

Domáca úloha č. 1

2-AIN-205, Leto 2015

Termín: 9.3.2015, 22:00, M-163 (pod dvere)

Skôr ako sa pustíte do riešenia domácej úlohy, oboznámte sa so všeobecnými pokynmi, ktoré sú priložené na konci tohto dokumentu. Riešenia, ktoré odovzdáte, musia byť vaše vlastné. Neopisujte a nesnažte sa nájsť riešenia v literatúre alebo na internete!

1. [20 bodov] **Nepodarené mince.** Jednocentové a dvojcentové mince sú drahšie na výrobu, ako je ich hodnota. Preto vrabce na strome čvirikajú, že v blízkej budúcnosti Európska komisia rozhodne o ich zrušení. Keďže v pokladniciach ostane voľná priehradka, namiesto nich sa uvažuje o zavedení 25 centovej mince. Výsledkom teda bude, že budeme mať mince o hodnote 5, 10, 20, 25, 50, 100 a 200 centov a všetky sumy sa budú zaokrúhľovať na najbližších 5 centov. Predavačky v obchodoch sú dnes zvyknuté vydávať jednoduchým greedy algoritmom, to znamená že opakovane vyplácajú vždy najvyššiu mincu, ktorá ešte nepresiahne sumu, ktorú treba vyplatiť. Tento spôsob doteraz viedol k tomu, že zákazník dostal svoj výdavok rozmenený na najmenší možný počet mincí.

- Ukážte, že po zavedení novej 25 centovej mince už greedy metóda nepovedie k vyplateniu najmenším možným počtom mincí.
- Dokážte, že to nie je až také zlé: Vždy vyplatíme nanaajvýš o jednu mincu viac, ako je optimálny počet.

2. [20 bodov] **Steinerove stromy.** Daný je súvislý neorientovaný graf G s ohodnotenými hranami, ktorých cena je nezáporné prirodzené číslo. Časť vrcholov grafu je označených ako "nevyhnutné" (ich množinu označme R). Steinerov strom T je podstrom grafu G , ktorý obsahuje všetky vrcholy z R a potenciálne aj nejaké iné.

Budeme sa zaoberať problémom hľadania najmenšieho Steinerovho stromu. Tento problém je NP-ťažký. V prípade metrického grafu, existuje ale veľmi jednoduchý 2-approximačný algoritmus: zoberieme minimálnu kostru na podgrafe, ktorý vznikne vymazaním všetkých vrcholov okrem tých z R .

Ak chceme ten istý problém riešiť v súvislom grafe, ktorý nie je metrický, môžeme to urobiť tak, že transformujeme pôvodný graf na úplný graf G' , pričom medzi každými dvoma vrcholmi u a v bude hrana, ktorej dĺžka bude dĺžkou najkratšej cesty z u do v v pôvodnom grafe.

- Ukážte, že graf G' bude metrický.
- Ukážte, že optimálny Steinerov strom v G' má rovnakú cenu ako optimálny Steinerov strom v pôvodnom grafe G .
- Ukážte, že výsledný algoritmus, ktorý pozostáva z transformácie problému na metrický a vyriešenie pomocou 2-approximačného algoritmu pre metrický problém, je 2-approximačný algoritmus pre všeobecný problém.

3. [20 bodov] **Programátorská úloha** (viď všeobecné pokyny). Na prednáške ste sa dozvedeli, že cena najlacnejšej kostry nám dáva dolné ohraničenie na cenu najlacnejšieho riešenia problému obchodného cestujúceho. V tejto úlohe sa budeme zaoberať tým, ako tento odhad zlepšiť.

Daný je kompletný neorientovaný graf G . Označme cenu hrany spájajúcej vrcholy u a v ako $c(u, v)$. Predstavme si, že okrem tejto funkcie máme funkciu $p(u)$, ktorá každému vrcholu priradí hodnotu.

Predstavte si teraz nový graf G' , pričom ceny jeho hrán budú $d(u, v) = c(u, v) + p(u) + p(v)$. Vašou úlohou je nájsť také ceny $p(u)$ pre jednotlivé vrcholy grafu, aby sa

- cena najlacnejšieho riešenia problému obchodného cestujúceho v grafe G bola rovnaká, ako cena najlacnejšieho riešenia problému obchodného cestujúceho v grafe G'
- a cena najlacnejšej kostry v G' bola ostro väčšia ako cena najlacnejšej kostry v G .

Hint: O koľko sa zmení cena riešenia problému obchodného cestujúceho, ak nastavíme pre všetky vrcholy u cenu $p(u) = 2$? Potrebujeme vedieť, aké to riešenie je, aby sme zodpovedali na predchádzajúcu otázku?

Zadanie nevyžaduje, aby cena najlacnejšej kostry v grafe G' bola najväčšia možná. Stačí nám, aby bola aspoň o niečo vyššia ako pôvodná.

Formát vstupu: Na prvom riadku je číslo n – počet vrcholov grafu. Nasleduje m riadkov, v každom z nich m čísel (vzdialenosti vrcholov); j -te číslo v i -tom riadku predstavuje $c(i, j)$. Platí $c(i, j) = c(j, i)$ a $c(i, i) = 0$.

Formát výstupu: Vypíšte jeden riadok s hodnotami $p(\cdot)$ oddelenými medzerou.

Obmedzenia a bodovanie: Na zisk plného počtu bodov je nutné, aby váš program dal správnu odpoveď pre vstupy, kde n je najviac 100. Môžete predpokladať, že vstupy sú zadané tak, že úloha má vždy riešenie.

Príklad vstupu:

```
5
0 7 8 10 1
7 0 8 8 5
8 8 0 10 10
10 8 10 0 7
1 5 10 7 0
```

Príklad výstupu:

```
1 -1 -1 -1 2
```

Všeobecné pokyny

Písomné úlohy. Písomné úlohy odovzdávajte *na papieri* (či už vytlačené alebo písané rukou) pod dvere kancelárie M-163 v stanovenom termíne. **Každý príklad odovzdajte na osobitnom liste papiera**, každý príklad bude opravovať iný človek. Na neskoro odovzdané riešenia sa nebude prihliadať. Nezabudnite na každý list jasne napísať svoje plné meno a priezvisko, v prípade že riešenie jedného príkladu je na viac listov, zopnite ich pevne spinkovacím strojčekom.

Píšte riešenia takým spôsobom, aby obsahovali všetku potrebnú informáciu na pochopenie vášho riešenia, ale súčasne aby boli stručné a ľahko pochopiteľné. Všetky tvrdenia je potrebné zdôvodniť (a to aj v prípade, že to nie je explicitne napísané v zadaní).

Ak sa v zadaní požaduje vyriešenie algoritmickej úlohy, odovzdajte najlepší algoritmus, aký viete navrhnúť. Základným kritériom na hodnotenie bude *správnosť algoritmu*, druhým kritériom bude jeho *časová, prípadne pamäťová zložitosť*. Správny ale pomalý algoritmus dostane podstatne viac bodov ako algoritmus, ktorý je síce rýchly, ale nedá správnu odpoveď na každý vstup. Neefektívne algoritmy spĺňajúce podmienky zadania dostanú cca 50% bodov. Súčasťou vášho riešenia musia byť nasledujúce časti:

- Najprv popíšte hlavnú myšlienku algoritmu.
- Vyjadrite algoritmus formou pseudokódu.
- Ak to nie je zrejmé na prvý pohľad, ukážte že váš algoritmus je správny.
- Nezabudnite na analýzu zložitosti algoritmu.

Programátorské úlohy. Pri programátorských úlohách je vašou úlohou odovzdať len funkčný program, nie je vyžadované písomné riešenie. Riešenie odovzdávate cez webové rozhranie `foja.dcs.fmph.uniba.sk/eval`, kde bude okamžite otestované na niekoľkých vstupoch a dozviete sa koľko bodov získalo (body získate, keď všetky vstupy z danej sady vyriešite správne v časovom limite). Riešenie môžete odovzdávať aj viackrát, hodnotí sa posledné riešenie odovzdané v stanovenom termíne. Navyše si dajte pozor, či v systéme máte správne vyplnené meno a priezvisko (sekcia Mój účet). Podrobnosti o tom, ako má váš program vyzeráť (vrátane povolených programovacích jazykov), nájdete v sekcii Návod.