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	3/1.5	4/2.5	5/2	6/2	7/2	8/1	9/6	10/3	Σ23

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

1. (1.5b+0.5b) Majme metódu `fraktal` uvedenú nižšie.
   a) Nakreslite obrázok, ktorý vznikne po volaní metódy `fraktal(4,100)`.
   b) Najviac koľko volaní metódy `fraktal` môžeme vidieť naraz v obsahu call stack-u počas vykonávania metódy `fraktal(4,100)`?

   Počet volaní: _____

```
public void fraktal(int uroven, double dlzka) {
      if (uroven > 0) {
            for (int i = 0; i < 4; i++) {
                  fraktal(uroven - 1, dlzka / 2);
                  step(dlzka);
                  turn(90);
            }
      }
```

2. (1b) Pri binárnom vyhľadávaní predpokladáme, že prvky v poli tvoria neklesajúcu postupnosť. Upravte kód nižšie, aby binárne vyhľadávanie fungovalo za predpokladu, že pole tvorí klesajúcu postupnosť.

```
public boolean jeVPoli(int[] pole, int odIdx, int poIdx, int cislo){
      if (odIdx > poIdx)
            return false;
      int stredIdx = (odIdx + poIdx) / 2;
      if (pole[stredIdx] == cislo)
            return true;
      if (cislo < pole[stredIdx])
            return jeVPoli(pole, odIdx, stredIdx - 1, cislo);
      else
            return jeVPoli(pole, stredIdx + 1, poIdx, cislo);
}
```

3. (1.5b) Metóda `vymen` vymení v poli `p` prvky na indexoch `i` a `i+1`. Dopíšte obsah poľa celých čísel tak, aby metóda `bubbleSort` na danom poli vykonala presne *10* výmen.

42						

```
public static void bubbleSort(int[] p) {
      boolean bolaVymena;
      do {
            bolaVymena = false;
            for (int i = 0; i < p.length - 1; i++)
                  if (p[i] > p[i + 1]) {
                        vymen(p, i, i + 1);
                        bolaVymena = true;
                  }
      } while (bolaVymena);
}
```

4. 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:

5. (1+1b) Ktorá z nižšie uvedených zložitostí najtesnejšie určuje asymptotickú časovú zložitosť metódy `zahada` vzhľadom na n - veľkosť poľa `p`. Svoju odpoveď zdôvodnite.

$O(1)$ $O(\log n)$ $O(n)$ $O(n.\log n)$ $O(n^2)$ $O(n^2.\log n)$ $O(2^n)$

```
public void zahada(int[] p) {
      int idx = 0;
      int pocet = p.length * p.length;
      for (int j = 0; j < pocet; j++) {
            p[idx] += j;
            idx = (idx + 1) % p.length;
            pocet = pocet / 2;
      }
}
```

6. (2b) Predpokladajte, že máte už naprogramovanú funkciu `pivotuj`, ktorá zrealizuje pivotizáciu v časti poľa `p` ohraničenom indexami `odIdx`, `poIdx` a vráti pozíciu pivota v poli po pivotizácií. S využitím tejto metódy napíšte metódu, ktorá zrealizuje triedenie `QuickSort`.

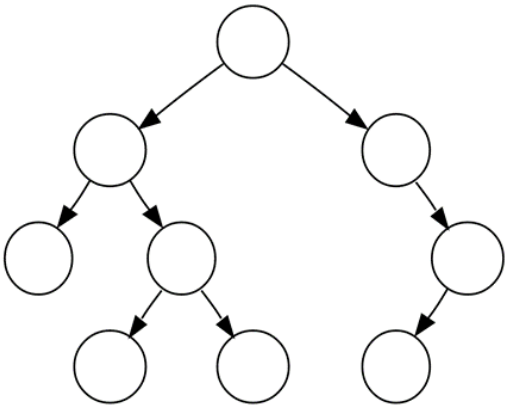
```
public static int pivotuj(int[] p, int odIdx, int poIdx);

public static void quickSort(int[] p, int odIdx, int poIdx) {




}
```

7. (2b) Nech **rad** je referencia na objekt realizujúci rad celých čísel. V akom poradí bude metóda `zahada` vypisovať čísla, ak je obsah radu *2, 3, 4, 4, 3, 2*?

Vypísané čísla: _____

```
public static void zahada(Queue<Integer> rad) {
	int zvysok = 0;
	while (!rad.isEmpty()) {
		if (rad.peek() % 2 == zvysok)
			System.out.println(rad.poll());
		else {
			int prvok = rad.poll();
			System.out.println(prvok);
			rad.offer(prvok);
		}
		zvysok = (zvysok + 1) % 2;
	}
}
```

8. (1b) Do stromu vpravo vpíšte navzájom rôzne prirodzené čísla tak, aby jeho preorder postupnosť bola usporiadaná od najmenšej hodnoty po najväčšiu.



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 $f(n) = O(g(n))$, potom $2 * f(n) + f(n) = O(g(n))$.

   Áno Nie

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

   Áno Nie

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

   Áno Nie

   D: Uvažujme 2 polia dĺžky n. V čase $O(n.logn)$ môžeme zistiť, či existuje hodnota, ktorá sa nachádza v oboch poliach.

   Áno Nie

   E: Ak MergeSort-om chceme usporiadať pole ktorého veľkosť *n* je násobkom mocniny dvojky, teda $n=k.2^i$ kde *k* a *i* sú prirodzené čísla. Potom počas zlučovania sa vždy zlučujú časti rovnakej dĺžky.

   Áno Nie

   F: V binárnej halde s maximom v koreni vieme nájsť maximum v čase $O(1)$, keďže sa nachádza v koreni. V n-prvkovej halde sa ale minimálna hodnota môže nachádzať až v $n/2$ rôznych uzloch haldy, t.j. časová zložitosť nájdenia minimálnej hodnoty v binárnej halde je v najhoršom prípade $\Omega(n)$.

   Áno Nie

10. (3b) Zdôvodnite svoju odpoveď k ľubovoľnému tvrdeniu A-C (1b) a k ľubovoľnému tvrdeniu D-F (2b) z predošlej úlohy - zakrúžkujte písmená zdôvodňovaných tvrdení.