



Polsemestrálny test teoretická časť



Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Píšte prosím čitateľne!

Hodnotenie, vyplní opravujúci:

Meno a priezvisko:	Skupina PAZ:	<input type="text"/>
--------------------	--------------	----------------------

1/1.5	2/1.5	3/1.5	4/1.5	5/1.5	6/4	7/2	8/6	9/3	Σ22.5
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

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

- (1.5b+0.5b) Majme metódu `fraktal` uvedenú nižšie.
 - Nakreslite obrázok, ktorý vznikne po volaní metódy `fraktal(3,100)`.
 - Najviac koľko volaní metódy `fraktal` môžeme vidieť naraz v obsahu call stack-u počas vykonávania metódy `fraktal(3,100)`?

Počet volaní: _____

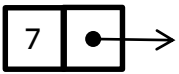
```
public void fraktal(int uroven, double dlzka) {
    if (uroven > 0) {
        for (int i = 1; i <= 4; i++) {
            step(dlzka);
            fraktal(uroven - 1, dlzka / 3);
            turn(90);
        }
    }
}
```

- (1b+0.5b) Pre dole uvedený kód a kladné parametre `zac` a `kon` odhadnite čo najtesnejšie asymptotickú časovú zložitost' a svoje tvrdenie zdôvodnite.

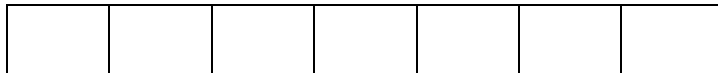
```
public void zlozitost(int zac, int kon) {
    if (zac > kon)
        return;
    if (kon % 2 == 0)
        zlozitost(zac * 2, kon);
    else
        zlozitost(zac, kon / 2);
}
```

3. (1,5b) Uvažujme metódu nižšie. Dokreslite čo najkratší spájaný zoznam, taký aby metóda zahada vrátila hodnotu 3 po spustení na tomto spájanom zozname.

```
public int zahada() {
    Uzol aktualny = prvvy;
    int vysledok = 0;
    while (aktualny != null) {
        if ((aktualny.dalsi != null) &&
            (aktualny.hodnota < aktualny.dalsi.hodnota)) {
            vysledok++;
            aktualny.dalsi = aktualny.dalsi.dalsi;
        }
        aktualny = aktualny.dalsi;
    }
    return vysledok;
}
```



4. (1.5b) Metóda selectionSort utriedi pole, navyše vypíše koľko výmen prvkov v poli nastalo. Dopíšte obsah poľa celých čísel tak, aby metóda selectionSort na danom poli na konci vypísala presne hodnotu 4. Pole nech obsahuje iba čísla 1, 2, ..., 7.



```
public static void selectionSort(int[] p) {
    int pocetVymen = 0;
    for (int i = 0; i < p.length - 1; i++) {
        int minIdx = i;
        for (int j = i + 1; j < p.length; j++)
            if (p[j] < p[minIdx])
                minIdx = j;
        if (minIdx != i) {
            pocetVymen++;
            int pom = p[i];
            p[i] = p[minIdx];
            p[minIdx] = pom;
        }
    }
    System.out.println(pocetVymen);
}
```

5. (0.5b+1b) Popíšte, čo počíta nasledujúca rekurzívna metóda a napíšte aká je jej zložitosť vzhľadom ku parametrom n a k.

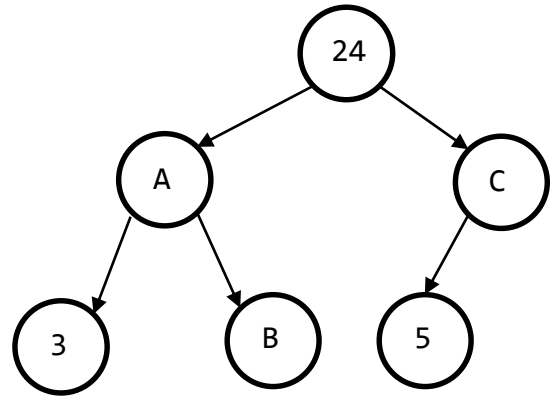
```
public static double zahada(double n, int k) {
    if (k == 0)
        return 1.0;
    if (k > 0)
        return n * zahada(n, k - 1);
    else
        return (1.0 / n) * zahada(n, k + 1);
}
```

6. (4b) Uvažujme 6-prvkovú haldu vpravo.

a) (0.5b) Prepíšte túto haldu do poľa.

b) (1b) Aká hodnota môže byť uložená v uzle A?

c) (1b) Môže byť hodnota C menšia ako B? Ak áno napíšte hodnoty A, B, C, po dosadení ktorých strom bude haldou. Ak nie tak napíšte stručné zdôvodnenie prečo nie.



d) (0.5b) Koľko uzlov treba pridať do tejto haldy aby sa jej výška zvýšila o 2?

e) (0.5b) Aká môže byť minimálna hodnota v tejto halde?

f) (0.5b) Napíšte postorder prechod haldou:

7. (2b) Implementujte triedu `MnozinaCisel`, ktorej objekty uchovávajú nejakú dynamickú (modifikovateľnú) podmnožinu množiny $\{0, 1, 2, \dots, n - 1\}$, pričom n je určené parametrom konštruktora. Implementujte túto triedu s pamäťovou zložitou $O(n)$ tak, aby každá z metód `pridaj` (pridá prvok do množiny), `odober` (odstráni prvok z množiny), `obsahuje` (vráti, či sa hodnota nachádza v množine) mala časovú zložitou $O(1)$.

```
public class MnozinaCisel {  
  
    public MnozinaCisel(int n) {  
  
    }  
  
    public void pridaj(int h) {  
  
    }  
  
    public void odober(int h) {  
  
    }  
  
    public boolean obsahuje(int h) {  
  
    }  
  
}
```

8. (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 $f(n) = O(g(n))$, potom $n + f(n) = O(g(n))$.

Áno Nie

B: Nech n je počet vrcholov binárneho vyhľadávacieho stromu. Potom pre každé n existuje taký BVS, že jeho preorder je usporiadaný.

Áno Nie

C: V halde je maximálna hodnota uložená v koreni. Druhá a tretia najväčšia hodnota v halde sa musia nachádzať v ľavom, resp. pravom synovi koreňa

Áno Nie

D: Uvažujme 2 usporiadané polia dĺžky n . V čase $O(n)$ môžeme zistiť, ktoré z polí má väčší počet záporných čísel a v asymptoticky lepšom čase sa to nedá.

Áno Nie

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

Áno Nie

F: Pri behu MergeSortu v časti, kde sa spájajú usporiadané podpolia, je vždy dĺžka podpolí rovnaká alebo sa líši iba o jedna.

Áno Nie

9. (3b) Zdôvodnite svoju odpoveď k ľubovoľnému tvrdeniu A-C, F (1b) a k ľubovoľnému tvrdeniu D-E (2b) z predošlej úlohy - zakrúžkujte písmená zdôvodňovaných tvrdení.