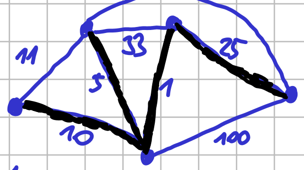
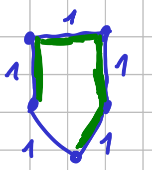


Problém minimální kostry

1926 Borůvka
1930 Jarník

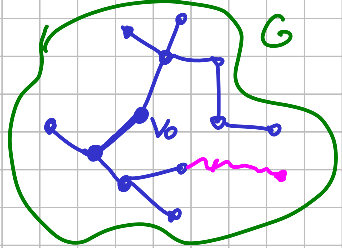
- je dán souvislý neorientovaný graf + váhy hran $w: E \to \mathbb{R}$
  BÚNO unikátní
- chceme najít kostru T t.ž. $w(T)$ je min.
  $w(T) := \sum_{e \in T} w(e)$



5 min. koster

Jarníkův algoritmus

- hladový algoritmus



pěstujeme strom T

- 1 krok:
  vybereme nejlehčí z hran mezi T a $V \setminus T$ a přidáme ji do T.

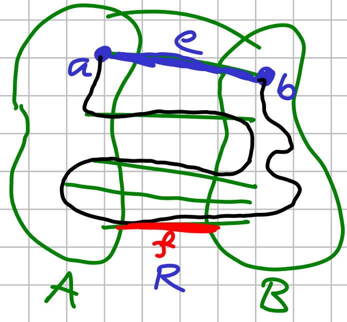
Lemmátko: Jarníkův alg. se zastaví, T je na konci kostra.

po max. n krocích

Df: Množina $R \subseteq E$ je elementární řez $\equiv \exists A \subseteq V,\ B = V \setminus A,\ A,B \neq \emptyset$ t.ž. $R = E(A,B)$

$E(A,B) = \{\{a,b\} \in E \mid a \in A \ \&\ b \in B\}$

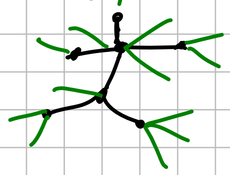
elem. řez



Řezové lemma: Nechť G je graf s unikátními vahami, R elementární řez v G, e nejlehčí hrana v R, T nějaká min. kostra v G. Pak $e \in T$.

Dk: Nechť T je kostra, $e \notin T$.
$R = E(A,B)$ pro A,B
$e = \{a,b\}$ pro $a \in A$, $b \in B$
$\exists C$ cesta v T mezi a,b
$\exists f \in C \cap R$
$T - f$ má 2 komponenty, v jedné je a, v druhé je b
$\check{T} := T - f + e$ je také kostra
$w(\check{T}) = w(T) \underbrace{- w(f) + w(e)}_{<0}$
$w(\check{T}) < w(T)$
$\Rightarrow T$ není minimální.
Q.E.D.

① J.a. najde min. kostru.



hrany mezi T a zbytkem grafu tvoří elem. řez

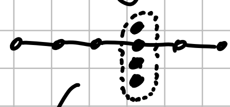
J.a. vybral nejlehčí hranu tohoto řezu

nalezená kostra $\subseteq$ každá min. kostra

② existuje jediná min. kostra

③ min. kostra je jednoznačně určena pořadím hran podle vah.

Odbočka: neunikátní váhy



lze rozhodnout "doporovnat" stejné váhy

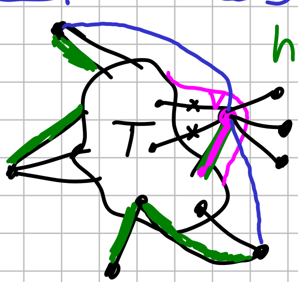
lin. uspořádání

čas. Složitost: $O(n \cdot m)$

\# kroků, čas na 1 krok

nic moc...

Verze podle Dijkstry



pro v sousedy T:
$h(v) :=$ nejlehčí hrana mezi v, T

1 krok:
najdu v s min. $h(v)$
přidám $h(v)$ do T
přepočítám $h(v)$

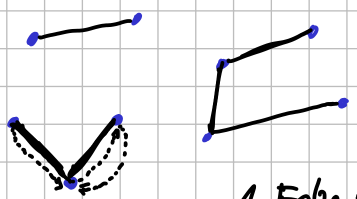
stavy:
- otevřený – soused T
- zavřený – součástí T
- nenalezený – ostatní

Implementace:

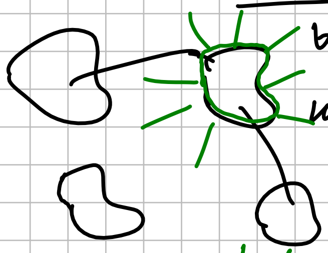
		pole	halda
Insert	$n \times$	1	$\log n$
ExtractMin	$n \times$	n	$\log n$
Decrease	$m \times$	1	$\log n$

Celý alg. $O(n^2 + m) = O(n^2)$ $\qquad O((n+m)\log n)$

Borůvkův alg.



1 Fáze:



∀ stromeček vybere min. hranu a tyto hrany přidáme

zastavíme, máme-li už jediný stromeček

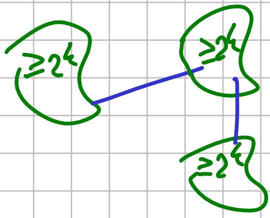
Složitost: $\Theta(m \cdot \log n)$ — m: 1 fáze, $\log n$: # fází

Lemma: # fází $\leq \log n$.

Dk: Na konci k-té fáze mají všechny stromečky aspoň 2^k vrcholů.

Indukcí: $k=0$ … $1 = 2^0$

krok: $(k+1)$-ní fáze



∀ stromeček na konci fáze vznikl složením aspoň 2 pův. stromečků

⇓

vrcholů $\geq 2^k + 2^k = 2^{k+1}$.

Lemma: výstup je min. kostra.

Dk: každá přidaná hrana je nejlehčí v elem. řezu mezi stromečkem a zbytkem grafu.
→ leží v (jediné) min. kostře.

Věta: Borůvkův alg. najde min. kostru v čase $\Theta(m \log n)$.

Kruskalův alg. (1956)

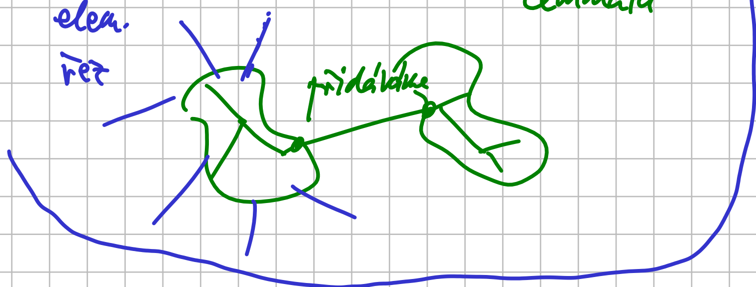
- setřídíme hrany od nejlehčí k nejtěžší $\Theta(m \log n)$
- postupně přidáváme do podgrafu

Vznikne cyklus? ANO → hranu zahodíme; NE → hranu přidáme

$m\times$ Find, $n\times$ Union

Lemma: Alg. najde min. kostru.

Dk: (1) podgraf je vždy les
(2) na konci strom (kostra)
(3) je min … díky řezovému lemmatu



Find(u,v): do u,v do kořenů keříků, kořeny porovnáme

složitost: O(hloubka keříku)

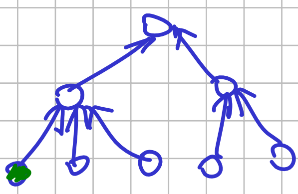
Union-Find (DFU)

- udržujeme komp. souvislosti
- operace: Find(u,v) … jsou u,v v téže komponentě?
  Union(u,v) … přidá hranu uv
- na začátku:
  pole: vrchol → číslo komponenty

  Find: $O(1)$, Union: $O(n)$ → Kruskal v čase $O(m \log n + m + n^2) = O(m \log n + n^2)$

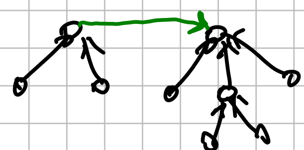
Cvičení: Přečíslujeme menší komponentu
→ 1 vrchol se přečísluje max. $\log n$ krát

Komponenty reprezentujeme keříky



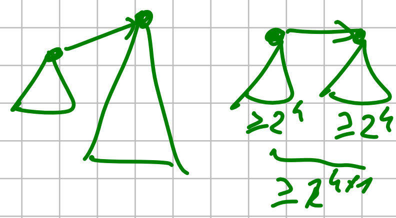
Strom orientovaný ke kořeni, jeho vrcholy tvoří komponentu — vrcholy si pamatují otce → 1 pole

Union(u,v):
najdeme kořeny u', v'
Pokud $u' = v'$: hotovo
Přidáme hranu mezi u', v'
nastavíme otce(u') na v'



složitost O(hloubka keříku)

Vylepšení: Udržuji si v kořenech hloubky keříků, vždy v Unionu připojíme mělčí pod hlubší.



Lemma: Keřík hloubky h má aspoň 2^h vrcholů.

↳ hloubky jsou $\leq \log n$.

Důsl.: Union i Find trvají $O(\log n)$

Kruskal trvá $O(\underbrace{m \log n}_{\text{sort}} + \underbrace{m \log n}_{\text{Findy}} + \underbrace{n \log n}_{\text{Uniony}}) = O(m \log n)$.

Umí se:
- $O(m)$ pro rovinné grafy
- $O(m)$ pro "aspoň trochu husté" ... už pro $m \geq n \cdot \log \log \log n$
- $O(m)$ pro celočíselné váhy
- $O(m)$ průměrně
- $O(m)$ pro seřazené hrany
- obecně: $O(m \cdot \log^* n)$
  $O(m \cdot \alpha(n))$ ← inverzní Ackermannova fce

$\log^*$ inverzní k $\left. 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} \right\} k$

$2 \uparrow k$

$$m \log m \leq m \underbrace{\log n^2}_{\leq 2 \log n} \in \Theta(m \cdot \log n)$$