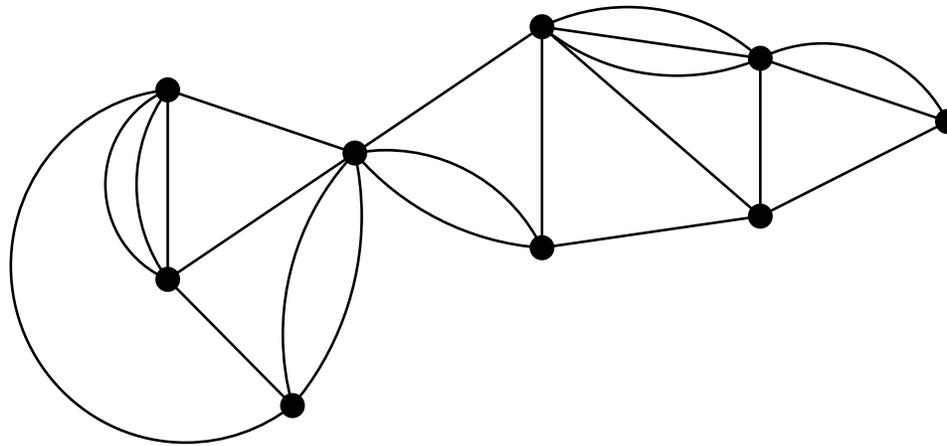


Análise de Algoritmos

**Estes slides são adaptações de slides
do Prof. Paulo Feofiloff e do Prof. José Coelho de Pina.**

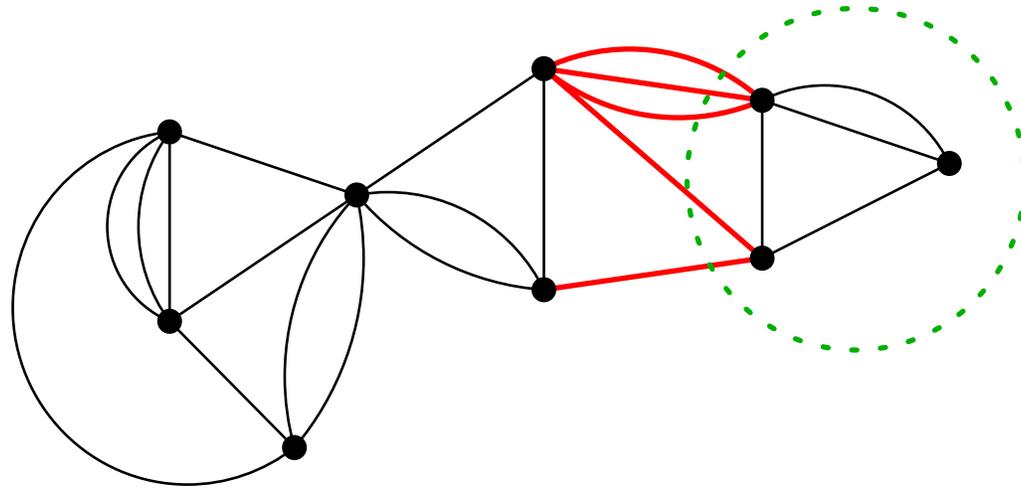
Cortes em grafos

G : grafo (não orientado) sem laços, possivelmente com arestas paralelas.



Cortes em grafos

G : grafo (não orientado) sem laços, possivelmente com arestas paralelas.



Conjunto de arestas C é um **corte** se existe um conjunto $S \subseteq V_G$ não vazio tal que

$$C = \{e \in E_G \mid e \text{ tem uma ponta em } S \text{ e outra fora de } S.\}$$

Ou seja, $C = \delta_G(S) = \delta_G(V_G \setminus S)$.

Corte mínimo

G : grafo sem laços, possivelmente com arestas paralelas.

$C \subseteq E_G$ é um **corte** se existe $S \subseteq V_G$ não vazio tal que

$$C = \{e \in E_G \mid e \text{ tem uma ponta em } S \text{ e outra fora de } S.\}$$

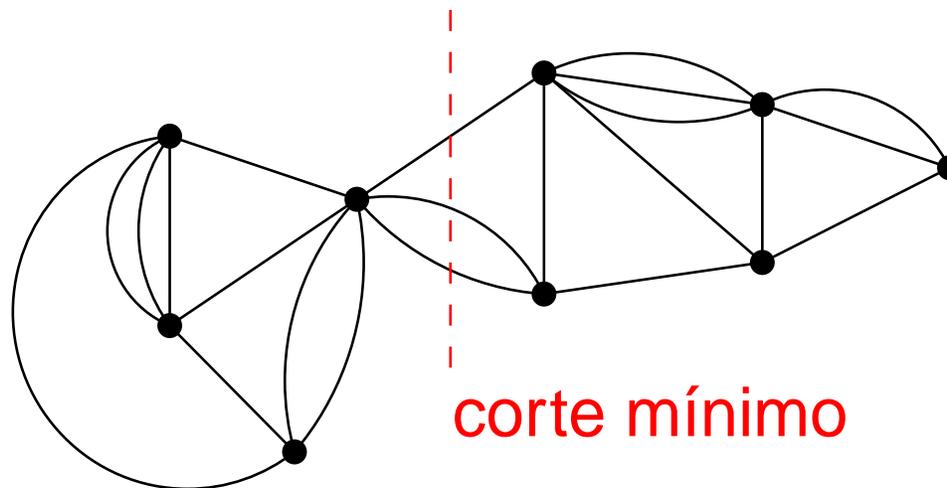
Corte mínimo

G : grafo sem laços, possivelmente com arestas paralelas.

$C \subseteq E_G$ é um **corte** se existe $S \subseteq V_G$ não vazio tal que

$$C = \{e \in E_G \mid e \text{ tem uma ponta em } S \text{ e outra fora de } S.\}$$

Problema: Dado G , encontrar um corte C com $|C|$ mínimo.



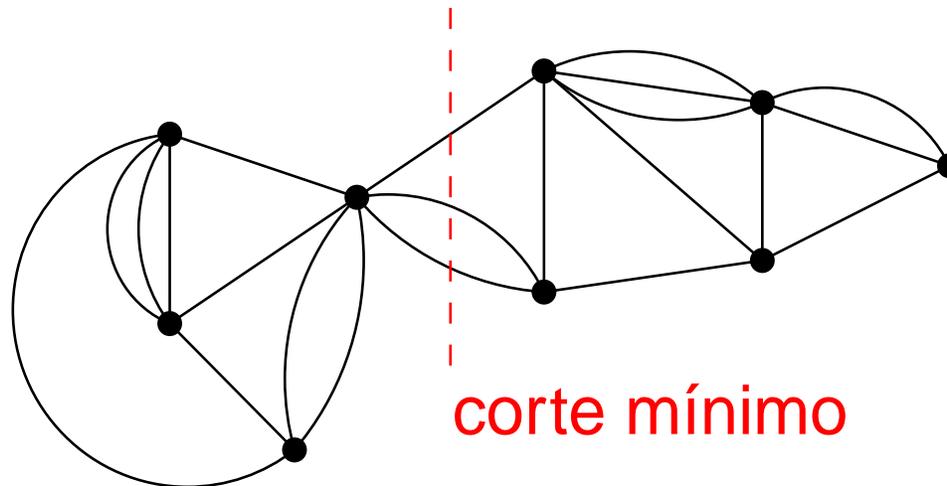
Corte mínimo

G : grafo sem laços, possivelmente com arestas paralelas.

$C \subseteq E_G$ é um **corte** se existe $S \subseteq V_G$ não vazio tal que

$$C = \{e \in E_G \mid e \text{ tem uma ponta em } S \text{ e outra fora de } S.\}$$

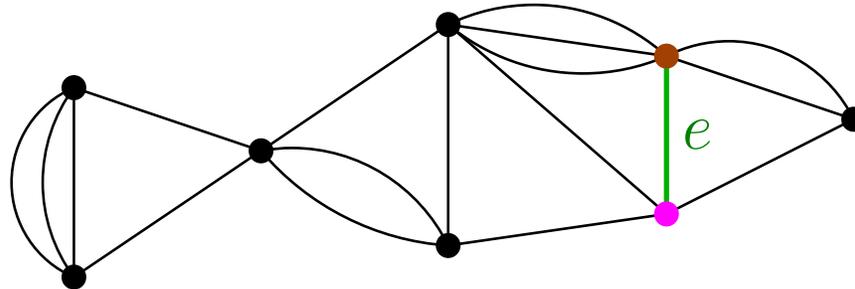
Problema: Dado G , encontrar um corte C com $|C|$ mínimo.



Esta aula: algoritmo aleatorizado que devolve um corte mínimo com alta probabilidade.

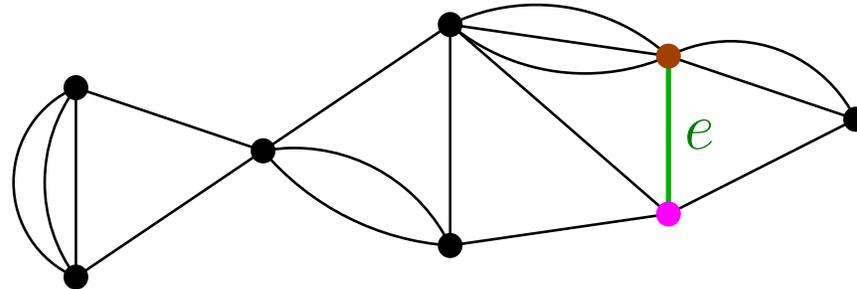
Contração de arestas

Dado um grafo G e uma aresta $e = uv$,
a **contração** $G|_e$ é o grafo (V', E') onde...

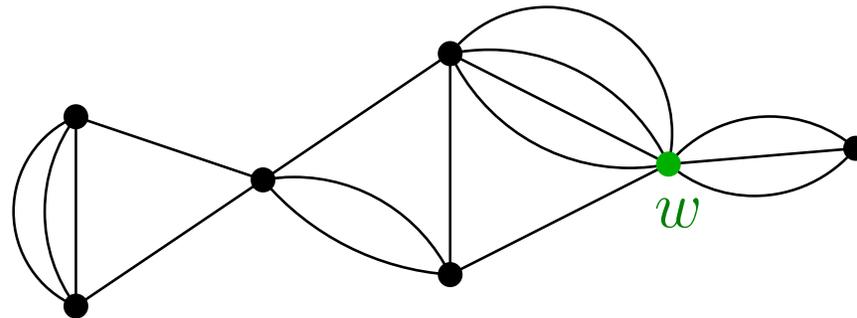


Contração de arestas

Dado um grafo G e uma aresta $e = uv$,
a **contração** $G|e$ é o grafo (V', E') onde...

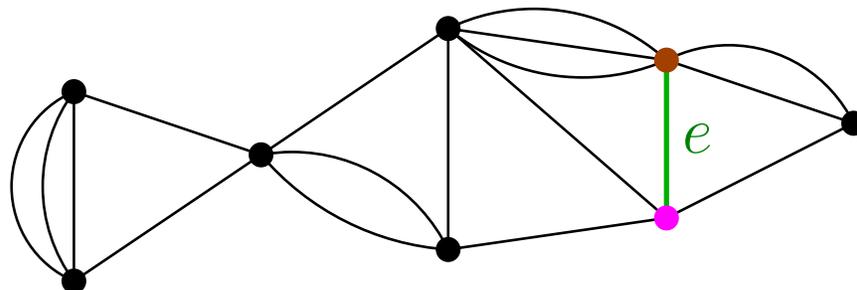


$V' = V \setminus \{u, v\} \cup \{w\}$ e $E' = E_G \setminus \{f \mid \text{pontas}(f) = \{u, v\}\}$
e cada aresta f em E' tem as mesmas pontas que f em E_G
exceto por u e v , que são substituídos por w .

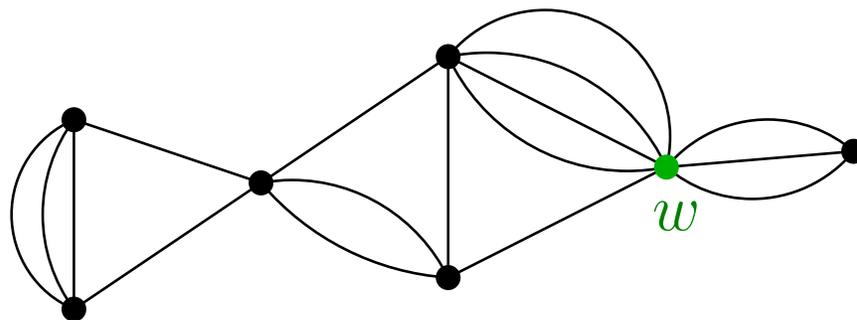


Contração de arestas

Dado um grafo G e uma aresta $e = uv$, a **contração** $G|e$ é o grafo (V', E') onde...



$V' = V \setminus \{u, v\} \cup \{w\}$ e $E' = E_G \setminus \{f \mid \text{pontas}(f) = \{u, v\}\}$ e cada aresta f em E' tem as mesmas pontas que f em E_G exceto por u e v , que são substituídos por w .



Proposição: Se C é corte de $G|e$, então C é corte de G .

Algoritmo de Karger

Problema: Dado G , encontrar um corte C com $|C|$ mínimo.

Algoritmo de Karger

Problema: Dado G , encontrar um corte C com $|C|$ mínimo.

Karger (G)

- 1 **enquanto** $|V_G| > 2$ **faça**
- 2 $e \leftarrow$ **RANDOM_EDGE**(E_G)
- 3 $G \leftarrow G|e$
- 4 **devolva** E_G

Algoritmo de Karger

Problema: Dado G , encontrar um corte C com $|C|$ mínimo.

Karger (G)

- 1 **enquanto** $|V_G| > 2$ **faça**
- 2 $e \leftarrow \text{RANDOM_EDGE}(E_G)$
- 3 $G \leftarrow G|e$
- 4 **devolva** E_G

Pela proposição, o algoritmo devolve de fato um corte de G .

Algoritmo de Karger

Problema: Dado G , encontrar um corte C com $|C|$ mínimo.

Karger (G)

- 1 **enquanto** $|V_G| > 2$ **faça**
- 2 $e \leftarrow$ **RANDOM_EDGE**(E_G)
- 3 $G \leftarrow G|e$
- 4 **devolva** E_G

Pela proposição, o algoritmo devolve de fato um corte de G .

Claramente ele consome tempo polinomial.
(O número de iteração é $|V_G| - 2$.)

Análise do algoritmo

Seja C_K o corte devolvido por Karger (G).

Vamos mostrar que

$$\Pr\{C_K \text{ não é um corte mínimo}\} < q(n) < 1,$$

onde $n = |V_G|$.

Análise do algoritmo

Seja C_K o corte devolvido por Karger (G).

Vamos mostrar que

$$\Pr\{C_K \text{ não é um corte mínimo}\} < q(n) < 1,$$

onde $n = |V_G|$.

Executa-se o algoritmo r vezes
e devolve-se o menor corte produzido.

A chance do corte devolvido não ser mínimo será $< q(n)^r$.

Análise do algoritmo

Seja C um corte mínimo de G .

Análise do algoritmo

Seja C um corte mínimo de G .

O corte C não é a resposta do algoritmo sse alguma aresta de C foi contraída.

Análise do algoritmo

Seja C um corte mínimo de G .

O corte C não é a resposta do algoritmo sse alguma aresta de C foi contraída.

Seja E_i o evento:

uma aresta de C não é sorteada na iteração i .

Análise do algoritmo

Seja C um corte mínimo de G .

O corte C não é a resposta do algoritmo sse alguma aresta de C foi contraída.

Seja E_i o evento:

uma aresta de C não é sorteada na iteração i .

Vamos delimitar **inferiormente** $\Pr\{E_1 \cap E_2 \cap \cdots \cap E_{n-2}\}$.

Análise do algoritmo

Seja C um corte mínimo de G .

O corte C não é a resposta do algoritmo sse alguma aresta de C foi contraída.

Seja E_i o evento:

uma aresta de C não é sorteada na iteração i .

Vamos delimitar **inferiormente** $\Pr\{E_1 \cap E_2 \cap \cdots \cap E_{n-2}\}$.

Tal probabilidade é igual a

$$\Pr\{E_1\} \cdot \Pr\{E_2|E_1\} \cdots \Pr\{E_{n-2}|E_1 \cap E_2 \cap \cdots \cap E_{n-3}\}.$$

Análise do algoritmo

Seja C um corte mínimo de G .

O corte C não é a resposta do algoritmo sse alguma aresta de C foi contraída.

Seja E_i o evento:

uma aresta de C não é sorteada na iteração i .

Vamos delimitar **inferiormente** $\Pr\{E_1 \cap E_2 \cap \dots \cap E_{n-2}\}$.

Tal probabilidade é igual a

$$\Pr\{E_1\} \cdot \Pr\{E_2|E_1\} \cdot \dots \cdot \Pr\{E_{n-2}|E_1 \cap E_2 \cap \dots \cap E_{n-3}\}.$$

Para começar, como delimitar $\Pr\{E_1\}$?

Análise do algoritmo

C : um corte mínimo de G e $k = |C|$.

E_i : uma aresta de C não é sorteada na iteração i .

Como delimitar $\Pr\{E_1\}$?

Análise do algoritmo

C : um corte mínimo de G e $k = |C|$.

E_i : uma aresta de C não é sorteada na iteração i .

Como delimitar $\Pr\{E_1\}$?

Como C é mínimo, $\delta_G(v) \geq k$ para todo vértice v .

Logo $|E_G| \geq kn/2$.

Análise do algoritmo

C : um corte mínimo de G e $k = |C|$.

E_i : uma aresta de C não é sorteada na iteração i .

Como delimitar $\Pr\{E_1\}$?

Como C é mínimo, $\delta_G(v) \geq k$ para todo vértice v .

Logo $|E_G| \geq kn/2$.

Portanto $\Pr\{\bar{E}_1\} \leq \frac{k}{\frac{kn}{2}} = \frac{2}{n}$.

Análise do algoritmo

C : um corte mínimo de G e $k = |C|$.

E_i : uma aresta de C não é sorteada na iteração i .

Como delimitar $\Pr\{E_1\}$?

Como C é mínimo, $\delta_G(v) \geq k$ para todo vértice v .

Logo $|E_G| \geq kn/2$.

Portanto $\Pr\{\bar{E}_1\} \leq \frac{k}{\frac{kn}{2}} = \frac{2}{n}$.

Ou seja, $\Pr\{E_1\} \geq 1 - \frac{2}{n}$.

Análise do algoritmo

C : um corte mínimo de G e $k = |C|$.

E_i : uma aresta de C não é sorteada na iteração i .

Como delimitar $\Pr\{E_1\}$?

Como C é mínimo, $\delta_G(v) \geq k$ para todo vértice v .

Logo $|E_G| \geq kn/2$.

Portanto $\Pr\{\bar{E}_1\} \leq \frac{k}{\frac{kn}{2}} = \frac{2}{n}$.

Ou seja, $\Pr\{E_1\} \geq 1 - \frac{2}{n}$.

Como delimitar $\Pr\{E_{i+1} | E_1 \cap E_2 \cap \dots \cap E_i\}$?

Análise do algoritmo

C : um corte mínimo de G e $k = |C|$.

E_i : uma aresta de C não é sorteada na iteração i .

Vimos que $\Pr\{E_1\} \geq 1 - \frac{2}{n}$.

Como delimitar $\Pr\{E_{i+1} | E_1 \cap E_2 \cap \dots \cap E_i\}$?

Análise do algoritmo

C : um corte mínimo de G e $k = |C|$.

E_i : uma aresta de C não é sorteada na iteração i .

Vimos que $\Pr\{E_1\} \geq 1 - \frac{2}{n}$.

Como delimitar $\Pr\{E_{i+1} | E_1 \cap E_2 \cap \dots \cap E_i\}$?

No começo da iteração $i + 1$, temos que $|E_G| \geq k(n - i)/2$.

Análise do algoritmo

C : um corte mínimo de G e $k = |C|$.

E_i : uma aresta de C não é sorteada na iteração i .

Vimos que $\Pr\{E_1\} \geq 1 - \frac{2}{n}$.

Como delimitar $\Pr\{E_{i+1} | E_1 \cap E_2 \cap \dots \cap E_i\}$?

No começo da iteração $i + 1$, temos que $|E_G| \geq k(n - i)/2$.

Logo $\Pr\{E_{i+1}^- | E_1 \cap E_2 \cap \dots \cap E_i\} \leq \frac{k}{\frac{k(n-i)}{2}} = \frac{2}{n-i}$.

Análise do algoritmo

C : um corte mínimo de G e $k = |C|$.

E_i : uma aresta de C não é sorteada na iteração i .

Vimos que $\Pr\{E_1\} \geq 1 - \frac{2}{n}$.

Como delimitar $\Pr\{E_{i+1} | E_1 \cap E_2 \cap \dots \cap E_i\}$?

No começo da iteração $i + 1$, temos que $|E_G| \geq k(n - i)/2$.

Logo $\Pr\{E_{i+1}^- | E_1 \cap E_2 \cap \dots \cap E_i\} \leq \frac{k}{\frac{k(n-i)}{2}} = \frac{2}{n-i}$.

Ou seja, $\Pr\{E_{i+1} | E_1 \cap E_2 \cap \dots \cap E_i\} \geq 1 - \frac{2}{n-i}$.

Análise do algoritmo

Seja C_K o corte devolvido por Karger (G).

C : um corte mínimo de G .

Queremos delimitar **inferiormente** $\Pr\{C_K = C\}$.

Análise do algoritmo

Seja C_K o corte devolvido por Karger (G).

C : um corte mínimo de G .

Queremos delimitar **inferiormente** $\Pr\{C_K = C\}$.

Ou seja, delimitar $\Pr\{E_1 \cap E_2 \cap \dots \cap E_{n-2}\}$.

Já sabemos que $\Pr\{E_{i+1} | E_1 \cap E_2 \cap \dots \cap E_i\} \geq 1 - \frac{2}{n-i}$.

Análise do algoritmo

Seja C_K o corte devolvido por Karger (G).

C : um corte mínimo de G .

Queremos delimitar inferiormente $\Pr\{C_K = C\}$.

Ou seja, delimitar $\Pr\{E_1 \cap E_2 \cap \dots \cap E_{n-2}\}$.

Já sabemos que $\Pr\{E_{i+1} | E_1 \cap E_2 \cap \dots \cap E_i\} \geq 1 - \frac{2}{n-i}$.

Então

$$\Pr\{C_K = C\} \geq \prod_{i=0}^{n-3} \left(1 - \frac{2}{n-i}\right)$$

Análise do algoritmo

Seja C_K o corte devolvido por Karger (G).

C : um corte mínimo de G .

Queremos delimitar **inferiormente** $\Pr\{C_K = C\}$.

Ou seja, delimitar $\Pr\{E_1 \cap E_2 \cap \dots \cap E_{n-2}\}$.

Já sabemos que $\Pr\{E_{i+1} | E_1 \cap E_2 \cap \dots \cap E_i\} \geq 1 - \frac{2}{n-i}$.

Então

$$\begin{aligned}\Pr\{C_K = C\} &\geq \prod_{i=0}^{n-3} \left(1 - \frac{2}{n-i}\right). \\ &= \left(1 - \frac{2}{n}\right) \left(1 - \frac{2}{n-1}\right) \left(1 - \frac{2}{n-2}\right) \dots \left(1 - \frac{2}{4}\right) \left(1 - \frac{2}{3}\right)\end{aligned}$$

Análise do algoritmo

Seja C_K o corte devolvido por Karger (G).

C : um corte mínimo de G .

Queremos delimitar inferiormente $\Pr\{C_K = C\}$.

Ou seja, delimitar $\Pr\{E_1 \cap E_2 \cap \dots \cap E_{n-2}\}$.

Já sabemos que $\Pr\{E_{i+1} | E_1 \cap E_2 \cap \dots \cap E_i\} \geq 1 - \frac{2}{n-i}$.

Então

$$\begin{aligned}\Pr\{C_K = C\} &\geq \prod_{i=0}^{n-3} \left(1 - \frac{2}{n-i}\right). \\ &= \left(1 - \frac{2}{n}\right) \left(1 - \frac{2}{n-1}\right) \left(1 - \frac{2}{n-2}\right) \dots \left(1 - \frac{2}{4}\right) \left(1 - \frac{2}{3}\right) \\ &= \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdot \frac{n-4}{n-2} \dots \frac{2}{4} \cdot \frac{1}{3}\end{aligned}$$

Análise do algoritmo

Seja C_K o corte devolvido por Karger (G).

C : um corte mínimo de G .

Queremos delimitar inferiormente $\Pr\{C_K = C\}$.

Ou seja, delimitar $\Pr\{E_1 \cap E_2 \cap \dots \cap E_{n-2}\}$.

Já sabemos que $\Pr\{E_{i+1} | E_1 \cap E_2 \cap \dots \cap E_i\} \geq 1 - \frac{2}{n-i}$.

Então

$$\begin{aligned}\Pr\{C_K = C\} &\geq \prod_{i=0}^{n-3} \left(1 - \frac{2}{n-i}\right). \\ &= \left(1 - \frac{2}{n}\right) \left(1 - \frac{2}{n-1}\right) \left(1 - \frac{2}{n-2}\right) \dots \left(1 - \frac{2}{4}\right) \left(1 - \frac{2}{3}\right) \\ &= \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdot \frac{n-4}{n-2} \dots \frac{2}{4} \cdot \frac{1}{3} = \frac{2}{n(n-1)}.\end{aligned}$$

Conclusão

$$\text{Logo } \Pr\{C_K \neq C\} \leq 1 - \frac{2}{n(n-1)} = q(n)$$

Conclusão

Logo $\Pr\{C_K \neq C\} \leq 1 - \frac{2}{n(n-1)} = q(n)$
e $\Pr\{C_K \text{ não é um corte mínimo}\} \leq q(n)$.

Conclusão

Logo $\Pr\{C_K \neq C\} \leq 1 - \frac{2}{n(n-1)} = q(n)$
e $\Pr\{C_K \text{ não é um corte mínimo}\} \leq q(n)$.

Se executarmos o algoritmo $r = c \frac{n(n-1)}{2} \ln n$ vezes, então

Conclusão

Logo $\Pr\{C_K \neq C\} \leq 1 - \frac{2}{n(n-1)} = q(n)$
e $\Pr\{C_K \text{ não é um corte mínimo}\} \leq q(n)$.

Se executarmos o algoritmo $r = c \frac{n(n-1)}{2} \ln n$ vezes, então

$$\begin{aligned}\Pr\{C_K^* \text{ não é um corte mínimo}\} &\leq q(n)^r \\ &= \left(1 - \frac{2}{n(n-1)}\right)^{c \frac{n(n-1)}{2} \ln n} \\ &= \left(\frac{1}{e}\right)^{c \ln n} = \frac{1}{n^c}.\end{aligned}$$

Conclusão

Logo $\Pr\{C_K \neq C\} \leq 1 - \frac{2}{n(n-1)} = q(n)$
e $\Pr\{C_K \text{ não é um corte mínimo}\} \leq q(n)$.

Se executarmos o algoritmo $r = c \frac{n(n-1)}{2} \ln n$ vezes, então

$$\begin{aligned}\Pr\{C_K^* \text{ não é um corte mínimo}\} &\leq q(n)^r \\ &= \left(1 - \frac{2}{n(n-1)}\right)^{c \frac{n(n-1)}{2} \ln n} \\ &= \left(\frac{1}{e}\right)^{c \ln n} = \frac{1}{n^c}.\end{aligned}$$

Erra com probabilidade exponencialmente pequena em n .

Conclusão

Logo $\Pr\{C_K \neq C\} \leq 1 - \frac{2}{n(n-1)} = q(n)$
e $\Pr\{C_K \text{ não é um corte mínimo}\} \leq q(n)$.

Se executarmos o algoritmo $r = c \frac{n(n-1)}{2} \ln n$ vezes, então

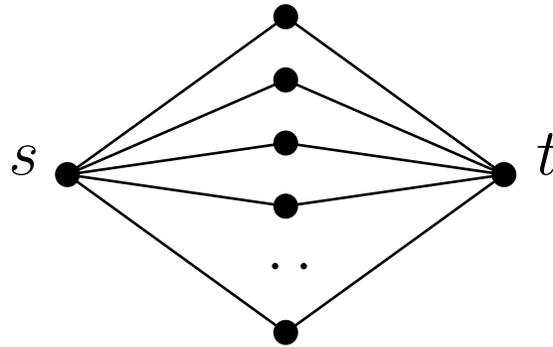
$$\begin{aligned}\Pr\{C_K^* \text{ não é um corte mínimo}\} &\leq q(n)^r \\ &= \left(1 - \frac{2}{n(n-1)}\right)^{c \frac{n(n-1)}{2} \ln n} \\ &= \left(\frac{1}{e}\right)^{c \ln n} = \frac{1}{n^c}.\end{aligned}$$

Erra com probabilidade exponencialmente pequena em n .

Como r é polinomial em n , o algoritmo é polinomial.

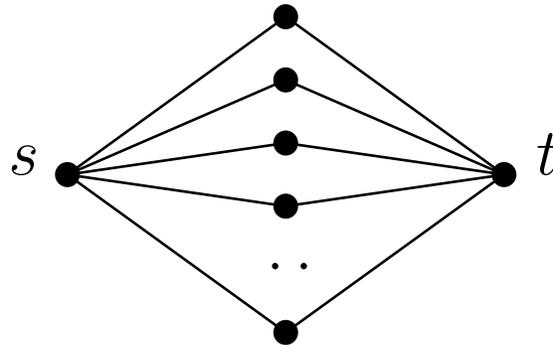
Uma pergunta...

Quantos st -cortes mínimos diferentes podem existir em G ?



Uma pergunta...

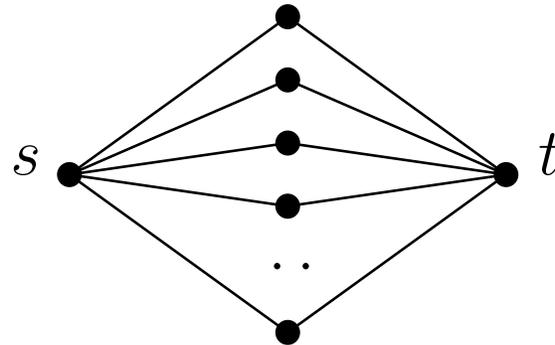
Quantos st -cortes mínimos diferentes podem existir em G ?



Um número exponencial em $n...$

Uma pergunta...

Quantos st -cortes mínimos diferentes podem existir em G ?

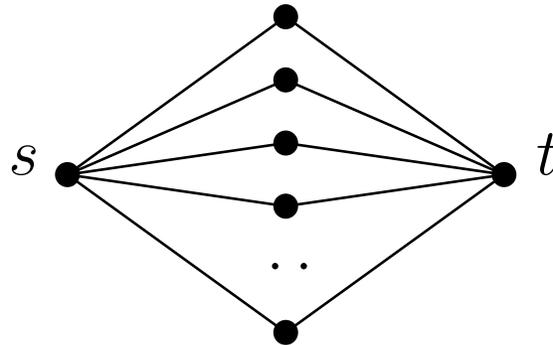


Um número exponencial em $n...$

Quantos cortes mínimos diferentes podem existir?

Uma pergunta...

Quantos st -cortes mínimos diferentes podem existir em G ?



Um número exponencial em n ...

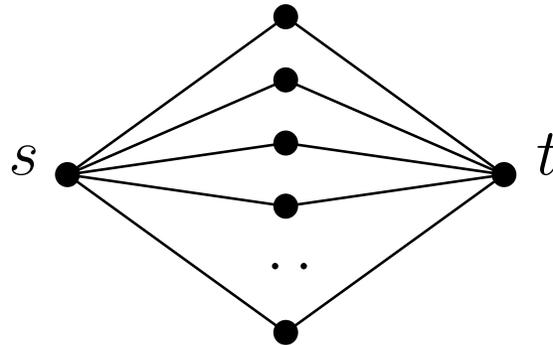
Quantos cortes mínimos diferentes podem existir?

Como $\Pr\{C_K = C\} \geq \frac{2}{n(n-1)} = 1/\binom{n}{2}$,

há no máximo $\binom{n}{2}$ cortes mínimos em G .

Uma pergunta...

Quantos st -cortes mínimos diferentes podem existir em G ?



Um número exponencial em n ...

Quantos cortes mínimos diferentes podem existir?

Como $\Pr\{C_K = C\} \geq \frac{2}{n(n-1)} = 1/\binom{n}{2}$,

há no máximo $\binom{n}{2}$ cortes mínimos em G .

Se G for por exemplo um circuito com n vértices...