

ALGORITMICKÉ RIEŠENIA ŤAŽKÝCH PROBLÉMOV

DOMÁCA ÚLOHA 1

Autor: Marián Kravec

Úloha 2 - Bureaucrat

Na začiatok predpoklad: žiadna úloha nemá zápornú dĺžku ($\forall a_i : a_i \geq 0$)

a)

Najskôr si zdefinujme ľahko vypočítateľné dolné ohraničenie optimálneho riešenia. Máme k byrokratov a n úloh dĺžok a_1, \dots, a_n , čiže celkový súčet úloh je $A = \sum_{i=1}^n a_i$. Bez ohľadu na to ako tieto úlohy rozdelíme tak v priemere bude mať byrokrat celkovú dĺžku úloh $\frac{A}{k}$. Keďže ide o priemer vieme, že maximum nemôže byť menšie ako tento priemer, takže pre každé riešenie platí, že musí existovať byrokrat ktorého celková dĺžka úloh bude aspoň priemerná dĺžka, z čoho dostávame dolné ohraničenie $OPT(X) \geq \frac{A}{k}$.

Teraz si ukážeme príklad kedy daný algoritmus nevráti optimálne rozdelenie úloh medzi byrokratov. Majme dvoch byrokratov a 6 úloh dĺžok 7, 5, 3, 3, 3, 3. Optimálne riešenie je v tomto prípade dať jednému byrokratovi úlohy dĺžky 7 a 5 (v súčte 12) a druhému všetky úlohy dĺžky 3 (v súčte 12), keďže celkový súčet úloh je 24 a máme dvoch byrokratov tak vieme riešenia zdola ohraničiť hodnotou 12 ($\frac{24}{2}$), naše riešenie túto hodnotu nadobúda preto o ňom môžeme povedať, že je optimálne. Teraz sa pozrime, čo spraví náš algoritmus:

dĺžka úlohy	byrokrat 1 pred	byrokrat 2 pred	kto dostane úlohu	byrokrat 1 po	byrokrat 2 po
7	0	0	byrokrat 1	7	0
5	7	0	byrokrat 2	7	5
3	7	5	byrokrat 2	7	8
3	7	8	byrokrat 1	10	8
3	10	8	byrokrat 2	10	11
3	10	11	byrokrat 1	13	11

Vidíme, že na konci algoritmu má byrokrat 1 celkovú sumu 13 a byrokrat 2 má sumu 11. Maximum z týchto súm je 13, čo je hodnota nášho riešenia, táto hodnota je vyššia ako hodnota optimálneho riešenia 12, čiže toto riešenie je suboptimálne.

Teraz sa pozrime, ako bude vyzeráť riešenie nášho algoritmu pre vstup v zadaní:

dĺžka úlohy	byrokrat 1 pred	byrokrat 2 pred	byrokrat 3 pred	kto dostane úlohu	byrokrat 1 po	byrokrat 2 po	byrokrat 3 po
14	0	0	0	byrokrat 1	14	0	0
10	14	0	0	byrokrat 2	14	10	0
7	14	10	0	byrokrat 3	14	10	7
4	14	10	7	byrokrat 3	14	10	11
2	14	10	11	byrokrat 2	14	12	11
2	14	12	11	byrokrat 3	14	12	13
1	14	12	13	byrokrat 2	14	13	13

Vidíme, že náš algoritmus dá riešenie 14 čo je optimálne riešenie. To že je to optimálne môžeme ukázať znovu použitím priemeru dĺžky úloh na byrokrata. V tomto prípade je súčet všetkých úloh 40, keďže máme troch byrokratov, priemer je 13.3 avšak keďže sú v tomto prípade všetky úlohy celočíselnej dĺžky aj optimum musí byť celočíselné preto optimum nemôže byť menšie ako 14, čo je výsledok nášho algoritmu.

b)

Chceme dokázať, že pre všetky a_i platí $a_i > \frac{t^*}{3}$ kde t^* je hodnota optimálneho riešenia, tak algoritmus vráti optimálne riešenie.

Najskôr si dokážme takéto tvrdenie ak platí $\forall a_i : a_i > \frac{t^*}{3}$ tak žiaden byrokrat nedostane viac ako 2 úlohy. (nulové úlohy ignorujeme)

Toto dokážeme sporom, čiže máme tvrdenie, že ak $\forall a_i : a_i > \frac{t^*}{3}$ tak môže existovať byrokrat ktorý dostane aspoň tri úlohy. Uvažujme, že máme byrokrata b (b označuje celkovú dĺžku práce byrokrata) ktorý má aspoň tri úlohy, medzi je úlohami sú úlohy i, j a k , čiže dĺžka je práce je aspoň súčet dĺžok týchto úloh $b \geq a_i + a_j + a_k$ avšak o každej z týchto úloh vieme povedať $a_i > \frac{t^*}{3}$ preto platí $b \geq a_i + a_j + a_k > \frac{t^*}{3} + \frac{t^*}{3} + \frac{t^*}{3} = t^*$, čiže $b > t^*$ čo je v spore z predpokladom, že t^* je optimálne riešenie, čiže neexistuje byrokrat ktorý má súčet dĺžok úloh väčší ako je táto hodnota. Preto platí, že každý byrokrat má nanajvýš 2 úlohy.

Teraz sa pozrime, čo spraví náš algoritmus. Bez ujmy na všeobecnosti si našich byrokratov nejak usporiadajme. Náš algoritmus najskôr postupne zľava doprava každému byrokratovi jednu najväčšiu zatiaľ nepridanú úlohu kým nemá každý práve jednu úlohu alebo nie sú pridané všetky úlohy (vieme to zaručiť lebo kým nepridá prvých k úloh existuje byrokrat s celkovou sumou 0 čo je najnižšia možná). Ďalej uvažujme o prípade, že každý byrokrat má pridelenú jednu úlohu a ešte existujú nepridelené úlohy, keďže sme pridávali úlohy zľava doprava od najväčšej sme teraz v situácii, že sú byrokрати zoradení podľa celkových dĺžok ich úloh. Preto keď algoritmus začne prideliť zvyšné úlohy, bude ich prideliť z sprava doľava po jednej úlohy (vieme, že iba po jednej lebo po pridelení ďalšej úlohy bude mať byrokrat 2 úlohy čo sme dokázali, že za daných podmienok je maximum).

Teraz potrebujeme ukázať, že toto riešenie je optimálne. Môžu nastať dve možnosti:

- maximálnu celkovú dĺžku úloh má byrokrat s jednou úlohou - tento prípad je jednoduchý keďže v takomto prípade, vieme, že optimum nemôže byť menšie ako je najdlhšia úloha a preto toto riešenie musí byť optimálne
- maximálnu celkovú dĺžku úloh má byrokrat s dvomi úlohami - na začiatok bez ujmy na všeobecnosti pridajme každému byrokratovi s iba jednou úlohou, druhú úlohu s nulovou dĺžkou, tým neovplyvníme naše ani optimálne riešenie a môžeme uvažovať, že každý byrokrat má 2 úlohy, zavedme si teraz takéto označenie úloh $a_{i,j}$ kde i je poradie byrokrata (na začiatku sme ich usporiadali) a j je poradie v ktorom bola úloha byrokratovi pridelená. Teraz vieme tieto úlohy takto usporiadať $a_{1,1} \geq a_{2,1} \geq a_{k,1} \geq a_{k,2} \geq a_{k-1,2} \geq a_{1,2}$ keďže prvé úlohy sú väčšie ako druhé a čím skôr je byrokrat v rade tým skôr svoju prvú úlohu dostal (keďže rozdávané úlohy od najväčšej) a podobne potom druhé úlohy sú menšie ako prvé a boli rozdávané byrokratom v obrátenom poradí (lebo byrokрати na konci mali najmenšie prvé úlohy, čiže najmenšiu celkovú dĺžku úloh). Teraz uvažujme, že maximum má byrokrat i , čiže $NASE(X) = a_{i,1} + a_{i,2}$