Assignment #6

Sorting/Binary Search Tree
Problem 2. Inventory Management

Score: 60 points

Due: Dec 11, 2020

Your friend has opened a new shop to buy and sell diamonds. Diamonds have various grades, and the smaller the grade number, the higher the value.

Your friend has bought and sold a large amounts of diamonds as the diamond business went well, and found managing the inventory is a difficult job. You want to help your friend by writing a inventory management program.

The inventory management program supports following five operations.

- Insert(n, m): Add m diamonds of grade n to the inventory.
- Delete(n, m): Delete m diamonds of grade n from the inventory.
- Count(n): Check how many grade n diamonds are left in stock.
- RangeCount(n, m): Check how many diamonds are in stock with grades between n and m (both inclusive).
- Top(n): Get the grade of the n-th highest valuable diamond.

Input

The first line contains a single integers N ($1 \le N \le 10^6$) — the number of operations.

Each of the next N lines describes one of the following operations, given a character and one or two integers.

- `I n m`: Insert(n, m). ($1 \le n \le 10^7$, $1 \le m \le 1,000$)
- `D n m`: Delete(n, m). ($1 \le n \le 10^7$, $1 \le m \le 1,000$)
- `C n`: Count(n). ($1 \le n \le 10^7$)
- `R n m`: RangeCount(n, m). ($1 \le n \le m \le 10^7$)
- `T n`: Top(n). ($1 \le n$)

Some test cases do not include Range and Top operations, so you can get partial scores even though you don't implement those operations.

Note: The grade numbers of input test cases are randomly distributed. Thus, if you implement using binary search tree, you don't have to consider the tree skew problem.

Output

For each Count, RangeCount and Top operation, print the single line with the returned value of each operation. Print `-1` for Top(n), if the number of diamonds left is less than n.

Sample Inputs

Sample Input 1

```
9
I 5 1
I 1 3
D 1 1
I 4 1
C 5
I 5 4
C 5
D 5 5
C 5
```

Sample Output 1

```
1
5
0
```

Sample Input 2

```
11
I 9 10
I 25 20
I 20 7
I 1 1
I 12 6
R 5 15
T 30
D 9 3
R 5 15
T 20
T 50
```

Sample Output 2

```
16
25
13
20
-1
```

Grading Policy

Submissions will be graded based on the number of test cases passed. There are execution time and memory limitations, so be sure to write efficient code. Submit your code on goormEDU (`https://skku.goorm.io`).

For late submission, grades will be deducted by 25% per late day. That is, after 4 days, the grade will be zero.

The assignment should be done by yourself. We use plagiarism detection tool on all submissions.

If you have any questions, leave them in the spreadsheet or send an email to the TA. Please do not use i-Campus message. (TA Youngjae Lee: `yjlee4154@gmail.com`)

Q&A Spreadsheet Link
`https://docs.google.com/spreadsheets/d/1RDbEqFMS1FUO-KHcqX3BvhMHNJIOY-wpOPXAmE-zdQU/edit?usp=sharing`