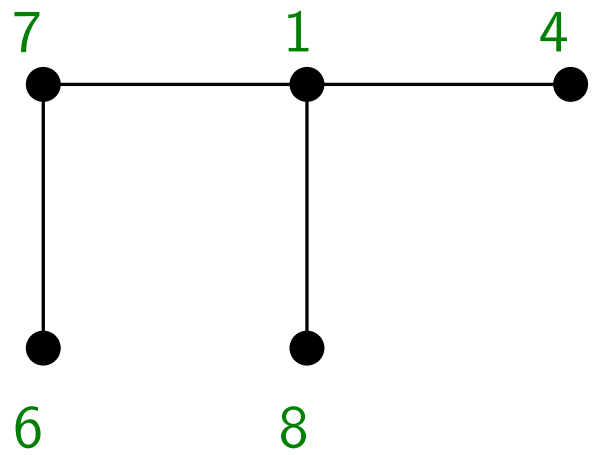


$a_3 = 4$ (vizinho da menor folha 5)

$f(T) = (7, 4, 4$

Exemplo do Código de Prüfer

$f(T) = (7, 4, 4$

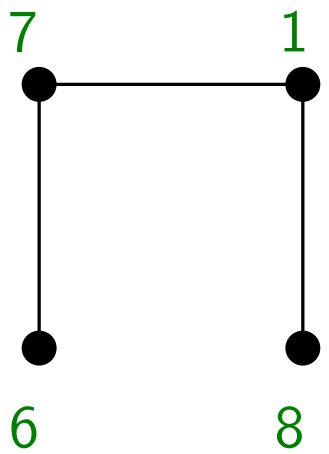


$a_4 = 1$ (vizinho da menor folha 4)

$f(T) = (7, 4, 4, 1$

Exemplo do Código de Prüfer

$f(T) = (7, 4, 4, 1$

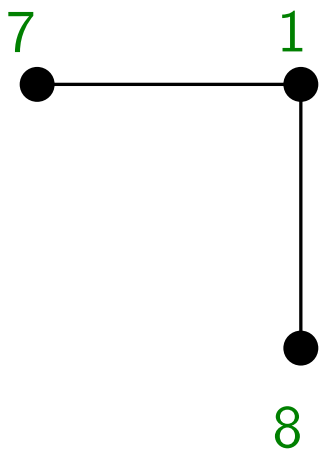


$a_5 = 7$ (vizinho da menor folha 6)

$f(T) = (7, 4, 4, 1, 7$

Exemplo do Código de Prüfer

$f(T) = (7, 4, 4, 1, 7$



$a_6 = 1$ (vizinho da menor folha 7)

$f(T) = (7, 4, 4, 1, 7, 1$

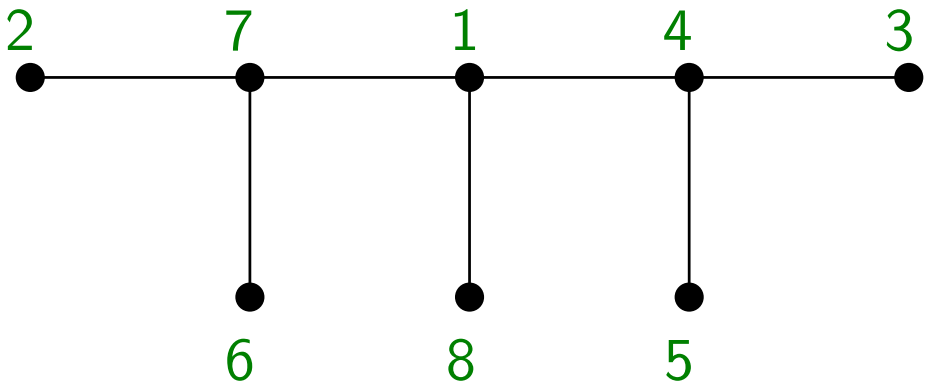
Exemplo do Código de Prüfer

$f(T) = (7, 4, 4, 1, 7, 1$

Algoritmo para.

$f(T) = (7, 4, 4, 1, 7, 1)$

Exemplo do Código de Prüfer



$f(T) = (7, 4, 4, 1, 7, 1)$

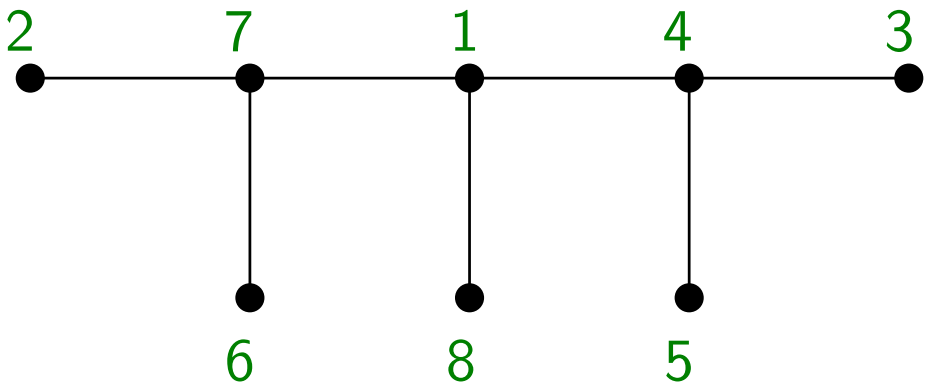
Código de Prüfer – versão recursiva

Código de Prüfer
Entrada: árvore T com $V(T) = S \subseteq \mathbb{N}$ e $|S| = n \geq 2$.
Saída: uma $(n-2)$-upla $f(T) = (a_1, a_2, \ldots, a_{n-2})$.

1. **se** $|S| = 2$ **então devolva** $(\ )$
2. seja v a menor folha de T
3. seja a_1 o **único vizinho** de v em T
3. $(a_2, \ldots, a_{n-2}) \leftarrow$ Código de Prüfer$(T - v)$
4. **devolva** $(a_1, a_2, \ldots, a_{n-2})$

Note que o algoritmo pode ser visto como um **método recursivo**: o algoritmo escolhe a_1 como o vizinho da menor folha v, recursivamente determina o código de Prüfer $a' := (a_2, \ldots, a_{n-2})$ de $T - v$ e devolve $f(T) := (a_1, a_2, \ldots, a_{n-2})$.

Código de Prüfer



$$f(T) = (7, 4, 4, 1, 7, 1)$$

Lema. Os vértices que aparecem em $f(T)$ são exatamente as não-folhas de T.

Prova. Note que o algoritmo só para quando a árvore tem apenas dois vértices. Assim, em cada iteração do algoritmo, o vértice escolhido a_i é necessariamente uma não-folha. Como em cada iteração o algoritmo remove uma folha, toda não-folha aparece em $f(T)$. ■

Código de Prüfer

Teorema (Prüfer, 1918). Seja $S \subseteq \mathbb{N}$ com $|S| = n$ e seja $\mathcal{T}$ o conjunto das árvores T com $V(T) = S$. Então para cada $a := (a_1, \ldots, a_{n-2}) \in S^{n-2}$, existe uma **única** árvore $T \in \mathcal{T}$ tal que $f(T) = a$. Em particular, temos que $|\mathcal{T}| = n^{n-2}$.

Prova. Provaremos o resultado por indução em n.

Base: $n = 2$. Neste caso, a é a sequência vazia $(\ )$, $\mathcal{T}$ contém uma única árvore T com $V(T) = S$ e $f(T) = a$.

Hipótese de indução: suponha que se $\mathcal{T}'$ é o conjunto das árvores T' com $V(T') = S'$, onde $S' \subseteq \mathbb{N}$ e $|S'| = n - 1$, então para cada $a' \in (S')^{n-3}$, existe uma única árvore $T' \in \mathcal{T}'$ tal que $f(T') = a'$.

Suponha que $n \geq 3$ e seja $a = (a_1, \ldots, a_{n-2}) \in S^{n-2}$. Provaremos que existe uma única árvore T tal que $f(T) = a$.

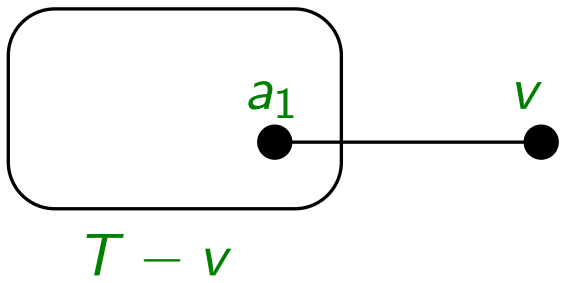
Código de Prüfer

Para isto é preciso mostrar que:

- existe alguma árvore $T \in \mathcal{T}$ tal que $f(T) = a$ e
- se $f(T_1) = f(T_2) = a$ com $T_1, T_2 \in \mathcal{T}$, então $T_1 = T_2$.

O que podemos dizer sobre uma árvore hipotética T tal que $f(T) = a$?

Pela definição do código de Prüfer, a_1 deve ser o vizinho da menor folha de T. Pelo Lema, $a_1, a_2, \ldots, a_{n-2}$ são as não-folhas de T. Logo, $v := \min(S - \{a_1, \ldots, a_{n-2}\})$ tem que ser a menor folha de T; logo $va_1 \in E(T)$. Além disso, pela definição do código de Prüfer, temos que $f(T - v) = (a_2, \ldots, a_{n-2})$.



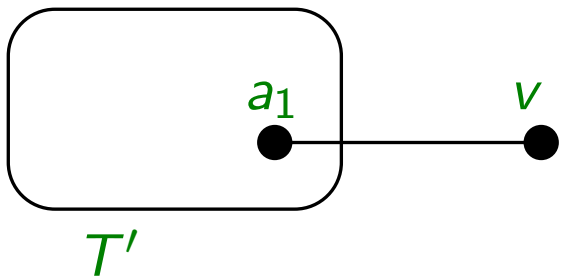
Código de Prüfer

Hipótese de indução: suponha que se $\mathcal{T}'$ é o conjunto das árvores T' com $V(T') = S'$, onde $S' \subseteq \mathbb{N}$ e $|S'| = n-1$, então para cada $a' \in (S')^{n-3}$, existe uma única árvore $T' \in \mathcal{T}'$ tal que $f(T') = a'$.

Seja $S' := S - v$ e seja $\mathcal{T}'$ o conjunto das árvores T' com $V(T') = S'$. Seja $a' := (a_2, \ldots, a_{n-2})$. Assim, $a' \in (S')^{n-3}$. Pela HI **existe** uma **(única) árvore** T' tal que $f(T') = a' = (a_2, \ldots, a_{n-2})$.

Seja $T = (V(T') \cup \{v\}, E(T') \cup \{va_1\})$. Claramente, $f(T) = a$.

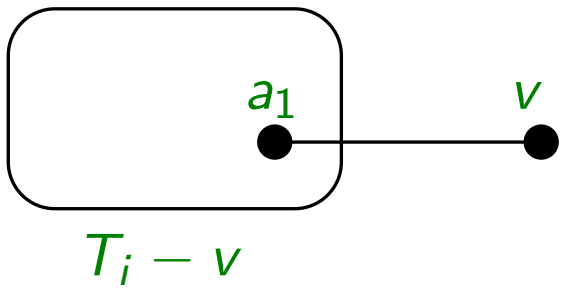
Agora resta mostrar que T é a única árvore em $\mathcal{T}$ tal que $f(T) = a$.



Código de Prüfer

Hipótese de indução: suponha que se $\mathcal{T}'$ é o conjunto das árvores T' com $V(T') = S'$, onde $S' \subseteq \mathbb{N}$ e $|S'| = n-1$, então para cada $a' \in (S')^{n-3}$, existe uma única árvore $T' \in \mathcal{T}'$ tal que $f(T') = a'$.

Sejam $T_1, T_2 \in \mathcal{T}$ tais que $f(T_1) = f(T_2) = a$. Pelo mesmo argumento anterior, segue que v é uma folha de T_i e va_1 é uma aresta de T_i. Além disso, $f(T_1 - v) = f(T_2 - v) = a'$. Por HI segue que $T_1 - v = T_2 - v$. Portanto, $T_1 = T_2$. Isto termina a prova. ■



Código de Prüfer (inversa)

INVERSA DO CÓDIGO DE PRÜFER ▷ versão recursiva
ENTRADA: $a = (a_1, \ldots, a_{n-2}) \in S^{n-2}$ e S onde $|S| = n$ e $n \geq 2$.
SAÍDA: árvore T com $V(T) = S$ tal que $f(T) = a$.

1. **se** $S = \{x, y\}$ ▷ i.e., $n = 2$ e $a = ()$
2. **então devolva** $T = (S, \{xy\})$
3. $v \leftarrow \min(S - \{a_1, a_2, \ldots, a_{n-2}\})$
4. $a' \leftarrow (a_2, \ldots, a_{n-2})$
5. $T' \leftarrow$ INVERSA DO CÓDIGO DE PRÜFER$(a', S - \{v\})$
6. $T \leftarrow (V(T') \cup \{v\}, E(T') \cup \{va_1\})$
7. **devolva** T

Exercício. Mostre que o algoritmo é de fato a inversa do Código de Prüfer.

Código de Prüfer (inversa)

INVERSA DO CÓDIGO DE PRÜFER ▷ versão iterativa
ENTRADA: $a = (a_1, \ldots, a_{n-2}) \in S^{n-2}$ onde $|S| = n$ e $n \geq 2$.
SAÍDA: árvore T com $V(T) = S$ tal que $f(T) = a$.

1. $T \leftarrow (S, \emptyset)$
2. $U \leftarrow S$
3. **para** $i \leftarrow 1$ **até** $n-2$ **faça**
4. $\quad v \leftarrow \min(U - \{a_i, a_{i+1}, \ldots, a_{n-2}\})$
5. $\quad T \leftarrow T + va_i$
6. $\quad U \leftarrow U - \{v\}$
7. sejam x, y os vértices de U
8. $T \leftarrow T + xy$
9. **devolva** T

Exercício. Mostre que o algoritmo é de fato a inversa do Código de Prüfer.

Contagem de árvores geradoras

Denote por $\tau(G)$ o número de árvores geradoras de um grafo G.

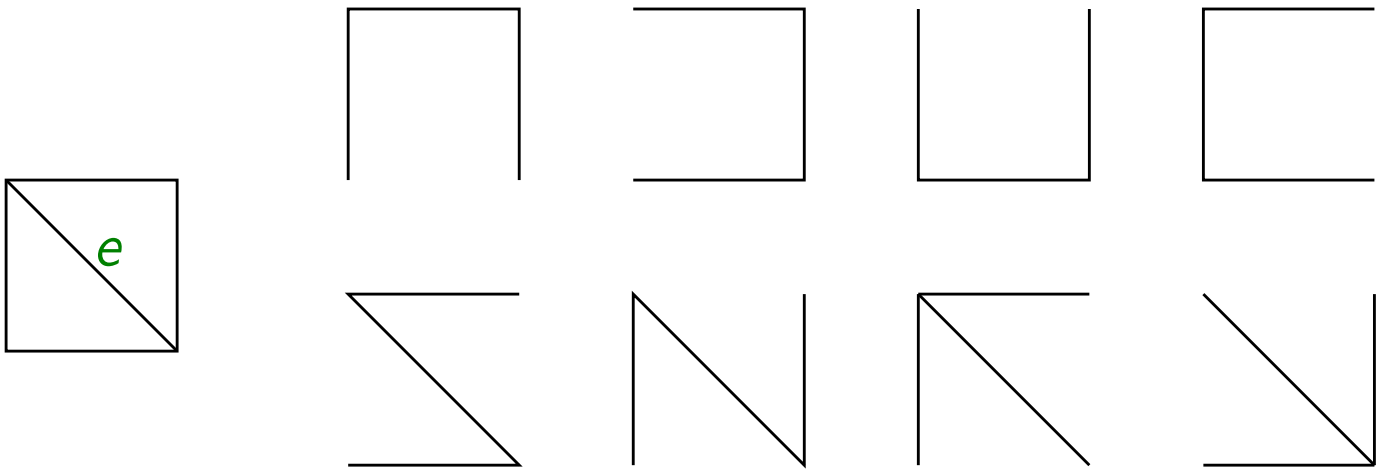
Uma forma equivalente de enunciar a Fórmula de Cayley é a seguinte.

Teorema. Seja $n \geq 2$ um inteiro. Então $\tau(K_n) = n^{n-2}$.

- Considere agora o problema mais geral de determinar $\tau(G)$ para um grafo arbitrário G.
- Não se pode esperar que haja uma fórmula simples em função apenas de $n(G)$ ou $m(G)$.
- Mostraremos uma recorrência que permite calcular (ineficientemente) $\tau(G)$ para qualquer grafo G.

Contagem de árvores geradoras

Seja G um grafo e seja $e \in E(G)$. Note que o número de árvores geradoras de G é igual ao número de árvores geradoras de G que não contém e **mais** o número de árvores geradoras de G que contém e.



$$\tau(G) = \tau(G - e) + \tau(??)$$

Identificação de vértices e contração de aresta

Identificar dois vértices não-adjacentes x e y de um grafo G é substituir x e y por um novo vértice incidente a todas as arestas que eram incidentes em G a x ou a y. Denotamos este grafo por $G/\{x, y\}$.

Contrair uma aresta e de um grafo G é remover e de G e então (se e não for um laço) identificar seus extremos. Denotamos este grafo por G/e.

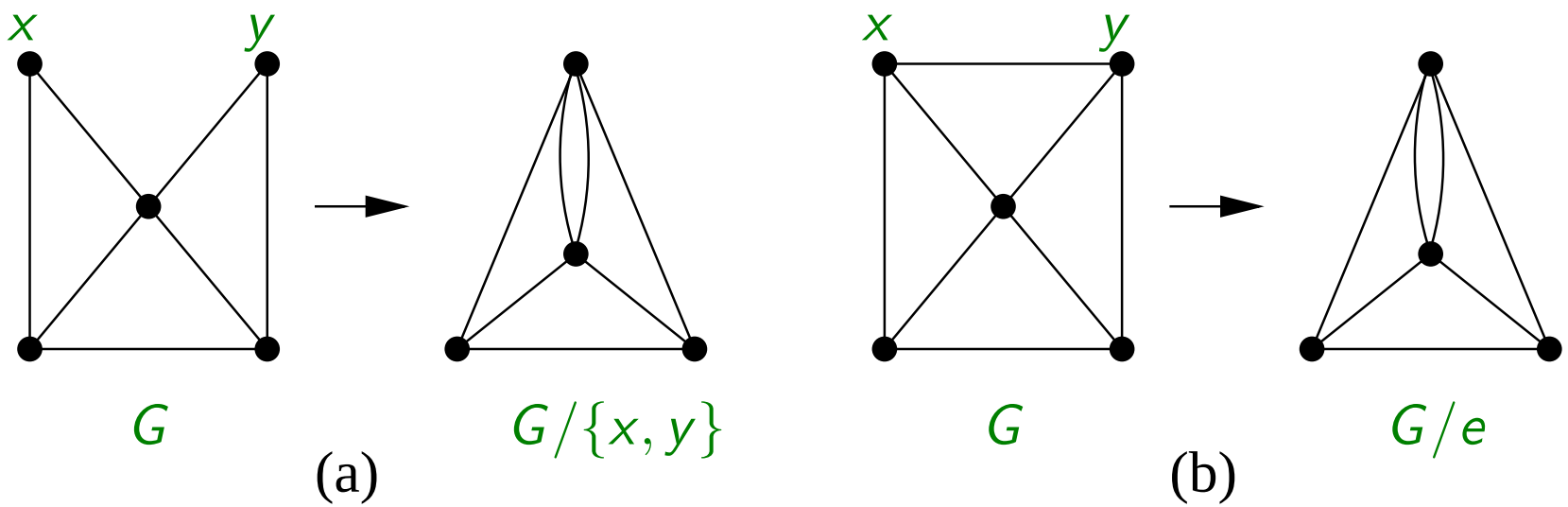
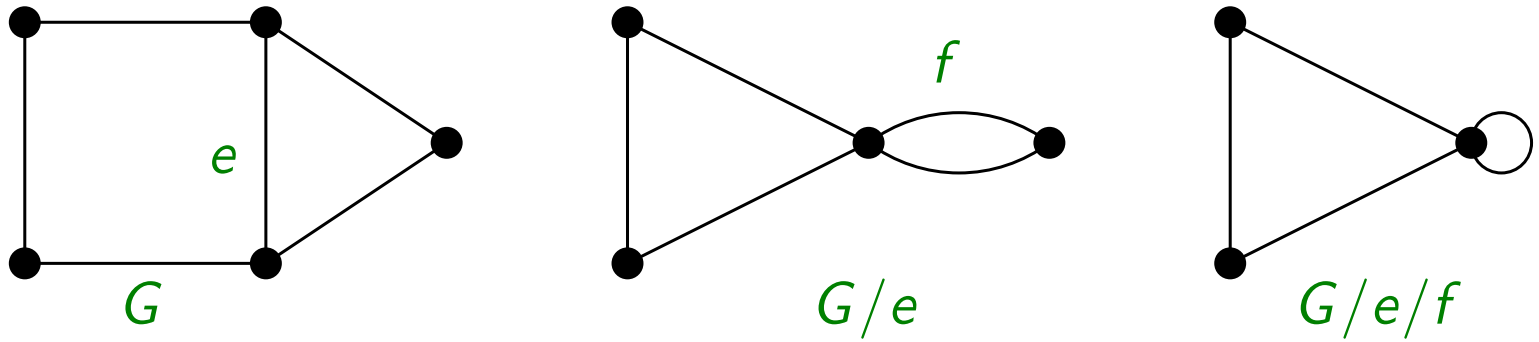


Figura: (a) Identificando dois vértices e (b) contraindo uma aresta.

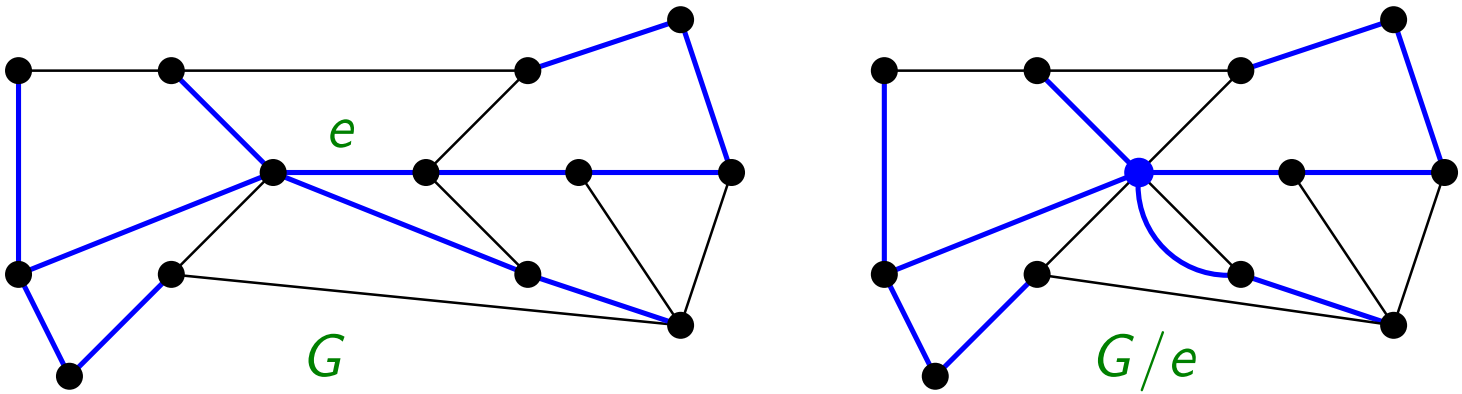
Contração de aresta



- Se e é um laço, então $G - e = G/e$.
- Se e é um laço, então e não pertence a nenhuma árvore geradora de G.

Contração de aresta e árvores geradores

Proposição. Seja G um grafo e seja e uma aresta de G que não é um laço. Então $\tau(G/e)$ é igual ao número de árvores geradoras de G que contém e. ■



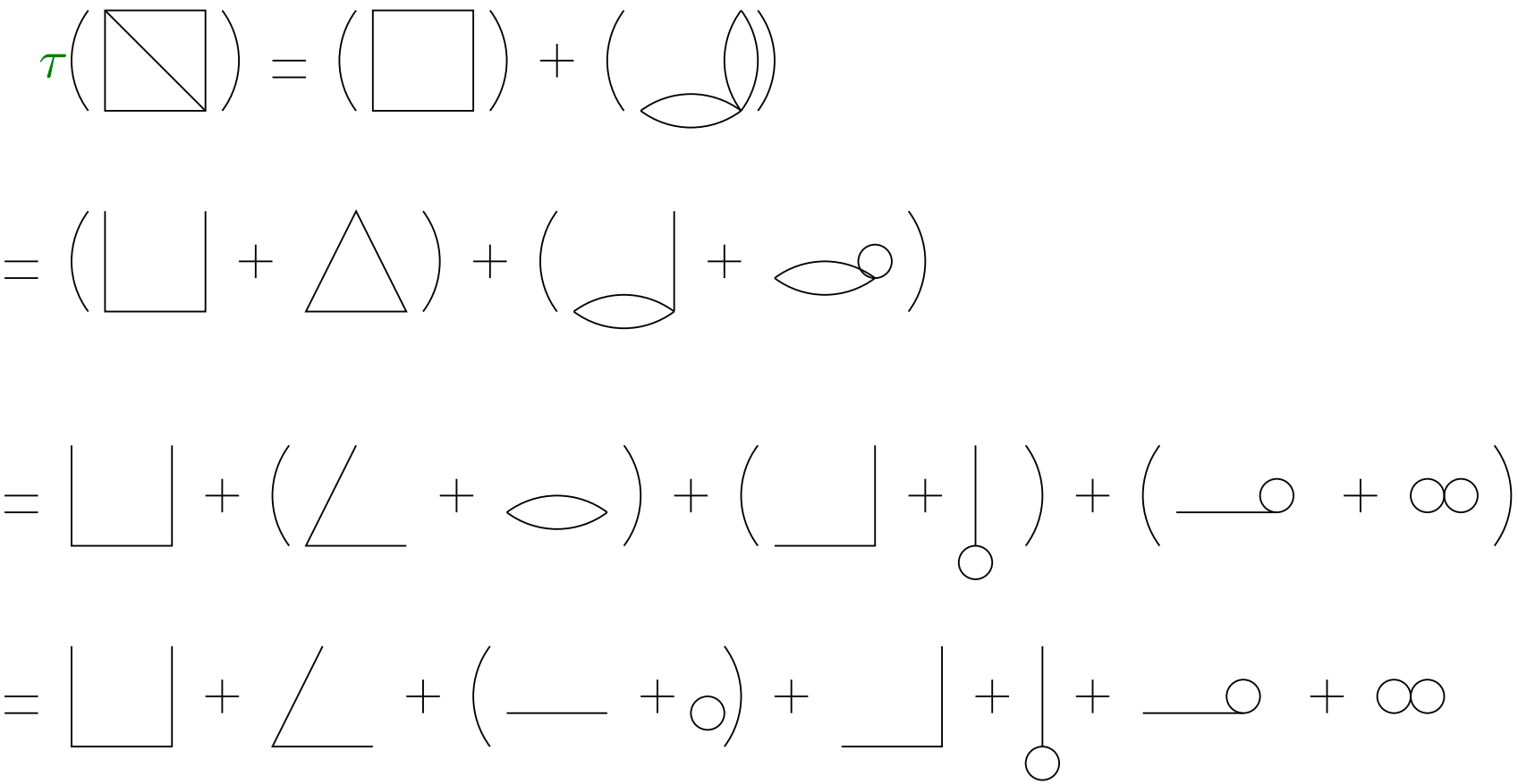
Contagem de árvores geradoras

Teorema. Seja G um grafo e seja e uma aresta de G que não é um laço. Então $\tau(G) = \tau(G - e) + \tau(G/e)$. ■

Em princípio, podemos usar este método para calcular $\tau(G)$. Entretanto, é fácil ver que isto pode gastar tempo exponencial em $m(G)$ (árvore binária de altura m tem 2^m nós).

Aplicação da recorrência

$\tau(K_4 - e) = 8$



Referências

- Em BM06, os autores descrevem outra forma bem interessante de mostrar que $\tau(K^n) = n^{n-2}$.
- Existe um algoritmo polinomial para determinar $\tau(G)$ para qualquer grafo G. A ideia é reduzir o problema a calcular o determinante de uma certa matriz. Veja West96.
- Usei principalmente as Seções 2.1 (Subseção Property of Trees), a Seção 2.2 e a Seção 2.3 (Subseção Minimum Spanning Trees) de West96 para preparar estes slides.