Assignment-Day 7 and 8

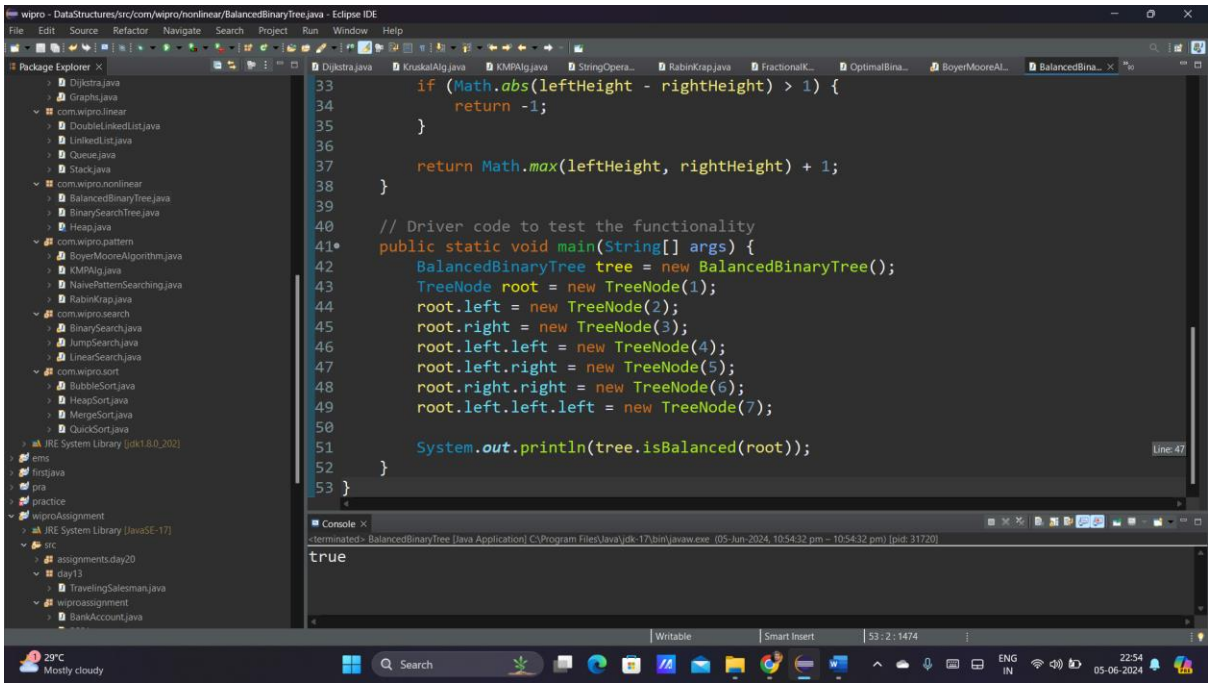
Day 7 and 8:

Task 1: Balanced Binary Tree Check

Write a function to check if a given binary tree is balanced. A balanced tree is one where the height of two subtrees of any node never differs by more than one.

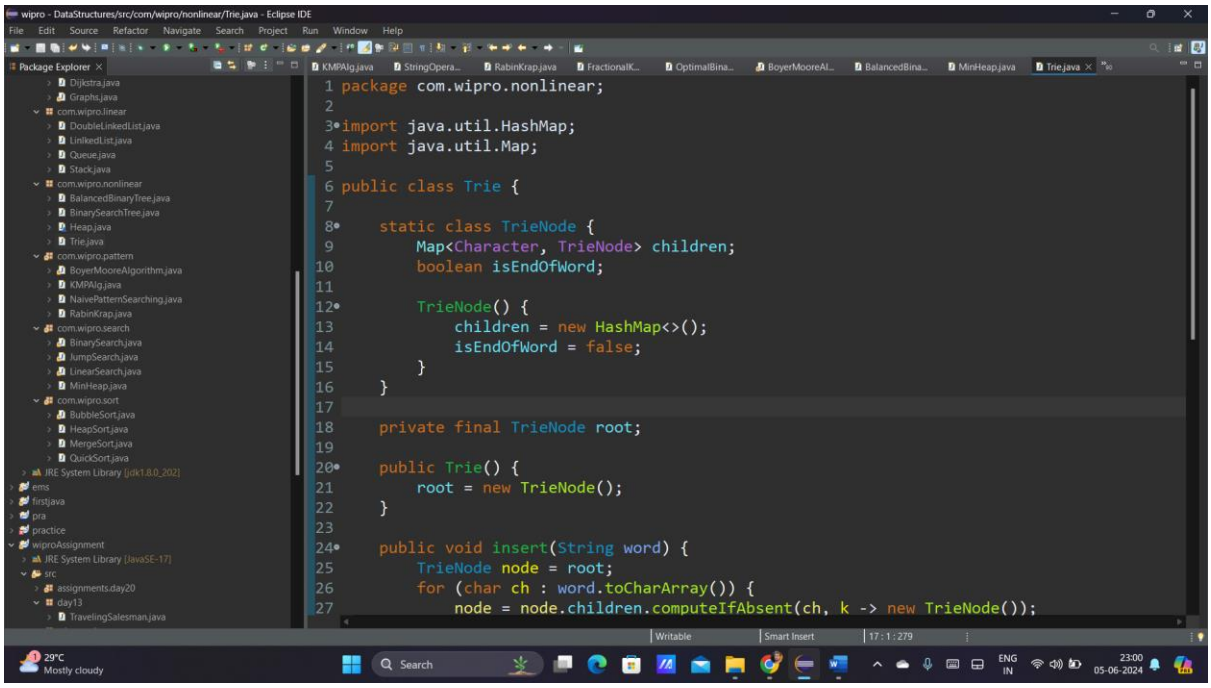
```java
package com.wipro.nonlinear;

public class BalancedBinaryTree {

    // TreeNode class definition
    static class TreeNode {
        int val;
        TreeNode left, right;

        TreeNode(int item) {
            val = item;
            left = right = null;
        }
    }

    // Method to check if the tree is balanced
    public boolean isBalanced(TreeNode root) {
        return checkHeight(root) != -1;
    }

    // Helper method to check the height of the tree
    private int checkHeight(TreeNode node) {
        if (node == null) {
            return 0;
        }

        int leftHeight = checkHeight(node.left);
```

```java
        int leftHeight = checkHeight(node.left);
        if (leftHeight == -1) return -1;

        int rightHeight = checkHeight(node.right);
        if (rightHeight == -1) return -1;

        if (Math.abs(leftHeight - rightHeight) > 1) {
            return -1;
        }

        return Math.max(leftHeight, rightHeight) + 1;
    }

    // Driver code to test the functionality
    public static void main(String[] args) {
        BalancedBinaryTree tree = new BalancedBinaryTree();
        TreeNode root = new TreeNode(1);
        root.left = new TreeNode(2);
        root.right = new TreeNode(3);
        root.left.left = new TreeNode(4);
        root.left.right = new TreeNode(5);
        root.right.right = new TreeNode(6);
        root.left.left.left = new TreeNode(7);

        System.out.println(tree.isBalanced(root));
    }
}
```



Task 2: Trie for Prefix Checking

Implement a trie data structure in C# that supports insertion of strings and provides a method to check if a given string is a prefix of any word in the trie.



wipro - DataStructures/src/com/wipro/nonlinear/Trie.java - Eclipse IDE

```java
            node = node.children.computeIfAbsent(ch, k -> new TrieNode());
        }
        node.isEndOfWord = true;
    }
    public boolean isPrefix(String prefix) {
        TrieNode node = root;
        for (char ch : prefix.toCharArray()) {
            node = node.children.get(ch);
            if (node == null) {
                return false;
            }
        }
        return true;
    }
    // Driver code to test the Trie
    public static void main(String[] args) {
        Trie trie = new Trie();
        trie.insert("apple");
        trie.insert("app");
        trie.insert("banana");

        System.out.println(trie.isPrefix("app")); // Output: true
        System.out.println(trie.isPrefix("ban")); // Output: true
        System.out.println(trie.isPrefix("bat")); // Output: false
    }
}
```

wipro - DataStructures/src/com/wipro/nonlinear/Trie.java - Eclipse IDE

```java
            node = node.children.computeIfAbsent(ch, k -> new TrieNode());
        }
        node.isEndOfWord = true;
    }
    public boolean isPrefix(String prefix) {
        TrieNode node = root;
        for (char ch : prefix.toCharArray()) {
            node = node.children.get(ch);
            if (node == null) {
                return false;
            }
        }
        return true;
    }
    // Driver code to test the Trie
    public static void main(String[] args) {
        Trie trie = new Trie();
        trie.insert("apple");
        trie.insert("app");
        trie.insert("banana");
```

Console

```
<terminated> Trie [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (05-Jun-2024, 11:01:27 pm – 11:01:31 pm) [pid: 11440]
true
true
false
```

Task 3: Implementing Heap Operations

Code a min-heap in C# with methods for insertion, deletion, and fetching the minimum element. Ensure that the heap property is maintained after each operation.

```java
package com.wipro.search;

import java.util.ArrayList;
import java.util.List;

public class MinHeap {

    private final List<Integer> heap;

    public MinHeap() {
        heap = new ArrayList<>();
    }

    public void insert(int val) {
        heap.add(val);
        heapifyUp(heap.size() - 1);
    }

    public int getMin() {
        if (heap.isEmpty())
            throw new IllegalStateException("Heap is empty");
        return heap.get(0);
    }

    public int removeMin() {
        if (heap.isEmpty())
            throw new IllegalStateException("Heap is empty");
```

```java
            throw new IllegalStateException("Heap is empty");
        int min = heap.get(0);
        int lastElement = heap.remove(heap.size() - 1);

        if (!heap.isEmpty()) {
            heap.set(0, lastElement);
            heapifyDown(0);
        }

        return min;
    }

    private void heapifyUp(int index) {
        while (index > 0) {
            int parentIndex = (index - 1) / 2;
            if (heap.get(index) >= heap.get(parentIndex))
                break;

            swap(index, parentIndex);
            index = parentIndex;
        }
    }

    private void heapifyDown(int index) {
        int size = heap.size();
        while (index < size) {
            int leftChildIndex = 2 * index + 1;
```

wipro - DataStructures/src/com/wipro/search/MinHeap.java - Eclipse IDE

```java
            int leftChildIndex = 2 * index + 1;
            int rightChildIndex = 2 * index + 2;
            int smallestIndex = index;

            if (leftChildIndex < size && heap.get(leftChildIndex) < heap.get(smallestIndex
                smallestIndex = leftChildIndex;
            }

            if (rightChildIndex < size && heap.get(rightChildIndex) < heap.get(smallestInd
                smallestIndex = rightChildIndex;
            }

            if (smallestIndex == index)
                break;

            swap(index, smallestIndex);
            index = smallestIndex;
        }
    }

    private void swap(int index1, int index2) {
        int temp = heap.get(index1);
        heap.set(index1, heap.get(index2));
        heap.set(index2, temp);
    }
```

wipro - DataStructures/src/com/wipro/search/MinHeap.java - Eclipse IDE

```java
            swap(index, smallestIndex);
            index = smallestIndex;
        }
    }

    private void swap(int index1, int index2) {
        int temp = heap.get(index1);
        heap.set(index1, heap.get(index2));
        heap.set(index2, temp);
    }

    // Driver code to test the MinHeap
    public static void main(String[] args) {
        MinHeap minHeap = new MinHeap();
        minHeap.insert(3);
        minHeap.insert(1);
        minHeap.insert(6);
        minHeap.insert(5);
        minHeap.insert(2);
        minHeap.insert(4);

        System.out.println(minHeap.getMin()); // Output: 1
        System.out.println(minHeap.removeMin()); // Output: 1
        System.out.println(minHeap.getMin()); // Output: 2
    }
}
```

wipro - DataStructures/src/com/wipro/search/MinHeap.java - Eclipse IDE

```java
        int temp = heap.get(index1);
        heap.set(index1, heap.get(index2));
        heap.set(index2, temp);
    }

    // Driver code to test the MinHeap
    public static void main(String[] args) {
        MinHeap minHeap = new MinHeap();
        minHeap.insert(3);
        minHeap.insert(1);
        minHeap.insert(6);
        minHeap.insert(5);
        minHeap.insert(2);
        minHeap.insert(4);

        System.out.println(minHeap.getMin()); // Output: 1
        System.out.println(minHeap.removeMin()); // Output: 1
        System.out.println(minHeap.getMin()); // Output: 2
    }
}
```

Console

<terminated> MinHeap [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (05-Jun-2024, 11:06:00 pm – 11:06:04 pm) [pid: 10348]

```
1
1
2
```