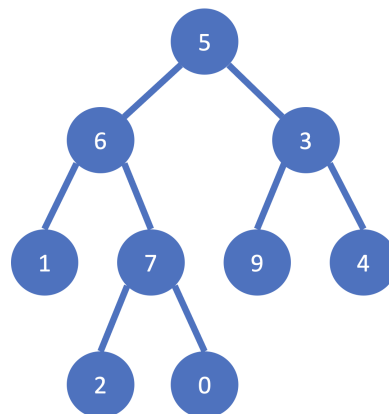


## Lowest common ancestor

**P49809\_en**

The lowest common ancestor (LCA) of two nodes  $x$  and  $y$  in a tree is the lowest (i.e. deepest) node that has both  $x$  and  $y$  as descendants, where we define each node to be a descendant of itself.

For instance, in the following tree, 5 is the LCA of 1 and 9, and 6 is the LCA of 1 and 0:



Write a function `Tree lowest_common_ancestor (Tree t, int x, int y)`; that returns the node that corresponds to the LCA of  $x$  and  $y$  in a binary tree of integers. You can assume that  $t$  contains both  $x$  and  $y$  and that  $t$  does not contain repeated elements.

Most of the program is already written for you. Download it! It reads several trees in pre-order with leaves marked with  $-1$  and, for each of these, reads several pairs of values and prints their LCA. You just have to specify and implement the `lowest_common_ancestor()` function (and other helper functions, should you need them). Also, write a comment with the time efficiency of your algorithm.

### Sample input

```

2
5 6 1 -1 -1 7 2 -1 -1 0 -1 -1 3 9 -1 -1 4 -1 -1

  1 9
  1 0
  6 3
  3 6
  5 5
  3 3
  5 0
  -1 -1

5 2 3 -1 -1 8 -1 -1 -1

  3 8
  3 2
  
```

```
3 5
2 5
8 5
-1 -1
```

### Sample output

```
5
6
5
5
5
3
5

2
2
5
5
5
```

### Problem information

Author : Jordi Petit

Generation : 2025-05-13 11:19:13

© *Jutge.org*, 2006–2025.

<https://jutge.org>