Polsemestrálny test
teoretická časť

Ústav informatiky
Prírodovedecká fakulta
UPJŠ v Košiciach

Píšte prosím čitateľne!

Hodnotenie, vyplní opravujúci:

Meno a priezvisko:	Skupina PAZ:	

1/2	2/2	3/2	4/2	5/2	6/1	7/2	8/1.5	9/8	Σ/22.5

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

1. (2b) Pri analýze algoritmu sa zistilo, že jeho časová zložitosť je vyjadriteľná vzorcom:
$T(1) = 1$
$T(n) = \min\{ T(k) + 2.T(n-k) \mid k = 1..n-1\}$

Napíšte (napr. rekurzívnu) metódu, ktorá presne vypočíta $T(n)$ pre zadané n.

```
public static int t(int n) {


}
```

2. (2b) Doplňte do poľa hodnoty tak, aby metóda zahada s parametrom referencujúcim toto pole vrátila hodnotu 3.

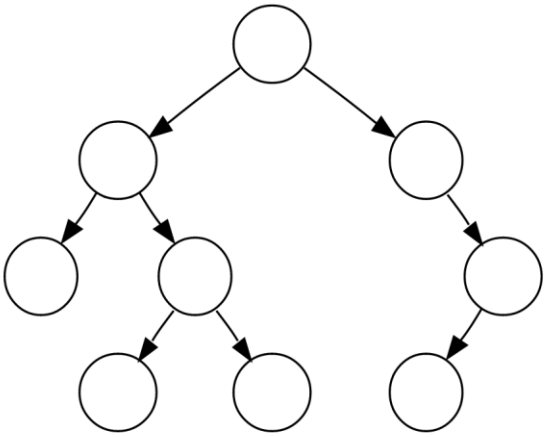
		8			*3*		*6*		*10*

```
int zahada(int[] pole) {
      int[] p = pole.clone();
      Arrays.sort(p);
      int vysledok = 0;
      int c = 1;
      for (int i = 1; i < p.length; i++) {
            if (p[i - 1] == p[i]) {
                  c++;
            } else {
                  c = 1;
            }
            vysledok = Math.max(c, vysledok);
      }
      return vysledok;
}
```

3. (2b) Doplňte do poľa hodnoty tak, aby toto pole reprezentovalo nejakú binárnu haldu s maximom v koreni, alebo zdôvodnite, že také doplnenie neexistuje.

		8			3		6		10

Zdôvodnenie neexistencie:



4. (2b) Do stromu vpravo vpíšte rôzne prirodzené čísla tak, aby strom bol binárnym vyhľadávacím stromom a v jeho preorder postupnosti bolo najprv 5 párnych čísel a potom 4 nepárne.

5. (2b) Zdôvodnite, že neplatí tvrdenie: *Ak má n-prvkový binárny vyhľadávací strom (BVS) aspoň $n/2$ listov, potom jeho výška je $O(\log n)$.*
Hint: Ukážte, že pre každé n existuje BVS s aspoň $n/2$ listami, ktorého výške je $\Omega(n)$.

6. (1b) 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) Nájdite a opravte chyby v metóde `kladneMinimum` triedy `SpajanyZoznam`, ktorá vráti minimálnu kladnú hodnotu uloženú v zozname (zoznam teda môže obsahovať aj záporné hodnoty) alebo -1, ak taká hodnota neexistuje. Kód je syntakticky korektný, t.j. kompilátor Javy nehlási žiadnu chybu.

```
public int kladneMinimum() {
      int vysledok = Integer.MAX_VALUE;
      boolean vysledokOK = false;
      Uzol aktualny = prvy;
      while (aktualny != null) {
            if (aktualny.hodnota > 0) {
                  vysledok = Math.min(0, aktualny.hodnota);
                  vysledokOK = true;
                  aktualny = aktualny.dalsi;
            }
      }

      if (vysledokOK) {
            return vysledok;
      } else {
            return -1;
      }
}
```

8. (1.5b) Máme dve verzie rekurzívnej metódy, ktoré počítajú presne to isté.

```
public static String funkcia1(int n) {
  if (n <= 0)
     return "*";
  else
     return funkcia1(n-1) + funkcia1(n-1);
}

public static String funkcia2(int n) {
  if (n <= 0)
     return "*";
  else {
     String pom = funkcia2(n-1);
     return pom + pom;
  }
}
```

Trvanie *funkcia1(7):*

Trvanie *funkcia2(7):*

Dĺžka reťazca vráteného metódou *funkcia1(7)*:

Predpokladajte, že mechanizmus jedného volania metódy *funkcia1* alebo *funkcia2*, a tiež operácia zreťazenia dvoch reťazcov „stoja“ počítač presne 0.001 sekundy. Všetky ostatné operácie a príkazy zanedbávame. Zistite, koľko bude trvať výpočet *funkcia1(7)* a *funkcia2(7)* a tiež, aký dlhý znakový reťazec sa pri tom vygeneruje.

Poznámka: netreba presnú hodnotu, stačí výraz (napr. $33^{42} - log\ 1024$*)*

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 usporiadané n-prvkové pole, potom v čase $O(log\, n)$ môžeme určiť počet záporných hodnôt v tomto poli.

Áno Nie

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

Áno Nie

C: V každom binárnom vyhľadávacom strome je maximálna hodnota uložená v uzle, ktorý nemá oboch synov (t.j. má jedného syna alebo je listom).

Áno Nie

D: Tvar binárneho vyhľadávacieho stromu je jednoznačne určený množinou prvkov, ktoré uchováva.

Áno Nie

E: V binárnej halde s maximom v koreni pre ľubovoľné dva prvky x, y tejto hlady platí: Ak $x < y$, potom x je sa nachádza v rovnakej alebo väčšej hĺbke (=úrovni stromu) ako y.

Áno Nie

F: $2\mathrm{n} + 3\log n = O(n)$

Áno Nie

Zdôvodnite svoju odpoveď k ľubovoľnému tvrdeniu A-F (2b, zakrúžkujte písmeno zdôvodňovaného tvrdenia):