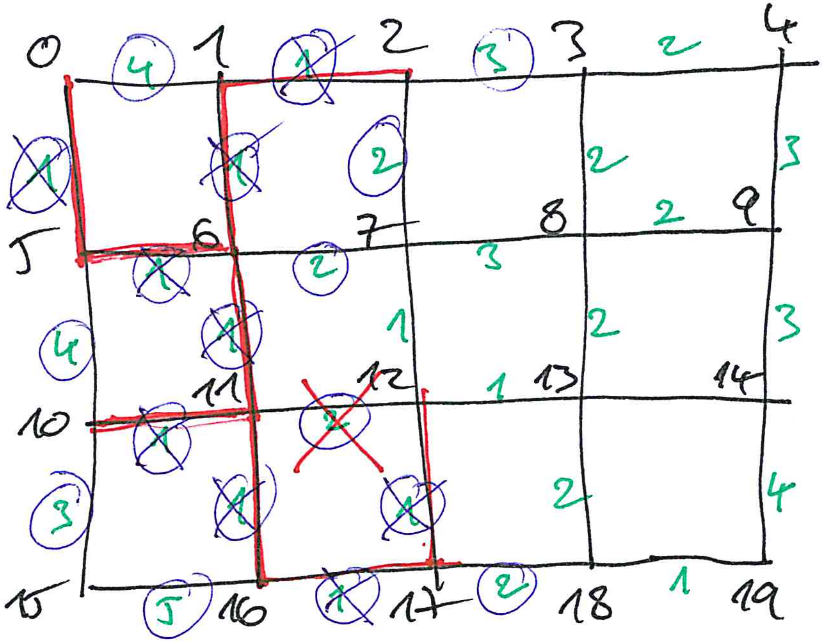

$V = \{0, \dots, 19\}$

Minimální kostra: $K = (V, E_K)$

Každé kostře (K) přiřadíme hodnotu, která je rovna součtu ohodnocení hran z E_K.

Mezi všemi kostrami grafu hledáme tu, která má nejmenší ohodnocení.

Jarníkův algoritmus:

Vybereme vrchol (libovolný) ~~a~~ v a do haldy vložíme ~~vs~~ pro sousedy s vrcholem v hrany vs.

v = 0, v haldě jsou hrany 01(4), 05(1)

(0,1 jsou vrcholy, 01 je hrana 4 je ohodnocení hrany)

z haldy odebereme minimální hranu: ~~01(4)~~ ~~to~~ 05(1) a vložíme do kostry (důležité vysvětlení proč, nechávám na později)

do haldy vložíme hrany, jejichž jeden vrchol je 5 a v haldě zatím nejsou

Znovu Jarníkův algoritmus   H ... halda

Dokud je H neprázdné:

odebereme minimální hranu ~~a pokud~~ ~~přidání do grafu~~ k dosud vybraným hranám nevytvoří kružnici, a přidáme hranu do kostry

~~další hrany do H~~

~~koncové~~

„další hrany" přidáme do haldy

V ... množina vrcholů grafu

$V = P \cup N$ na začátku $P = \{v\}$, $N = V \setminus P$ (z V odstraníme prvek P.)

$E_K = \emptyset$ (prázdná množina)

H .. halda, která obsahuje hrany

1. Dokud je H neprázdná (tj obsahuje alespoň jeden prvek):
2. — — — odeber minimální hranu v_1, v_2 z H   haldové operace
3. — — — pokud $v_1 \in P$ i $v_2 \in P$:
4. — — — — ukonči tuto smyčku (vrať se 3 řádky výš na řádek 1)
5. — — — pokud $v_1 \notin P$:
   ~~přidej do H hrany $v_1\Delta$ pro sousedy Δ vrcholu v_1~~
6. — — — — pro Δ sousedy v_1:
7. — — — — — přidej $v_1\Delta$ do H
8. — — — — $v_1 v_2$ přidej do E_K (hrany kostry)
9. — — —
10. —

9. pokud $v_2 \notin P$:

10. pro sousedy Δ vrcholu v_2:

11. ~~Δ~~ přidej $v_2\Delta$ do H

12. $v_1 v_2$ přidej do E_K