

Course Project

Deadline: Tir 2nd 11:59 pm

The goal of this project is to develop a system that maintains and processes patients' information. For each patient we have a unique patient ID which is a positive integer (between 0 and 10^9), and a health measure which is an integer (between -10^9 and 10^9). Design and implement a data structure that supports the following queries:

- Add X Y: This operation inserts a patient with patient ID X and health measure Y to the data structure. This operation is called upon the arrival of a patient.
- Serve First: This operation serves the patient in the data structure that arrived the earliest, and removes her from the dataset.
- Serve Sickest: This operation serves the patient in the data structure that has the lowest health measure and removes her from the dataset.
- Update X Y: This operation updates the health measure of the patient X to value Y.

The running time of each operation should be either $O(\log(n))$ in the worst case, or $O(1)$ in expectation. Note that you are supposed to implement any data structure that you need yourself. You are not allowed to use data structures from the standard library of your programming language or from the internet.

The input is a list of queries, with one query per line. Whenever a patient is served (either through Serve First or Serve Sickest), write the patient's ID followed by a space and the patient's health measure at the serving time, and move to the next line.

The queries are set such that the number of patients in the data structure is always less than or equal to 10^6 . However, the total number of patients that are inserted to the data structure and served is not limited.

Hint 1: you need to use multiple data structures.

Hint 2: you do not need red-black tree.

Programming language: Python/C++

Sample Input 1:

```
Add 10 7
Add 4 -3
Serve First
Serve First
```

Sample Output 1:

```
10 7
4 -3
```

Sample Input 2:

```
Add 10 7
Add 4 -3
Serve Sickest
```

Serve Sickest

Sample Output 2:

4 -3

10 7

Sample Input 3:

Add 10 7

Add 4 -3

Add 7 13

Update 4 15

Serve First

Serve First

Serve First

Sample Output 3:

10 7

4 15

7 13

Sample Input 4:

Add 10 7

Add 4 -3

Add 7 13

Add 5 4

Update 7 -8

Serve Sickest

Serve Sickest

Serve First

Serve First

Sample Output 4:

7 -8

4 -3

10 7

5 4