Hľadanie najlacnejšej kostry

Daný je graf $G = (V, E)$
ohodnotené hrany: $w : E \to R^+$

Vyberte podmnožinu hrán $F \subseteq E$ tak, aby:

- (V, F) bol acyklický graf
- pridanie ľubovoľnej hrany $\Rightarrow$ cyklus
- $w(F)$ bola minimálna možná

Ak je G súvislý, tak F tvorí strom
Ináč je F les (strom v každom komponente G)

(Pre jednoduchosť predpokladáme, že všetky váhy sú rôzne.)

Pravidlo inklúzie

Ak pre nejaký vrchol v je $e = \arg\min_{\{u,v\}\in E}\{w(u,v)\}$ potom hrana e patrí do najlacnejšej kostry.

Po skontrahovaní takej hrany môžeme hľadať ďalšiu takým istým spôsobom.

Kruskalov algoritmus (1956)

```
MST-KRUSKAL(E):
  repeat:
    (u,v) := hrana s minimálnou cenou
    T := T + {(u,v)}
    skontrahuj hranu (u,v)
```

Časová zložitosť: $O(m \log n)$
(použi dátovú štruktúru pre UNION/FIND-SET)

Primov algoritmus (1957)

```
MST-PRIM(E):
  s := ľubovoľný počiatočný vrchol
  repeat
    (s,v) := hrana s minimálnou cenou z vrcholu s
    T := T + {(s,v)}
    skontrahuj hranu (s,v)
```

Časová zložitosť: $O(m \log n)$

ak použijeme Fibonacciho heap: $O(n \log n + m)$

Borůvkov algoritmus (1926)

```
MST-BORUVKA(E):
  repeat
    pre každý vrchol v[i] nájdi z neho vychádzajúcu hranu
    e[i] s minimálnou cenou;
    T := T + {e[1],e[2],...}
    skontrahuj hrany e[1],e[2],...
```

Časová zložitosť: $O(m \log n)$

(v každom kroku minimálne polovicu vrcholov odstránime)

Čo sa stane, ak použijeme Borůvkov algoritmus $\log\log n$ krát

- V každom kroku počet vrcholov na polovicu
- Po $\log\log n$ krokoch: $n' = n/2^{\log\log n} = n/\log n$ vrcholov
- Na tento graf použijeme Primov algoritmus: $O(n' \log n' + m)$
  $n' \log n' = \frac{n}{\log n} \cdot \log \frac{n}{\log n} \leq \frac{n}{\log n} \cdot \log n = n$

Celková časová zložitosť: $O(m \log\log n + m + n) = O(n + m \log\log n)$

Pravdepodobnostný algoritmus (Karger et al. 1994)

```
MST-RANDOMIZED(E):
  repeat
1:   urob 2x Borůvkov krok
2:   R := náhodný podgraf (každá hrana s pravdepodobnosťou p)
3:   F := MST-RANDOMIZED(R)
4:   H := ťažké hrany z E vzhľadom ku F
5:   E := E - H
```

Čo sú ťažké hrany?

Hrana (u, v) je **ťažká** vzhľadom na F, akk:

- vrcholy u a v sú spojené v F
- hrana (u, v) má väčšiu váhu ako
  všetky hrany na ceste z u do v v F

Čo sú ťažké hrany?

Hrana (u, v) je **ťažká** vzhľadom na F, akk:

- vrcholy u a v sú spojené v F
- hrana (u, v) má väčšiu váhu ako všetky hrany na ceste z u do v v F

(možno nájsť všetky ťažké hrany v čase $O(m + n)$ - netriviálne)

Pravidlo exklúzie: Ak je v grafe G cyklus C a hrana e má najväčšiu váhu z C, potom e nemôže byť súčasťou minimálnej kostry.

Koľko je ťažkých hrán?

Predstavme si, že by sme mali hrany zotriedené podľa váhy.

```
F:={}; R:={}
pre všetky hrany e od najmenšej po najväčšiu:
  ak e neurobí cyklus v F, OZNAČKUJ hranu
  s pravdepodobnosťou p:
     R:=R U {e}
     ak je e označkovaná, F:=F U {e}
```

R je náhodný podgraf z kroku 2
F je jeho najlacnejšia kostra
Len označkované hrany môžu byť ľahké

$\#$ ľahkých hrán $\leq \#$ značiek
$E[\#$ ľahkých hrán$] \leq E[\#$ značiek$] = \frac{n-1}{p}$
$\Rightarrow$ v každom kroku vyhodíme aspoň $n - \frac{n-1}{p}$ ťažkých hrán!

Nastavme $\boxed{p = 0.5}$

```
MST-RANDOMIZED(E):
  repeat
1:  urob 2x Borůvkov krok
2:  R := náhodný podgraf (každá hrana s pravdepodobnosťou p)
3:  F := MST-RANDOMIZED(R)   # m' hrán, n/4 vrcholov
4:  H := ťažké hrany z E vzhľadom ku F
5:  E := E - H               # m'' hrán, n/4 vrcholov
```

$T(m,n)$: očakávaný čas pre m hrán a n vrcholov

$T(m,n) \leq c(m+n) + T(m', n/4) + T(m'', n/4)$

$T(m,n)$: očakávaný čas pre m hrán a n vrcholov

$T(m,n) \le c(m+n) + T(m', n/4) + T(m'', n/4)$

$E[m'] = m/2 \quad E[m''] \le n/2$

Ukážeme indukciou, že $T(m,n) \le d(m+n)$ pre $d > 4c$

$$\begin{aligned} T(m,n) &\le c(m+n) + T(m', n/4) + T(m'', n/4) \\ &\le c(m+n) + E[d(m' + n/4)] + E[d(m'' + n/4)] \\ &\le c(m+n) + dn/2 + dE[m'] + dE[m''] \\ &\le cm + cn + dn/2 + dm/2 + dn/2 \\ &\le dn + dm/2 + 2cm \le d(m+n) \ \square \end{aligned}$$

$\Rightarrow$ Las Vegas algoritmus s očakávanou zložitosťou $O(m+n)$

Zhrnutie: Hľadanie najlacnejšej kostry v lineárnom čase

- Dve pravidlá: inklúzia a exklúzia
- Deterministické algoritmy založené na inklúzii:
  Kruskalov algoritmus: $O(m \log n)$
  Primov algoritmus: $O(n \log n + m)$
  Borůvkov algoritmus: $O(m \log n)$
- Pravdepodobnostný algoritmus založený
  na použití Borůvkových krokov:
  najprv graf "zahustíme" (pomocou inklúzie),
  potom sa zbavíme veľa hrán "naraz" (pomocou exklúzie)
- Výsledná časová zložitosť $O(m + n)$ (očakávaná)