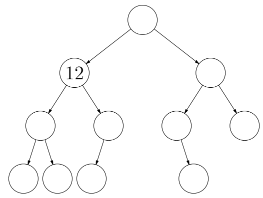
Písomná skúška z ADŠ dňa 18. 1. 2012

1 (True or False) and Justify [20 bodov]

1. Pomocou spájaného zoznamu sa dá efektívne simulovať vektor.
2. Pomocou vektoru sa dá efektívne simulovať zásobník (stack).
3. Pomocou deque (obojsmernej fronty) sa dá efektívne simulovať vektor.
4. Pomocou setu (resp. multisetu) sa dá efektívne triediť – v čase $\Theta(n \log n)$.
5. Do hešovacej tabuľky veľkosti n ideme uložiť menej ako $n/2$ prvkov. Ak použijeme dobrú hešovaciu funkciu (napr. univerzálne hešovanie), s pravdepodobnosťou $> 1/2$ nenastanú žiadne kolízie.
6. Ak binárny vyhľadávací strom nie je vyvážený, tak nevieme zostrojiť jeho pre-order zápis v čase $O(n)$.
7. Daná je postupnosť n prvkov. Prvok, ktorý sa v nej najviackrát opakuje, vieme nájsť v čase $O(n \log n)$.
8. Zoznam n živých ľudí chceme usporiadať podľa dátumu narodenia. Najlepší algoritmus má čas. zlož. $\Theta(n \log n)$.

2 Pohľad zvnútra: binárny vyhľadávací strom [3+7 bodov]



Na obrázku je binárny vyhľadávací strom.
a) Je vyvážený? Prečo?
b) Najmenšie prvky väčšie ako 12 sú 14, 20, 42 a 64. Doplňte ich do obrázka.

3 Pohľad zvnútra: optimalizácia výroby [15 bodov]

Kleofáš vyrába ručne maľované ťuplíky. Prieskum trhu ukázal, že ak ich bude predávať za cenu x, predá ich ročne $a/(x^b \ln(x+3))$. Ak Kleofáš ročne vyrobí t ťuplíkov, tak bude mať výrobné náklady $c + d/(t \ln(t+3))$ na jeden ťuplík. Pri riešení predpokladajte, že sa dá vyrobiť a predať aj neceločíselný počet ťuplíkov.
Napíšte program, ktorý načíta reálne čísla a, b, c, d (všetky sú väčšie ako 1) a hodnotu $\varepsilon > 0$ a s presnosťou ε vypočíta cenu $\leq 10^9$, za ktorú má Kleofáš predávať ťuplíky, ak chce mať maximálny možný čistý zisk.
Hint: Funkcia udávajúca celkový zisk v závislosti od ceny má na $[c, 10^9]$ jediné lokálne (a zároveň globálne) maximum.

4 Pohľad zvnútra: náhodný výber [5+5+(5+bonus) bodov]

a) V pamäti je pole $A[0..n-1]$. Napíšte program, ktorý ho náhodne preusporiada.
b) V pamäti je pole $A[0..n-1]$ a číslo k ($k < n$). Napíšte program, ktorý náhodne vyberie k prvkov poľa A. (Pole A je dovolené meniť.)
c) Na vstupe bude postupne prichádzať n prvkov. Napíšte program, ktorý použije len $O(1)$ pamäte a náhodne vyberie jeden z nich. Bonusové body za všeobecnejšie riešenie: pomocou $O(k)$ pamäte vybrať k-ticu náhodných prvkov.
(Preusporiadania a výbery musia byť **rovnomerne** náhodné – všetky možnosti musia byť rovnako pravdepodobné.)

5 Pohľad zvonka: rozvrh [20 bodov]

Na matfyze sa v budúcom semestri bude prednášať n prednášok. Každá má presne stanovený čas začiatku $Z[i]$ a konca $K[i]$ (pre jednoduchosť nech sú to prirodzené čísla udávajúce offset od začiatku týždňa). Taktiež pre jednoduchosť predpokladajme, že každá prednáška môže byť robená v každej miestnosti.
Ako pseudokód alebo kus programu napíšte čo najefektívnejší algoritmus, ktorý zistí, koľko najmenej rôznych miestností treba na to, aby sa všetky prednášky mohli uskutočniť. (Stačí zistiť počet miestností, netreba vypisovať konkrétne rozmiestnenie predmetov.) Nezabudnite na dôkaz správnosti a odhad časovej zložitosti.

Príklad: Ak majú byť prednášky v časoch $[30, 70]$, $[20, 80]$, $[72, 95]$ a $[65, 73]$, tak treba 3 miestnosti (prvá a tretia prednáška môžu byť po sebe v tej istej miestnosti).

6 Pohľad zvonka: hokej [4 × 5 bodov]

Hokejový zápas sa skončil 4:1. Ľuboško by chcel zrekonštruovať priebeh zápasu. Pamätá si len to, že nikdy nebol stav 3:0 ani 0:1. Koľko rôznych priebehov zápasu vyhovuje tomu, čo si Ľuboško pamätá?
(Odpoveď je 2: buď dali hostia gól za stavu 1:0, alebo za stavu 2:0.)
Budeme riešiť všeobecnú verziu tejto úlohy. Cieľom je zistiť počet možných priebehov zápasu pre dané záverečné skóre $d : h$, číslo n a polia $A[0..n-1]$ a $B[0..n-1]$. Pre každé i si je Ľuboško istý, že stav nikdy nebol $A[i] : B[i]$.

a) Napíšte (ako pseudokód alebo kus programu) rekurzívny algoritmus, ktorý vyskúša a overí všetky možné priebehy zápasu a spočíta, koľko z nich vyhovuje. (Odporúčanie: začne tým, že vyskúša obe možnosti, kto strelil posledný gól zápasu.)
b) Pridajte do predchádzajúceho algoritmu memoizáciu tak, aby vznikol algoritmus s časovou zložitosťou polynomiálnou od d, h, n. Odhadnite jeho časovú zložitosť.
c) Uveďte ekvivalentný algoritmus, ktorý túto úlohu rieši pomocou dynamického programovania.
d) Zistite, koľko rôznych priebehov mohol mať zápas, ktorý skončil 4:6, pričom nikdy nebolo skóre 2:2 ani 1:5. (Ľubovoľný postup vedúci k správnej odpovedi je OK. Jednou z možností je simulovať jeden z algoritmov b), c).)