

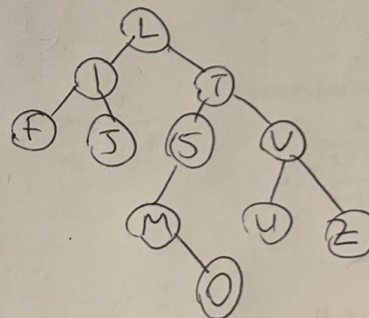
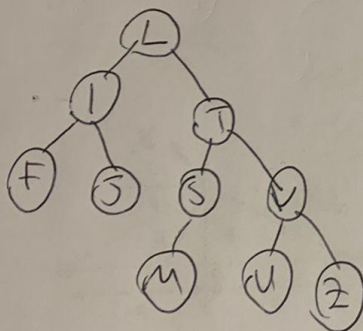
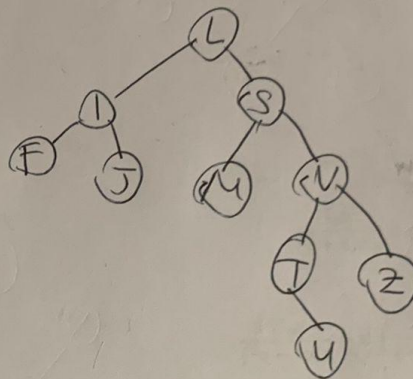
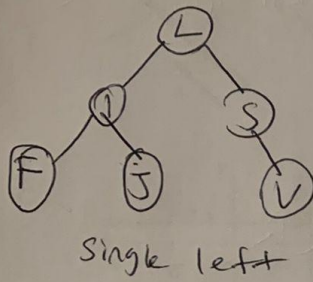
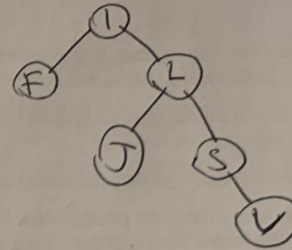
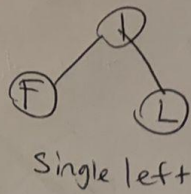
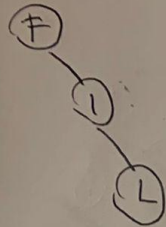
Name: Kutay Şenyiğit

ID: 21902377

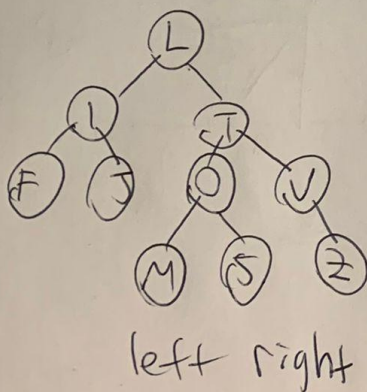
Section: 1

Homework: 3

Question 1



right left



```

double computeMedian () {
    if (size == 0) {
        cout << "Illegal State";
    }

    if (size % 2 == 0) {
        int first = getNode (root, (size/2) - 1).key;
        int second = getNode (root, size/2).key;
        int result = (first + second) / 2;
        return result;
    }
    else {
        int result = getNode (root, size/2).key;
        return result;
    }
}

```

Median is calculated by finding the middle of the element of odd numbers and finding the middle and middle-1 element's average. Note that these middle elements should be found in sorted numbers. By using getNode method, ~~providing~~ giving the necessary indexes we can find the median. It's time complexity is  $O(\log n)$

```

bool checkAVL(Node* root) {
    if (root == NULL) return true; // it's AVL
    int leftHeight,
    int rightHeight;

    leftHeight = height(root->left);
    rightHeight = height(root->right);
    if (checkAVL(root->left) && checkAVL(root->right)) {
        int diff = abs(leftHeight - rightHeight);
        if (diff <= 1) { return true; }
    }
    return false;
}

```

In order to be AVL tree. The tree should be balanced. To be able to understand if the tree is balanced or not, we can simply look at its number of left child nodes and as well as the number of the right child nodes. If the difference more than 1 node the tree cannot be a AVL tree. Thus, cannot be balanced. In the worst case of full binary tree, time complexity is  $O(n^2)$

### Question 3

As we learned from previous algorithms in the class, it would not be efficient because when we start to increase computer count one by one we have  $O(n)$  complexity. Therefore, we should try to use an efficient algorithm that will have  $O(\log n)$  time complexity. We can use divide and conquer techniques by using recursive implementations.