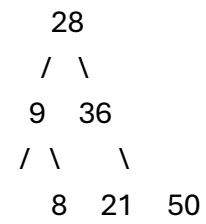
Version 1

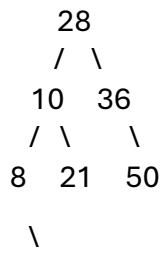
Q1) O(logN)

Q2) Any element greater than 50

Q3) ***Award full credit for either of the first two trees:***



Award only 2 pts for the following tree:



Q4) Award 2 pts for each correct condition. ***No partial grading for single incomplete conditions.***

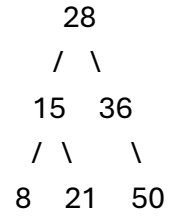
```
// Condition 1
If (balance < -1 && getBalance(root → right) <= 0)
     return leftRotate(root);
// Condition 2
if (balance > 1 && getBalance(root → left) < 0)
     return leftRightRotate(root);
```

Version 2

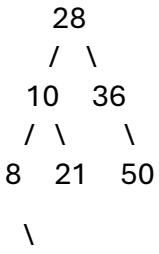
Q1) O(logN)

Q2) Any element greater than 36 but less than 50

Q3) ***Award full credit for:***



Award only 2 pts for the following tree:



Q4) Award 2 pts for each correct condition. ***No partial grading for single incomplete conditions.***

```
// Condition 1
If (balance > 1 && getBalance(root → left) >= 0)
      return rightRotate(root);
// Condition 2
if (balance < -1 && getBalance(root → right) > 0)
      return rightLeftRotate(root);
```

```
Node* rebalance(Node* root) {
     if ( root == nullptr)
          return root;
      updateHeight(root);
      int bal = getBalance(Node* root);
      // Condition 1
      if

          return leftRotate(root);
     // Condition 2
     if

          return leftRightRotate(root);
}
```