

[Pregledna plošča](#) / [Moji predmeti](#) / [aps2uni](#) / 11. marec - 17. marec / [Izziv 2](#)

Začeto dne	torek, 12. marec 2019, 10:24
Stanje	Zaključeno
Dokončano dne	torek, 12. marec 2019, 21:52
Porabljeni čas	11 ure 27 min
Točke	1,00/1,00
Ocena	10,00 od možne ocene 10,00 (100%)

Napišite program, ki na standardnem vhodu najprej dobi dolžino tabele n , nato pa iz preostanka standardnega vhoda prebere tabelo celih števil dolžine n .

Prebrano tabelo uredite s kopico.

Izpisujte **sled algoritma** na naslednji način:

- vsakič, ko imate v tabeli pravo kopico jo izpišite (po gradnji kopice, nato pa po vsakem pogrezanju),
- vsako kopico izpišite v svojo vrstico,
- kopico izpišite tako, da vsak nivo drevesa ločite z navpično črto. Primer: 7 | 6 5 | 4 3 2, predstavlja kopico, ki ima v korenu 7, na prvem nivoju števil 6 in 5, na zadnjem pa 4, 3 in 2.

Nasvet: Metodo za urejanje s kopico implementirajte postopoma:

- Najprej implementirajte metodo za pogrezanje, ki delno pokvarjeno kopico popravi (za podrobnosti glej predavanja). Metodo dobro pretestirajte, saj na njej temelji celoten nadaljnji postopek. Podpis te metode je sledeči: `void pogreznj(int a[], int i, int dolzKopice)`, pri čemer je `a` tabela v kateri pogrezamo, `i` je indeks elementa, ki ga pogrezamo in `dolzKopice` je dolžina podtabele, ki trenutno predstavlja kopico.
- Nato implementirajte metodo, ki iz poljubne tabele zgradi veljavno kopico (seveda z uporabo metode razvite v predhodni točki).
- Nazadnje uporabite prej definirani metodi za ureditev poljubne tabele s kopico.

For example:

Input	Result
8 0 13 0 6 15 22 7 9	22 15 7 9 13 0 0 6 15 13 7 9 6 0 0 13 9 7 0 6 0 9 6 7 0 0 7 6 0 0 6 0 0 0 0 0

Answer: (penalty regime: 0 %)

```

68         pogreznj(desniSin, dolzinaKopice);
69     }
70 }
71 }
72
73 private static void pogrezanje (int koren, int dolzinaKopice) {
74     int leviSin = 2* koren +1;
75     int desniSin = 2* koren +2;
76     int temp = -1;
77
78     if (leviSin <= dolzinaKopice)
79         pogrezanje(leviSin, dolzinaKopice);
80     else return;
81
82     if (desniSin <= dolzinaKopice)
83         pogrezanje(desniSin, dolzinaKopice);
84
85     if (desniSin > dolzinaKopice) {
86         if (tabela[leviSin] > tabela[koren]) {
87             //zamenjamo levi sin & koren
88             temp = tabela[leviSin];

```

	Input	Expected	Got	
✓	8 5 7 2 4 9 1 7 6	9 7 7 6 5 1 2 4 7 6 7 4 5 1 2 7 6 2 4 5 1 6 5 2 4 1 5 4 2 1 4 1 2 2 1 1	9 7 7 6 5 1 2 4 7 6 7 4 5 1 2 7 6 2 4 5 1 6 5 2 4 1 5 4 2 1 4 1 2 2 1 1	✓

	Input	Expected	Got	
✓	8 0 13 0 6 15 22 7 9	22 15 7 9 13 0 0 6 15 13 7 9 6 0 0 13 9 7 0 6 0 9 6 7 0 0 7 6 0 0 6 0 0 0 0 0	22 15 7 9 13 0 0 6 15 13 7 9 6 0 0 13 9 7 0 6 0 9 6 7 0 0 7 6 0 0 6 0 0 0 0 0	✓
✓	10 26 22 28 27 29 13 28 27 10 17	29 27 28 27 22 13 28 26 10 17 28 27 28 27 22 13 17 26 10 28 27 17 27 22 13 10 26 27 27 17 26 22 13 10 27 26 17 10 22 13 26 22 17 10 13 22 13 17 10 17 13 10 13 10 10	29 27 28 27 22 13 28 26 10 17 28 27 28 27 22 13 17 26 10 28 27 17 27 22 13 10 26 27 27 17 26 22 13 10 27 26 17 10 22 13 26 22 17 10 13 22 13 17 10 17 13 10 13 10 10	✓
✓	12 31 7 6 2 5 1 13 11 18 14 12 12	31 18 13 11 14 12 6 7 2 5 12 1 18 14 13 11 12 12 6 7 2 5 1 14 12 13 11 5 12 6 7 2 1 13 12 12 11 5 1 6 7 2 12 11 12 7 5 1 6 2 12 11 6 7 5 1 2 11 7 6 2 5 1 7 5 6 2 1 6 5 1 2 5 2 1 2 1 1	31 18 13 11 14 12 6 7 2 5 12 1 18 14 13 11 12 12 6 7 2 5 1 14 12 13 11 5 12 6 7 2 1 13 12 12 11 5 1 6 7 2 12 11 12 7 5 1 6 2 12 11 6 7 5 1 2 11 7 6 2 5 1 7 5 6 2 1 6 5 1 2 5 2 1 2 1 1	✓
✓	15 40 14 3 1 34 34 16 18 11 22 32 28 40 15 21	40 34 40 18 32 34 21 1 11 22 14 28 3 15 16 40 34 34 18 32 28 21 1 11 22 14 16 3 15 34 32 34 18 22 28 21 1 11 15 14 16 3 34 32 28 18 22 16 21 1 11 15 14 3 32 22 28 18 15 16 21 1 11 3 14 28 22 21 18 15 16 14 1 11 3 22 18 21 11 15 16 14 1 3 21 18 16 11 15 3 14 1 18 15 16 11 1 3 14 16 15 14 11 1 3 15 11 14 3 1 14 11 1 3 11 3 1 3 1 1	40 34 40 18 32 34 21 1 11 22 14 28 3 15 16 40 34 34 18 32 28 21 1 11 22 14 16 3 15 34 32 34 18 22 28 21 1 11 15 14 16 3 34 32 28 18 22 16 21 1 11 15 14 3 32 22 28 18 15 16 21 1 11 3 14 28 22 21 18 15 16 14 1 11 3 22 18 21 11 15 16 14 1 3 21 18 16 11 15 3 14 1 18 15 16 11 1 3 14 16 15 14 11 1 3 15 11 14 3 1 14 11 1 3 11 3 1 3 1 1	✓

Passed all tests! ✓

Question author's solution:

```

import java.util.Scanner;

public class Izziv2 {
    public static void dump_heap(int n, int a[]) {
        System.out.print(a[0]);

        for (int i = 1, items_on_level = 2; i < n; items_on_level *= 2) {
            System.out.print(" |");
            for (int j = 0; j < items_on_level && i < n; j++, i++)
                System.out.print(" " + a[i]);
        }
        System.out.println();
    }

    public static void swap(int a[], int i, int j) {
        int tmp = a[i];
        a[i] = a[j];
        a[j] = tmp;
    }

    public static void sink(int n, int a[], int i) {
        while (true) {
            int l_anc = 2 * i + 1;
            if (l_anc >= n)
                break;

            int max = l_anc;
            int r_anc = l_anc + 1;
            if (r_anc < n && a[l_anc] < a[r_anc])
                max = r_anc;

            if (a[i] >= a[max])
                break;

            swap(a, i, max);
            i = max;
        }
    }

    public static void heapify(int n, int a[]) {
        for (int i = (n - 1) / 2; i >= 0; i--) {
            sink(n, a, i);
        }
    }

    public static void heapsort(int n, int a[]) {
        heapify(n, a);
        dump_heap(n, a);

        for (int i = n - 1; i > 0; i--) {
            swap(a, 0, i);
            sink(i, a, 0);
            dump_heap(i, a);
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();

        int[] a = new int[n];
        for (int i=0; i < n; i++) {
            a[i] = sc.nextInt();
        }
        heapsort(n, a);
    }
}

```

Pravilno

Marks for this submission: 1,00/1,00.

◀ Kviz 2

Skok na...

Kviz 3 ▶

