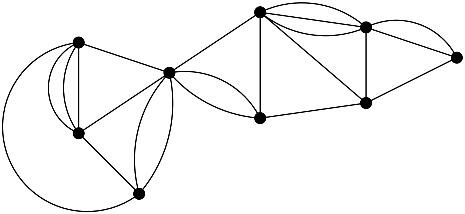
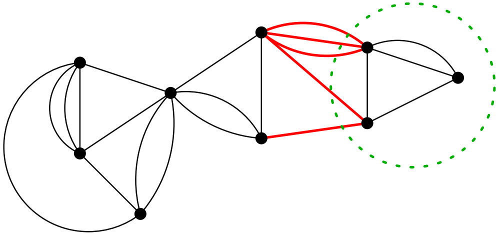
Cortes em grafos

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

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

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

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

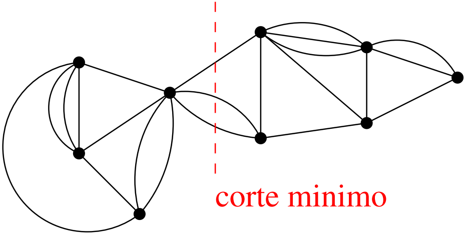
Corte mínimo

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

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

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



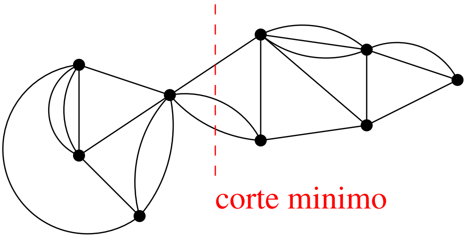
Corte mínimo

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

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

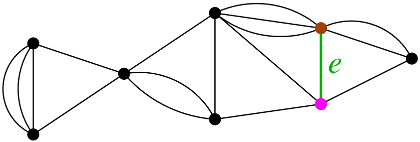
Problema: Dado G, encontrar um corte C com $|C|$ mínimo.



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

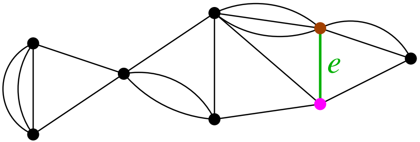
Contração de arestas

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

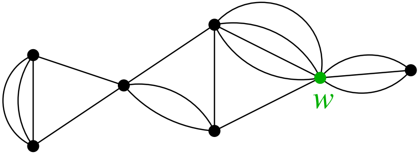


Contração de arestas

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

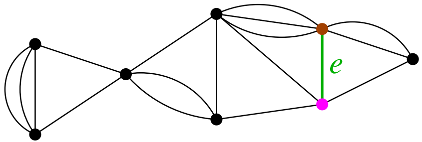


$V' = V \setminus \{u, v\} \cup \{w\}$ e $E' = E_G \setminus \{f | \mathrm{pontas}(f) = \{u, v\}\}$
e cada aresta f em E' tem as mesmas pontas que f em E_G exceto por u e v, que são substituídos por w.

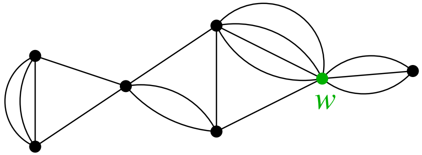


Contração de arestas

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 $E' = E_G \setminus \{f | \mathrm{pontas}(f) = \{u, v\}\}$
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)
  enquanto $|V_G| > 2$ **faça**
    $e \leftarrow$ RANDOM-E(E_G)
    $G \leftarrow G|e$
  devolva E_G

Algoritmo de Karger

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

KARGER (G)
    enquanto $|V_G| > 2$ **faça**
        $e \leftarrow$ RANDOM-E(E_G)
        $G \leftarrow G|e$
    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)
    enquanto |V_G| > 2 faça
        e ← RANDOM-E(E_G)
        G ← G|e
    devolva E_G
```

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

Claramente ele consume 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 $\Pr\{E_1 \cap E_2 \cap \cdots \cap E_{n-2}\}$.

Tal probabilidade é igual a

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

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 $\Pr\{E_{i+1}|E_1 \cap E_2 \cap \cdots \cap E_i\}$?

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 $\Pr\{E_{i+1} | E_1 \cap E_2 \cap \cdots \cap E_i\}$?

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 $\Pr\{E_{i+1} | E_1 \cap E_2 \cap \cdots \cap E_i\}$?

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 $\Pr\{E_{i+1} | E_1 \cap E_2 \cap \cdots \cap E_i\}$?

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

Logo $\Pr\{\bar{E}_{i+1} | E_1 \cap E_2 \cap \cdots \cap E_i\} \leq \frac{k}{\frac{k(n-i)}{2}} = \frac{2}{n-i}$.

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 $\Pr\{E_{i+1} | E_1 \cap E_2 \cap \cdots \cap E_i\}$?

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

Logo $\Pr\{\bar{E}_{i+1} | E_1 \cap E_2 \cap \cdots \cap E_i\} \leq \frac{k}{\frac{k(n-i)}{2}} = \frac{2}{n-i}$.

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

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

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

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

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

Então

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

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

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

Então

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

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

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

Então

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

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

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

Então

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

Conclusão

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 $\Pr\{C_K$ não é um corte mínimo$\}$

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 $\Pr\{C_K$ 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 $\Pr\{C_K$ 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 $\Pr\{C_K$ 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 c.

Conclusão

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

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

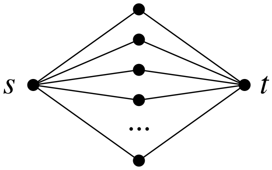
$$
\begin{aligned}
\Pr\{C_K^* \text{ não é um corte mínimo}\} &\leq q(n)^r \\
&= \left(1 - \frac{2}{n(n-1)}\right)^{c\frac{n(n-1)}{2}\ln n} \\
&< \left(\frac{1}{e}\right)^{c \ln n} = \frac{1}{n^c}.
\end{aligned}
$$

Erra com probabilidade exponencialmente pequena em c.

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

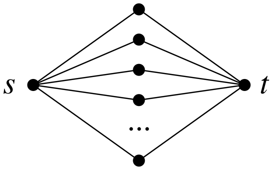
Uma pergunta...

Quantos *st*-cortes mínimos diferentes podem existir em G?



Uma pergunta...

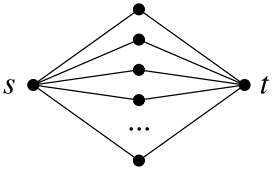
Quantos st-cortes mínimos diferentes podem existir em G?



Um número exponencial em n...

Uma pergunta...

Quantos *st*-cortes mínimos diferentes podem existir em G?

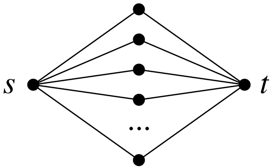


Um número exponencial em n...

Quantos cortes mínimos diferentes podem existir?

Uma pergunta...

Quantos *st*-cortes mínimos diferentes podem existir em G?



Um número exponencial em n...

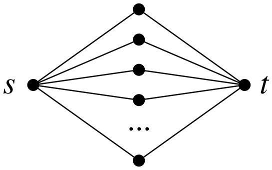
Quantos cortes mínimos diferentes podem existir?

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...

Hashing universal

U: conjunto universo (contém todas as possíveis chaves).

n: um número muito menor que $|U|$.

$\mathcal{H}$: conjunto de funções de U em $\{0, \ldots, n-1\}$.

Hashing universal

U: conjunto universo (contém todas as possíveis chaves).

n: um número muito menor que $|U|$.

$\mathcal{H}$: conjunto de funções de U em $\{0, \ldots, n-1\}$.

Se, para cada par de chaves k, ℓ em U, o número de funções h em $\mathcal{H}$ tais que $h(k) = h(\ell)$ é no máximo $|\mathcal{H}|/n$, então $\mathcal{H}$ é uma coleção universal de hashing (para U e n).

Hashing universal

U: conjunto universo (contém todas as possíveis chaves).

n: um número muito menor que $|U|$.

$\mathcal{H}$: conjunto de funções de U em $\{0, \ldots, n-1\}$.

Se, para cada par de chaves k, ℓ em U, o número de funções h em $\mathcal{H}$ tais que $h(k) = h(\ell)$ é no máximo $|\mathcal{H}|/n$, então $\mathcal{H}$ é uma coleção universal de hashing (para U e n).

Teorema: Seja $\mathcal{H}$ uma coleção universal de hashing para U e n, seja $S \subseteq U$ tal que $|S| \leq n$ e $u \in U$. Se h é escolhida aleatoriamente de $\mathcal{H}$ e X é o número de elementos s em S tais que $h(s) = h(u)$, então $\mathrm{E}[X] \leq 1$ se $u \notin S$ e $\mathrm{E}[X] \leq 2$ se $u \in S$.

Hashing universal

U: conjunto universo (contém todas as possíveis chaves).

n: um número muito menor que $|U|$.

$\mathcal{H}$: conjunto de funções de U em $\{0, \ldots, n-1\}$.

Se, para cada par de chaves k, ℓ em U, o número de funções h em $\mathcal{H}$ tais que $h(k) = h(\ell)$ é no máximo $|\mathcal{H}|/n$, então $\mathcal{H}$ é uma coleção universal de hashing (para U e n).

Teorema: Seja $\mathcal{H}$ uma coleção universal de hashing para U e n, seja $S \subseteq U$ tal que $|S| \leq n$ e $u \in U$. Se h é escolhida aleatoriamente de $\mathcal{H}$ e X é o número de elementos s em S tais que $h(s) = h(u)$, então $\mathrm{E}[X] \leq 1$ se $u \notin S$ e $\mathrm{E}[X] \leq 2$ se $u \in S$.

Esboço da prova: Seja X_s a variável binária que vale 1 se $h(s) = h(u)$.

Hashing universal

U: conjunto universo (contém todas as possíveis chaves).

n: um número muito menor que $|U|$.

$\mathcal{H}$: conjunto de funções de U em $\{0, \ldots, n-1\}$.

Se, para cada par de chaves k, ℓ em U, o número de funções h em $\mathcal{H}$ tais que $h(k) = h(\ell)$ é no máximo $|\mathcal{H}|/n$, então $\mathcal{H}$ é uma coleção universal de hashing (para U e n).

Teorema: Seja $\mathcal{H}$ uma coleção universal de hashing para U e n, seja $S \subseteq U$ tal que $|S| \leq n$ e $u \in U$. Se h é escolhida aleatoriamente de $\mathcal{H}$ e X é o número de elementos s em S tais que $h(s) = h(u)$, então $\mathrm{E}[X] \leq 1$ se $u \notin S$ e $\mathrm{E}[X] \leq 2$ se $u \in S$.

Esboço da prova: Seja X_s a variável binária que vale 1 se $h(s) = h(u)$. Note que $\Pr\{X_u = 1\} = 1$ e $\Pr\{X_s = 1\} \leq 1/n$ se $s \neq u$.

Hashing universal

U: conjunto universo (contém todas as possíveis chaves).

n: um número muito menor que $|U|$.

$\mathcal{H}$: conjunto de funções de U em $\{0, \ldots, n-1\}$.

Se, para cada par de chaves k, ℓ em U, o número de funções h em $\mathcal{H}$ tais que $h(k) = h(\ell)$ é no máximo $|\mathcal{H}|/n$, então $\mathcal{H}$ é uma coleção universal de hashing (para U e n).

Teorema: Seja $\mathcal{H}$ uma coleção universal de hashing para U e n, seja $S \subseteq U$ tal que $|S| \leq n$ e $u \in U$. Se h é escolhida aleatoriamente de $\mathcal{H}$ e X é o número de elementos s em S tais que $h(s) = h(u)$, então $\mathrm{E}[X] \leq 1$ se $u \notin S$ e $\mathrm{E}[X] \leq 2$ se $u \in S$.

Esboço da prova: Seja X_s a variável binária que vale 1 se $h(s) = h(u)$. Note que $\Pr\{X_u = 1\} = 1$ e $\Pr\{X_s = 1\} \leq 1/n$ se $s \neq u$. Logo $\mathrm{E}[X] = \sum_{s \in S} \mathrm{E}[X_s] = \sum_{s \in S} \frac{1}{n} = |S|/n \leq 1$ se $u \notin S$ e

Hashing universal

U: conjunto universo (contém todas as possíveis chaves).

n: um número muito menor que $|U|$.

$\mathcal{H}$: conjunto de funções de U em $\{0, \ldots, n-1\}$.

Se, para cada par de chaves k, ℓ em U, o número de funções h em $\mathcal{H}$ tais que $h(k) = h(\ell)$ é no máximo $|\mathcal{H}|/n$, então $\mathcal{H}$ é uma coleção universal de hashing (para U e n).

Teorema: Seja $\mathcal{H}$ uma coleção universal de hashing para U e n, seja $S \subseteq U$ tal que $|S| \leq n$ e $u \in U$. Se h é escolhida aleatoriamente de $\mathcal{H}$ e X é o número de elementos s em S tais que $h(s) = h(u)$, então $\mathrm{E}[X] \leq 1$ se $u \notin S$ e $\mathrm{E}[X] \leq 2$ se $u \in S$.

Esboço da prova: Seja X_s a variável binária que vale 1 se $h(s) = h(u)$. Note que $\Pr\{X_u = 1\} = 1$ e $\Pr\{X_s = 1\} \leq 1/n$ se $s \neq u$. Logo $\mathrm{E}[X] = \sum_{s \in S} \mathrm{E}[X_s] = \sum_{s \in S} \frac{1}{n} = |S|/n \leq 1$ se $u \notin S$ e $\mathrm{E}[X] = 1 + \sum_{s \in S \setminus \{u\}} \frac{1}{n} < 1 + |S|/n \leq 2$ se $u \in S$. $\square$

Exemplo de coleção universal de hashing

Seja p um primo tal que $U \subseteq \{0, \ldots, p-1\}$.

$\mathbb{Z}_p$: conjunto $\{0, \ldots, p-1\}$.
$\mathbb{Z}_p^*$: conjunto $\{1, \ldots, p-1\}$.

Exemplo de coleção universal de hashing

Seja p um primo tal que $U \subseteq \{0, \ldots, p-1\}$.

$\mathbb{Z}_p$: conjunto $\{0, \ldots, p-1\}$.
$\mathbb{Z}_p^*$: conjunto $\{1, \ldots, p-1\}$.

Para todo a em $\mathbb{Z}_p^*$ e b em $\mathbb{Z}_p$, seja

$$h_{a,b}(k) \; = \; ((ak + b) \bmod p) \bmod n,$$

para todo k em U.

Exemplo de coleção universal de hashing

Seja p um primo tal que $U \subseteq \{0, \ldots, p-1\}$.

$\mathbb{Z}_p$: conjunto $\{0, \ldots, p-1\}$.
$\mathbb{Z}_p^*$: conjunto $\{1, \ldots, p-1\}$.

Para todo a em $\mathbb{Z}_p^*$ e b em $\mathbb{Z}_p$, seja

$$h_{a,b}(k) \;=\; ((ak+b) \bmod p) \bmod n,$$

para todo k em U.

A coleção $\mathcal{H} \;=\; \{h_{a,b} : a \in \mathbb{Z}_p^* \text{ e } b \in \mathbb{Z}_p\}$ é universal.

Esboço da prova no próximo slide.

Exemplo de coleção universal de hashing

Seja p um primo tal que $U \subseteq \{0, \ldots, p-1\}$.

$\mathbb{Z}_p$: conjunto $\{0, \ldots, p-1\}$.
$\mathbb{Z}_p^*$: conjunto $\{1, \ldots, p-1\}$.

Para todo a em $\mathbb{Z}_p^*$ e b em $\mathbb{Z}_p$, seja

$$h_{a,b}(k) \;=\; ((ak + b) \bmod p) \bmod n,$$

para todo k em U.

A coleção $\mathcal{H} \;=\; \{h_{a,b} : a \in \mathbb{Z}_p^* \text{ e } \; b \in \mathbb{Z}_p\}$ é universal.

Esboço da prova no próximo slide.

(KT Sec 13.6 para hashing)

Exemplo de coleção universal de hashing

Seja p um primo tal que $U \subseteq \{0, \ldots, p-1\}$.

Para todo a em $\mathbb{Z}_p^*$ e b em $\mathbb{Z}_p$, seja

$$h_{a,b}(k) \;=\; ((ak+b) \bmod p) \bmod n,$$

para todo k em U.

A coleção $\mathcal{H} \;=\; \{h_{a,b} : a \in \mathbb{Z}_p^* \text{ e } b \in \mathbb{Z}_p\}$ é universal.

Esboço da prova: Sejam $k, \ell \in U$ tais que $ak + b = a\ell + b \bmod p$.
Então $a(k-\ell) = 0 \bmod p$, o que implica que $a = 0$ ou $k = \ell$.

Exemplo de coleção universal de hashing

Seja p um primo tal que $U \subseteq \{0, \ldots, p-1\}$.

Para todo a em $\mathbb{Z}_p^*$ e b em $\mathbb{Z}_p$, seja

$$h_{a,b}(k) \;=\; ((ak+b) \bmod p) \bmod n,$$

para todo k em U.

A coleção $\mathcal{H} \;=\; \{h_{a,b} : a \in \mathbb{Z}_p^* \text{ e } b \in \mathbb{Z}_p\}$ é universal.

Esboço da prova: Sejam k, $\ell \in U$ tais que $ak + b = a\ell + b \bmod p$.
Então $a(k - \ell) = 0 \bmod p$, o que implica que $a = 0$ ou $k = \ell$.
Logo, como $a \neq 0$, se $k \neq \ell$, então $r = ak + b \neq a\ell + b = s$.

Exemplo de coleção universal de hashing

Seja p um primo tal que $U \subseteq \{0, \dots, p-1\}$.

Para todo a em $\mathbb{Z}_p^*$ e b em $\mathbb{Z}_p$, seja

$$h_{a,b}(k) \;=\; ((ak+b) \bmod p) \bmod n,$$

para todo k em U.

A coleção $\mathcal{H} \;=\; \{h_{a,b} : a \in \mathbb{Z}_p^* \text{ e } \; b \in \mathbb{Z}_p\}$ é universal.

Esboço da prova: Sejam $k, \ell \in U$ tais que $ak + b = a\ell + b \bmod p$.
Então $a(k-\ell) = 0 \bmod p$, o que implica que $a = 0$ ou $k = \ell$.
Logo, como $a \neq 0$, se $k \neq \ell$, então $r = ak + b \neq a\ell + b = s$.
Ademais, para cada par (a, b) está associado um par distinto (r, s), com $r \neq s$, já que $a = (r-s)(k-\ell)^{-1} \bmod p$ e $b = r - ak \bmod p$.

Exemplo de coleção universal de hashing

Seja p um primo tal que $U \subseteq \{0, \ldots, p-1\}$.

Para todo a em $\mathbb{Z}_p^*$ e b em $\mathbb{Z}_p$, seja

$$h_{a,b}(k) \;=\; ((ak+b) \bmod p) \bmod n,$$

para todo k em U.

A coleção $\mathcal{H} \;=\; \{h_{a,b} : a \in \mathbb{Z}_p^* \text{ e } \; b \in \mathbb{Z}_p\}$ é universal.

Esboço da prova: Sejam $k, \ell \in U$ tais que $ak + b = a\ell + b \bmod p$.
Então $a(k-\ell) = 0 \bmod p$, o que implica que $a = 0$ ou $k = \ell$.
Logo, como $a \neq 0$, se $k \neq \ell$, então $r = ak + b \neq a\ell + b = s$.
Ademais, para cada par (a, b) está associado um par distinto (r, s), com $r \neq s$, já que $a = (r-s)(k-\ell)^{-1} \bmod p$ e $b = r - ak \bmod p$.
Para cada r, temos $p-1$ valores possíveis para s em $\mathbb{Z}_p$.
Destes, não mais que p/n são iguais a $r \bmod n$.
Ou seja, $h(k) = h(\ell)$ para no máximo $|\mathcal{H}|/n$ das funções h de $\mathcal{H}$. $\square$