# Minimální kostra v grafu
# Jarníkův algoritmus s haldou/listem

VSTUP:
Souvislý graf $G = (V, E)$, ohodnocení hran $w(e)$ pro $e \in E$.

VÝSTUP:
Kostra grafu (tj podgraf $(V, E')$, který je stromem a kde $E' \subseteq E$) s co nejmenším součtem ohodnocení hran, tj. s minimálním součtem $\Sigma_{e \in E'} w(e)$.

PSEUDOKÓD Jarníkova algoritmu s haldou:
Ke každému vrcholu si budeme udržovat stav:
nepřipojený, v haldě, připojený.

Do haldy budeme vkládat trojice: $(u, v, w(uv))$, kde $u$ je připojený, $v$ je nepřipojený.

Kostru stavíme z vrcholu $v_0$ (ten lze zvolit libovolně).

1. pro $v \in V$
2.   stav $v$ = nepřipojený
3. stav $v_0$ = připojený
4. vytvoř prázdnou haldu $H$
5. vytvoř prázdnou kostru $K$
6. pro $v_0 s \in E$
7.   vlož $(v_0, s, w(v_0 s))$ do haldy $H$
8.   stav $s$ = v haldě
9. Dokud je halda $H$ neprázdná
10.   odeber z $H$ trojici $(u, v, w(uv))$ s minimální hodnotou $w(uv)$
11.   stav $v$ = připojený
12.   vlož do $K$ trojici $(u, v, w(uv))$
13.   pro $vs \in E$
14.     pokud stav $s$ = v haldě s hodnotou hrany $w(e)$ a zároveň $w(vs) < w(e)$
15.       zmenši v haldě hodnotu na $w(vs)$ – změň trojici na $(v, s, w(vs))$
16.     jinak pokud stav $s$ = nepřipojený
18.       vlož do haldy $(v, s, w(vs))$
19.       stav $s$ = v haldě
20. výstup algoritmu: kostra $K$

Za zmínku stojí podobnost Jarníkova algoritmu s Dijkstrovým algoritmem na hledání nejkratší cesty. Stačí nahradit váhu hrany $w(uv)$ délkou cesty

$l(v_0v)$. V pseudokódu na řádcích 14, 15, 18 nahradíme $w(vs)$ součtem $l(v_0s) = l(v_0v) + w(vs)$ a na řádku 14 hodnotu $w(e)$ dosavadní hodnotou $l(v_0s)$.

ČASOVÁ SLOŽITOST:

Označíme $n = |V|$ počet vrcholů, $m = |E|$ počet hran, stupeň vrcholu $\deg(v)$ značí počet hran, které vrchol $v$ obsahují. Součet stupňů $\sum_{v\in V} \deg(v) = 2m$, protože každá hrana obsahuje dva vrcholy.

Určíme, kolikrát se provedou cykly v algoritmu:

For cyklus na řádcích 1 – 2 se provede $n$-krát.

For cyklus na řádcích 6 – 8 se provede $\deg(v_0)$-krát.

Vrcholy postupně projdou stavy: nepřipojený, v haldě, připojený. Každý vrchol vložíme do haldy jen jednou. While cyklus na řádcích 9 – 19 se tedy provede $n$-krát.

For cyklus na řádcích 13 – 19 se provede $\deg(v)$-krát. Celkem se tedy tento cyklus ve while cyklu provede $\sum_{v\in V, v\neq v_0} \deg(v)$.

Přidáme-li počet cyklů na řádcích 6 – 8, dostaneme, že v algoritmu se provede celkem $\sum_{v\in V} \deg(v) = 2m$-krát přidání prvku do haldy/zmenšení hodnoty v haldě a $n$-krát odebrání minimálního prvku z haldy.

Všechny tři operace – přidání prvku, odebrání minimálního prvku, zmenšení hodnoty prvku mají pro haldu velikosti $n$ stejnou časovou složitost $O(\log(n))$.

Místo haldy lze použít list délky $n$: na pozici s indexem $s$ si uložíme stav vrcholu $s$ a trojici $(v, s, w(vs))$.

Následující tabulka shrnuje operace, jejich časovou složitost pro haldu/list a kolikrát se operace v algoritmu vykoná:

| | | *časová složitost* | |
|---:|:---:|:---:|:---:|
| *operace* | *počet* | *halda* | *list* |
| odebrání minimálního prvku | $n$ | $\log(n)$ | $O(n)$ |
| vložení/zmenšení hodnoty prvku | $2m$ | $\log(n)$ | $O(1)$ |
| za celý algoritmus | | $O((n+m)\log(n))$ | $O(n^2+m)$ |
| po úpravě | | $O(m\log(n))$ | $O(n^2)$ |

Vysvětlení tabulky:

$n$, $2m$ ve sloupci *počet* uvádí, kolikrát se daná operace provede během celého algoritmu. Ve sloupci *časová složitost* jsou na prvních dvou řádcích časové složitosti jedné operace, na dalších řádcích pak vynásobením počtem operací dostaneme časovou složitost za celý algortimus.

TYPY GRAFŮ – POČTY HRAN – VYSVĚTLENÍ ÚPRAVY

Obecně platí $m \leq \binom{n}{2} = \frac{n(n-1)}{2}$, rovnost nastane pro tzv. úplný graf, odtud

dostaneme $O(n^2 + m) = O(n^2)$.
Pro strom platí: $m = n - 1$ a pro souvislý graf platí: $m \geq n - 1$. Odtud dostaneme pro souvislý graf $O((n + m)\log(n)) = O(m \log(n))$

POROVNÁNÍ LISTU S HALDOU

Pro hustý graf, kde je $m$ řádově $n^2$ je výhodnější list.
Pro grafy s počtem hran $m = O(n)$ je výhodnější použití haldy. Takovými grafy jsou například rovinné grafy.