AtCoder Beginner Contest 212

Problema D: *Querying Multiset*

Prof. Edson Alves – UnB/FCTE

Problema

Takahashi has many balls, on which nothing is written, and one bag. Initially, the bag is empty. Takahashi will do Q operations, each of which is of one of the following three types.

- Type 1: Write an integer X_i on a blank ball and put it in the bag.
- Type 2: For each ball in the bag, replace the integer written on it with that integer plus X_i.
- Type 3: Pick up the ball with the smallest integer in the bag (if there are multiple such balls, pick up one of them). Record the integer written on this ball and throw it away.

For each $1 \leq i \leq Q$, you are given the type P_i the i-th operation and the value of X_i the operation is of Type 1 or 2. Print the integers recorded in the operations of Type 3 in order

Entrada e saída

Constraints

- $1 \leq Q \leq 2 \times 10^5$
- $1 \leq P_i \leq 3$
- $1 \leq X_i \leq 10^9$
- All values in input are integers.
- There is one or more i such that $P_i = 3$.
- If $P_i = 3$, the bag contains at least one ball just before the i-th operation.

Entrada e saída

Input

Input is given from Standard Input in the following format:

Q
query_1
query_2
$\vdots$
query_Q

Entrada e saída

Each query$_i$ in the 2-nd through $(Q+1)$-th lines is in the following format:

```
1 X_i
2 X_i
3
```

The first number in each line is $1 \leq P_i \leq 3$, representing the type of the operation. If $P_i = 1$ or $P_i = 2$, it is followed by a space, and then by X_i.

Output

For each operation with $P_i = 3$ among the Q operations, print the recorded integer in its own line.

Exemplo de entradas e saídas

Exemplo de Entrada

```
5
1 3
1 5
3
2 2
3
```

Exemplo de Saída

```
3
7
```

Solução com complexidade $O(Q \log Q)$

- O problema descreve as três operações fundamentais de um *venice set*: adição de um elemento, somar um valor constante a todos os elementos do conjunto e a extração do menor elemento
- Em um *venice set*, a soma de constante tem complexidade $O(1)$ e a adição e remoção de elementos tem complexidade $O(\log N)$, onde N é o número de elementos armazenados no conjunto
- O *venice set* mantém um delta que abstrai o nível da água na cidade de Veneza
- A ideia é que, para adicionar um andar um andar em todos os prédios, basta reduzir o nível da água em um andar
- É preciso utilizar o tipo `long long` para evitar erros de *overflow*

Solução com complexidade $O(Q \log Q)$

```cpp
#include <bits/stdc++.h>

using namespace std;

template<typename T>
struct VeniceSet
{
    multiset<T> s;
    T delta = 0;

    void emplace(T x) { s.emplace(x + delta); }
    void add_all(T x) { delta -= x; }

    T pop()
    {
        auto m = *s.begin() - delta;
        s.erase(s.begin());
        return m;
    }
};
```

Solução com complexidade $O(Q \log Q)$

```
22 int main()
23 {
24     ios::sync_with_stdio(false);
25
26     int Q;
27     cin >> Q;
28
29     VeniceSet<long long> s;
30
31     while (Q--)
32     {
33         int cmd;
34         long long X;
35
36         cin >> cmd;
```

Solução com complexidade $O(Q \log Q)$

```
38          if (cmd == 1)
39          {
40              cin >> X;
41              s.emplace(X);
42          } else if (cmd == 2)
43          {
44              cin >> X;
45              s.add_all(X);
46          } else
47              cout << s.pop() << '\n';
48      }
49
50      return 0;
51  }
```