# HW2

2021.11.01

助教

吳承翰

# Announcements

- 作業的繳交格式請參照第一週發布的Homework rules and Online judge guide
- 盡量不要看其他人或網路上的code，至少不要讓TA們覺得有抄襲
- 原則上STL模板是禁止使用的，若可使用會在題目中提及
- 請確定OJ最後上傳的程式碼是您所要交的程式碼
- 跟題目相關的問題可以盡量問，但請不要跟TA們提奇怪的要求

# HW0

Deadline: 11/10 08:00 AM

| | | | | | |
|---|---|---|---|---|---|
| ✓ | HW2-1… | 瓦基 the gardener | Low | 14 | 21.43% |
| ✓ | HW2-2… | Students' score managment system | Low | 6 | 66.67% |

# HW2-1: 瓦基 the gardener🌳🪚 (5pts)

## Description

Thanks to your help, 瓦基 has found the shortest path out of 魯拉拉's house. However, when 瓦基 tries to go through 魯拉拉's garden, he is hindered by the overgrown Binary Search Trees.瓦基 must prune the BSTs ASAP so as to escape from 魯拉拉's house keepers. There are 5 operations that 瓦基 can do:

1. Insertion: 瓦基 inserts a number, $I$, to the BST. Smaller numbers will be on the left. If there are duplicated numbers, insert the latter one to the left subtree of the former one.
2. Deletion: 瓦基 deletes a number, $D$, from the BST. If $D$ has both left and right child nodes, find the greatest number in the left subtree to replace $D$. If $D$ has only one child node, link the child node to its parent node. If $D$ does not have any child node, simply eliminate $D$ . If duplicated $D$s are found in the tree, eliminate the one nearest to the root.
3. Print: 瓦基 traverses the BST in preorder. The traversal should be noted in format root(left__child_node()())(right_child_node()()).
4. remove value with Least node: 瓦基 wants to make the total value of the BST at most $V$, please return the least number of nodes he needs to remove.
5. remove value with Most node: 瓦基 wants to make the total value of the BST at least $V$, please return the most number of nodes he needs to remove.

Please help 瓦基 estimate the result of certain commands.

## Input

Each test case contains one BST only. The first line contains a positive integer $N$, the number of nodes of the initial BST, where $1 \leq N \leq 50000$ . The second line contains $N$ integers $a_1, a_2, ..., a_n$, which is the preorder list of the value of nodes in the BST. Then, the test case contains $M$ lines of commands, where $1 \leq M \leq 1000$.

1. I: The insertion command. The command is followed by an integer $input$, the desired number to be inserted into the BST. Please insert $input$ into the BST.
2. D: The deletion command. The command is followed by an integer $input$, the desired number to be deleted from the BST. Please delete $input$ from the BST.
3. P: The print command. Please print the BST in the format root(left_child_node()())(right_child_node()()).
4. L: The command estimates the least number of nodes that need to be removed. The command is followed by an integer $V_max$, the desired maximum total value of the tree. Please output the least number of nodes that need to be removed to make the sum of the tree at most $V$.
5. M: The command estimates the most number of nodes that can be removed. The command is followed by an integer $V_min$, the desired minimum total value of the tree. Please output the most number of nodes that can be removed to make the sum of the tree at least $V$.

Please note that the number of $M$ will not be given, so the program should stop automatically upon EOF is read.

## Output

Please print the results of the corresponding command line by line.

# HW2-1: 瓦基 the gardener🌳🪚 (5pts)

## Sample Input 1

```
7
39 27 45 18 29 40 54
P
I 21
I 46
P
D 27
P
D 45
P
D 40
P
```

## Sample Output 1

```
39(27(18()())(29()()))(45(40()())(54()()))
39(27(18()(21()()))(29()()))(45(40()())(54(46()())())))
39(21(18()())(29()()))(45(40()())(54(46()())()))
39(21(18()())(29()()))(40()(54(46()())()))
39(21(18()())(29()()))(54(46()())())
```

## Sample Input 2

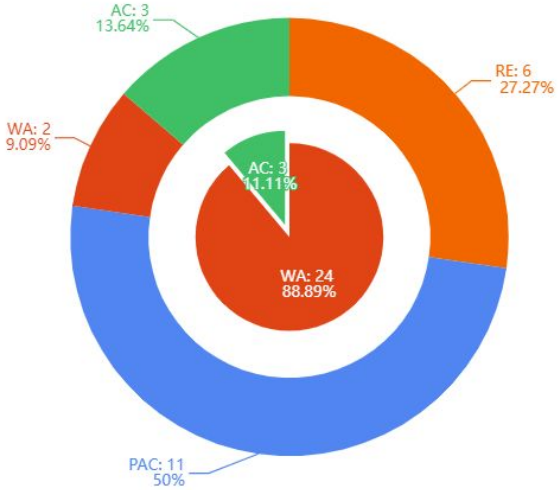
```
7
39 21 45 18 29 40 54
I 27
I 46
L 273
L 219
L 218
M 301
M 250
M 1
```

## Sample Output 2

```
1
2
3
1
3
8
```

# HW2-1: 瓦基 the gardener🌳🪚 (5pts)

- (2pts) support commands I, D, P.
- (2pts) support commands L, M.
- (1pts) support all commands and work in good efficiency.
    - HINT: 節點中除了該點的值還能放其他資料



# HW2-2: Students' score management system💯 (3pts)

## Description

The TAs have finally done correcting all the midterm exam papers, only one problem remains. Since Data Structure is such an intriguing subject that everyone wants to study, there are a great deal of students taking this course. Managing the students' score thus becomes a burden. Fortunately, Hank designs a system to manage the scores with AVL tree. The system includes the following functions.

1. Insertion: insert a number, $I$, to the AVL tree. Smaller numbers will be on the left. If there are duplicated numbers, insert the latter one to the left subtree of the former one.
2. Deletion: delete a number, $D$, from the AVL tree. If $D$ has both left and right child nodes, find the greatest number in the left subtree to replace $D$. If $D$ has only one child node, li nk the child node to its parent node. If $D$ does not have any child node, simply eliminate $D$. If duplicated $D$s are found in the tree, eliminate the one nearest to the root.
3. Print in preorder: traverse the AVL tree in preorder. The traversal should be noted in format root(left__child_node()())(right_child_node()()).
4. Print in inorder: traverse the AVL tree in inorder. The traversal should be noted in format (()left__child_node())root(()right_child_node()).
5. Print in postorder: traverse the AVL tree in postorder. The traversal should be noted in format (()()left__child_node)(()()right_child_node)root.

Please help Hank implement the system.

## Input

Each test case contains one AVL tree only. The first line contains a positive integer $N$, the number of nodes of the initial AVL tree, where $1 \leq N \leq 50000$. The second line contains $N$ integers $a_1, a_2, ..., a_n$, which is the ordered list of the value of nodes in the AVL tree. Then, the test case contains $M$ lines of commands, where $1 \leq M \leq 1000$.

1. I: The insertion command. The command is followed by an integer $input$, the desired number to be inserted into the AVL tree. Please insert $input$ into the AVL tree.
2. D: The deletion command. The command is followed by an integer $input$ the desired number to be deleted from the AVL tree. Please delete $input$ from the AVL tree.
3. P: The print command. The command is followed by an integer $input/in 1, 2, 3$, referring to preorder, inorder, and postorder traversal respectively. Please print the AVL tree in th e corresponding format.

Please note that the number of $M$ will not be given, so the program should stop automatically upon EOF is read.

## Output

Please print the results of the corresponding command line by line.

# HW2-2: Students' score management system💯 (3pts)

## Sample Input 1

```
8
63 9 19 27 18 108 99 81
I 110
I 5
P 1
```

## Sample Output 1

```
19(9(5()())(18()()))(99(63(27()())(81()()))(108()(110()()))))
```

## Sample Input 2

```
8
63 9 19 27 18 108 99 81
P 2
I 110
D 99
P 2
```

## Sample Output 2

```
(()9(()18()))19(((()27())63(((()81())99(()108())))
(()9(()18()))19((((()27())63())81(()108(()110())))
```

## Sample Input 3

```
8
63 9 19 27 18 108 99 81
P 3
D 18
I 100
P 3
```

## Sample Output 3

```
((()(()()18)9)((((()27)(((()()81)(()()108)99)63)19
((((()9)(()()27)19)((((()81)((((()100)()108)99)63
```

# HW2-2: Students' score management system💯 (3pts)

- (1pts) support insertion, deletion, and preorder traversal.
- (1pts) support insertion, deletion, and inorder traversal.
- (1pts) support insertion, deletion, and postorder traversal.