

1 (True or False) and Justify

[20 bodov]

1. Máme pole, ktoré obsahuje n troj písmenných kódov letísk. Dá sa toto pole usporiadať v čase lepšom ako $\Theta(n \log n)$?
2. Ak je pri hešovaní počet prvkov ostro menší ako veľkosť tabuľky, v ktorej sú uložené, tak nenastanú žiadne kolízie.
3. Pomocou dokonalej mince sa dá rovnomerne náhodne rozhodnúť medzi tromi možnosťami.
4. Z hodnoty x vieme hodnotu x^{34} získať pomocou menej ako 10 násobení.
5. Dané sú dve binárne haldy obsahujúce po n prvkov. Dá sa z nich v $O(n)$ spraviť usporiadané pole obsahujúce všetkých $2n$ prvkov?
6. Dané sú dve binárne haldy obsahujúce po n prvkov. Dá sa z nich v $O(n)$ spraviť binárna halda obsahujúca všetkých $2n$ prvkov?
7. Dané sú dva binárne vyhľadávacie stromy obsahujúce po n prvkov. Dá sa z nich v $O(n)$ spraviť usporiadané pole obsahujúce všetkých $2n$ prvkov?
8. Dané sú dva **ľubovoľné** binárne vyhľadávacie stromy obsahujúce po n prvkov. Dá sa z nich v $O(n)$ spraviť **vyvážený** binárny vyhľadávací strom obsahujúci všetkých $2n$ prvkov?

2 Pohľad zvnútra: hrb

[10+10 bodov]

Pole $A[1..n]$ obsahuje hrb. Presnejšie, obsahuje celé čísla, o ktorých sa vie, že najskôr ostro rastú a potom ostro klesajú. A ešte presnejšie, existuje z také, že $1 \leq z \leq n$ a platí $\forall i < z : A[i] < A[i + 1]$ a $\forall i > z : A[i] < A[i - 1]$.

Detailne implementujte funkciu, ktorá (pokiaľ možno v lepšom ako lineárnom čase) nájde v poli A maximum. Môžete predpokladať, že pole A už máte načítané v pamäti. Nezabudnite na dôkaz správnosti a odhad časovej zložitosti.

Následne dokážte, že keby sme zo zadania odstránili slová „ostro“ (t.j. zmenili $A[i] < A[i \pm 1]$ na $A[i] \leq A[i \pm 1]$), tak nebude existovať riešenie v lepšom ako lineárnom čase.

3 Pohľad zvnútra: porovnanie trojice

[10+10 bodov]

Budeme sa hrať nasledovnú hru. Mám pole a v ňom $n \geq 3$ rôzne drahých vecí, očíslovaných od 1 do n . V každom kole hry mi poviete tri rôzne čísla vecí a ja vám poviem, ktorá z tých troch je najdrahšia a ktorá najlacnejšia. Cieľom hry je usporiadať veci od najdrahšej po najlacnejšiu.

Príklad: Mám veci s cenami (10, 47, 1, 2, 42). Poviete mi čísla 1, 4, 5. Odpoviem: najdrahšia je vec 5 a najlacnejšia vec 4.

- a) Dokážte, že neexistuje stratégia, ktorá zaručuje výhru v tejto hre a pritom vždy položí nanaajvýš $7n$ otázok.
- b) Nájdite a detailne popíšte (alebo naprogramujte) stratégiu, ktorou túto hru vyhráte, pričom položíte $O(n \log n)$ otázok.

4 Pohľad zvonka: zlato

[15 bodov]

Veštica Teodora, ktorá sa nikdy nemýli, vyveštila ceny zlata na najbližších n dní: o i dní bude gram zlata stáť $C[i]$ centov. Teraz máte kapitál 100 eur = 10 000 centov. Môžete ľubovoľne veľa krát nakupovať a predávať zlato, a to ľubovoľne malé časti gramu. Ako pseudokód alebo kus programu napíšte čo najefektívnejší algoritmus, ktorý zistí, koľko najviac peňazí sa dá mať o n dní. Nezabudnite na dôkaz správnosti a odhad časovej zložitosti. Zaokrúhľovacie chyby ignorujte.

Príklad: Ak budú v najbližších dňoch ceny zlata (4200, 4000, 4355, 4700, 3950), tak najlepšie je nakúpiť o 2 dni 2.5 gramu zlata a o ďalšie 2 dni všetko predať. Výsledný kapitál: $2.5 \times 4700 = 11750$ centov, teda 117.5 eura.

5 Pohľad zvonka: anonymný list

[15+10 bodov]

Máte časopis C , t. j. veľmi dlhý reťazec znakov. Chcete z neho vystrihnúť niektoré písmená a z tých poskladať správu S . (Pre jednoduchosť predpokladajme, že z druhej strany každého textu je vytlačená nejaká fotka, takže vystrihnutím písmena nijaké iné nestratíte.)

- a) Na vstupe je dané malé číslo k a polia C a S , predstavujúce dané dva reťazce C a S . Každý prvok v poliach C a S je číslo z množiny $\{1, 2, \dots, k\}$: rôzne čísla predstavujú rôzne znaky.

Ako pseudokód alebo kus programu napíšte čo najefektívnejší algoritmus, ktorý zistí, či sa daná správa dá z daného časopisu vyrobiť. Ak áno, váš program by mal pre každé písmeno správy vypísať, koľké písmeno časopisu naň použij. Nezabudnite na dôkaz správnosti a odhad časovej zložitosti.

Príklad: Vstup $C = abeceda$ a $S = baca$ môže byť zadaný ako $k = 5$, $C = (1, 2, 5, 3, 5, 4, 1)$ a $S = (2, 1, 3, 1)$. Jeden správny výstup pre tento vstup je postupnosť 2,7,4,1.

- b) Na vstupe sú reťazce C a S . Predpokladáme, že vstupné reťazce sú v tom istom európskom jazyku, teda používajú nejakú malú podmnožinu Unicode. (Každé $C[i]$ a $S[i]$ je jedno písmeno, trebárs v kódovaní UTF-8.)

Ako pseudokód alebo kus programu napíšte čo najefektívnejší algoritmus, ktorý zistí počet k rôznych písmen v $C + S$ a vyrobí polia čísel C a S , ktoré budú spolu s číslom k vstupom pre podúlohu a).