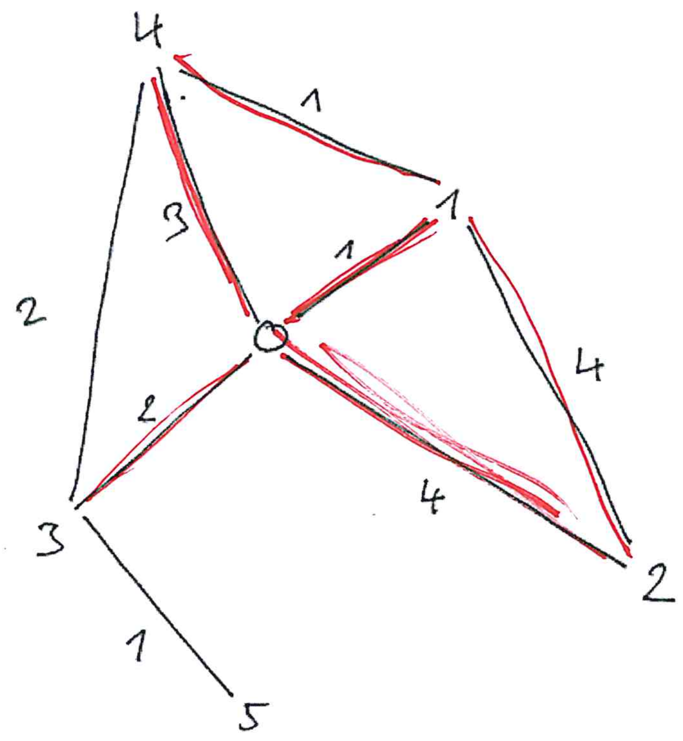


Na vstupu:

Vrcholy: 0 1 2 3 4 5

Hrany a (jejich ohodnocení):

01 (1) 02 (4) 03 (2) 04 (3)

12 (4) 14 (1) 34 (2) 35 (1)

Počáteční vrchol: 0

Hledáme pro každý vrchol

v nejkratší cestu z 0 do v.

Dokud je fronta neprázdná (dokud obsahuje alespoň jeden uzel)

v vyndám z fronty: ~~0~~, ~~1~~, ~~2~~ ~~0~~, ~~1~~ 0, 4, 5 ~~1~~ ~~2~~ 3

Pro sousedy s vrcholu v:

porovnám vzd. (s, 0) s vzd(v, 0) + délka(s, v)

$\boxed{\text{pokud} \ > \text{:}}$

aktualizuji vzdálenost (s, 0)

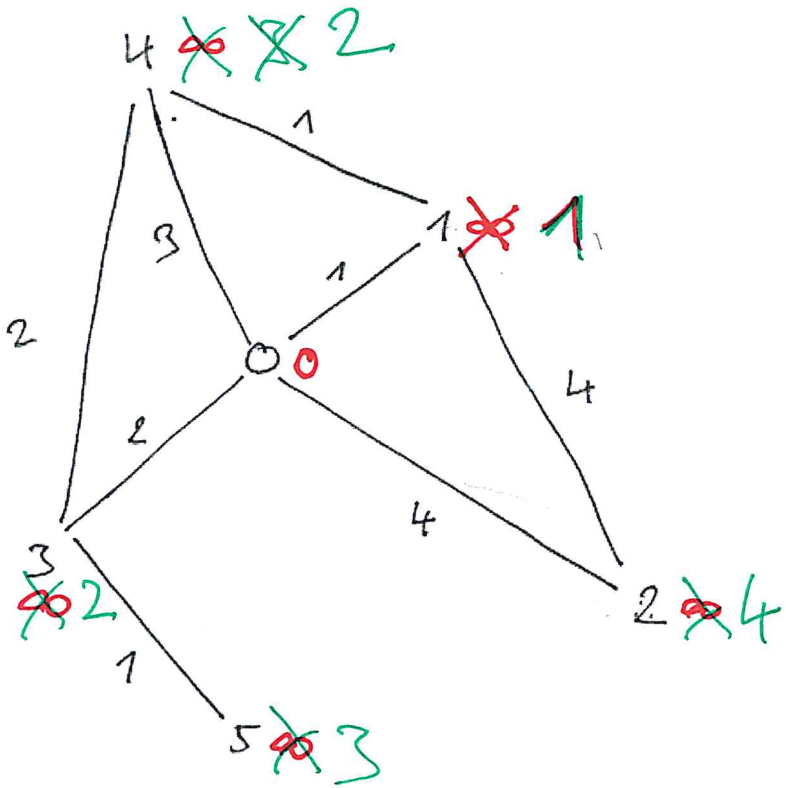
$\boxed{\text{tj. vzd}(s,0) = \text{vzd}(v,0) + \text{délka}(s,v)}$

pokud s není ve frontě:

vložím do fronty

Bellman-Ford algoritmus

1. Dokázat, že algoritmus vrací správné vzdálenosti
2. Spočítat* horní odhad do rychlosti běhu algoritmu *v závislosti na vstupu



Na vstupu:

Vrcholy: 0 1 2 3 4 5

Hrany a (jejich ohodnocení):

01 (1) 02(4) 03(2) 04(3)

12 (4) 14(1) 34 (2) 35 (1)

Počáteční vrchol: 0

Hledáme pro každý vrchol v nejkratší cestu z 0 do v.

Fronta:

~~0~~ ~~1~~, ~~2~~, ~~3~~, 4, 5

① Víme, že do každého vrcholu v existuje nejkratší cesta.

Označme $P(v)$ předchozí vrchol na této cestě, tedy nejkratší cesta z 0 do v: $0, v_1, \dots, P(v), v$

~~Tvrdím:~~

Označme $d(v)$ délku nejkratší cesty z 0 do v

Tvrdím $d(v) = d(P(v)) +$ délka hrany z $P(v)$ do v

~~Tvrzení:~~

Co se stane, když vyndám vrchol v z fronty a zpracuji jej?

Tvrzení:

~~Po zpracování bodow bude mít správné vzdálenosti v grafu, který obsahuje hrany, jejichž jer obstará jeden koncový vrchol je než zpracovaní vrcholy.~~

Po zpracování bude mít minimum přes cesty,

0, v1, ..., v takové, že ~~je než~~

obsahuje zpracovaní hrany

[([[1,1], [2,4], [3,2], [4,3]], 0, 'F', '⊔'), iterace a přidáním vrchol

soused délka hrany

vzdálenost

je ve frontě

$i=0$

([[0,1], [2,4], [4,1]], ∞, '-', '+'),

ní ve frontě

$i=1$