

Domáca úloha č. 2

2-AIN-205, Leto 2014

Termín: 24.3.2014, 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!

- [20 bodov] Delenie koristi.** Dve sovy nalovili n myší o veľkostiach $s_1 \leq s_2 \leq \dots \leq s_n$ (pre jednoduchosť predpokladajme, že všetky veľkosti sú kladné celé čísla) a teraz sedia a delia si korisť na dve kopy A a B , aby ju mohli zaniest mladým. Vždy zoberú najmenšiu myš a hodia ju na menšiu kopy. Cieľom je rozdeliť korisť čo najspravodlivejšie, čo sa dá vyjadriť napríklad tak, že sa snažia, aby $\max\{\sum_{i \in A} s_i, \sum_{i \in B} s_i\}$ bolo najmenšie možné.
 - Ukážte, že sovy vymysleli k -aproximačný algoritmus s konštantným aproximačným faktorom. Aký je tento aproximačný faktor?
 - Ukážte, že váš odhad aproximačného faktoru je tesný. To znamená, že nájdete príklad, kde podiel medzi riešením algoritmu a optimálnym riešením dosiahne k .

- [20 bodov] Efektívne pokrývanie množinami.** Pripomeňme si, že pri pokrývaní množinami (Set Cover) máme daných k množín S_1, \dots, S_k , kde $S_i \subseteq \{1, \dots, n\}$ a našou úlohou je nájsť najmenší možný počet množín, ktorých zjednotením sú všetky prvky $\{1, \dots, n\}$. $O(\log n)$ -aproximačný greedy algoritmus z prednášky funguje tak, že v každom kroku vyberie množinu, ktorá pokrýva najväčší počet ešte nepokrytých prvkov.

Navrhňte dátové štruktúry a napíšte podrobný pseudokód, ktorý implementuje tento greedy algoritmus tak, aby výsledná časová zložitosť bola čo najlepšia. Časovú zložitosť odhadnite vzhľadom ku parametrom k , n a u , kde $u = \sum_i |S_i|$.

- [20 bodov] Programátorská úloha** (viď všeobecné pokyny). Na vstupe je zadaný kompletný orientovaný graf. Vašou úlohou je z neho odstrániť čo najmenej vrcholov tak, aby výsledný graf bol acyklický. Vaše riešenie môže použiť najviac trikrát viac vrcholov ako optimálne riešenie.

Formát vstupu: Na prvom riadku je počet vrcholov grafu n . Následuje $n(n-1)/2$ riadkov, na každom z nich sú dve čísla a, b , ktoré hovoria, že z vrchola a ide hrana do vrchola b .

Formát výstupu: Na prvý riadok vypíšte počet odstránených vrcholov k . Na druhý riadok vypíšte k medzerou oddelených čísel, čo sú čísla odstránených vrcholov.

Obmedzenia a bodovanie: Na zisk plného počtu bodov je nutné, aby váš program dal správnu odpoveď pre vstupy, kde n je najviac 200.

Príklad vstupu:

4
1 2
2 3
3 1
1 4
2 4
3 4

Príklad výstupu:

2
1 2

Pozn.: Stačí odstrániť aj jeden vrchol (napr. 2).

Hint: Určite treba odstrániť všetky cykly dĺžky 3. Čo s ostatnými cyklami? Spomeňte si na 2-aproximačný algoritmus pre vrcholové pokrytie.

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 strojkom.

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.