

## Assignment on Red Black Tree

A super-computer named TEUB has many running processes. Each process has a priority  $x$ . Two active processes in TEUB cannot have equal priorities. When a process finishes its task, it gets terminated. Sometimes, the programmers of TEUB want to know how many running processes have priorities less than  $y$ . The programmers hired you to help them using Red-Black tree.

The input has four types of commands.

- Initiation of a program
- Termination of a program
- Searching for a program
- Find the programs with less priority

### Input

First line of input shows the total number of commands (N).

Each of the following N commands, has two integers  $e_i$  and  $x_i$ .

| $e_i$ | Meaning   |
|-------|---|
| 0     | Terminate the program with priority $x_i$ .                 |
| 1     | Initiate a program with priority $x_i$ .                    |
| 2     | Search the program with priority $x_i$ .                    |
| 3     | Find the number of programs with priority less than $x_i$ . |

### Output

First line of input shows the number of output lines.

For each command, you have to print three integers  $e_i$ ,  $x_i$  and  $r_i$ .

$r_i$  signifies the result of the corresponding command.

| $e_i$ | $r_i$   |
|-------|---|
| 0     | 1 if successful termination.<br>0 if there is no program with priority $x_i$      |
| 1     | 1 if successful initiation<br>0 if there is already a program with priority $x_i$ |
| 2     | 1 if found<br>0 if not found  |

|   |  |
|---|--|
| 3 | The number of programs with priority $< x_i$ . |
|---|--|

### Sample I/O

| Sample Input | Sample Output | Explanation                      |
|--------------|---------------|----------------------------------|
| 1 1          | 1 1           | Line count                       |
| 1 1          | 1 1 1         | Successful initiation 1          |
| 1 2          | 1 2 1         | Successful initiation 2          |
| 1 3          | 1 3 1         | Successful initiation 3          |
| 1 1          | 1 1 0         | Same priority (1) exists         |
| 0 1          | 0 1 1         | Successful termination 1         |
| 0 4          | 0 4 0         | No priority (4) exists           |
| 2 3          | 2 3 1         | Priority 3 found                 |
| 2 5          | 2 5 0         | Priority 5 not found             |
| 1 1          | 1 1 1         | Successful initiation 1          |
| 3 3          | 3 3 2         | 2 programs having priority $< 3$ |
| 3 6          | 3 6 3         | 3 programs having priority $< 6$ |

### Constraints

$$1 \leq N \leq 10^5$$

$$1 \leq x_i \leq 10^6$$

$$0 \leq e_i \leq 3$$

Each of the commands has to be answered in logarithmic time.

### More instructions

- Write the program in such a way to accept input from file
- Write Red-Black tree codes in such a way that it can be reused for other tasks during online evaluation.

### Submission

- Include only source files
- Do not include executable binaries, input/output files
- Place your files in a folder named 1905XXX
- Zip the folder
- Submit to Moodle after renaming it to 1905XXX.zip