Domáca úloha č. 4

2-AIN-205, Leto 2016

Termín: 30.5.2016, 23:59, M-163 (pod dvere)
alebo pred termínom skúšky (čokoľvek je skôr)

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*] **Kariérny postup.** V nemenovanej firme majú n zamestnancov a množstvo kariérnych stupňov očíslovaných $1, 2, 3, \ldots$. Zamestnanec môže byť z kariérneho stupňa i povýšený do kariérneho stupňa $i-1$, z kariérneho stupňa 1 zamestnanec postupuje do správnej rady, kde sa môže tváriť múdro a dostávať až do smrti rozprávkový plat bez toho, aby čo i len pohol prstom. Nech a_i označuje počet zamestnancov na kariérnom stupni i.

Kariérny postup funguje nasledovne: odborári každý rok navrhnú množinu S zamestnancov určených na kariérny postup. Správna rada si preštuduje návrh a vyberie jednu z dvoch alternatív: zamestnancov z množiny S povýši a všetkých ostatných zamestnancov prepustí alebo zamestnancov z množiny S prepustí a všetkých ostatných zamestnancov povýši. Cieľom odborárov je dostať aspoň jedného zamestnanca do správnej rady. Naopak, hlavnou úlohou správnej rady je nepripustiť do svojich radov ďalších členov.

Ide o hru, kde obaja hráči majú úplnú informáciu (počty zamestnancov na jednotlivých kariérnych stupňoch). Keďže hra nepozná remízu, musí mať pre každý počiatočný stav $a_1, a_2, \ldots$ jeden z hráčov víťaznú stratégiu (t.j. vyhrá bez ohľadu na to, čo robí druhý hráč).

- **a)** Predpokladajme, že pre niektoré k platí $a_k \geq 2^k$. Ukážte, že v takom prípade majú odborári víťaznú stratégiu.
- **b)** Predpokladajme, že správna rada rozhoduje náhodne, t.j. s pravdepodobnosťou 1/2 množinu S povýši a s pravdepodobnosťou 1/2 množinu S prepustí. Aká je pravdepodobnosť, že sa zamestnanec, ktorý začne na kariérnom stupni k, napokon dostane do správnej rady?
- **c)** Ukážte, že stredná hodnota počtu zamestnancov, ktorí sa dostanú do správnej rady je v takomto prípade $\sum_k a_k/2^k$.
- **d)** **(bonus)** Argumentujte, že ak na začiatku $\sum_k a_k/2^k < 1$, tak správna rada má víťaznú stratégiu. Hint: Aká by bola stredná hodnota z časti c), ak by mali vyhrávajúcu stratégiu odborári?

2. [*20 bodov*] **Najdlhšia jednoduchá cesta.** Uvažujme nasledujúci problém: Daný je neorientovaný ohodnotený graf $G = (V, E)$ a dva vrcholy u a v. Váhy hrán môžu byť kladné aj záporné. Úlohou je nájsť najdlhšiu jednoduchú cestu z vrcholu u do vrcholu v. (Jednoduchá cesta je taká cesta, v ktorej sa žiaden vrchol neopakuje.)

- a) Sformulujte tento problém ako rozhodovací problém.
- b) Ukážte, že váš rozhodovací problém je v NP.
- c) Ukážte, že váš rozhodovací problém je NP-ťažký.

3. [*20 bodov*] **Vrcholové pokrytie, nezávislá množina a klika.** Nech $G = (V, E)$ je neorientovaný graf. Spomeňte si, že *vrcholové pokrytie* grafu je taká množina vrcholov, v ktorej má každá hrana aspoň jeden koniec, že *nezávislá množina* je taká množina vrcholov, že žiadne dva z nich nie sú spojené hranou a naopak *klika* v grafe je taká podmnožina vrcholov, z ktorých každé dva sú spojené hranou.

Množina vrholov $U \subseteq V$ je nezávislá množina práve vtedy, keď $V - U$ je vrcholové pokrytie grafu. U je tiež nezávislá množina v grafe G práve vtedy, keď U je klika v doplnku grafu G.

Hľadanie najmenšieho vrcholového pokrytia, najväčšej nezávislej množiny, či najväčšej kliky v grafe sú všetko NP-ťažké problémy a vyššieuvedené tvrdenia nám umožňujú vytvoriť jednoduché polynomiálne redukcie medzi týmito problémami. Zachovávajú tieto redukcie aproximačný faktor?

a) Ak by sme mali polynomiálny algoritmus s konštantným aproximačným faktorom pre hľadanie najväčšej nezávislej množiny, dáva nám vyššieuvedená redukcia polynomiálny algoritmus s konštantným aproximačným faktorom pre hľadanie najväčšej kliky?

b) Ak by sme mali polynomiálny algoritmus s konštantným aproximačným faktorom pre hľadanie najväčšej nezávislej množiny, dáva nám vyššieuvedená redukcia polynomiálny algoritmus s konštantným aproximačným faktorom pre hľadanie najmenšieho vrcholového pokrytia?

4. [*20 bodov*] **Programátorská úloha** (viď všeobecné pokyny). Na vstupe máte n reťazcov zložených zo znakov $0, 1$. Vašou úlohou je povedať, či majú spoločnú vybranú podpostupnosť dlhú aspoň k.

Formát vstupu: V prvom riadku vstupu je počet reťazcov n a číslo k. V ďalších n riadkoch sa nachádza n reťazcov.

Formát výstupu: Pokiaľ dané reťazce majú spoločnú vybranú podpostupnosť, vypíšte YES, ináč vypíšte NO.

Obmedzenia a bodovanie: Na zisk plného počtu bodov je potrebné, aby váš program dal korektný výsledok pre vstupy, kde $n \leq 10, k \leq 18$ a dĺžka reťazcov je najviac 1000.

Príklad vstupu:

```
3 3
01010
00000
10100
```

Príklad výstupu:

```
YES
```

Poznámka: Spoločná vybraná podpostupnosť je 000.

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 Môj účet). Podrobnosti o tom, ako má váš program vyzerať (vrátane povolených programovacích jazykov), nájdete v sekcii Návod.