Assignment 5

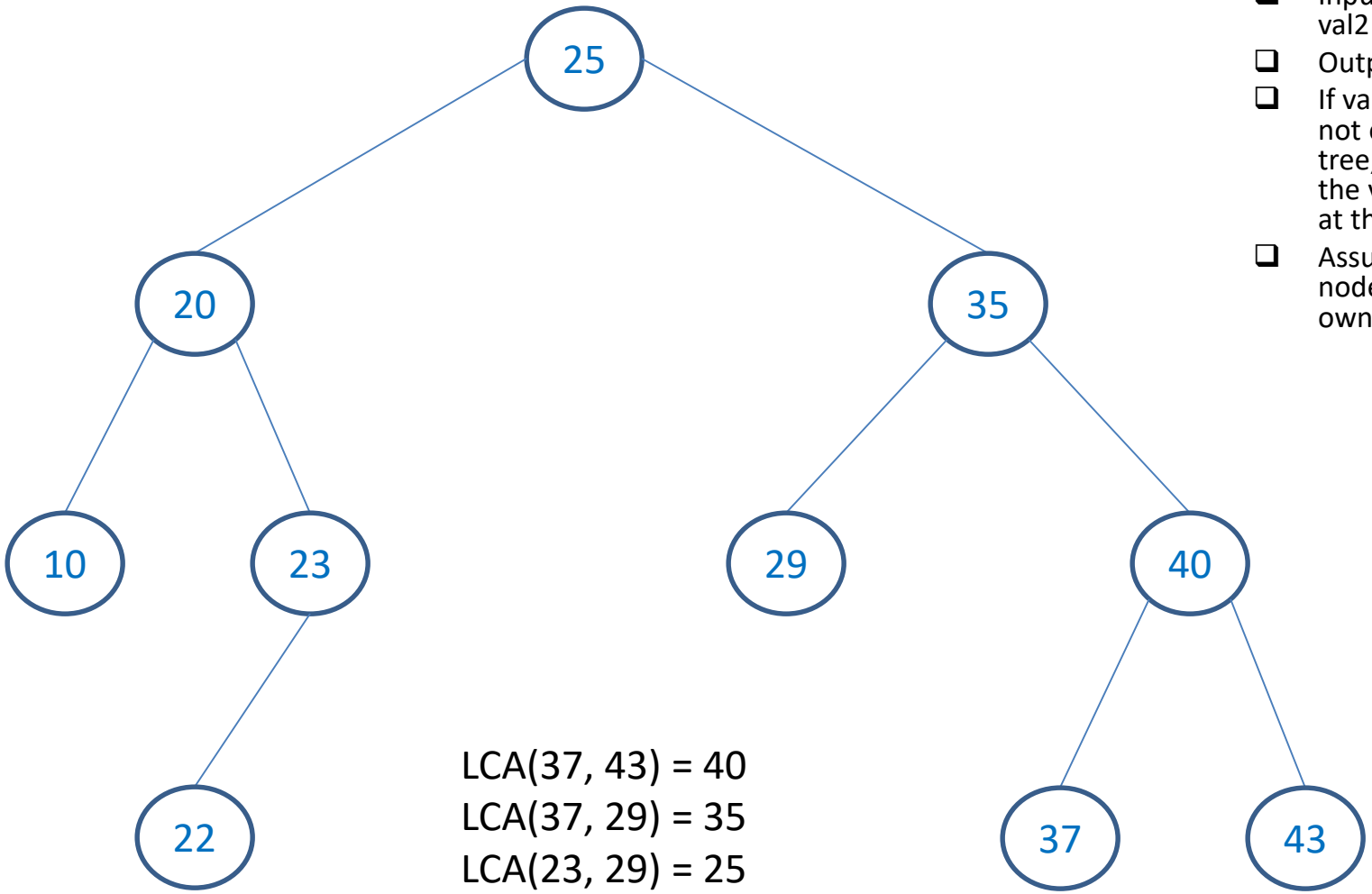
A5_ROLLNO.c/cpp

pds2016autumn@gmail.com

Binary Search Tree

- Given the preorder traversal, construct the binary search tree.
  - Merge Sort to get the inorder traversal, and construct the binary search tree.
  - Recursive postorder traversal

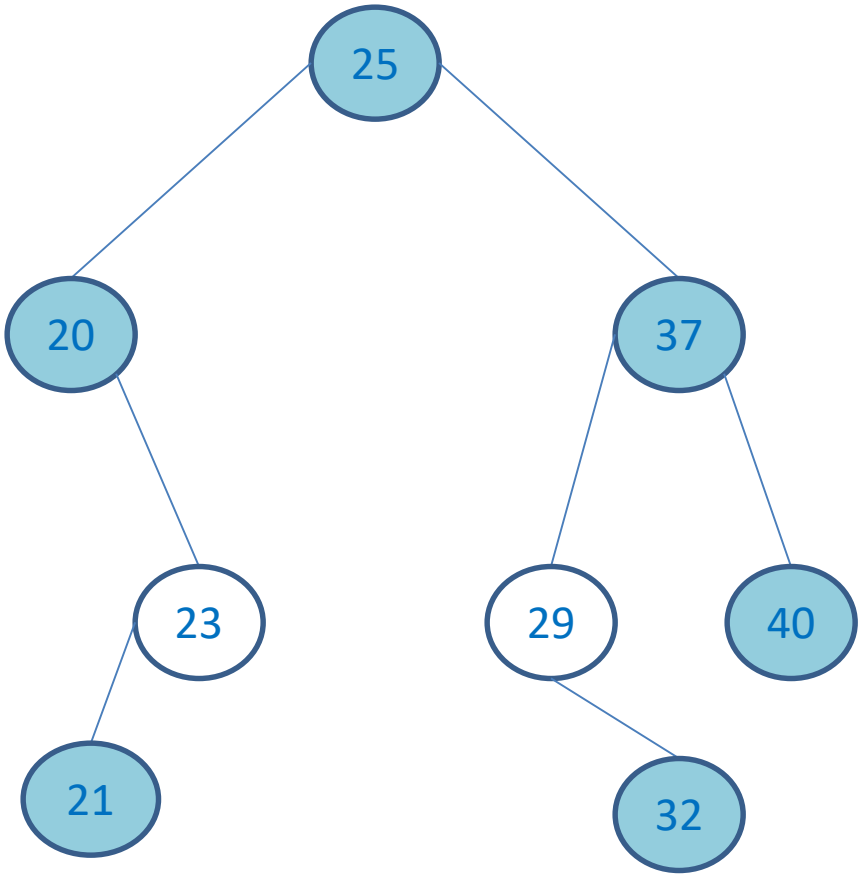
Lowest Common Ancestor



- Input: Root, val1, val2
- Output: val / -1
- If val1/val2 does not exist in the tree, or one of the value is value at the root
- Assumption: A node can't be its own ancestor

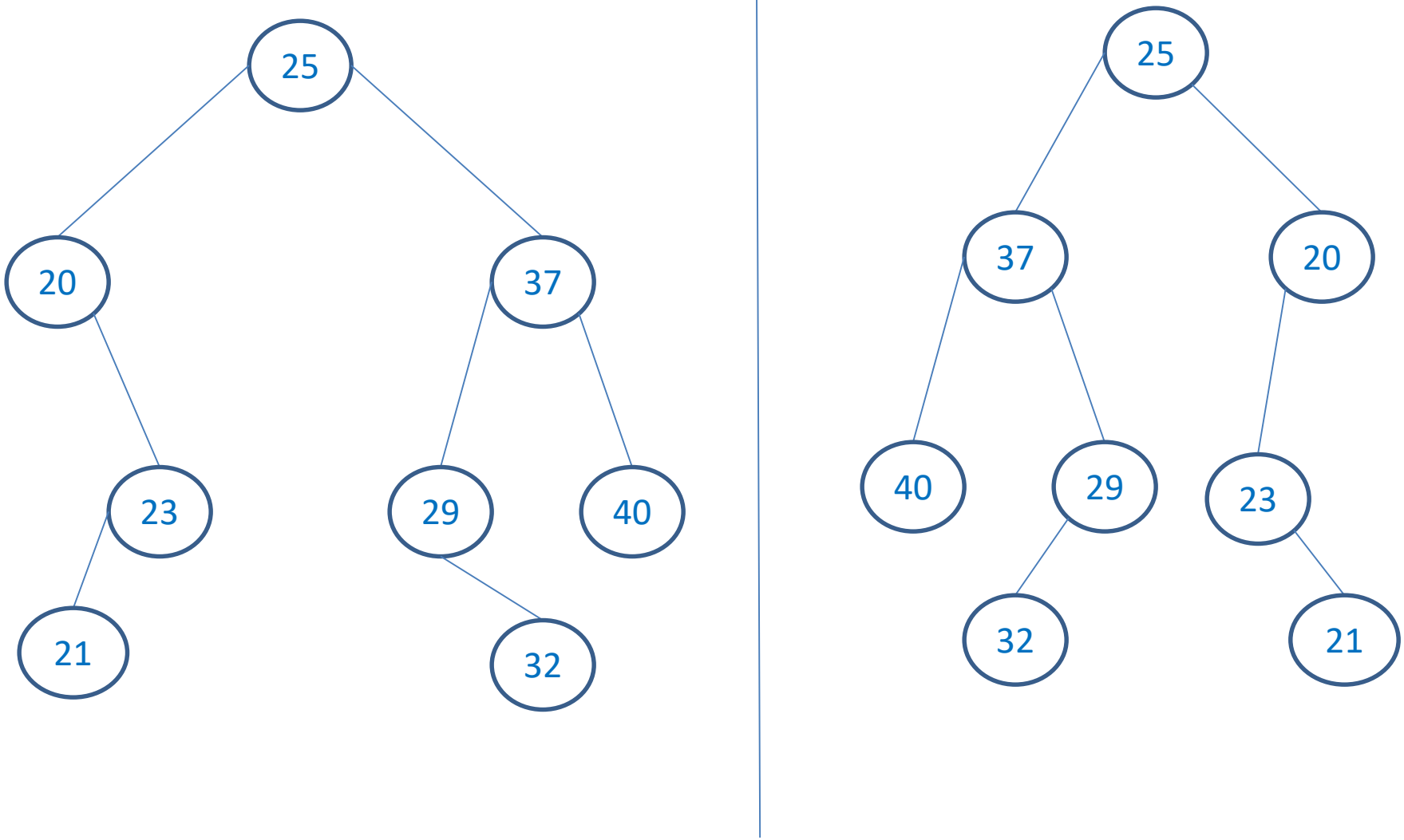
LCA(37, 43) = 40
LCA(37, 29) = 35
LCA(23, 29) = 25
LCA(22, 20) = 25
LCA(25, 29) = -1

Border Traversal



- Print data in the border nodes in anti-clockwise fashion
  - Left border top to bottom
  - Right border bottom to top
  - Leaf nodes from left to right
  - No node should be repeated
  - 25 20 21 32 40 37

Mirror Tree of Binary Tree



Counting Inversions

- Inversion: If i < j and A[i] > A[j], then (i, j) is called an inversion.
- Given an array A of n integers, write a C/C++ program that counts the number of inversions in A.
- Example:
- A[] = {1, 1, 3, 5} Output: 0
- A[] = {10, 30, 20} Output: 1
- A[] = {6, 4, 3, 2} Output: 6
- A[] = {6, 3, 4, 2} Output: 5
- Modify the merge sort routine to count the number of inversions in O(n log n) time.