Announcements

1. Exam 2 ends today! (M-W)

Join Code: **225**

Extra Credit Survey!

Warm-Up Question: What happens with a binary tree if I insert the elements in ascending order? Ex. 1, 2, 3, 4, 5

Balanced Binary Search Trees (BBSTs)

Learning Objectives

1. Define Balance of a Node
2. Evaluate whether a tree is balanced or not
3. Know how to perform rotations in a tree

Implementation of Lambda Functions

[ ]( ){ }

Example Function

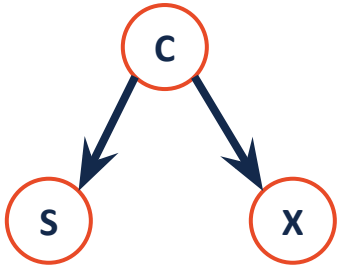
```
1   int big;
2   std::cout << "How big is big? "; // say input is 10
3   std::cin >> big;
4
5   auto isbig = [big](int num) { return (num > big); };
6   std::vector<int> values = {3, 7, 10, 15, 2, 20, 8};
7   std::cout << "Numbers greater than " << big << ": ";
8   for (int v : values) {
9     if (isbig(v)) {
10        std::cout << v << " ";
11    }
12  }
13
14  std::cout << std::endl;
15
16 }
```

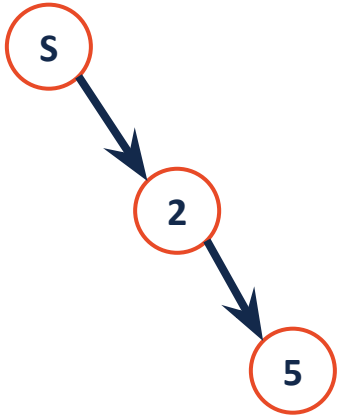
Motivation

Binary Search Trees

- Find
- Insert
- Remove

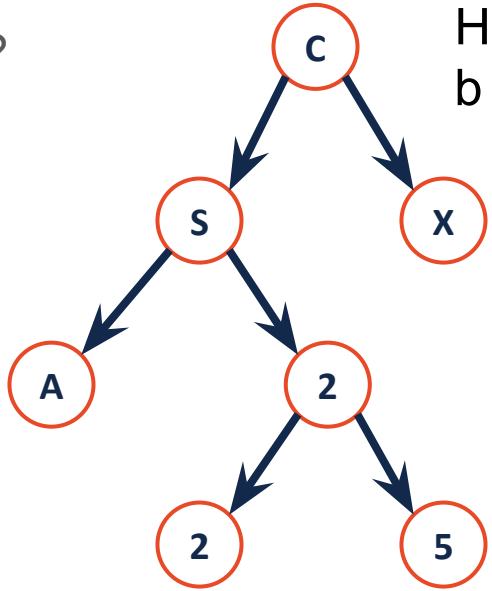
Balanced





Balanced

What is the balance of node s?



Height balance:
b = height(T_R) - height(T_L)

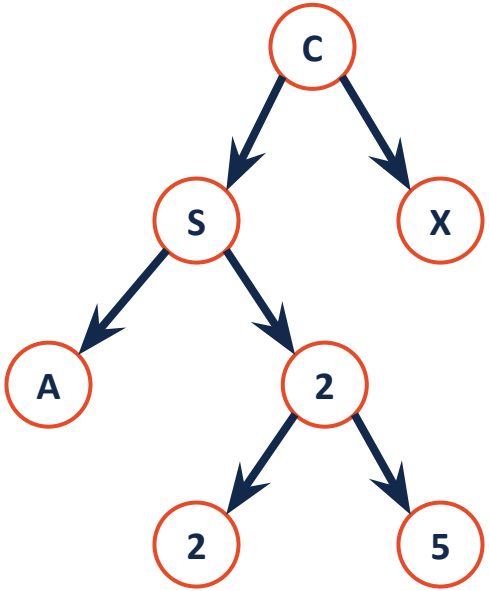
Is this tree balanced?

Join Code: **225**

Tree Balanced

A tree is height balanced if: |b| <= 1

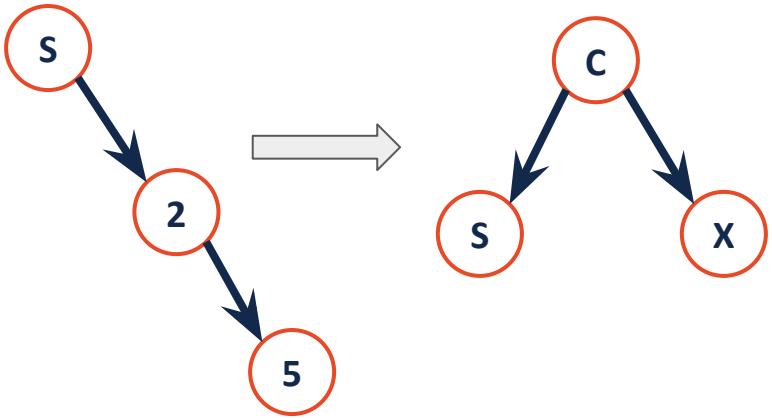
No!

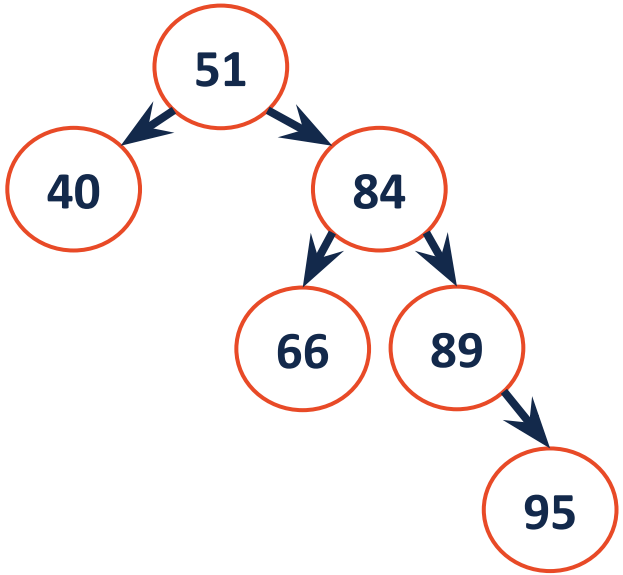


Goal of Rotations

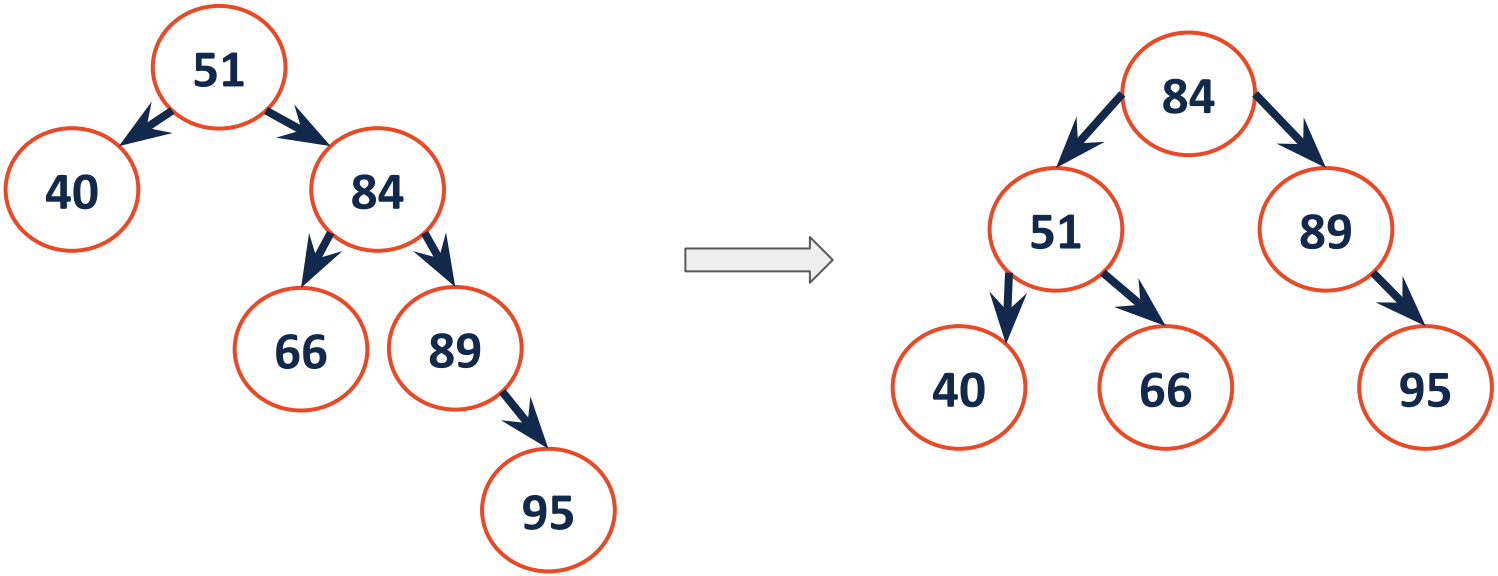
1. BST Property

2. Changes sticks into mountains

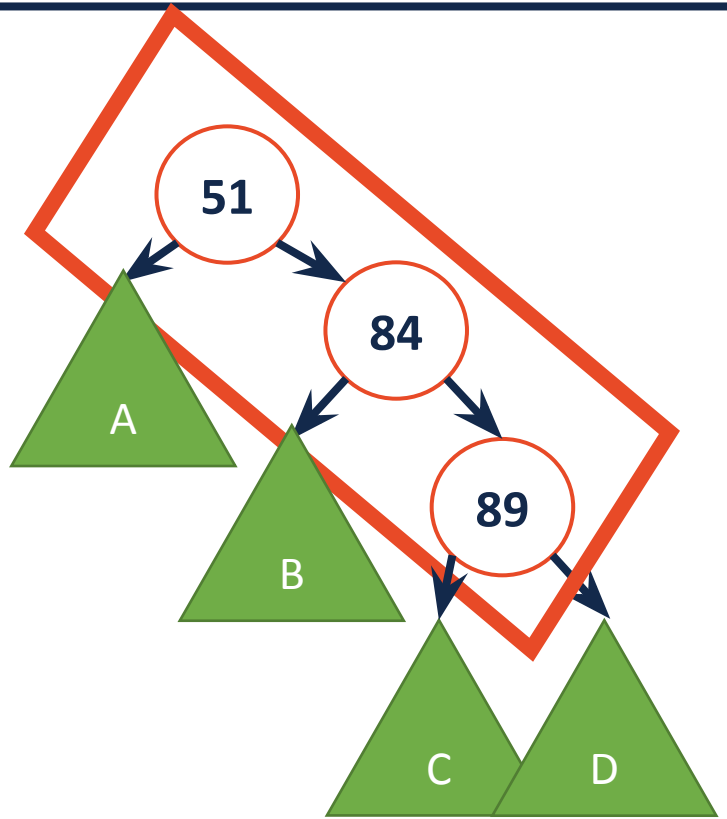



51
40
84
66
89
95

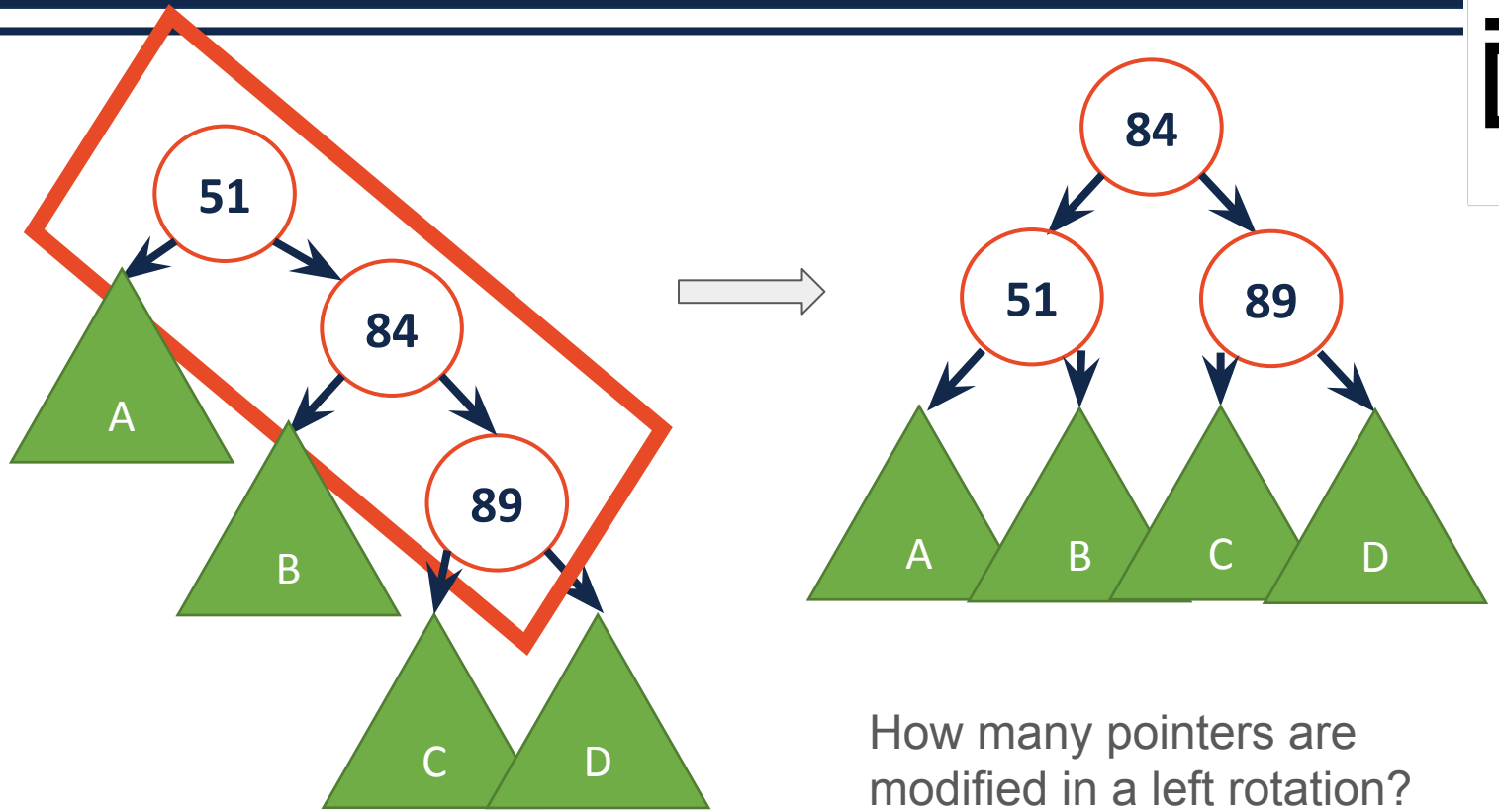
Left Rotation



Left Rotation



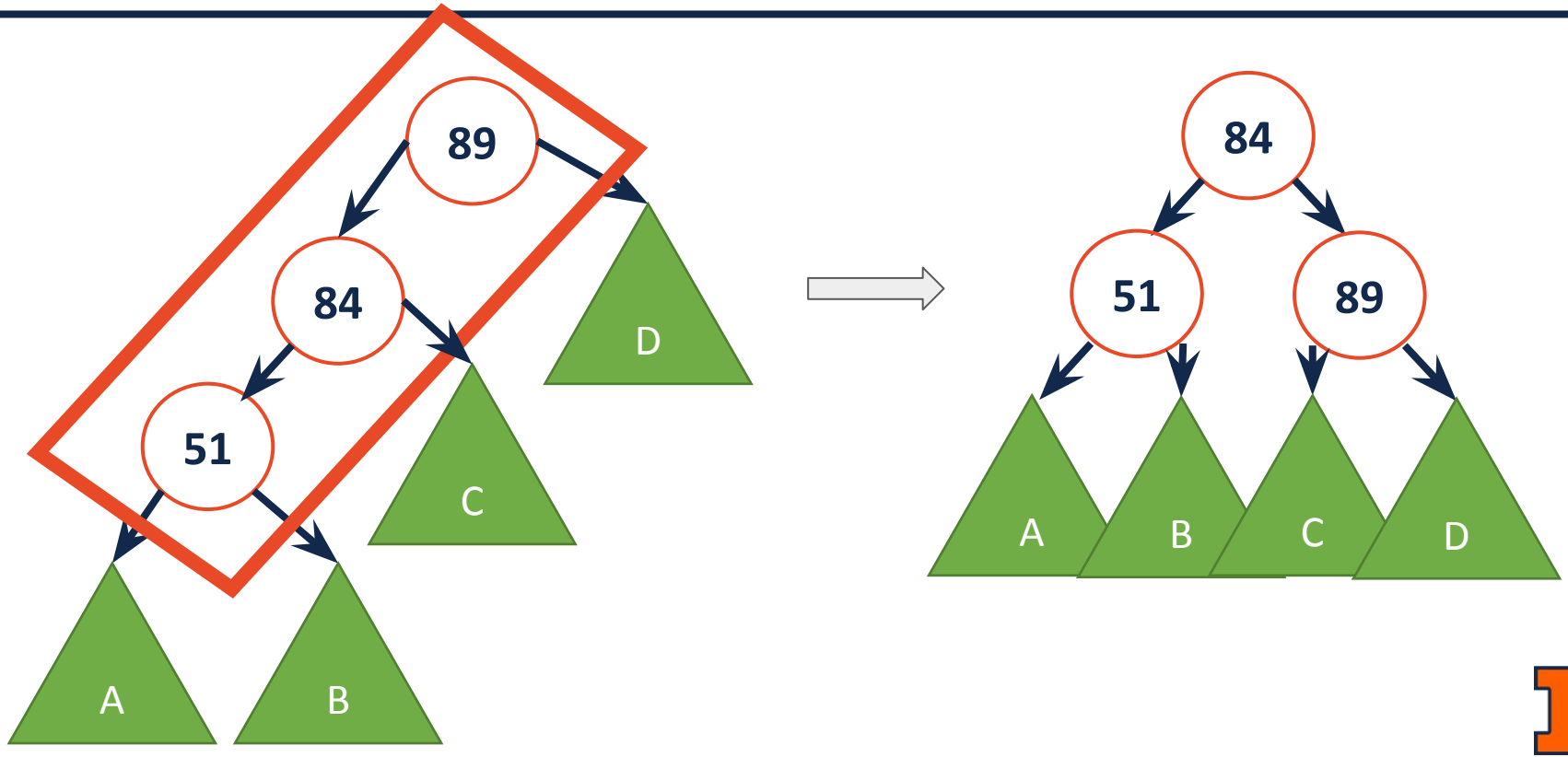
Left Rotation

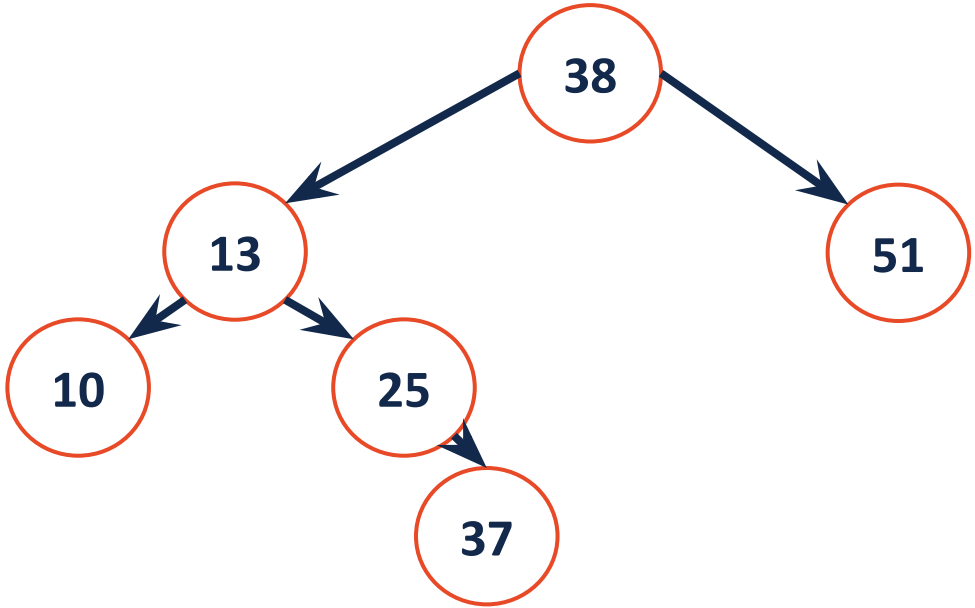


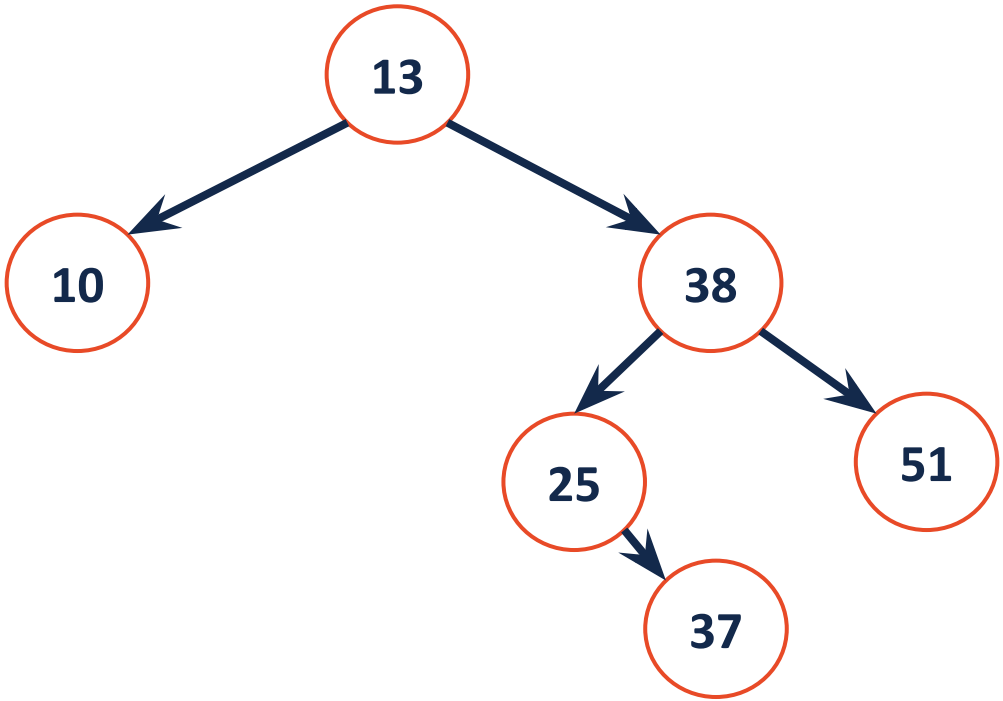
How many pointers are modified in a left rotation?

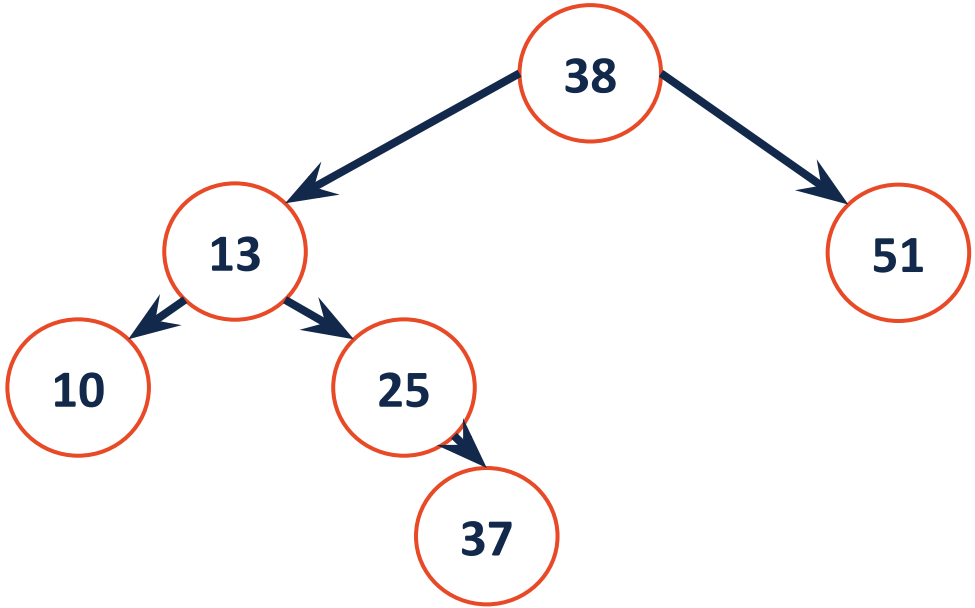
Join Code: **225**

Right Rotation

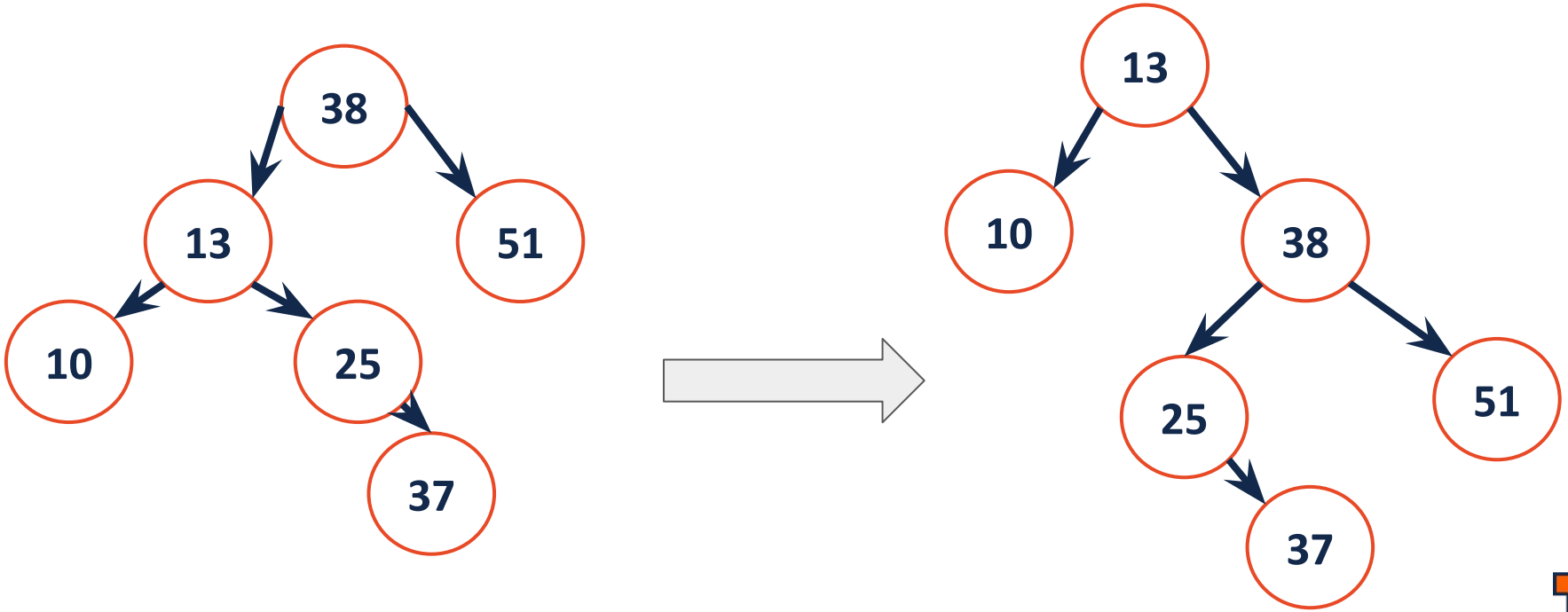



38
13
51
10
25
37

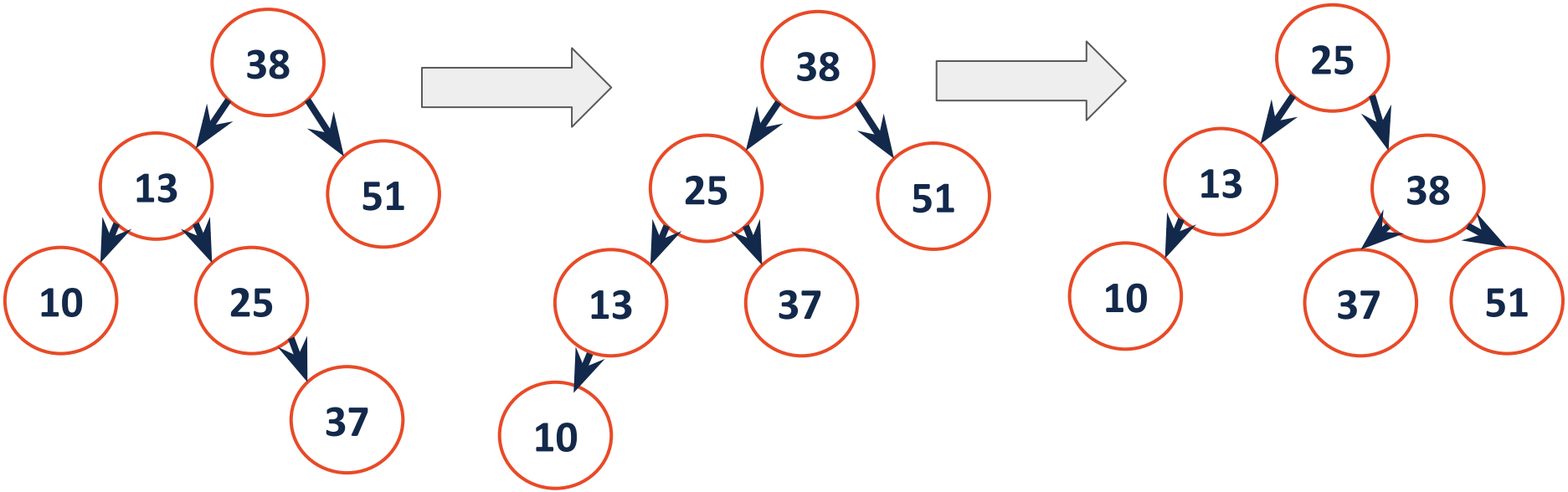

13
10
38
25
51
37


38
13
51
10
25
37

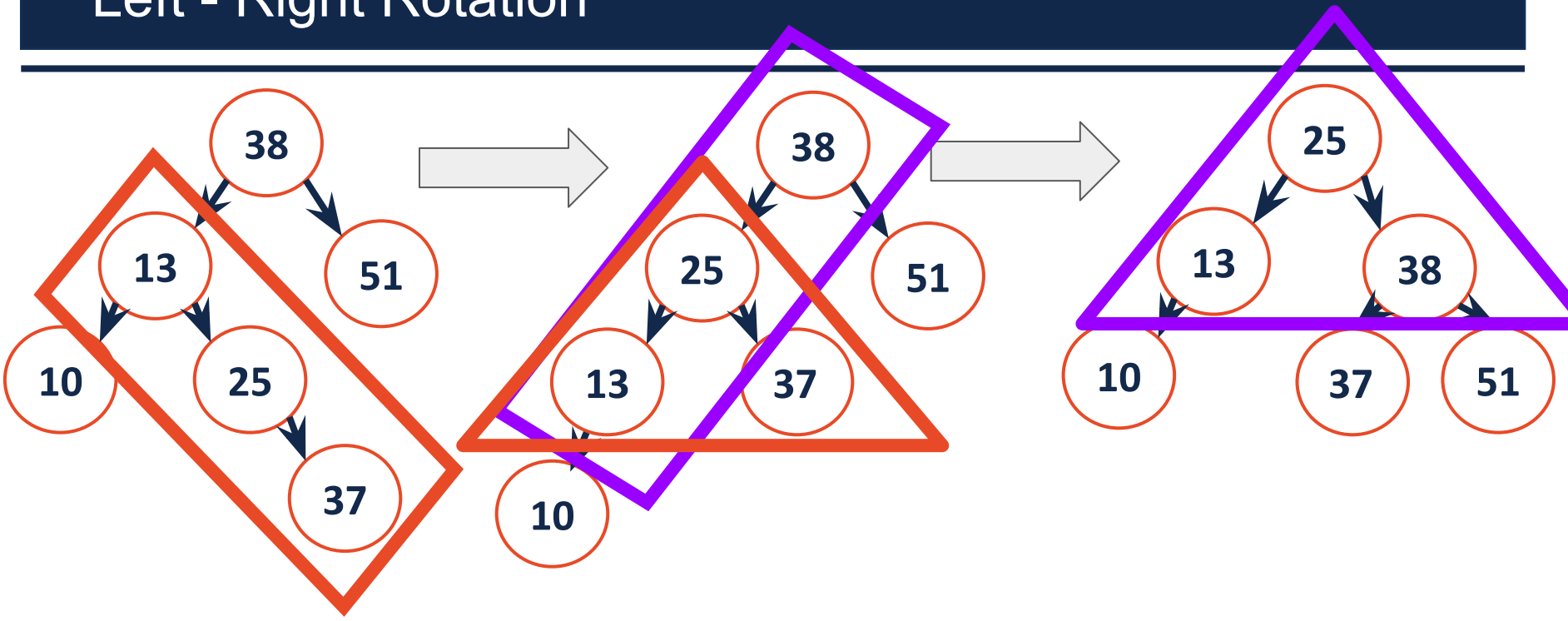
What to do in this case?



Left - Right Rotation



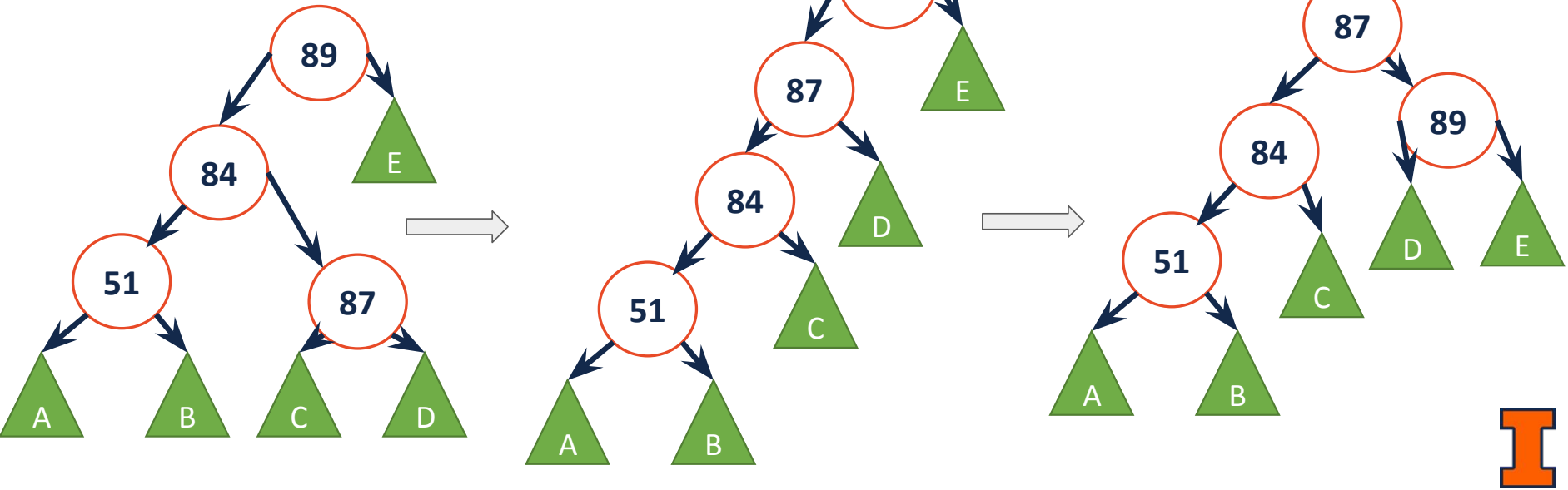
Left - Right Rotation



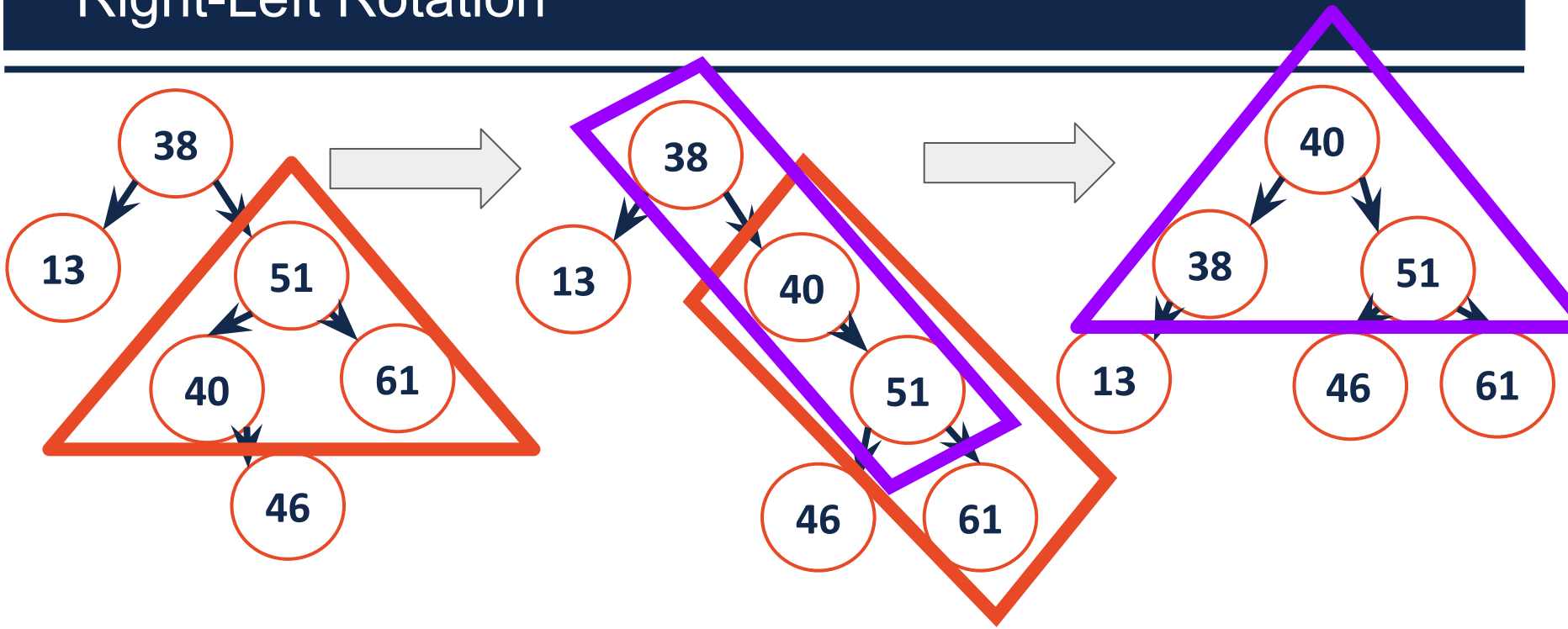
Left-Right Rotation

How many pointers are modified in a left rotation?

Join Code: **225**



Right-Left Rotation



BBST Rotations Summary

Four kinds of rotations (L, R, LR, RL)

All rotations are local (subtrees are not impacted)

All rotations are constant time: O(1)

BST property maintained

GOAL:

We call these trees:

AVL Trees

Three issues for consideration:

1. Detecting Imbalance - How do I detect an imbalance?
2. Rotations - How do they work?
3. Selecting a rotation - Which one do I choose?