

# Тренировки по алгоритмам 3.0 от Яндекса — Дивизион В

13 мар 2023, 16:13:06  
старт: 13 фев 2023, 13:00:00  
финиш: 13 мар 2023, 18:59:59  
до финиша: 02:46:06  
начало: 13 фев 2023, 13:00:00  
конец: 13 мар 2023, 18:59:59  
длительность: 28д. 5ч.

## 19. Хипуй

Ограничение времени	2 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В этой задаче вам необходимо самостоятельно (не используя соответствующие классы и функции стандартной библиотеки) организовать структуру данных `Heap` для хранения целых чисел, над которой определены следующие операции: а) `Insert(k)` — добавить в `Heap` число `k` ; б) `Extract` достать из `Heap` наибольшее число (удалив его при этом).

### Формат ввода

В первой строке содержится количество команд  $N$  ( $1 \leq N \leq 100000$ ), далее следуют  $N$  команд, каждая в своей строке. Команда может иметь формат: “0 <число>” или “1”, обозначающий, соответственно, операции `Insert(<число>)` и `Extract`. Гарантируется, что при выполнении команды `Extract` в структуре находится по крайней мере один элемент.

### Формат вывода

Для каждой команды извлечения необходимо отдельной строкой вывести число, полученное при выполнении команды `Extract`.

#### Пример 1

Ввод <input type="text"/>	Вывод <input type="text"/>
2	10000
0 10000	
1	

#### Пример 2

Ввод <input type="text"/>	Вывод <input type="text"/>
14	345
0 1	4346
0 345	2435
1	365
0 4346	235
1	5
0 2435	1
1	
0 235	
0 5	

Ввод Вывод 

0 365  
1  
1  
1  
1

Язык 

```
1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
4
5 public class Yandex {
6     static int[] heap = new int[100000];
7     static int heapLength = 0;
8     public static void main(String[] args) throws IOException {
9         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
10        int n = Integer.parseInt(br.readLine());
11        StringBuilder sb = new StringBuilder();
12        for (int i = 0; i < n; i++) {
13            processQuery(br.readLine(), heap, sb);
14        }
15        System.out.println(sb);
16    }
17
18    static void processQuery(String query, int[] heap, StringBuilder result) {
19        String[] queryParams = query.split(" ");
20        if (Integer.parseInt(queryParams[0]) == 0) {
21            heap[heapLength] = Integer.parseInt(queryParams[1]);
22            heapLength++;
23            pushUp(heap, heapLength - 1);
24        } else {
25            result.append(extractMax(heap)).append("\n");
26        }
27    }
28
29    static int extractMax(int[] arr) {
30        int max = arr[0];
31        arr[0] = arr[heapLength - 1];
32        heapLength--;
33        heapify(arr, 0);
34        return max;
35    }
36
37    static void heapify(int[] arr, int i) {
38        while (2 * i + 1 < heapLength) {
```