Data Structure

Lab Session #6: Binary Search Trees

U Kang

Seoul National University

Goals

- Implement "**Binary Search Tree**"
  - Fill your code in the methods in BinaryTree.Node class.
  - Reuse the insert, delete, find, traversal code in lab session5
  - **New function:** Lowest Common Ancestor (LCA), FlattenBinaryTree

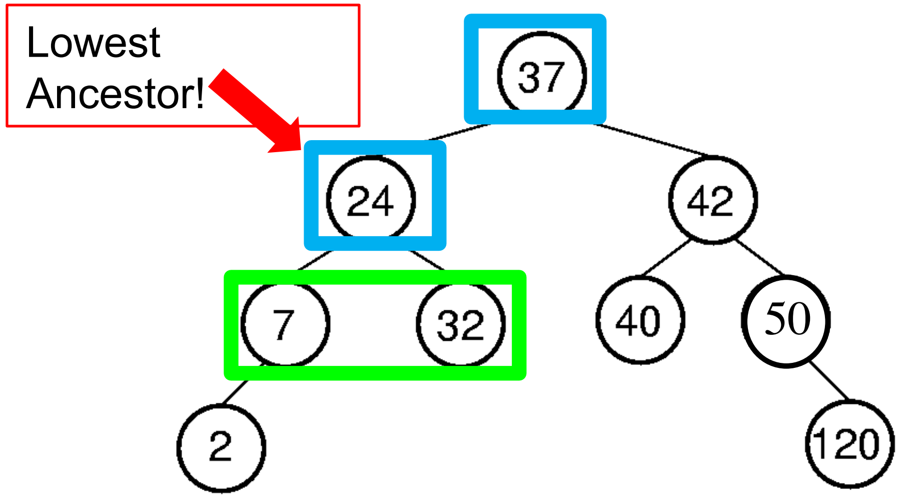
- Print the sample output corresponding to the sample input
  - Please carefully observe the I/O specification.

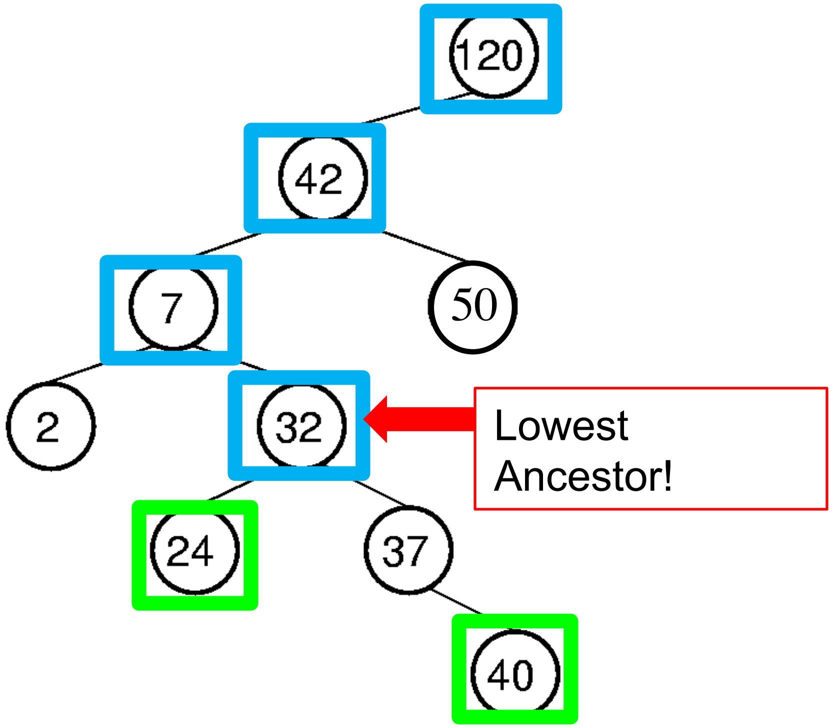
Build a project

- Download the project for this lab from eTL.
- Extract the project, and open it in InteliJ
  - See the slide of 1st lab session to check how to open the project in InteliJ.

Lowest Common Ancestor

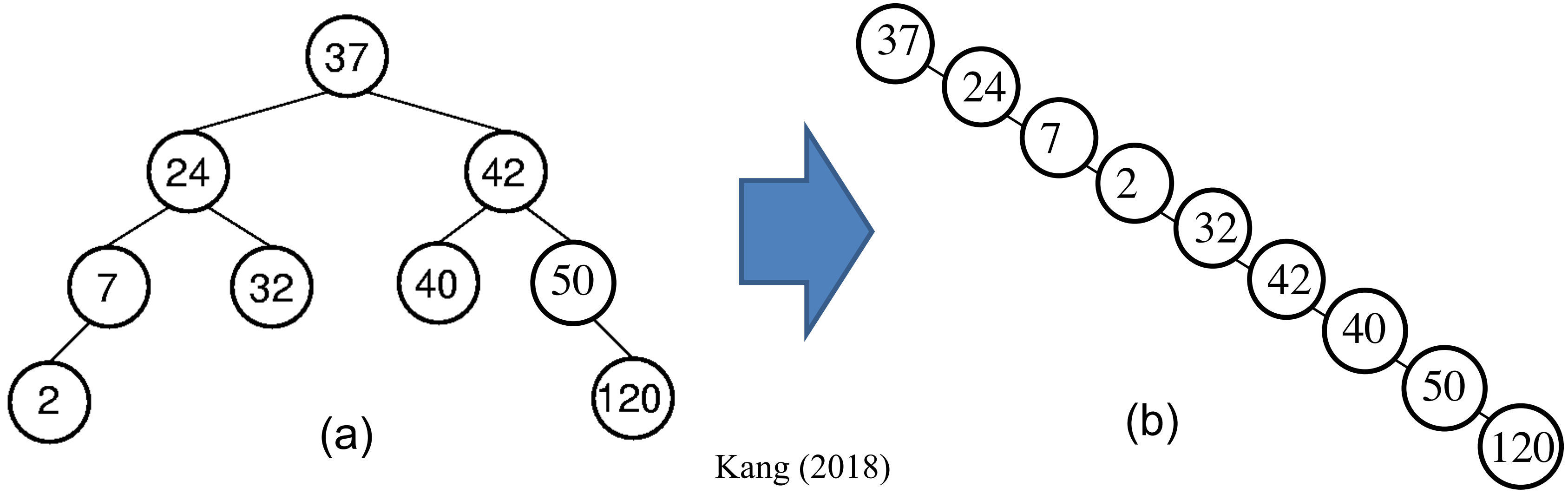
- Lowest Common Ancestor (LCA) of two nodes *v* and *w* in a tree *T* is the lowest (i.e. deepest) node that has both v and w as descendants.





Flatten Binary Tree

- Flatten binary tree is converting a binary tree into a special binary tree where each node has only one child.
  - Use constant extra space.
  - Flatten the tree (a) in preorder traversal sequence.
- After flattening, the left child of each node should point to NULL and the right child should point to the next node in the next level.



Kang (2018)

I/O Specification

- lowestCommonAncestor

Input form	Output form
`LCA (key1) (key2)`	`LCA of (key1) and (key2) : [(key3):(E3)]`

Description
- Given the tree root, key1 and key2, find the LCA of these keys. - Return the LCA node.

Example Input	Example Output
`LCA 3 1`	`LCA of 3 and 1: [2:Blackpink]`

I/O Specification

- flattenBinaryTree

Input form	Output form
`flatten`	`Flatten: [(key1):(E2)] [(key2):(E2)] [(key3):(E3)]…`

Description
- Given the tree root, flatten the tree in preorder traversal sequence with constant extra space.

Example Input	Example Output
`flatten`	`Flatten:[1:BTS][2:Blackpink]`

Sample Input

```
insert 4 TWICE
insert 2 Blackpink
insert 3 Redvelvet
insert 1 BTS
insert 6 EXO
insert 5 IZONE
insert 7 GFriend
preorder
inorder
LCA 1 7
flatten
preorder
inorder
```

Sample Output

```
preorder : [4:TWICE][2:Blackpink][1:BTS][3:Redvelvet][6:EXO][5:IZONE][7:GFriend]
inorder : [1:BTS][2:Blackpink][3:Redvelvet][4:TWICE][5:IZONE][6:EXO][7:GFriend]
LCA of 1 and 7 : [4:TWICE]
Flatten binary seatch tree
preorder : [4:TWICE][2:Blackpink][1:BTS][3:Redvelvet][6:EXO][5:IZONE][7:GFriend]
inorder : [4:TWICE][2:Blackpink][1:BTS][3:Redvelvet][6:EXO][5:IZONE][7:GFriend]
```

Questions?