

1 (True or False) and Justify

[20 bodov]

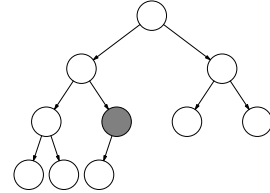
- Existuje dátová štruktúra, ktorá v konštantnom čase zvláda vloženie prvku aj nájdenie maxima. (Prvky netreba vedieť odstraňovať. Ani nájdenie maxima ho z dátovej štruktúry neodstráni.)
- Existuje dátová štruktúra, ktorá v konštantnom čase zvláda vloženie prvku aj odstránenie maxima.
- Vektor obsahujúci n prvkov, ktoré majú každý b bajtov, zaberá vždy $O(bn)$ bajtov pamäte.
- Vyvážený binárny vyhľadávací strom obsahujúci n takých prvkov zaberá $\Theta(n \log n)$ bajtov pamäte.
- Pole A má 5 prvkov, sú navzájom rôzne. Existuje algoritmus, ktorý A usporiada pomocou ≤ 6 porovnaní dvojíc.
- V úlohe o stabilných manželstvách existuje vždy práve jedno riešenie.
- Každý algoritmus iterujúci cez všetky podmnožiny množiny $\{0, 1, \dots, n - 1\}$ má časovú zložitosť $\Omega(n!)$.
- Prioritnú frontu vieme efektívne implementovať ako vyvažovaný binárny vyhľadávací strom.

2 Pohľad zvnútra: halda

[5+5 bodov]

Na obrázku je binárna halda s maximom v koreni. Sú v nej uložené čísla 1 až 10.

- Doplňte čísla do haldy tak, aby vyfarbené pole malo čo najväčšie číslo.
 - Doplňte čísla do haldy tak, aby vyfarbené pole malo čo najmenšie číslo.
- Stručne zdôvodnite, prečo tam väčšie ani menšie číslo od tých vašich byť nemôže.



3 Pohľad zvnútra: časová zložitosť v praxi

[10+5 bodov]

Anička napísala program, ktorý načíta n -prvkové pole čísel, $1000 \times$ ho vypíše na výstup a potom spočíta duplikáty v poli tak, že každý prvok postupne porovnáva so všetkými predchádzajúcimi.

Odhadnite, ako dlho bude Aničkin program bežať na súčasnom bežnom počítači pre $n = 10, 100, 1000, \dots, 1\,000\,000$.

Boris spustil Aničkin program na počítači, ktorý má v pivnici. Pre $n = 1\,000\,000$ na ňom tento program bežal asi 1.2 sekundy. Ako dlho by tento program bežal na Borisovom počítači pre $n = 3\,000\,000$?

4 Pohľad zvnútra: binárny strom

[15 bodov]

Napište (nejaký, čím efektívnejší, tým lepšie) algoritmus, ktorý k danému vrcholu binárneho vyhľadávacieho stromu nájde vrchol s nasledujúcou väčšou hodnotou. Inými slovami, napíšte (do detailov) program, ktorý sa spustí vždy, keď inkrementujem iterátor ukazujúci na prvok v sete.

(Vstupom je ukazovateľ na vrchol v strome, návratovou hodnotou je ukazovateľ na iný vrchol. Detaily uloženia stromu v pamäti si vhodne zvolte.)

5 Pohľad zvonka: jedenkrát zlato

[15 bodov]

V poli $Z[0..n - 1]$ máme od veštice Teodory presné ceny zlata (v eurách za gram) na dnes a nasledujúcich $n - 1$ dní. Máme 1000 eur. Aby sme neboli podozriví, môžeme *len raz* za všetky peniaze (alebo prípadne len ich časť) zlato nakúpiť a niekedy *neskôr* (alebo prípadne hneď v ten istý deň) ho zas predať. Navrhňte algoritmus, ktorý zistí, koľko najviac peňazí môžeme mať na konci. Dokážte jeho správnosť a odhadnite časovú zložitosť. (Môžete sa bez rozpisovania odvolávať na algoritmy a dôkazy z prednášok.)

Príklad: ak $Z = (30, 29, 33, 37, 35)$, tak je najlepšie nakúpiť zajtra zlato za všetky peniaze a o dva dni neskôr ho predať.

6 Pohľad zvonka: kozmonaut

[5 × 5 bodov]

Howard ide na vesmírnu stanicu. Treba mu nabaliť čo najviac jedla. V ponuke je n typov túb s jedlom. Každá tuba i -teho typu má kladnú celočíselnú hmotnosť m_i (v gramoch) a kladné reálne číslo c_i udávajúce počet kalórií. Každé tuby si Howard môže nabrať, koľko kusov chce, dokopy sa ale musí zmestiť do celkového hmotnostného limitu M , lebo viac ruská raketa nevniesie. Zistite, koľko najviac kalórií môže Howard zobrať so sebou.

Príklad: pre $M = 65000$, $n = 3$ a typy vecí s $(m_i, c_i) = (30000, 1000), (16000, 120), (4047, 1)$ Howard vezme 2 veci prvého typu a 1 vec tretieho typu. Celková hmotnosť bude 64047, celkové množstvo kalórií 2001.

- Napište (ako pseudokód alebo kus programu) rekurzívny algoritmus skúšajúci všetky možnosti ako vybrať jedlo. (Začne napr. tým, že pre n -tý typ tuby postupne rekurzívne vyskúša možnosti „ešte jednu takúto vezmem“ a „už žiadnu takúto nevezmem“.) Zdôvodnite, že váš program pre ľubovoľný možný vstup naozaj skončí.
- Pridajte do predchádzajúceho algoritmu memoizáciu tak, aby vznikol algoritmus s časovou zložitosťou polynomiálnou od počtu typov túb n . Odhadnite jeho časovú zložitosť.
- Uveďte ekvivalentný algoritmus, ktorý túto úlohu rieši pomocou dynamického programovania.
- Existujú situácie, kedy sa môžeme pozrieť na hmotnosti túb a množstvá kalórií a z nich priamo usúdiť, že niektoré tuby určite v optimálnom riešení nepoužijeme. Pre nasledujúcu sadu typov túb nájdite dva typy túb, ktoré Howard určite brať nebude (bez ohľadu na M). Vyslovte kritérium, ktoré ste použili, všeobecne. Dokážte jeho správnosť. Máme $n = 7$ typov túb s $(m_i, c_i) = (3200, 1001), (447, 1), (3000, 1000), (1600, 120), (1310, 98), (2320, 84), (415, 1)$.
- Dokážte, že nefunguje pažravý algoritmus, ktorý typy túb usporiada zostupne podľa hodnoty c_i/m_i a následne zoberie každé z nich čo najviac kusov.