

# Domáca úloha č. 4

2-AIN-205, Leto 2014

Termín: 28.4.2014, 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!

1. [20 bodov] **Náhodné permutácie.** Budeme generovať náhodné permutácie nasledujúcim spôsobom. Začneme s poľom  $S$ , ktoré nainicializujeme tak, že  $S[i] = i$ . Potom pre všetky  $i$  od 2 po  $n$  vymeníme prvok  $S[i]$  s prvkom  $S[\text{rand}(1, i)]$ , pričom  $\text{rand}(a, b)$  vráti náhodné celé číslo medzi  $a$  a  $b$ .

a) Ukážte, že tento algoritmus môže vygenerovať ľubovoľnú z  $n!$  permutácií a že každú z týchto permutácií vygeneruje s rovnakou pravdepodobnosťou.

b) Ak predpokladáme, že náhodné číslo  $\text{rand}(a, b)$  vieme vygenerovať v čase  $O(1)$ , potom tento algoritmus zjavne funguje v čase  $O(n)$ .

Ako by ste implementovali  $\text{rand}(a, b)$ , ak by ste mohli naraz vygenerovať len jeden náhodný bit? Analyzujte najhoršiu a očakávanú časovú zložitosť.

2. [20 bodov] **Náhodná prechádzka po strome.** Majme binárny strom s  $n$  vrcholmi, kde každý vrchol má práve dve deti. Začíname v jeho koreni a v každom kroku si zvolíme náhodne ľavé alebo pravé dieťa a posunieme sa do neho. Toto opakujeme, až kým sa neocitneme v liste stromu. Dokážte, že očakávaný počet krokov tohto algoritmu je zhora ohraničený hodnotou  $\log_2(n + 1)$ .

**Hint:** Pozor! Strom nemusí byť vyvážený. Skúste to indukciou vzhľadom na počet vrcholov stromu.

3. [20 bodov] **Programátorská úloha** (viď všeobecné pokyny). Na vstupe máte  $n$  neprázdnych množín (očíslovaných  $1, 2, \dots, n$ ), ktoré obsahujú celé čísla. Podobnosť medzi množinami  $A, B$  definujeme ako  $\text{SIM}(A, B) = |A \cap B| / |A \cup B|$ .

Vašou úlohou je napísať program, ktorý spracúva nasledovné požiadavky:

- Pridaj do množiny s číslom  $a$  prvok  $b$  (môžete predpokladať, že toto číslo sa tam predtým nevyskytvalo).
- Nájdí ku množine s číslom  $a$  množinu, ktorá má s ňou podobnosť aspon 0.85 (a nie je to množina  $a$ ). V prípade, že taká neexistuje podajte o tom správu.

Navyše vášmu programu dovolíme sa občas zmýliť. Presnejšie: V 10% prípadov, keď podobná množina existuje, môže vyhlásiť, že neexistuje.

**Formát vstupu:** V prvom riadku vstupu je počet množín  $n$  a počet požiadaviek  $q$ . V ďalších  $n$  riadkoch sa nachádza popis začiatočného obsahu množín. Každý riadok začína číslom  $k_i$  (počet prvkov v množine  $i$ ) a nasleduje  $k_i$  prvkov, ktoré sú v množine  $i$ .

V ďalších  $q$  riadkoch sú jednotlivé požiadavky. Každá požiadavka je popísaná dvoma číslami  $a, b$ . Pokiaľ  $a = -1$ , tak hľadáme podobnú množinu k množine  $b$ , inak chceme pridať prvok  $b$  do množiny  $a$ .

Môžete predpokladať, že prvky množín sa zmestia do bežnej 32-bitovej premennej.

**Formát výstupu:** Pre každú požiadavku na nájdenie podobnej množiny vypíšte buď číslo jej podobnej množiny, alebo  $-1$  ak ste takú množinu nenašli.

**Obmedzenia a bodovanie:** Na získanie plného počtu bodov je potrebné, aby váš program dal korektný výsledok pre vstupy, kde  $n \leq 30000, q \leq 120000$ .

### Príklad vstupu:

```
3 4
10 1 2 3 4 5 6 7 8 9 10
9 1 2 3 4 5 6 7 8 9
5 1 2 3 4 5
-1 1
2 11
-1 1
-1 3
```

### Príklad výstupu:

```
2
-1
-1
```

## 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 zřejmé 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.