



Polsemestrálny test teoretická časť



Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Píšte prosím čitateľne!

Hodnotenie, vyplní opravujúci:

Meno a priezvisko:	Skupina PAZ:	
--------------------	--------------	--

1/2	2/1,5	3/2	4/1,5	5/2	6/2,5	7/2	8/1	9/6	10/2	Σ22,5

Zdôvodnenia majú byť stručné (1-3 vety) a zachycujúce podstatné argumenty.

1. (1b+1b) Majme metódu `fraktal` uvedenú nižšie.
 - a) Nakreslite obrázok, ktorý vznikne po volaní metódy `fraktal(3,100)`.
 - b) Koľkokrát sa zavolá metóda `fraktal` s úrovňou nula počas vykonávania metódy `fraktal(3,100)`?

Počet volaní: _____

```
public void fraktal(int uroven, double dlzka) {
    if (uroven > 0) {
        turn(-90);
        for (int i = 0; i < 3; i++) {
            turn(45);
            if (uroven % 2 == 0 && i == 1)
                continue;
            step(dlzka);
            fraktal(uroven - 1, dlzka / 3);
            step(-dlzka);
        }
        turn(-45);
    }
}
```

2. (1,5b) Navrhните spôsob, ako možno v čase $O(n \cdot \log n)$ zistiť, či n -prvkové pole obsahuje nejaké 2 rovnaké hodnoty.

3. (2b) Rozhodnite o pravdivosti doleuvedených tvrdení pravda/nepravda. (+0,5 boda za správnu odpoveď, -0,5 boda za nesprávnu odpoveď, 0 bodov za žiadnu. Neprehliadnite rozdiel medzi notáciami Θ , O , Ω . Pre matematikov znakom „=“ myslíme „ \subset “.)

Ak $g(n) = O(f(n))$ tak aj $4 \cdot g(n) = O(f(n))$ pravda / nepravda

Ak $g(n) = O(f(n))$ tak aj $g(n) = O(2 \cdot f(n))$ pravda / nepravda

Ak $g(n) = O(f(n))$ tak aj $g(n) = \Theta(f(n))$ pravda / nepravda

Ak $g(n) = \Omega(f(n))$ tak aj $g(n) = O(f(n))$ pravda / nepravda

4. (1,5b) Vyberte si jedno z tvrdení v úlohe 3, (označte ho hviezdíčkou, nech vieme ktoré ste si vybrali) a dokážte jeho pravdivosť alebo dokážte jeho nepravdivosť (dôkaz alebo kontrapríklad):

5. (2b) Nech `rad` je referencia na objekt realizujúci rad celých čísel. Napíšte aké čísla bude obsahovať list ktorý vráti metóda `zahada`, ak je obsah radu 2, 3, 4, 4, 3, 2?

Čísla v list-e: _____

```
public static List<Integer> zahada(Queue<Integer> rad) {
    List<Integer> list = new LinkedList<Integer>;
    int zvysoak = 0;
    while (!rad.isEmpty()) {
        if (rad.peek() % 2 == zvysoak)
            list.add(rad.poll());
        else {
            int prvok = rad.poll();
            list.add(prvok);
            rad.offer(prvok);
        }
        zvysoak = (zvysoak + 1) % 2;
    }
    return list;
}
```

6. Uvažujme binárnu haldu s maximom v koreni uloženú v poli, pričom koreň sa nachádza na indexe 0 (úplne vľavo). Táto halda obsahuje hodnoty 0, 1, ..., 10 (každú práve raz). Označme si H najväčšiu hodnotu uloženú v listoch haldy.

- (1b) Krížikom označte tie pozície v poli, ktoré zodpovedajú listom haldy.

--	--	--	--	--	--	--	--	--	--	--

- (0,5b) Akú najväčšiu hodnotu H je možné dosiahnuť? H: _____

- (1b) Nájdite takú haldu, v ktorej je hodnota H najväčšia možná a zaznačte jej hodnoty v poli:

--	--	--	--	--	--	--	--	--	--	--

7. (2b) Naprogramujte metódu, ktorá zrealizuje spojenie dvoch utriedených polí p1 a p2 do utriedeného výsledného poľa v čase $O(n)$, napríklad spôsobom ako to robí MergeSort.

```
public static int[] zluc(int[] p1, int[] p2){
```

```
}
```

8. (1b) Pri binárnom vyhľadávaní predpokladáme, že prvky v poli tvoria neklesajúcu postupnosť. Upravte kód z cvičenia nižšie, aby binárne vyhľadávanie fungovalo za predpokladu, že pole tvorí nerastúcu postupnosť.

```
public int indexVPoli(int[] p, int hodnota, int odIdx, int poIdx) {
    while (odIdx <= poIdx) {
        int stredIdx = (odIdx + poIdx) / 2;
        if (p[stredIdx] == hodnota) {
            return stredIdx;
        }
        if (hodnota < p[stredIdx]) {
            poIdx = stredIdx - 1;
        } else {
            odIdx = stredIdx + 1;
        }
    }
    return -1;
}
```

9. (1b za správnu, -0.5b za nesprávnu a 0b za žiadnu odpoveď) **Rozhodnite** o pravdivosti tvrdení vyznačením (napr. zakrúžkovaním) možnosti Áno alebo Nie:

A: Ak máme výpisy z preorder a inorder prechodov binárnym stromom, tak vieme jednoznačne zrekonštruovať tento binárny strom.

Áno Nie

B: Ak MergeSort-om chceme usporiadať pole ktorého veľkosť n je mocninou dvojky, teda $n=2^i$ kde i je prirodzené číslo. Potom počas zlučovania sa vždy zlučujú časti rovnakej dĺžky.

Áno Nie

C: Ak má n -prvkový binárny vyhľadávací strom aspoň $n/2$ listov, potom jeho výška je $O(\log n)$.

Áno Nie

D: Uvažujme ľubovoľnú n -prvkovú postupnosť celých čísel uloženú v poli. Potom existuje algoritmus v čase $O(n)$ a s pamäťou $O(1)$ (t.j. bez pomocného poľa) taký, že dokáže preusporiadať čísla v poli tak, že najprv budú v poli párne čísla a potom nepárne čísla.

Áno Nie

E: Maximálna hodnota v každom binárnom vyhľadávacom strome je uložená v uzle, ktorý nemá pravého potomka.

Áno Nie

F: Uvažujme haldu uloženú v poli. Potom k -ty najväčší prvok v halde sa nachádza medzi prvými k prvkami poľa.

Áno Nie

10. (1b+1b) Zdôvodnite svoju odpoveď ku ľubovoľným dvom tvrdeniam z predošlej úlohy - napíšte ktoré úlohy zdôvodňujete.