

# Codechef RRATING

*Restaurant Rating*

---

Prof. Edson Alves – UnB/FCTE

# Problema

Chef has opened up a new restaurant. Like every other restaurant, critics critique this place. The Chef wants to gather as much positive publicity as he can. Also, he is very aware of the fact that people generally do not tend to go through all the reviews. So he picks out the positive reviews and posts it on the website of the restaurant. A review is represented by an integer which is the overall rating of the restaurant as calculated by that particular review. A review is considered to be positive if it is among the top one-third of the total reviews when they are sorted by their rating. For example, suppose the ratings given by 8 different reviews are as follows:

2 8 3 1 6 4 5 7

Then the top one-third reviews will be 8 and 7. Note that we considered one-third to be  $8/3=2$  top reviews according to integer division. (see Notes)

So here is what the Chef wants from you: Given the reviews (ratings) by different critics, the Chef needs you to tell him what is the minimum rating that his website will be displaying. For example in the above case, the minimum rating that will be displayed is 7. Also, different critics keep reviewing the restaurant continuously. So the new reviews keep coming in. The Chef wants your website ratings to be up-to-date. So you should keep updating the ratings there. At any point of time, the Chef might want to know the minimum rating being displayed. You'll have to answer that query. An interesting thing to note here is that a review might be in the website for some time and get knocked out later because of new better reviews and vice-versa.

Notes: To be precise, the number of reviews displayed website will be  $\text{floor}(n/3)$ , where  $n$  denotes the current number of all reviews.

## Input

First line of the input file consists of a single integer  $N$ , the number of operations to follow. The next  $N$  lines contain one operation each on a single line. An operation can be of 2 types:

- 1  $x$  : Add a review with rating ' $x$ ' to the existing list of reviews ( $x$  is an integer)
- 2 : Report the current minimum rating on the website

## Output

For every test case, output a single integer each for every operation of type 2 mentioned above. If no review qualifies as a positive review, print "No reviews yet".

## Constraints

$$1 \leq N \leq 250000$$

$$1 \leq x \leq 1000000000$$

# Exemplo de entradas e saídas

## Sample Input

```
10
1 1
1 7
2
1 9
1 21
1 8
1 5
2
1 9
2
```

## Sample Output

```
No reviews yet
9
9
```

## Solução com complexidade $O(N \log N)$

- Este é um problema dinâmico, onde é preciso manter o registro de um terço dos melhores *ratings* obtidos até o momento
- Para tal serão utilizadas duas *heap* binárias:  $l_o$ , uma *max heap* que manterá os piores *ratings* (2/3 do total); e  $h_i$ , uma *min heap* que manterá os melhores *ratings* (1/3 do total)
- Para manter o invariante destas *heaps*, é preciso, ao processar comandos do tipo 1, inserir os elementos em  $l_o$  e atualizar o total  $R$  de *ratings* já inseridos
- Se o tamanho de  $h_i$  for inferior a  $\lfloor R/3 \rfloor$ , é preciso passar o topo de  $l_o$  para  $h_i$

## Solução com complexidade $O(N \log N)$

- Um último cuidado é necessário: uma inserção pode fazer com que  $l_0$  tenha elementos maiores do que  $h_i$ , violando o invariante
- Assim, se  $h_i$  não estiver vazia, enquanto o topo  $x$  de  $l_0$  for maior do que o topo  $y$  de  $h_i$ , eles elementos devem passar de uma *heap* para a outra
- Como cada elemento será inserido, no máximo, 2 vezes em  $l_0$ , a complexidade desta solução é  $O(R \log R)$ , onde  $R$  é o número de *ratings* obtidos

# Solução AC com complexidade $O(R \log R)$

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 int main()
6 {
7     ios::sync_with_stdio(false);
8
9     int N;
10    cin >> N;
11
12    priority_queue<int> lo;
13    priority_queue<int, vector<int>, greater<int>> hi;
14
15    size_t total = 0;
16
17    while (N--)
18    {
19        int cmd, x;
20        cin >> cmd;
```



## Solução AC com complexidade $O(R \log R)$

```
22     switch (cmd) {
23     case 1:
24         cin >> x;
25
26         lo.push(x);
27         ++total;
28
29         if (hi.size() < total / 3)
30         {
31             hi.push(lo.top());
32             lo.pop();
33         }
34
35         while (not hi.empty() and lo.top() > hi.top())
36         {
37             auto x = lo.top();
38             auto y = hi.top();
39
40             lo.pop();
41             hi.pop();
```

## Solução AC com complexidade $O(R \log R)$

```
43         hi.push(x);
44         lo.push(y);
45     }
46
47     break;
48
49     default:
50         if (hi.empty())
51             cout << "No reviews yet\n";
52         else
53             cout << hi.top() << '\n';
54     }
55 }
56
57 return 0;
58 }
```