

**1 (True or False) and Justify**

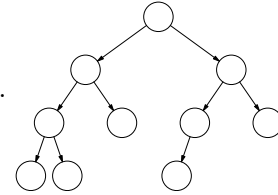
[20 bodov]

1. Strom z úlohy 2 je vyvážený (v zmysle používanom pri AVL stromoch).
2. Obsah poľa, v ktorom máme uloženú haldu, je jednoznačne určený množinou prvkov, ktoré sú v tejto halde uložené.
3. Tvar vyváženého binárneho stromu je jednoznačne určený množinou prvkov, ktorá je v ňom uložená.
4. Každý binárny strom s  $n$  listami má hĺbku  $O(\log n)$ .
5. Časová zložitosť MergeSortu na poli s  $n$  prvkami je  $O(n^3)$ .
6. Aladár mal pole veľkosti  $n$ , v ktorom boli celé čísla z rozsahu od 1 po  $n^2$ . Na usporiadanie tohto poľa použil CountSort: ku každej hodnote si spočítal počet výskytov a podľa toho vyplnil výstupné pole. Jeho algoritmus mal lepšiu časovú zložitosť ako by mal v tejto situácii klasický MergeSort.
7. Dá sa z haldy obsahujúcej  $n$  prvkov vyrobiť v čase  $O(n)$  (ľubovoľný, nie nutne vyvážený) binárny vyhľadávací strom obsahujúci tých istých  $n$  prvkov?
8. Dá sa z (ľubovoľného, nie nutne vyváženého) binárneho vyhľadávacieho stromu obsahujúceho  $n$  prvkov vyrobiť v čase  $O(n)$  halda obsahujúca tých istých  $n$  prvkov?

**2 Pohľad zvnútra: BST**

[7 bodov]

Na obrázku je binárny vyhľadávací strom obsahujúci 10 navzájom rôznych prvkov. Vyfarbite všetky vrcholy, kde sa môže nachádzať tretí najmenší prvok. Veľmi stručne slovne zdôvodnite správnosť svojej odpovede.



**3 Pohľad zvnútra: náhodná fronta**

[20 bodov]

Popíšte, ako by ste implementovali dátovú štruktúru *náhodná fronta*, ktorá podporuje operácie „vložiť prvok“ a „vyber náhodný prvok“. Pri výbere musia mať všetky prvky rovnakú pravdepodobnosť, že budú vybrané. Snažte sa dosiahnuť čo najlepšiu časovú zložitosť pomalšej z operácií.

Môžete predpokladať, že máte k dispozícii funkciu generujúcu vhodné náhodné čísla. Ak neviete úlohu riešiť vo všeobecnosti, pridajte si predpoklad, že v našej fronte sa nikdy naraz neobjaví viac ako 100 000 prvkov.

**4 Pohľad zvnútra: jackpot**

[10+3 bodov]

Na digitálnom displeji pred kasínom sa zobrazuje výška jackpotu. Každú sekundu narastie jackpot o 1 euro. Aby bol stále jackpot korektne zobrazený, je potrebné každú sekundu niektoré segmenty na displeji zhasnúť a niektoré iné rozsvietiť. Displej má  $d$  cifier, každá cifra je zobrazená pomocou klasických 7 segmentov. Práve niekto vyhral jackpot, takže  $d - 1$  cifier nesvieti a na poslednej pozícii svieti 0.

Nájdite asymptotický odhad počtu segmentov, ktoré sa dokopy prepnú počas nasledujúcich  $n$  sekúnd. (Môžete predpokladať, že  $n \leq 10^d$ .) Potom uveďte, čo sa zmení, keď vynecháme podmienku, že na začiatku je na displeji nula.

**5 Pohľad zvonka: krówki**

[20 bodov]

Mišof je závislý na króvkach. Musí každú minútu zjesť jednu, inak umrie v bolestivých kŕčoch. Krówki nakupuje úplne čerstvé od poľských šmelinárov. Čerstvá krówka vydrží  $d$  minút, potom stvrdne a vyláme si na nej zuby. Na vstupe je číslo  $d$ , číslo  $n$  a pole obsahujúce ceny króviok počas najbližších  $n$  minút. Nájdite efektívny algoritmus, ktorý spočíta najmenšiu celkovú cenu króviok pre Mišofa. Dokážte jeho správnosť a odhadnite časovú zložitosť.

(Na plný počet bodov stačí čokoľvek použiteľné pre  $d, n \leq 10^6$ . Bonusové body za asymptoticky optimálne riešenie!)

Príklad: pre  $d = 3$  a ceny (10, 1, 5, 2, 5, 10, 10) optimálnu cenu 18 dosiahne takto: 1. minútu kúpi 1 króvkku a zje ju, 2. minútu ich kúpi 4 (viac nestihne zjesť, kým stvrdnú) a počas 4. minúty dokúpi ďalšie 2 (tie zje počas 6. a 7. minúty).

**6 Pohľad zvonka: reklamy v telke**

[4 × 5 bodov]

Predávaš zubné kefky zahnuté do praveho uhla, ale moc nejdú na odbyt. Ešteže televízia Matfýza v tomto predvianočnom období nepretržite vysiela bloky reklám. O každom bloku  $i$  je známy počet ľudí  $P[i]$ , ktorí ho budú sledovať. Chcel by si si dať pustiť reklamu na svoje kefky v niektorých blokoch. Pritom sleduješ dva ciele. Prvý: aby neliezla ľuďom príliš na nervy. Ten dosiahneš tak, že medzi každými dvomi jej vysielaniami bude aspoň jeden blok, v ktorom tvoja reklama vysielaná nebude. Druhý cieľ je, aby ju dokopy videlo čo najviac ľudí. Treba teda vybrať takú množinu blokov, aby súčet ich hodnôt  $P$  bol najväčší možný.

Príklad: pre  $P = (70\ 000, 30\ 000, 100\ 000, 4\ 500, 9\ 000, 470\ 000)$  by sme vybrali prvý, tretí a šiesty blok.

- a) Dokážte alebo vyvráťte správnosť pažravého algoritmu: Kým máme na výber, tak v každom kroku vyberieme ten blok, ktorý nesusedí so žiadnym skôr vybratým blokom a zo všetkých takých má najviac divákov.
- b) Napíšte (ako pseudokód alebo kus programu) rekurzívny algoritmus skúšajúci všetky možnosti ako vybrať povolenú sadu blokov. Odhadnite jeho časovú zložitosť zhora. (Bonusové body za asymptoticky tesný odhad!)
- c) Pridajte do predchádzajúceho algoritmu memoizáciu tak, aby vznikol algoritmus s časovou zložitosťou polynomiálnou od počtu blokov  $n$ . Odhadnite jeho časovú zložitosť.
- d) Uveďte ekvivalentný algoritmus, ktorý túto úlohu rieši pomocou dynamického programovania.