**CS1020E | Lab 5 | Exercise 1**

**Binary Search Tree**

**Objective**

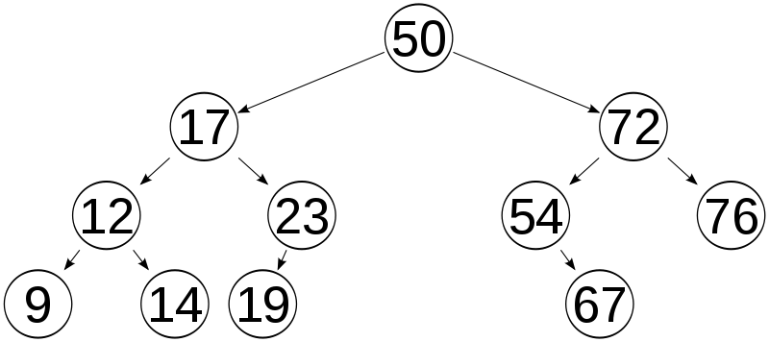
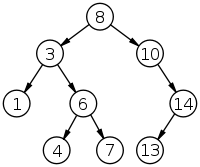
The objective of this exercise is to learn how to **manipulate simple non-linear linked structures**.

**Problem Description**

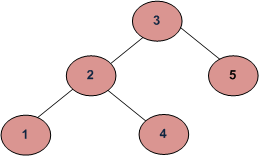
A *Binary Search Tree* is a data structure for maintaining a list of integers, where each node in the tree consists of a single integer, has at most two children, and holds the following condition:

The integer in the node must be **greater than or equal** to every integer in the *left subtree* of this node, and must be **smaller than** every integer in the *right subtree* of this node.

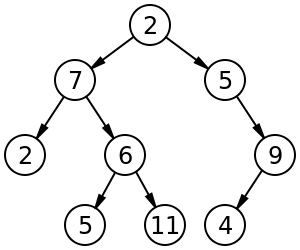
For example, these two trees are binary search trees:



while these two trees are NOT binary search tree:



(Because 4 ≥ 3; 4 should not be in the left subtree of 3.)



(This is obviously not a binary search tree.)

There will be *N* queries given to you. Each query will be one of the following types:

1. **Insert an integer** to the binary tree. You must insert a new element at the *bottom of the tree*, and insert it at the correct position such that the conditions of a binary search tree must be fulfilled all the time.

2. **Find the minimum** integer in the binary tree, or return 0 if the tree is empty.

3. **Find the maximum** integer in the binary tree, or return 0 if the tree is empty.

4. **Search an integer** in the binary tree. You should output the *depth* of the node nearest to the *root* that has that integer inside, or output 0 if the integer is not in the tree. The root has depth = 1. The root is the topmost node in the tree.

**Add your code only to the parts of the files indicated. Do not modify any other part of the given code, and do not add new files.**

**Inputs**

The first line consists of an integer *N*, which indicate the number of queries.

*N* lines follow. Each line represents a query, and is in one of the following formats:

* **INSERT *<integer>*** indicates the query of the first type.
* **FINDMIN** indicates the query of the second type.
* **FINDMAX** indicates the query of the third type.
* **FIND *<integer>*** indicates the query of the fourth type.

*N* will be between 1 and 3000 inclusive. Each integer in the query will be between 1 and 1000000 inclusive

**Outputs**

For each FIND, FINDMIN, FINDMAX query, you have to output an integer as explained above.

**Sample Input**

8

INSERT 3

INSERT 1

INSERT 2

FINDMIN

FIND 4

INSERT 4

FINDMAX

FIND 4

**Sample Output**

1

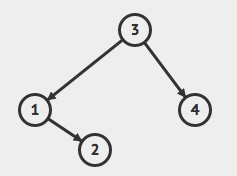
0

4

2

**Explanation**

The tree after all the queries will look like this:



It can be seen that 4 has depth = 2.

**Submission**

You need to submit **ALL** your completed skeleton **\*.cpp** and **\*.h** files to CodeCrunch (<https://codecrunch.comp.nus.edu.sg/>) before the specified deadline. We will take only your latest submission.

Late submissions will not be accepted. The submission system in CodeCrunch will automatically close at the deadline.