Splay Trees 1

locked

by maruthisarat

Implement a splay tree, using following operations.

1. **INSERT k** :
   - If only one tree t1 is available, insert k into t1.
   - If two trees are available, and if k < threshold, then insert into t1 otherwise into t2.
   - Splay the corresponding tree after every insertion.
2. **SEARCH k** :
   - Search for k, if found, splay with k and print "YES<space>G", where G = number of keys > k in the corresponding tree.
   - If not found, do not splay, print "NO" in a new line.
3. **SPLIT k**:
   - Splay the tree with k, split resulting tree into two parts.
   - First part is t1, contains root without right subtree.
   - Second part is t2, contains root's right subtree (can also be empty).
   - Store k as "threshold" for INSERT, JOIN operation until both trees get merged.
   - It is guaranteed that when this command is given, only one tree i.e. t1 will be available.
4. **JOIN** :
   - Splay t1 with the stored threshold, add t2 as t1's right child.
   - It is guaranteed that when this command is given, two trees will be available.
5. **PRE-ORDER** :
   - Print triplet (key<space>left_subtree_size<space>right_subtree_size) in preorder of resulting tree in a new line.
   - It is guaranteed that when this command is given, only one tree i.e. t1 will be available.

Note :

- Test cases 0-16, are private test cases for top-down approach. (60 points)
- Test cases 17-33, are private test cases for bottom-up approach. (60 points)
- Test cases 34-36, are public test cases for top-down approach.
- Test cases 37-39, are public test cases for bottom-up approach.
- Assignemnt is set to 120 points. You have to score 60 points in order to complete this assignment.

Input Format

First line contains N denots number of commands, Next N lines, each line contains one of the 5 commands.

Constraints

- 1 <= N <= 10^4

- 1 <= k <= 10^6

Output Format

- Search
  - If found, splay and print "YES<space>G", where G = number of keys > k in the corresponding tree.
  - If not found, "NO".
- PRE-ORDER
  - Print triplet (key<space>left_subtree_size<space>right_subtree_size) in preorder of resulting tree in a new line.

Note: - Sample output 0, 1, 2 are for top-down approach. - Sample output 3, 4, 5 are for bottom-up approach.

Sample Input 0

```
10
INSERT 838014
INSERT 889246
INSERT 658449
INSERT 517413
INSERT 251640
INSERT 81455
INSERT 907169
INSERT 330137
INSERT 673787
PRE-ORDER
```

Sample Output 0

```
(673787 5 3) (330137 2 2) (251640 1 0) (81455 0 0) (658449 1 0) (517413 0 0) (907169 2 0) (838014 0 1) (889246 0 0)
```

Sample Input 1

```
20
INSERT 33759
INSERT 170151
INSERT 477309
SEARCH 170151
INSERT 558177
INSERT 510617
INSERT 249253
INSERT 612180
SEARCH 33759
INSERT 339838
SEARCH 510617
INSERT 513848
SEARCH 249253
SEARCH 33759
INSERT 102043
SEARCH 249253
INSERT 515452
SEARCH 170151
INSERT 925931
PRE-ORDER
```

Sample Output 1

```
YES 1
YES 6
YES 2
YES 6
YES 8
YES 6
YES 8
(925931 11 0) (510617 6 4) (170151 2 3) (102043 1 0) (33759 0 0) (249253 0 2) (339838 0 1) (477309 0 0) (558177 2 1) (515452 1 0)
(513848 0 0) (612180 0 0)
```

Sample Input 2

```
50
INSERT 139734
INSERT 634606
```

```
INSERT 990979
SEARCH 634606
SEARCH 634606
INSERT 35751
INSERT 227155
SEARCH 139734
INSERT 97929
SEARCH 97929
SPLIT 139734
SEARCH 634606
INSERT 424180
SEARCH 424180
INSERT 733433
INSERT 900632
INSERT 921768
SEARCH 921768
INSERT 960032
SEARCH 921768
INSERT 442217
SEARCH 442217
SEARCH 634606
INSERT 776756
INSERT 331817
INSERT 487403
SEARCH 776756
INSERT 838279
INSERT 459680
INSERT 705135
JOIN
SEARCH 705135
SEARCH 442217
INSERT 691347
INSERT 89556
INSERT 207701
INSERT 855843
INSERT 790013
INSERT 373730
INSERT 85236
SPLIT 97929
INSERT 496507
SEARCH 97929
INSERT 996857
SEARCH 89556
SEARCH 35751
INSERT 350255
INSERT 76498
JOIN
PRE-ORDER
```

Sample Output 2

```
YES 1
YES 1
YES 3
YES 4
YES 1
YES 2
YES 1
YES 2
YES 6
YES 5
YES 4
YES 7
YES 11
YES 0
YES 1
YES 3
(97929 4 24) (85236 2 1) (76498 1 0) (35751 0 0) (89556 0 0) (350255 4 19) (331817 3 0) (227155 2 0) (207701 1 0) (139734 0 0)
(790013 11 7) (373730 0 10) (496507 4 5) (442217 1 2) (424180 0 0) (459680 0 1) (487403 0 0) (691347 1 3) (634606 0 0) (705135 0
2) (776756 1 0) (733433 0 0) (996857 6 0) (900632 2 3) (855843 1 0) (838279 0 0) (960032 1 1) (921768 0 0) (990979 0 0)
```

Sample Input 3

```
10
INSERT 838014
INSERT 889246
INSERT 658449
INSERT 517413
INSERT 251640
INSERT 81455
INSERT 907169
INSERT 330137
```

```
INSERT 673787
PRE-ORDER
```

Sample Output 3

```
(673787 5 3) (330137 2 2) (251640 1 0) (81455 0 0) (658449 1 0) (517413 0 0) (907169 2 0) (838014 0 1) (889246 0 0)
```

Sample Input 4

```
20
INSERT 33759
INSERT 170151
INSERT 477309
SEARCH 170151
INSERT 558177
INSERT 510617
INSERT 249253
INSERT 612180
SEARCH 33759
INSERT 339838
SEARCH 510617
INSERT 513848
SEARCH 249253
SEARCH 33759
INSERT 102043
SEARCH 249253
INSERT 515452
SEARCH 170151
INSERT 925931
PRE-ORDER
```

Sample Output 4

```
YES 1
YES 6
YES 2
YES 6
YES 8
YES 6
YES 8
(925931 11 0) (170151 2 8) (102043 1 0) (33759 0 0) (515452 5 2) (249253 0 4) (513848 3 0) (339838 0 2) (510617 1 0) (477309 0 0)
(612180 1 0) (558177 0 0)
```

Sample Input 5

```
50
INSERT 139734
INSERT 634606
INSERT 990979
SEARCH 634606
SEARCH 634606
INSERT 35751
INSERT 227155
SEARCH 139734
INSERT 97929
SEARCH 97929
SPLIT 139734
SEARCH 634606
INSERT 424180
SEARCH 424180
INSERT 733433
INSERT 900632
INSERT 921768
SEARCH 921768
INSERT 960032
SEARCH 921768
INSERT 442217
SEARCH 442217
SEARCH 634606
INSERT 776756
INSERT 331817
INSERT 487403
SEARCH 776756
INSERT 838279
INSERT 459680
INSERT 705135
JOIN
SEARCH 705135
SEARCH 442217
```

```
INSERT 691347
INSERT 89556
INSERT 207701
INSERT 855843
INSERT 790013
INSERT 373730
INSERT 85236
SPLIT 97929
INSERT 496507
SEARCH 97929
INSERT 996857
SEARCH 89556
SEARCH 35751
INSERT 350255
INSERT 76498
JOIN
PRE-ORDER
```

Sample Output 5

```
YES 1
YES 1
YES 3
YES 4
YES 1
YES 2
YES 1
YES 2
YES 6
YES 5
YES 4
YES 7
YES 11
YES 0
YES 1
YES 3
(97929 4 24) (76498 1 2) (35751 0 0) (89556 1 0) (85236 0 0) (350255 4 19) (207701 1 2) (139734 0 0) (331817 1 0) (227155 0 0)
(996857 18 0) (496507 5 12) (373730 0 4) (459680 2 1) (442217 1 0) (424180 0 0) (487403 0 0) (691347 1 10) (634606 0 0) (855843 5
4) (790013 3 1) (776756 2 0) (705135 0 1) (733433 0 0) (838279 0 0) (921768 1 2) (900632 0 0) (990979 1 0) (960032 0 0)
```

Submissions: 64
Max Score: 120
Difficulty: Medium

Rate This Challenge:

More

Current Buffer (saved locally, editable) C++

```cpp
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;


int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code