Domáca úloha č. 2

1-AIN-105, Zima 2022

Termín: 28.10.2022, 22:00

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*] **Dvojsmerne triedené pole.** Dvojsmerne triedené pole (2STP) je pole $m \times n$ čísel, kde prvky v každom riadku a v každom stĺpci sú utriedené od najväčšieho po najmenšie. Aby sme vedeli v 2TSP ukladať aj menej ako mn čísel, tak niektoré prvky môžu byť $-\infty$. Takéto prvky však tiež musia spĺňať podmienky na utriedenie riadkov a stĺpcov, tzn. že musi byť bud na konci riadku alebo na spodku stĺpca. Príklad, ako môže vyzerať 2STP:

$$\begin{pmatrix} 15 & 12 & 10 & 9 \\ 14 & 11 & 8 & 7 \\ 13 & 6 & 2 & 1 \\ 5 & 3 & -\infty & -\infty \\ 4 & -\infty & -\infty & -\infty \end{pmatrix}$$

a) Ukážte, ako môžete použiť 2STP ako implementáciu abstraktného dátového typu prioritná fronta s operáciami `insert` a `extractMax`. Obe operácie (insert, extractMax) by ste mali vedieť implementovať v čase $O(m+n)$. Môžete predpokladať, že na začiatku je celá matica nainicializovaná hodnotami $-\infty$ a že v každom okamihu je v matici aspoň jeden prvok $-\infty$ (t.j. nedôjde k preplneniu dátovej štruktúry)

b) Pripomeňme si, že triediť pomocou prioritnej formy môžeme tak, že najprv vložíme do dátovej štruktúry všetkých N prvkov pomocou operácie `insert` a potom vyťahujeme zo štruktúry utriedené prvky pomocou operácie `extractMax`. Ak by ste použili implementáciu z časti a) na takéto triedenie, akú časovú zložitosť triedenia by ste vedeli dosiahnuť?

Vaša odpoveď by mala závisieť len od čísla N, preto súčasťou vašej odpovede by malo byť, ako zvoliť rozmery m a n matice pre 2STP.

c) Implementácia prioritnej fronty pomocou 2STP má výhodu, že okrem operácií `insert` a `extractMax` viete implementovať aj nejaké ďalšie operácie v dobrej časovej zložitosti. Ukážte, ako môžete nájsť v čase $O(m+n)$ najmenší prvok uložený v prioritnej fronte (t.j. najmenší prvok, ktorý je uložený v 2STP matici, ktorý nie je $-\infty$).

2. [*20 bodov*] **Takmer utriedené polia.** Prof. Premúdrelý prednášal o dolnom odhade $\Omega(n \log n)$ pre triedenia založené na porovnávaní prvkov a na konci prednášky dostal geniálny nápad: možno by sa dala dosiahnuť aj lepšia časová zložitosť, ak by sme na výstupe požadovali iba "takmer utriedené" pole.

Takmer utriedené pole je pole, v ktorom sa každý prvok nachádza nanajvýš vo vzdialenosti k od svojej pozície v utriedenom poli pre nejakú konštantu $k > 0$. *Takmer triediaci algoritmus* má na vstupe pole n čísel, ktoré preusporiada tak, že na konci bude pole takmer utriedené; takmer triediaci algoritmus môže len porovnávať prvky poľa navzájom a môže ich vymieňať.

Ukážte, že prof. Premúdrelý nemá pravdu: aj ak vyžadujeme iba takmer utriedené pole na výstupe, ľubovoľný takmer triediaci algoritmus potrebuje v najhoršom prípade aspoň $\Omega(n \log n)$ porovnaní.

3. [*20 bodov*] **O e-shope (programátorská úloha).** Máme produkty v e-shope. Každý produkt má svoju cenu - nejaké prirodzené číslo. Vyrobte program, ktorý bude vedieť filtrovať produkty na základe ceny tak, že dostane hodnoty a a b a zistí počet produktov, ktorých cena je v rozmedzí a až b, **vrátane** a a b (teda cena patrí do $\langle a, b \rangle$).

Vstup. Na prvom riadku vstupu je číslo N - počet produktov, ($1 \leq N \leq 2\,000\,000$). Na druhom riadku je N medzerou oddelených čísel c_i - ceny produktov. Platí, že $1 \leq c_i \leq 10\,000$.

Na treťom riadku je číslo t ($1 \leq t \leq 100\,000$), označujúce počet požiadaviek na filtrovanie. Nasleduje t riadkov, kde každý obsahuje 2 medzerou oddelené čísla, a_i - začiatok intervalu a b_i - koniec intervalu cien. Platí, že $a_i \leq b_i$.

Výstup. Vypíšte t riadkov. i-ty riadok obsahuje jedno číslo - odpoveď na i-tu požiadavku, teda počet produktov, ktorých cena patrí do intervalu $\langle a_i, b_i \rangle$.

Príklady:

Vstup:

```
10
553 828 297 626 31 637 670 847 216 264
3
50 407
386 657
84 138
```

Výstup:

```
3
3
0
```

Vstup:

```
10
13 481 116 783 403 145 701 45 143 974
2
116 549
238 729
```

Výstup:

```
5
3
```

Vstup:

```
10
80 949 219 149 311 716 922 244 509 25
4
40 738
103 333
339 522
59 864
```

Výstup:

```
7
4
1
7
```

Vstup:

```
7
5 1 1 2 3 4 4
3
1 5
2 3
100 110
```

Výstup:

```
7
2
0
```

Odovzdávanie: Odkaz na odovzdanie programátorskej úlohy : `https://testovac.ksp.sk/tasks/eads2022-du2a/`

4. [*20 bodov*] **Dámy na šachovnici (programátorská úloha).** Máme šachovnicu s rozmermi $10^9 \times 10^9$. Na šachovnici je umiestnených N dám. Dostanete pozície políčok, kde stoja jednotlivé dámy. Vašou úlohou je zistiť, koľko dvojíc dám sa navzájom ohrozuje. Dve dámy sa ohrozujú vtedy, ak sa nachádzajú v rovnakom riadku, stĺpci alebo na rovnakej diagonále, a **nie je medzi nimi žiadna iná dáma**.

Vstup. Na prvom riadku vstupu je číslo N. Nasleduje N riadkov, každý z nich obsahuje 2 čísla r_i a s_i označujúce riadok a stĺpec, kde sa nachádza i-ta dáma. Platí, že $1 \leq N \leq 10^5$, $1 \leq r_i \leq 10^9$ a $1 \leq s_i \leq 10^9$.

Výstup. Vypíšte jedno číslo, počet dvojíc dám, ktoré sa ohrozujú.

Príklady:

Vstup:

```
4
1 7
3 7
7 7
103 107
```

Výstup:

```
3
```

Dvojice, ktoré sa ohrozujú sú 1-2, 2-3, 2-4. Všimnite si, že dámy 1 a 3 sa neohrozujú, keďže medzi nimi je dáma 2.

Vstup:

```
2
3 3
5 2
```

Výstup:

```
0
```

Odovzdávanie: Odkaz na odovzdanie programátorskej úlohy : `https://testovac.ksp.sk/tasks/eads2022-du2b/`

Všeobecné pokyny

Písomné úlohy. Písomné úlohy odovzdávajte *do Google Classroom* ako PDF súbory v stanovenom termíne. **Každý príklad odovzdajte v osobitnom PDF súbore.** Na neskoro odovzdané riešenia sa nebude prihliadať. 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.

Ak nie je povedané inak, logaritmy majú základ 2.

Programátorské úlohy. Pri programátorských úlohách je Vašou úlohou odovzdať len funkčný program, nie je vyžadované písomné riešenie. Riešenie odovzdávate cez webové rozhranie `https://testovac.ksp.sk/tasks/`, 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. Na odovzdávanie riešení (keďže na testovač nefungujú univerzitné prihlasovacie údaje) je nutné sa na stránke zaregistrovať (vľavo na stránke testovača). Pri vytváraní účtu nastavte správne meno a priezvisko, a ako používateľské meno nastavte Váš univerzitný login. Nezabudnite tiež napísať Vaše používateľské meno do PDF súboru k ostatným úlohám, ktoré odovzdávate. Podrobnosti o tom, ako má váš program vyzerať (vrátane povolených programovacích jazykov), nájdete v sekcii "Čo odovzdávať?". Informácie o testovači nájdete v sekcii "Odpovede testovača".

Na zoznámenie sa s rozhraním testovača si môžete vyskúšať naprogramovať niektoré z úloh z časti "Úvod do programovania".