

- [主页](#)
- [题目集](#)
 - [基本信息](#)
 - [题目列表](#)
 - [提交列表](#)
 - [排名](#)
- [帮助](#)

1143. Lowest Common Ancestor (30)

时间限制

200 ms

内存限制

65536 kB

代码长度限制

16000 B

判题程序

Standard

作者

CHEN, Yue

The lowest common ancestor (LCA) of two nodes U and V in a tree is the deepest node that has both U and V as descendants.

A binary search tree (BST) is recursively defined as a binary tree which has the following properties:

- The left subtree of a node contains only nodes with keys less than the node's key.
- The right subtree of a node contains only nodes with keys greater than or equal to the node's key.
- Both the left and right subtrees must also be binary search trees.

Given any two nodes in a BST, you are supposed to find their LCA.

Input Specification:

Each input file contains one test case. For each case, the first line gives two positive integers: M (≤ 1000), the number of pairs of nodes to be tested; and N (≤ 10000), the number of keys in the BST, respectively. In the second line, N

distinct integers are given as the preorder traversal sequence of the BST. Then M lines follow, each contains a pair of integer keys U and V. All the keys are in the range of **int**.

Output Specification:

For each given pair of U and V, print in a line "LCA of U and V is A." if the LCA is found and A is the key. But if A is one of U and V, print "X is an ancestor of Y." where X is A and Y is the other node. If U or V is not found in the BST, print in a line "ERROR: U is not found." or "ERROR: V is not found." or "ERROR: U and V are not found.".

Sample Input:

```
6 8
6 3 1 2 5 4 8 7
2 5
8 7
1 9
12 -3
0 8
99 99
```

Sample Output:

```
LCA of 2 and 5 is 3.
8 is an ancestor of 7.
ERROR: 9 is not found.
ERROR: 12 and -3 are not found.
ERROR: 0 is not found.
ERROR: 99 and 99 are not found.
```

[提交代码](#)
