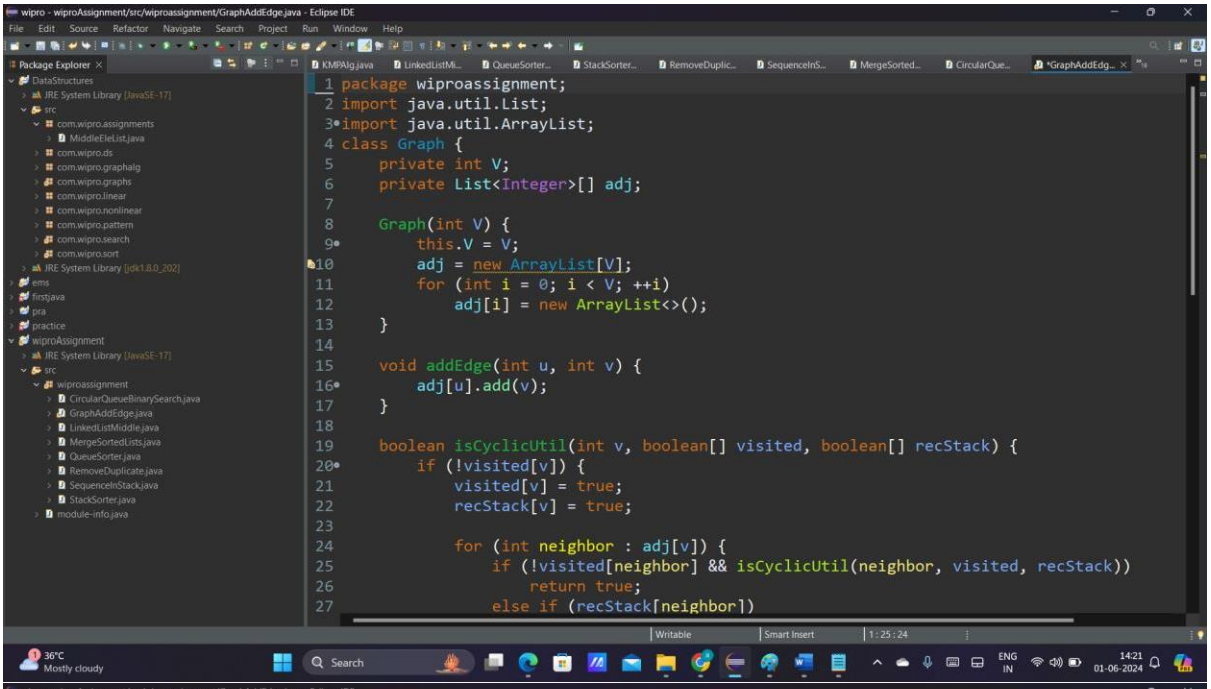
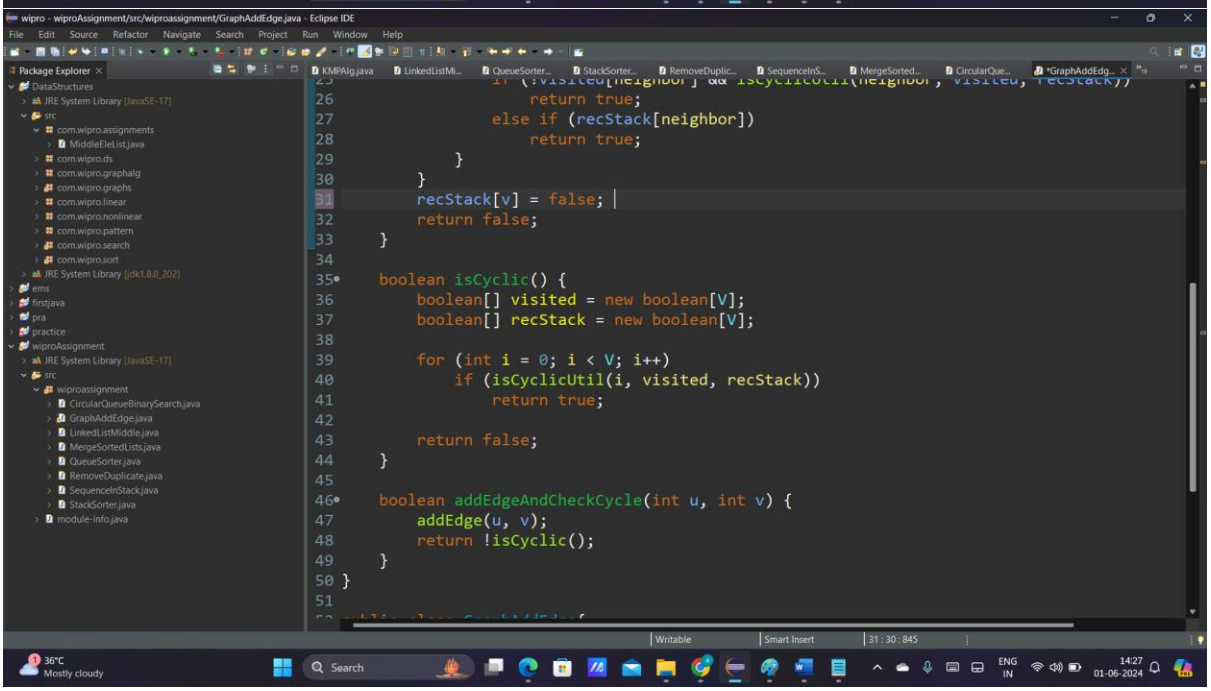
Task 4: Graph Edge Addition Validation

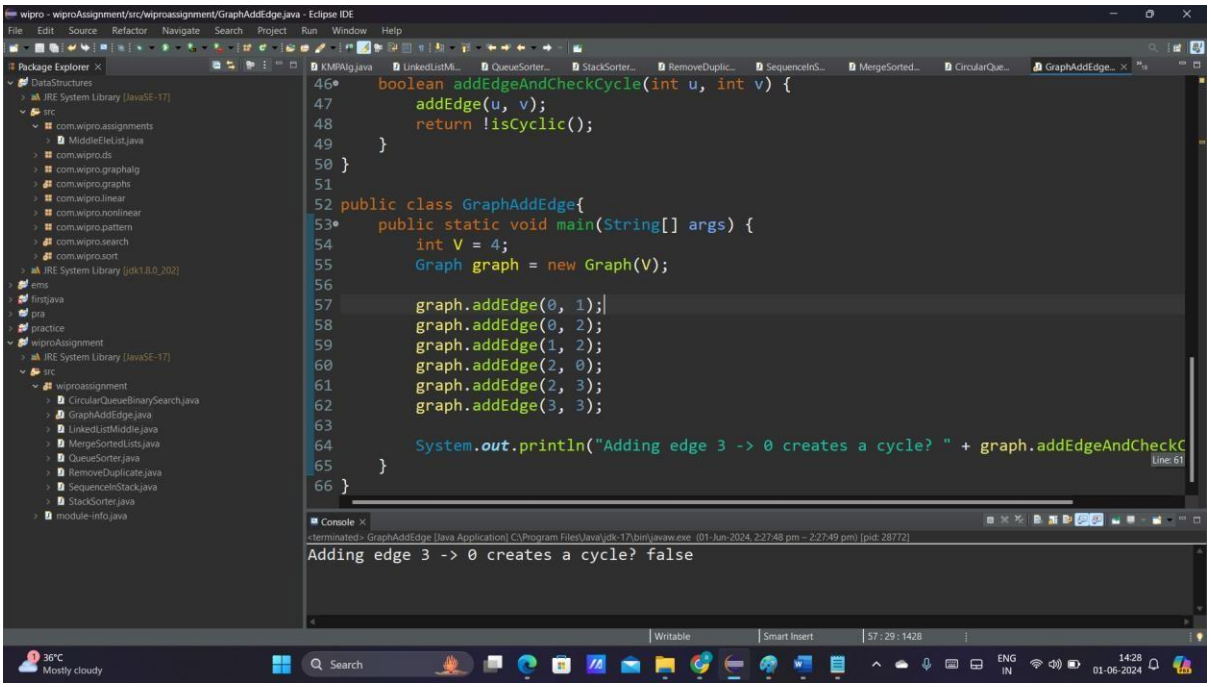
Given a directed graph, write a function that adds an edge between two nodes and then checks if the graph still has no cycles. If a cycle is created, the edge should not be added.



```java
package wiproassignment;
import java.util.List;
import java.util.ArrayList;
class Graph {
    private int V;
    private List<Integer>[] adj;

    Graph(int V) {
        this.V = V;
        adj = new ArrayList[V];
        for (int i = 0; i < V; ++i)
            adj[i] = new ArrayList<>();
    }

    void addEdge(int u, int v) {
        adj[u].add(v);
    }

    boolean isCyclicUtil(int v, boolean[] visited, boolean[] recStack) {
        if (!visited[v]) {
            visited[v] = true;
            recStack[v] = true;

            for (int neighbor : adj[v]) {
                if (!visited[neighbor] && isCyclicUtil(neighbor, visited, recStack))
                    return true;
                else if (recStack[neighbor])
```

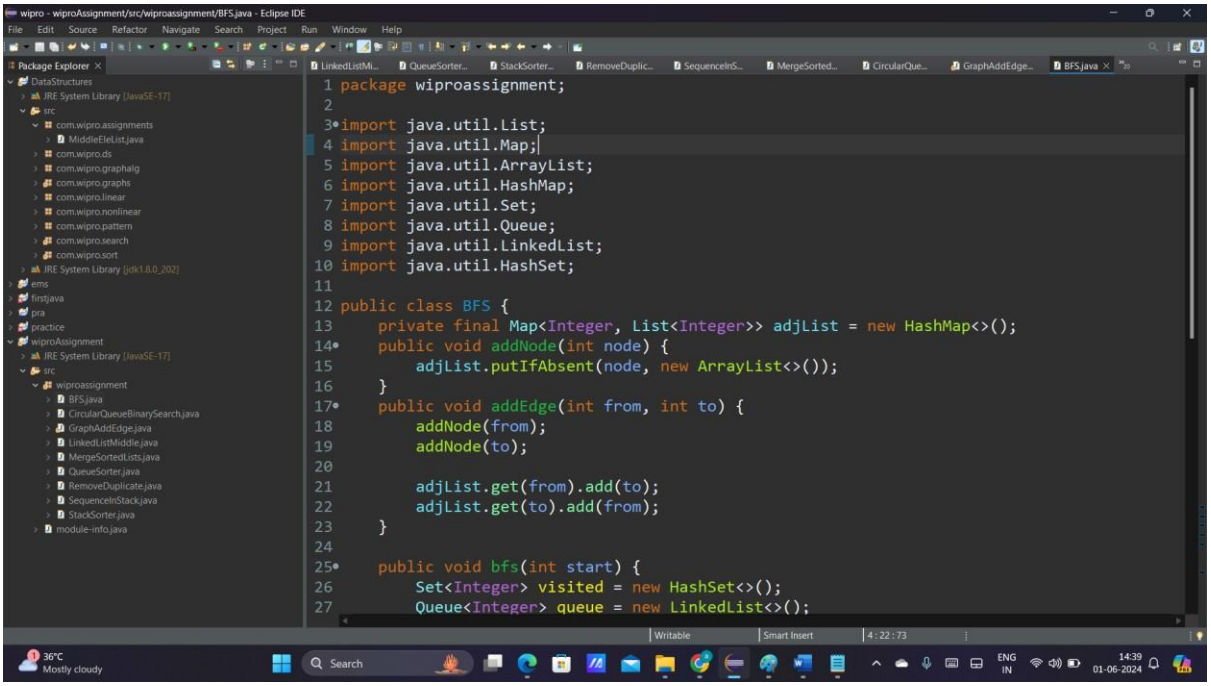


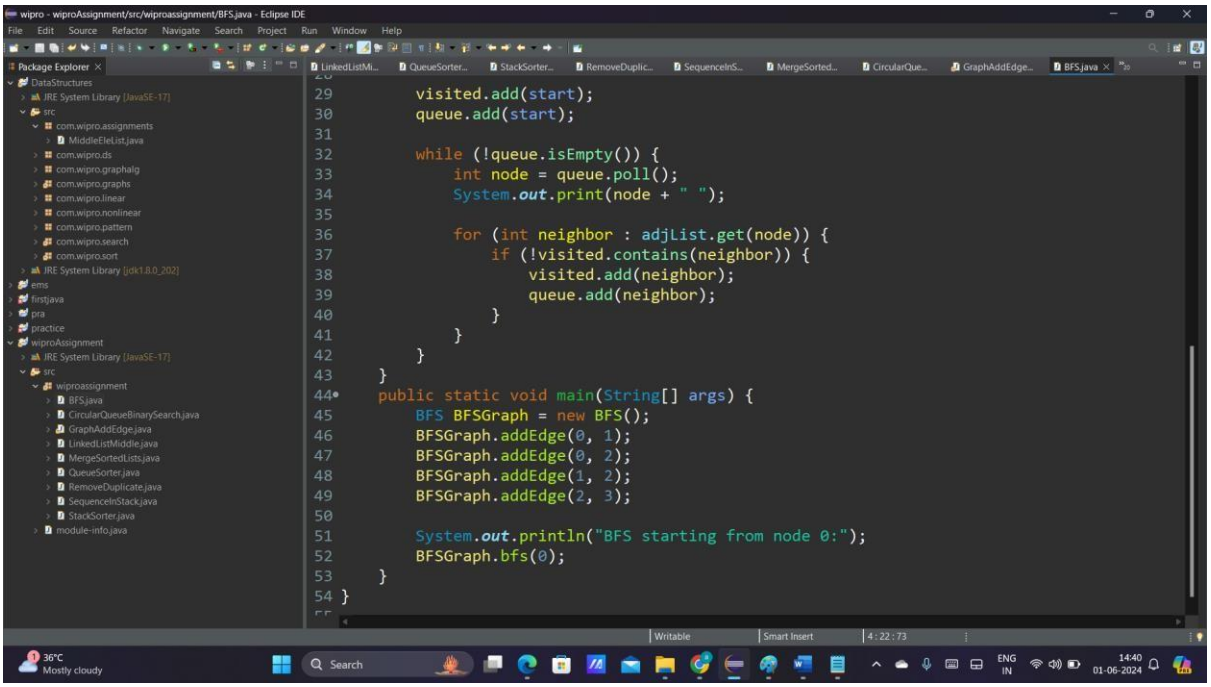
```java
                    return true;
                else if (recStack[neighbor])
                    return true;
            }
        }
        recStack[v] = false;
        return false;
    }

    boolean isCyclic() {
        boolean[] visited = new boolean[V];
        boolean[] recStack = new boolean[V];

        for (int i = 0; i < V; i++)
            if (isCyclicUtil(i, visited, recStack))
                return true;

        return false;
    }

    boolean addEdgeAndCheckCycle(int u, int v) {
        addEdge(u, v);
        return !isCyclic();
    }
}
```

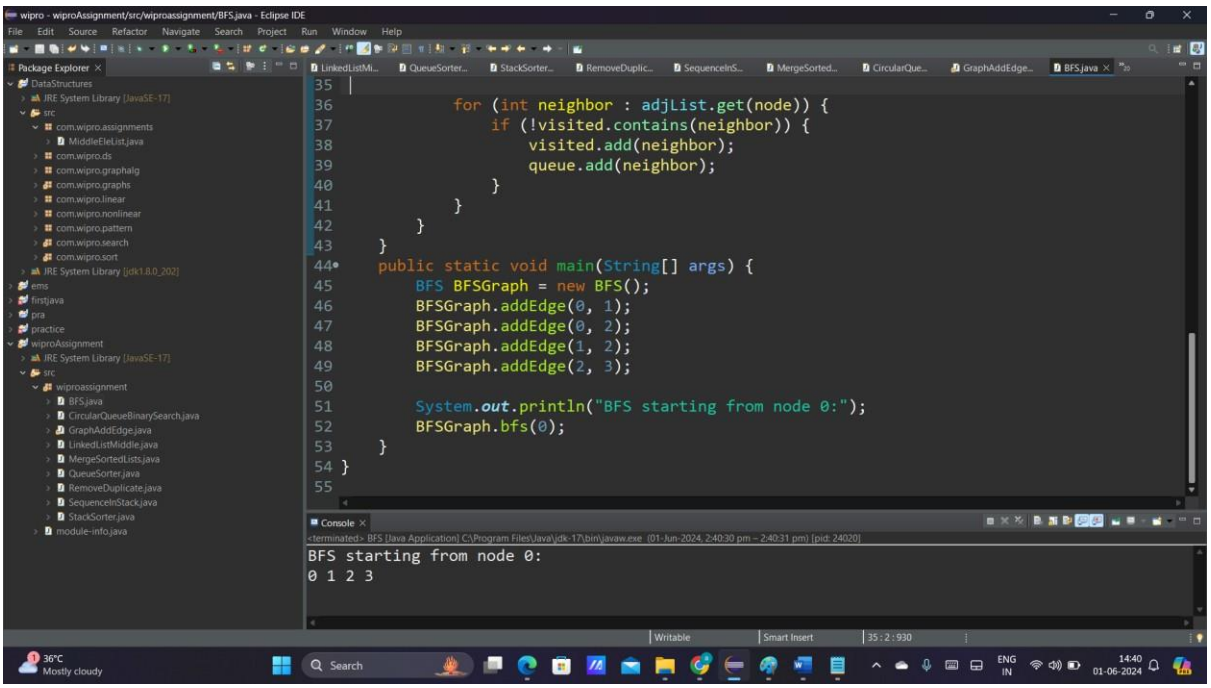


Task 5: Breadth-First Search (BFS) Implementation

For a given undirected graph, implement BFS to traverse the graph starting from a given node and print each node in the order it is visited.

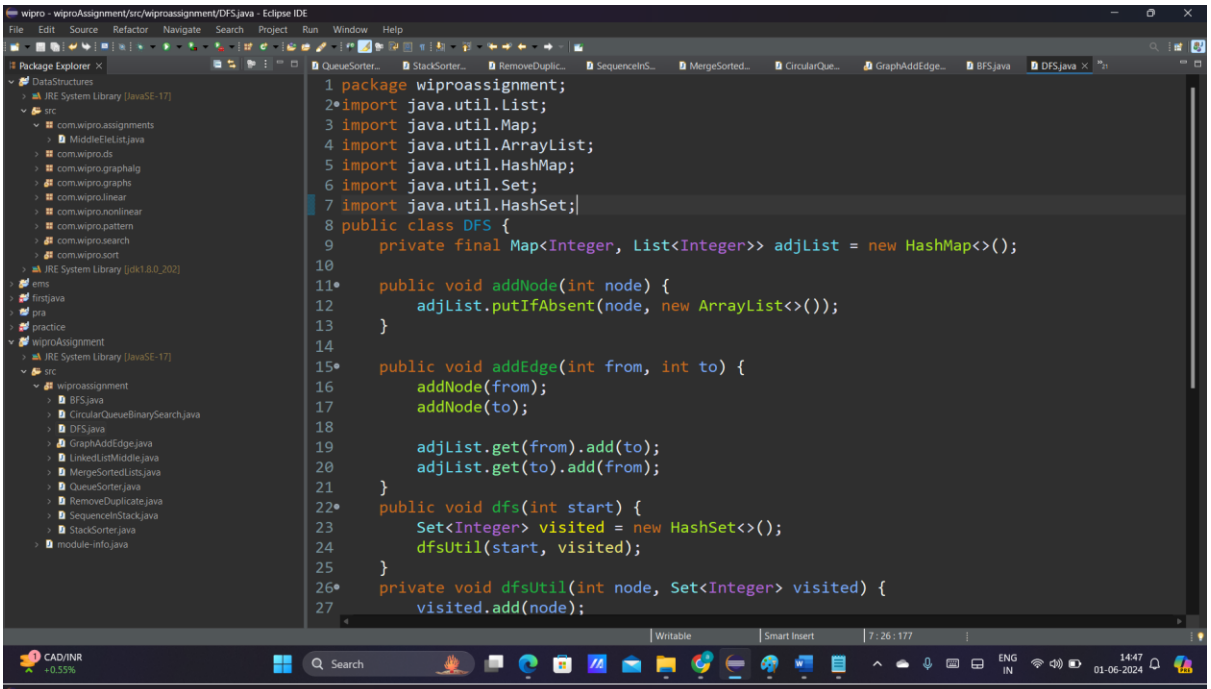


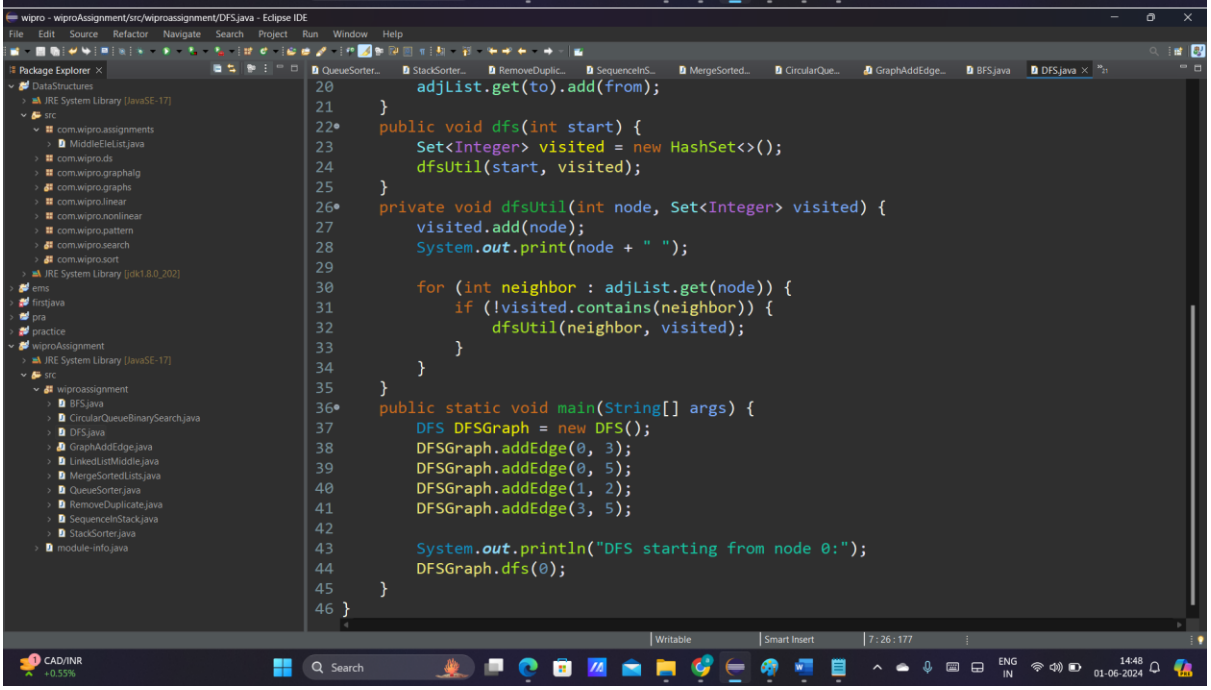


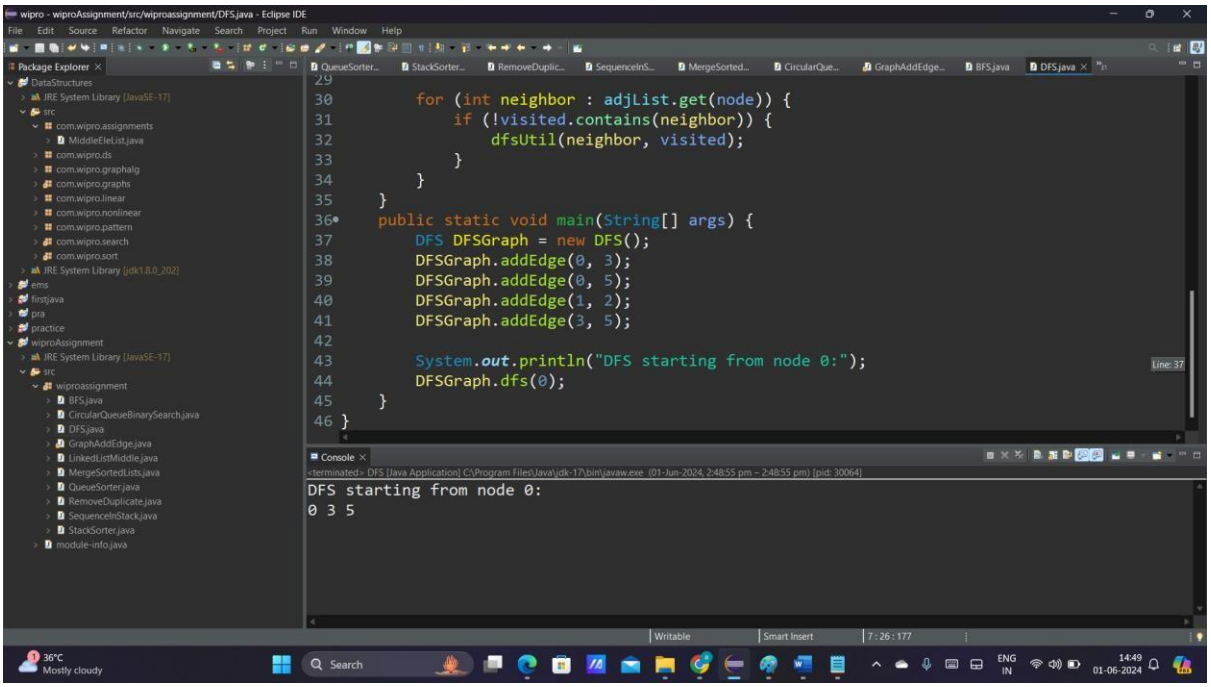


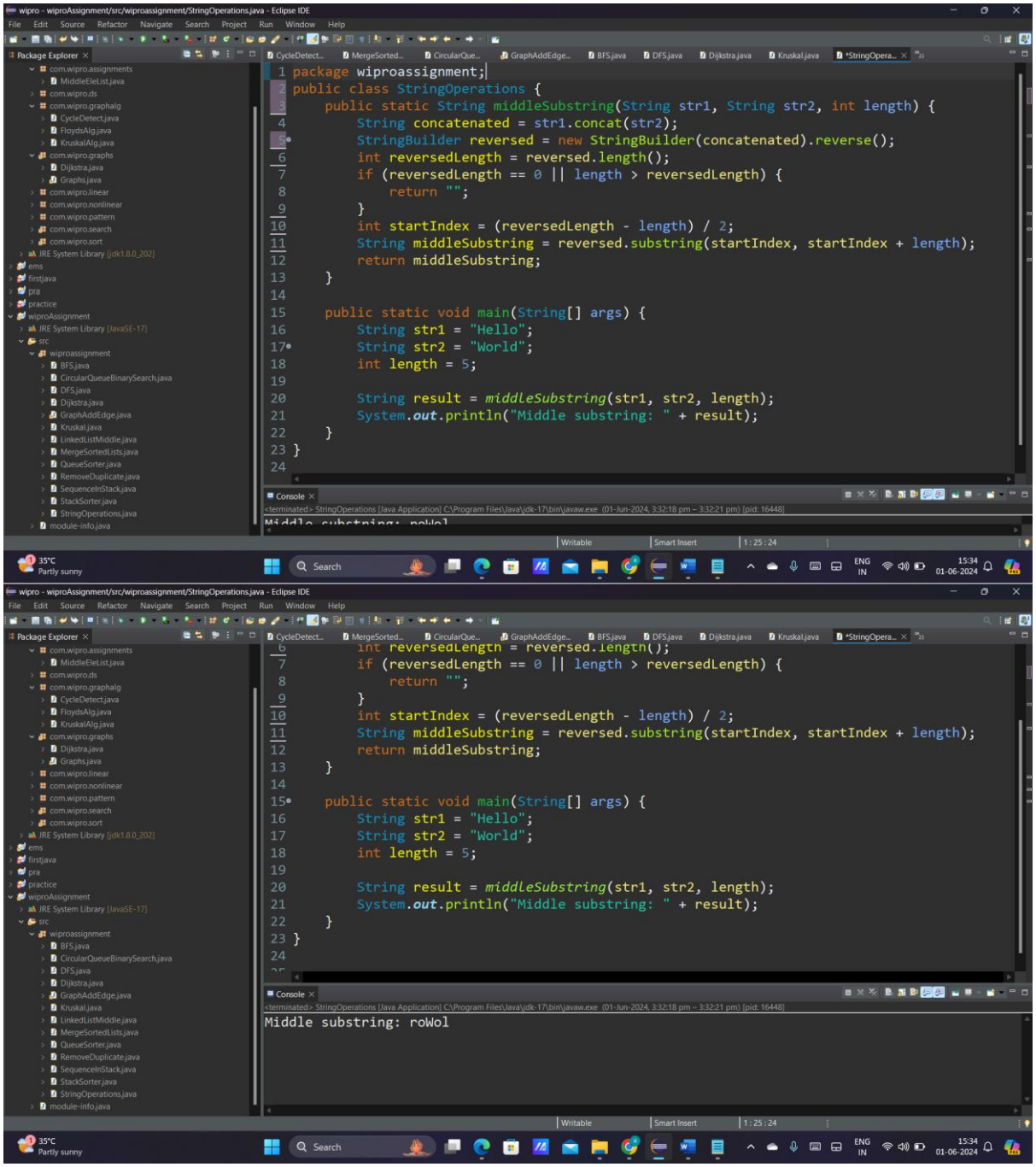
Task 6: Depth-First Search (DFS) Recursive

Write a recursive DFS function for a given undirected graph. The function should visit every node and print it out.


wipro - wiproAssignment/src/wiproassignment/DFS.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
DataStructures
JRE System Library [JavaSE-17]
src
com.wipro.assignments
MiddleEleList.java
com.wipro.ds
com.wipro.graphalg
com.wipro.graphs
com.wipro.linear
com.wipro.nonlinear
com.wipro.pattern
com.wipro.search
com.wipro.sort
JRE System Library [jdk1.8.0_202]
ems
firstjava
pra
practice
wiproAssignment
JRE System Library [JavaSE-17]
src
wiproassignment
BFS.java
CircularQueueBinarySearch.java
DFS.java
GraphAddEdge.java
LinkedListMiddle.java
MergeSortedLists.java
QueueSorter.java
RemoveDuplicate.java
SequenceInStack.java
StackSorter.java
module-info.java
QueueSorter... StackSorter... RemoveDuplic... SequenceInS... MergeSorted... CircularQue... GraphAddEdge... BFS.java DFS.java
1 package wiproassignment;
2 import java.util.List;
3 import java.util.Map;
4 import java.util.ArrayList;
5 import java.util.HashMap;
6 import java.util.Set;
7 import java.util.HashSet;
8 public class DFS {
9 private final Map<Integer, List<Integer>> adjList = new HashMap<>();
10
11 public void addNode(int node) {
12 adjList.putIfAbsent(node, new ArrayList<>());
13 }
14
15 public void addEdge(int from, int to) {
16 addNode(from);
17 addNode(to);
18
19 adjList.get(from).add(to);
20 adjList.get(to).add(from);
21 }
22 public void dfs(int start) {
23 Set<Integer> visited = new HashSet<>();
24 dfsUtil(start, visited);
25 }
26 private void dfsUtil(int node, Set<Integer> visited) {
27 visited.add(node);
Writable Smart Insert 7 : 26 : 177
CAD/INR +0.55%
Search
ENG IN
14:47 01-06-2024


wipro - wiproAssignment/src/wiproassignment/DFS.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
DataStructures
JRE System Library [JavaSE-17]
src
com.wipro.assignments
MiddleEleList.java
com.wipro.ds
com.wipro.graphalg
com.wipro.graphs
com.wipro.linear
com.wipro.nonlinear
com.wipro.pattern
com.wipro.search
com.wipro.sort
JRE System Library [jdk1.8.0_202]
ems
firstjava
pra
practice
wiproAssignment
JRE System Library [JavaSE-17]
src
wiproassignment
BFS.java
CircularQueueBinarySearch.java
DFS.java
GraphAddEdge.java
LinkedListMiddle.java
MergeSortedLists.java
QueueSorter.java
RemoveDuplicate.java
SequenceInStack.java
StackSorter.java
module-info.java
QueueSorter... StackSorter... RemoveDuplic... SequenceInS... MergeSorted... CircularQue... GraphAddEdge... BFS.java DFS.java
20 adjList.get(to).add(from);
21 }
22 public void dfs(int start) {
23 Set<Integer> visited = new HashSet<>();
24 dfsUtil(start, visited);
25 }
26 private void dfsUtil(int node, Set<Integer> visited) {
27 visited.add(node);
28 System.out.print(node + " ");
29
30 for (int neighbor : adjList.get(node)) {
31 if (!visited.contains(neighbor)) {
32 dfsUtil(neighbor, visited);
33 }
34 }
35 }
36 public static void main(String[] args) {
37 DFS DFSGraph = new DFS();
38 DFSGraph.addEdge(0, 3);
39 DFSGraph.addEdge(0, 5);
40 DFSGraph.addEdge(1, 2);
41 DFSGraph.addEdge(3, 5);
42
43 System.out.println("DFS starting from node 0:");
44 DFSGraph.dfs(0);
45 }
46 }
Writable Smart Insert 7 : 26 : 177
CAD/INR +0.55%
Search
ENG IN
14:48 01-06-2024


wipro - wiproAssignment/src/wiproassignment/DFS.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
DataStructures
JRE System Library [JavaSE-17]
src
com.wipro.assignments
MiddleEleList.java
com.wipro.ds
com.wipro.graphalg
com.wipro.graphs
com.wipro.linear
com.wipro.nonlinear
com.wipro.pattern
com.wipro.search
com.wipro.sort
JRE System Library [jdk1.8.0_202]
ems
firstjava
pra
practice
wiproAssignment
JRE System Library [JavaSE-17]
src
wiproassignment
BFS.java
CircularQueueBinarySearch.java
DFS.java
GraphAddEdge.java
LinkedListMiddle.java
MergeSortedLists.java
QueueSorter.java
RemoveDuplicate.java
SequenceInStack.java
StackSorter.java
module-info.java
QueueSorter...
StackSorter...
RemoveDuplic...
SequenceInS...
MergeSorted...
CircularQue...
GraphAddEdge...
BFS.java
DFS.java
29
30 for (int neighbor : adjList.get(node)) {
31 if (!visited.contains(neighbor)) {
32 dfsUtil(neighbor, visited);
33 }
34 }
35 }
36 public static void main(String[] args) {
37 DFS DFSGraph = new DFS();
38 DFSGraph.addEdge(0, 3);
39 DFSGraph.addEdge(0, 5);
40 DFSGraph.addEdge(1, 2);
41 DFSGraph.addEdge(3, 5);
42
43 System.out.println("DFS starting from node 0:");
44 DFSGraph.dfs(0);
45 }
46 }
Line: 37
Console
<terminated> DFS [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (01-Jun-2024, 2:48:55 pm – 2:48:55 pm) [pid: 30064]
DFS starting from node 0:
0 3 5
Writable
Smart Insert
7 : 26 : 177
36°C
Mostly cloudy
Search
ENG
IN
14:49
01-06-2024


wipro - wiproAssignment/src/wiproassignment/StringOperations.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
CycleDetect... MergeSorted... CircularQue... GraphAddEdge... BFS.java DFS.java Dijkstra.java Kruskal.java *StringOpera...
1 package wiproassignment;
2 public class StringOperations {
3 public static String middleSubstring(String str1, String str2, int length) {
4 String concatenated = str1.concat(str2);
5 StringBuilder reversed = new StringBuilder(concatenated).reverse();
6 int reversedLength = reversed.length();
7 if (reversedLength == 0 || length > reversedLength) {
8 return "";
9 }
10 int startIndex = (reversedLength - length) / 2;
11 String middleSubstring = reversed.substring(startIndex, startIndex + length);
12 return middleSubstring;
13 }
14
15 public static void main(String[] args) {
16 String str1 = "Hello";
17 String str2 = "World";
18 int length = 5;
19
20 String result = middleSubstring(str1, str2, length);
21 System.out.println("Middle substring: " + result);
22 }
23 }
24
Console
<terminated> StringOperations [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (01-Jun-2024, 3:32:18 pm – 3:32:21 pm) [pid: 16448]
Middle substring: roWol
Writable Smart Insert 1 : 25 : 24
35°C Partly sunny
Search
ENG IN
15:34 01-06-2024