

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

[20 bodov]

O každom tvrdení uveďte, či je pravdivé, a tromi vetami svoj názor zdôvodnite. Úplne správna odpoveď: 2.5 boda. Správna odpoveď bez zdôvodnenia: 0.5 boda. Nesprávna odpoveď: 0 bodov. Správna odpoveď s úplne zlým zdôvodnením: 0 bodov.

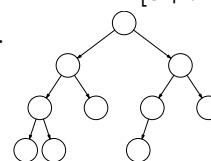
1. Ak budeme v MergeSorte pole zakaždým deliť na tretinu a dve tretiny, bude časová zložitosť naďalej  $O(n \log n)$ .
2. Vo vektore už máme 47 prvkov. Postupne na jeho koniec pridáme  $n$  ďalších. Toto celé má časovú zložitosť  $O(n)$ .
3. Usporiadať ľubovoľné pole veľkosti  $4\sqrt{n}$  vieme v čase  $O(n)$ .
4. Hodnoty v *listoch* binárneho vyhľadávacieho stromu sú usporiadané podľa veľkosti zľava doprava.
5. Vo vyváženom binárnom strome sa hĺbky každých dvoch listov líšia nanaajvýš o 1.
6. Z binárnej haldy s minimom v koreni sa dá v čase  $O(n)$  spraviť binárna halda s maximom v koreni.
7. Vieme si rovnomerne náhodne vybrať z 5 možností, ak smieme počas rozhodovania max.  $47 \times$  hodiť klasickou kockou?
8. Pri univerzálnom hešovaní do poľa veľkosti  $n > 1$  vie nepriateľ vopred pripraviť dvojicu prvkov, ktorá vyrobí kolíziu.

**2 Pohľad zvnútra: binárny strom**

[3+7 bodov + 10 bonus]

Na obrázku je binárny vyhľadávací strom obsahujúci *párne* čísla od 2 do 20, vrátane.

- a) Ktoré hodnoty môžu byť v koreni?
- b) Nájdite jedno možné poradie, v ktorom mohli byť hodnoty (bez vyvažovania stromu) vkladané, aby mal tento tvar.
- c) Koľko rôznych poradí vyhovuje ako odpoveď v časti b)?



**3 Pohľad zvnútra: nefronta nezásobník**

[20 bodov]

Detailne implementujte čo najefektívnejšiu dátovú štruktúru, podporujúcu nasledovné operácie: i) vloženie prvku, ii) výber niektorého prvku. Pri operácii typu ii) je nám skoro jedno, ktorý prvok bude vybraný, máme len jedno obmedzenie: ak sú práve v dátovej štruktúre viac ako dva prvky, tak vybraný prvok nesmie byť ani prvý vložený, ani posledný vložený prvok – teda ani ten prvok, ktorý by bol vybraný, keby šlo o frontu, ani ten, ktorý by sme vybrali, keby šlo o zásobník. Odhadnite časovú zložitosť každej z operácií pri vašej implementácii.

(Nanaajvýš 14 bodov môžete dostať za riešenie ľahšej verzie úlohy. V nej môžete bez uvádzania detailov implementácie použiť dátové štruktúry fronta, zásobník, vektor a spájaný zoznam.)

**4 Pohľad zvnútra: záporné čísla**

[10 bodov]

V pamäti máme pole  $A[0..n - 1]$ , v ktorom je *ostro rastúca* postupnosť reálnych čísel. Napíšte funkciu, ktorá (ideálne v lepšom ako lineárnom čase) zistí, koľko z týchto čísel je záporných.

**5 Pohľad zvonka: behy**

[15 bodov]

*Beh* je maximálny usporiadaný úsek poľa. Napr. pole (10, 20, 30, 5, 6, 7, 7, 8, 3, 8) má tri behy. Dokážte alebo vyvráťte tvrdenie: existuje algoritmus, ktorý ľubovoľné pole s  $n$  prvkami a  $b$  behmi usporiada v čase  $O(n \log b)$ .

**6 Pohľad zvonka: zber známok**

[15/25 bodov]

Deti zbierajú známky a nosia ich mame. Navrhňte pre mamu čo najefektívnejší algoritmus, ktorý bude evidovať udalosti nižšie popísaných typov. Mená detí sú ľubovoľné (rozumne dlhé) reťazce, známky majú 9-ciferné evidenčné čísla. Môžete sa bez rozpísovania odvolávať na algoritmy a dôkazy z prednášok.

Ľahšia verzia (15 bodov): Jediný typ operácie je „dieťa s menom  $x$  prinieslo známku číslo  $y$ “. Pri jej spracovaní je potrebné upozorniť, ak dané dieťa už má rovnakú známku.

Ťažšia verzia (25 bodov): Treba efektívne implementovať nasledovné operácie:

- „dieťa s menom  $x$  prinieslo známku číslo  $y$ “,
- „dieťa s menom  $x_1$  dalo dieťaťu s menom  $x_2$  známku číslo  $y$ “,
- „chceme zoznam všetkých známok, ktoré má dieťa s menom  $x$ “.