

Jméno: Karel Kubíček

UČO: 408351

Jméno: Henrich Lauko

UČO: 410438

a) Pri generovaní prípustneho toku môžeme modifikovať myšlienku hľadania maximálneho toku, kde zvyšujeme tok kým neexistuje nezaplnená cesta medzi počiatkom a koncom toku na to, že algoritmus zvyšuje ohodnotenie toku kým na každej hrane není zaplnená požiadavka. Algoritmus môže vyzeráť nasledovne:

Algoritmus 1: ACCEPTABLEFLOW(G, p)

```
vstup: graf  $G = (E, V)$ , a funkcia  $p$  definujúca požiadavky na každú hranu  
výstup: nejaký tok  $f$  splňujúci požiadavky  
// inicializácia toku na všetkých hranách  
1 foreach  $(i, j) \in E$  do  
2    $f(i, j) \leftarrow 0$   
3 end foreach  
4 while existuje cesta  $c$  medzi  $s$  a  $t$  v  $G$  na ktorej je hrana kde  $f(i, j) < p(i, j)$  do  
   // navýš hodnoty  $f$  na tejto ceste o najvyššiu požiadavku na tejto ceste  
5    $r_f \leftarrow \max\{p(i, j) - f(i, j) : (i, j) \in c\}$   
6   foreach  $(i, j) \in c$  do  
7      $f(i, j) \leftarrow f(i, j) + r_f$   
8   end foreach  
9 od  
10 return  $f$ 
```

Tento algoritmus vychádza z modifikácie Ford-Fulkersonovho algoritmu. Najprv inicializuje nulový tok, následne hľadá cesty medzi počiatkom s a koncom toku t , také, že na nich existuje aspoň jedna hrana s nesplneným požiadavkom.

Korektnosť Algoritmus iteruje dokým existujú nejaké nesplnené cesty, pri každej iterácii algoritmus splní požiadavok aspoň jednej hrany, keďže počet hran je konečný, tak aj algoritmus je konečný. Aby bol tok prípustný musí splňovať požiadavok na každej hrane, z algoritmu vyplíva, že navýšovaním toku si predchádzajúce vyriešené hrany nemôžeme pokaziť, keďže algoritmus naplní každú hranu tak aj celkový tok bude prípustný, teda algoritmus je korektný.

Zložitosť Inicializácia toku prebehne v lineárnom čase voči počtu hrán - $\mathcal{O}(|E|)$. Z predpokladu, že každá iterácia cyklu v algoritme splní aspoň jednu požiadavku, teda sa vykoná $\mathcal{O}(|E|)$ krát. Operácie v cykle tiež majú časovú zložitosť $\mathcal{O}(|E|)$, keďže hľadanie maximálnej požiadavky a navýšenie cesty prechádza lineárne cez cestu grafu v najhoršom prípade cez celý graf. Z čoho vyplíva výsledná zložitosť $\mathcal{O}(|E|) + \mathcal{O}(|E|) \cdot \mathcal{O}(|E|) = \mathcal{O}(|E|^2)$

b) Algoritmus pre nájdenie toku s minimálnou hodnotou využíva algoritmus z predchádzajúceho príkladu, kde získa nejaký prípustný tok. Následne napočítaný tok minimalizuje (v tomto kroku sa použije algoritmus MAXFLOW)

Jméno: Karel Kubíček

UČO: 408351

Jméno: Henrich Lauko

UČO: 410438

Algoritmus 2: MINIMALFLOW(G, p)

vstup: graf $G = (E, V)$, a funkcia p definujúca požiadavky na každú hranu
výstup: minimálny tok f splňujúci požiadavky

```
1  $f \leftarrow \text{ACCEPTABLEFLOW}(G, p)$ 
2 foreach  $(i, j) \in E$  do
3    $\text{additionalFlow}(i, j) \leftarrow f(i, j) - p(i, j)$  // rozdiel tokov
4 end foreach
5  $r \leftarrow \text{MAXFLOW}(G, \text{additionalFlow})$ 
6 foreach  $(i, j) \in E$  do
7    $f \leftarrow f(i, j) - r(i, j)$  // rozdiel povodneho toku a prebitku
8 end foreach
9 return  $f$ 
```

Korektnosť Nejaký prípustný tok získame pomocou predchádzajúceho algoritmu ACCEPTABLEFLOW, ktorého korektnosť máme dokázanú vyššie. Tento algoritmus vráti tok nie nutne najmenší, preto treba detekovať, ktoré cesty v grafe je možné znižovať (zbaviť sa nabitokného toku). Vytvoríme teda dočasný tok, kde si spočítame nadbitokný tok. Tento nadbitok chceme následne minimalizovať teda určiť koľko toku je možné odobrať aby sme neporušili poptávky. Ak v nadbitoknom toku existuje nejaká cesta z s do t to znamená, že celú túto cestu je možné odobrať z pôvodného toku. Nás zaujíma koľko najviac tohto nadbitku dokážeme odobrať, čo vykoná algoritmus MAXFLOW, ktorý vráti najväčší možný zbitkový tok, ktorý následne odčítame od pôvodného toku. Dostávame korektný minimálny tok. Mimo operácie MAXFLOW o ktorej predpokládame, že je konečná je aj zvyšok algoritmu konečný, čo vyplýva z konečnosti forcyklov a dôkazu ACCEPTABLEFLOW z predchádzajúceho príkladu.

Časová zložitosť Nie je možné určiť vzhľadom k neurčeniu náročnosti MAXFLOW. Zložitosť môže ovplyvniť ešte hľadanie prípustného toku, ktorú sme odvodili v minulom príklade na $\mathcal{O}(|E^2|)$, zvyšné operácie sú lineárne voči počtu hrán.