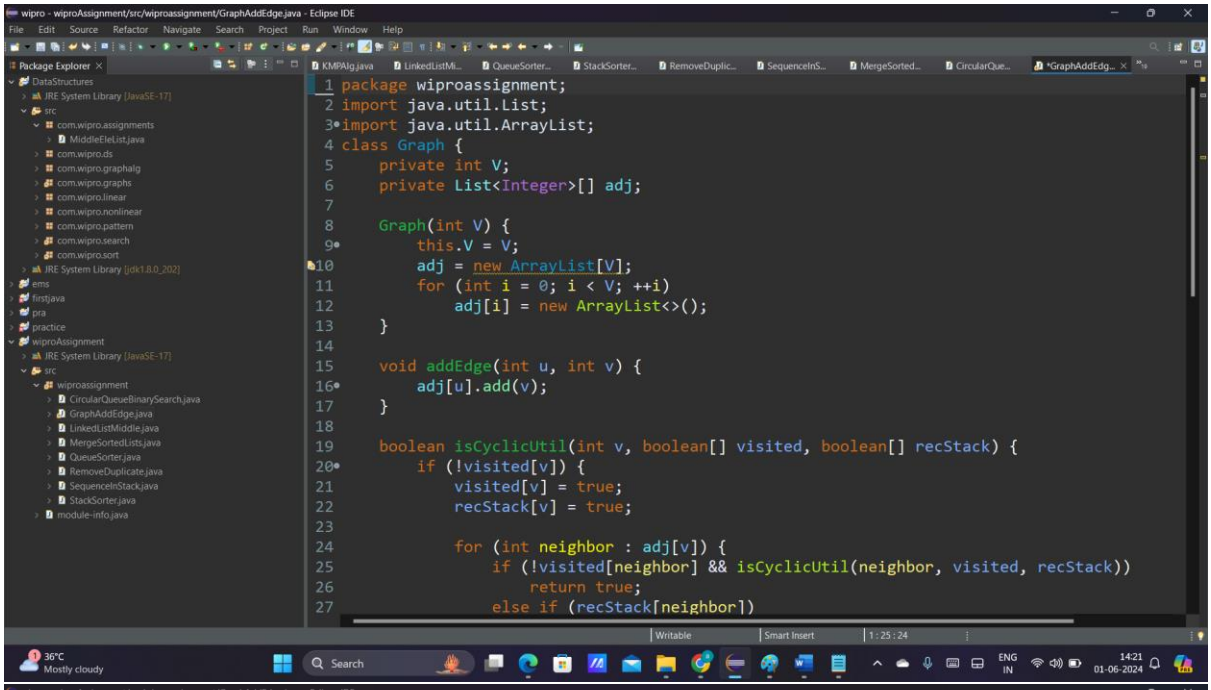
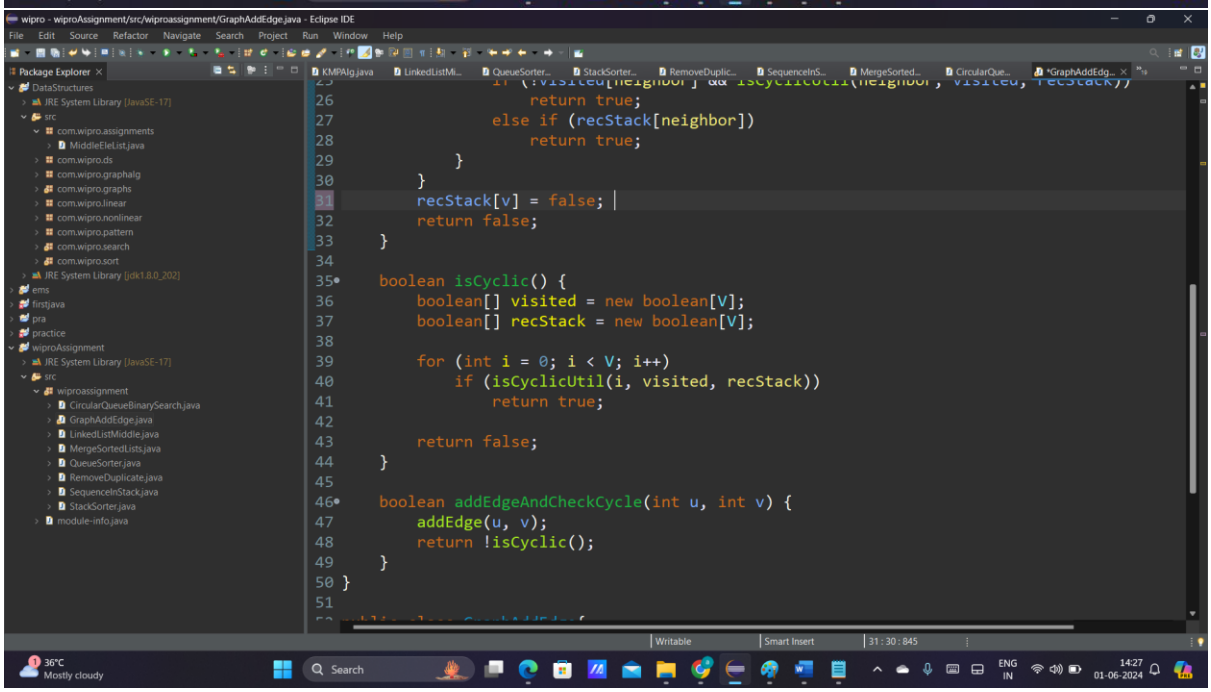
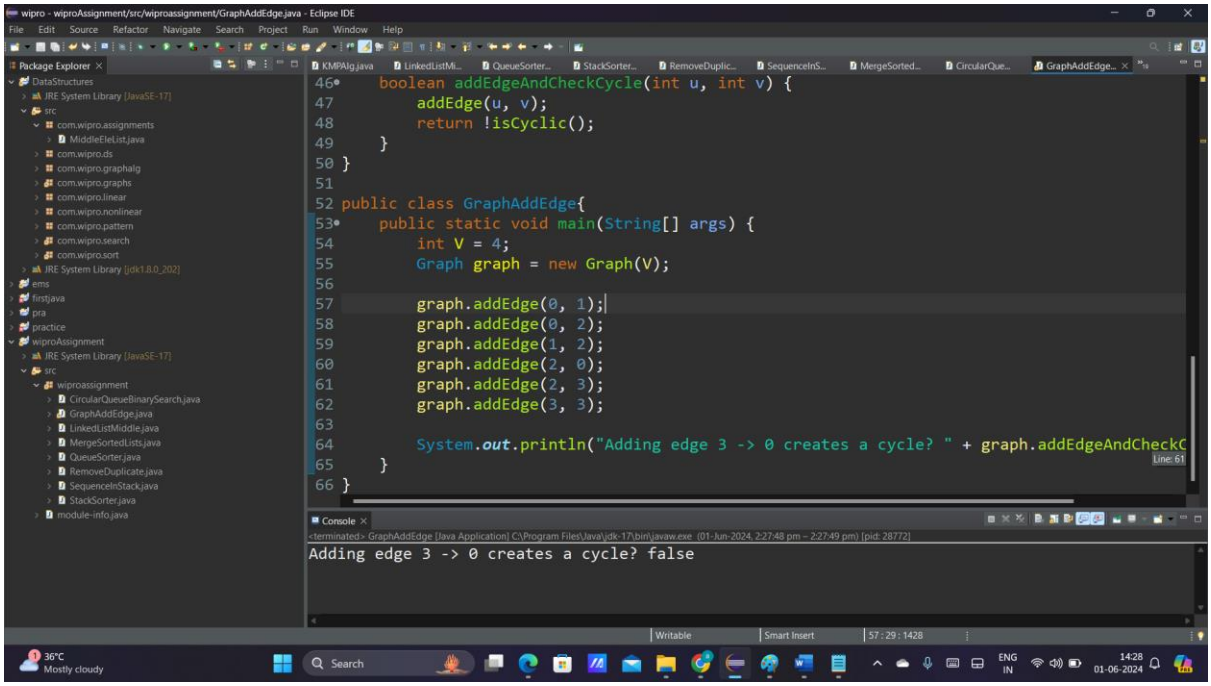
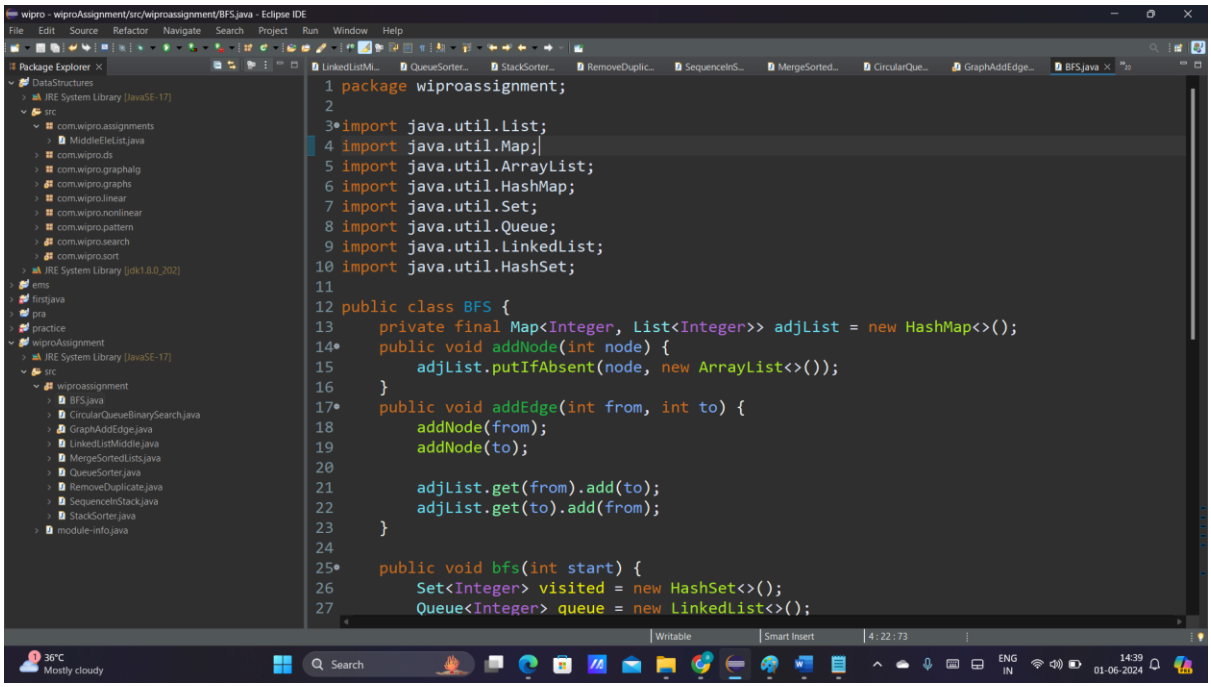

wipro - wiproAssignment/src/wiproassignment/GraphAddEdge.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
DataStructures
JRE System Library [JavaSE-17]
src
com.wipro.assignments
MiddleEleList.java
com.wipro.ds
com.wipro.graphalg
com.wipro.graphs
com.wipro.linear
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
CircularQueueBinarySearch.java
GraphAddEdge.java
LinkedListMiddle.java
MergeSortedLists.java
QueueSorter.java
RemoveDuplicate.java
SequenceInStack.java
StackSorter.java
module-info.java
KMPAlg.java
LinkedListMi...
QueueSorter...
StackSorter...
RemoveDuplic...
SequenceInS...
MergeSorted...
CircularQue...
*GraphAddEdg...
1 package wiproassignment;
2 import java.util.List;
3 import java.util.ArrayList;
4 class Graph {
5 private int V;
6 private List<Integer>[] adj;
7
8 Graph(int V) {
9 this.V = V;
10 adj = new ArrayList[V];
11 for (int i = 0; i < V; ++i)
12 adj[i] = new ArrayList<>();
13 }
14
15 void addEdge(int u, int v) {
16 adj[u].add(v);
17 }
18
19 boolean isCyclicUtil(int v, boolean[] visited, boolean[] recStack) {
20 if (!visited[v]) {
21 visited[v] = true;
22 recStack[v] = true;
23
24 for (int neighbor : adj[v]) {
25 if (!visited[neighbor] && isCyclicUtil(neighbor, visited, recStack))
26 return true;
27 else if (recStack[neighbor])
Writable
Smart Insert
1 : 25 : 24
36°C
Mostly cloudy
Search
ENG
IN
14:21
01-06-2024

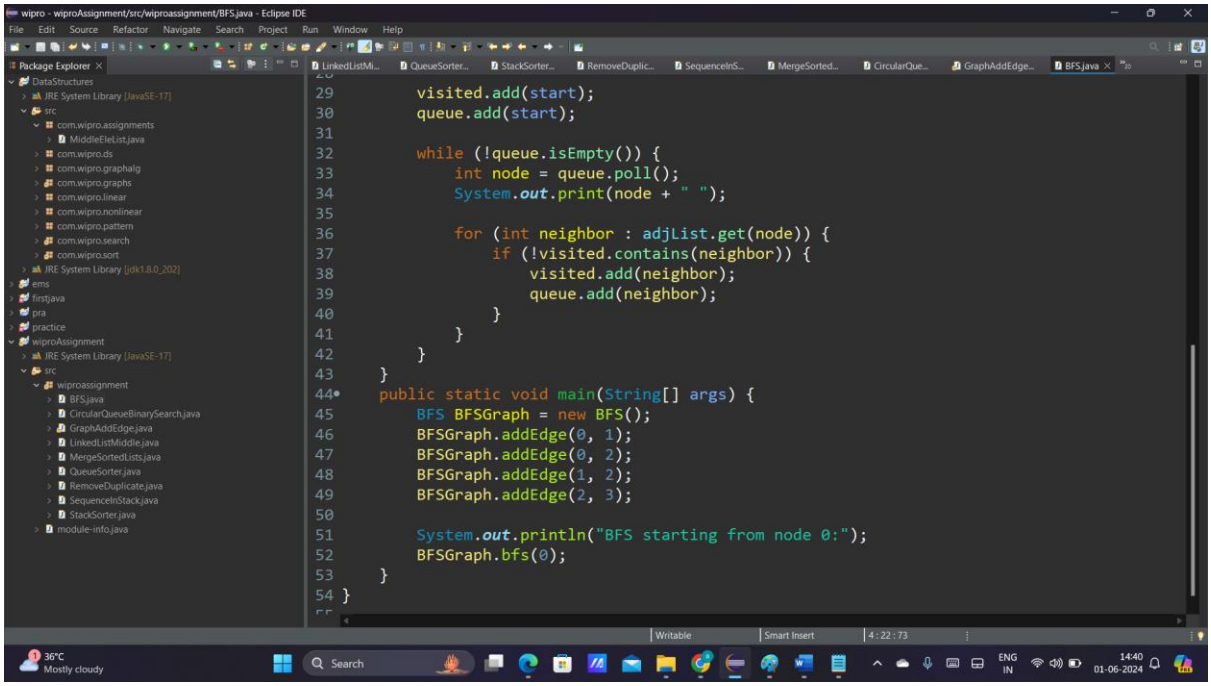

wipro - wiproAssignment/src/wiproassignment/GraphAddEdge.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
DataStructures
JRE System Library [JavaSE-17]
src
com.wipro.assignments
MiddleEleList.java
com.wipro.ds
com.wipro.graphalg
com.wipro.graphs
com.wipro.linear
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
CircularQueueBinarySearch.java
GraphAddEdge.java
LinkedListMiddle.java
MergeSortedLists.java
QueueSorter.java
RemoveDuplicate.java
SequenceInStack.java
StackSorter.java
module-info.java
KMPAlg.java
LinkedListMi...
QueueSorter...
StackSorter...
RemoveDuplic...
SequenceInS...
MergeSorted...
CircularQue...
*GraphAddEdg...
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();
}
}
Writable
Smart Insert
31 : 30 : 845
36°C
Mostly cloudy
Search
ENG
IN
14:27
01-06-2024

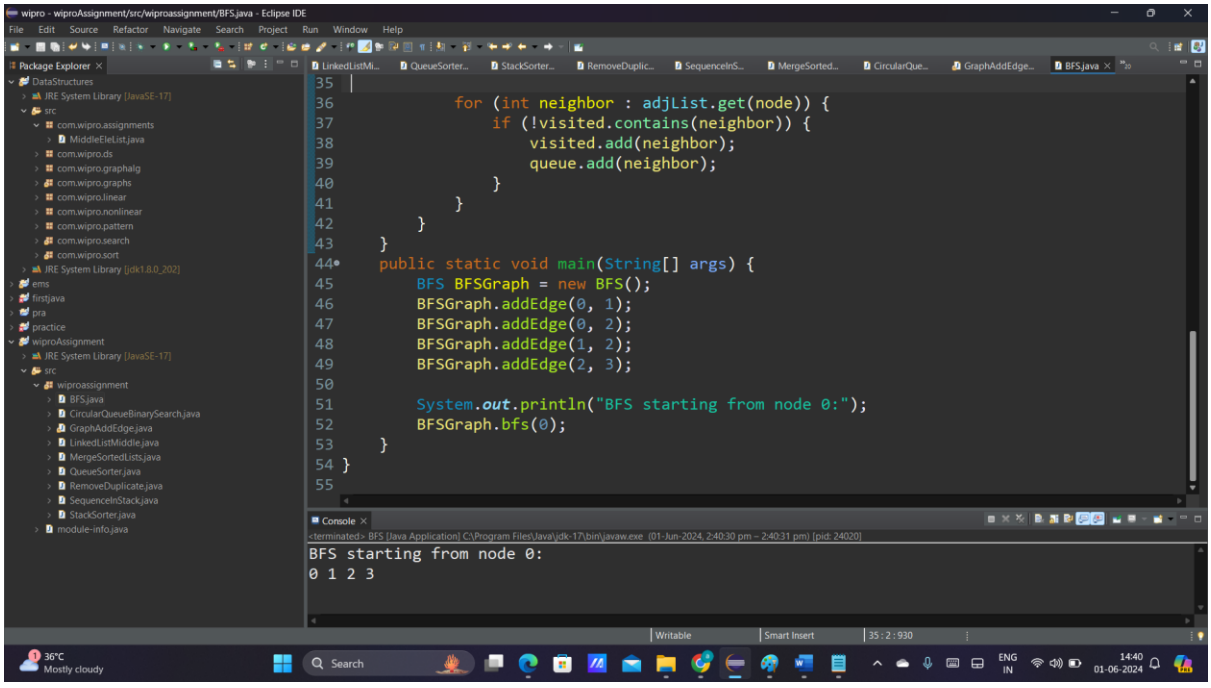



Task 5: Breadth-First Search (BFS) Implementation

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.

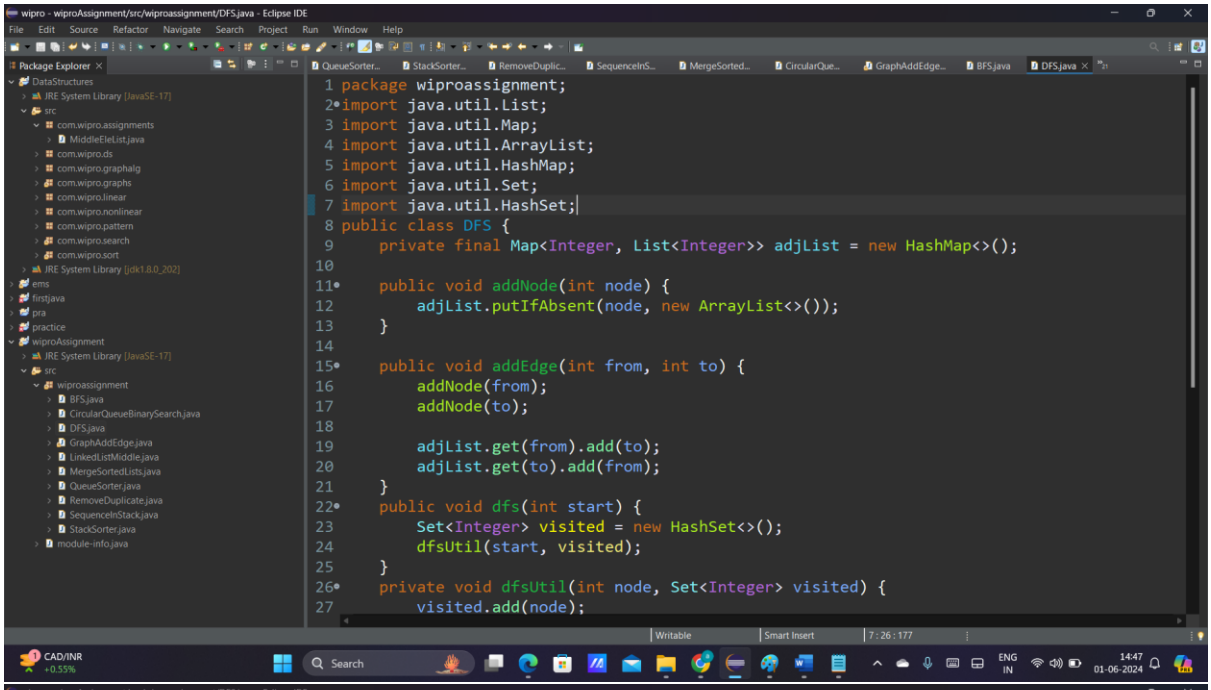


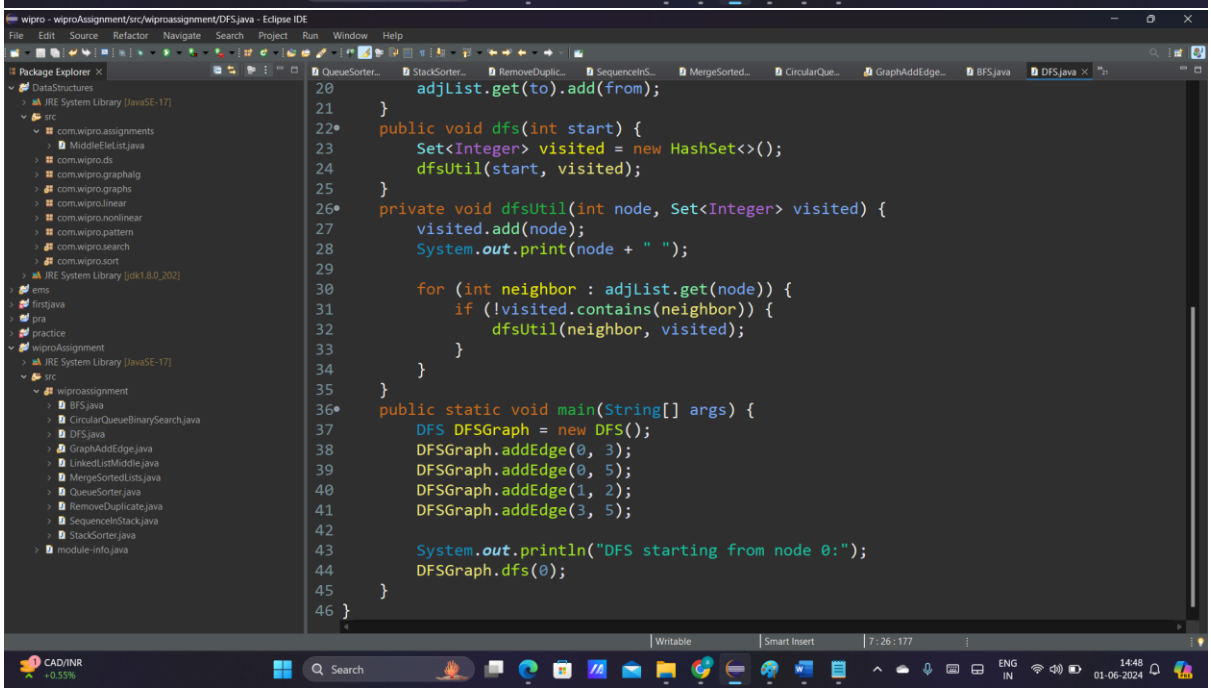


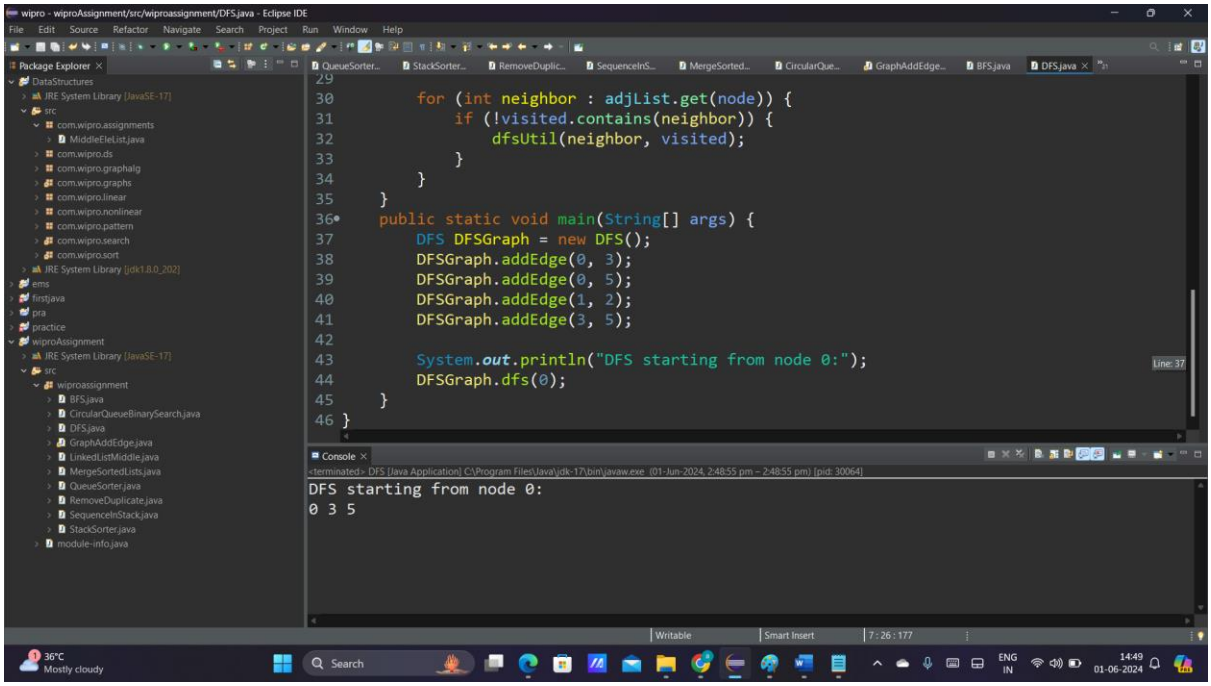




Task 6: Depth-First Search (DFS) Recursive

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

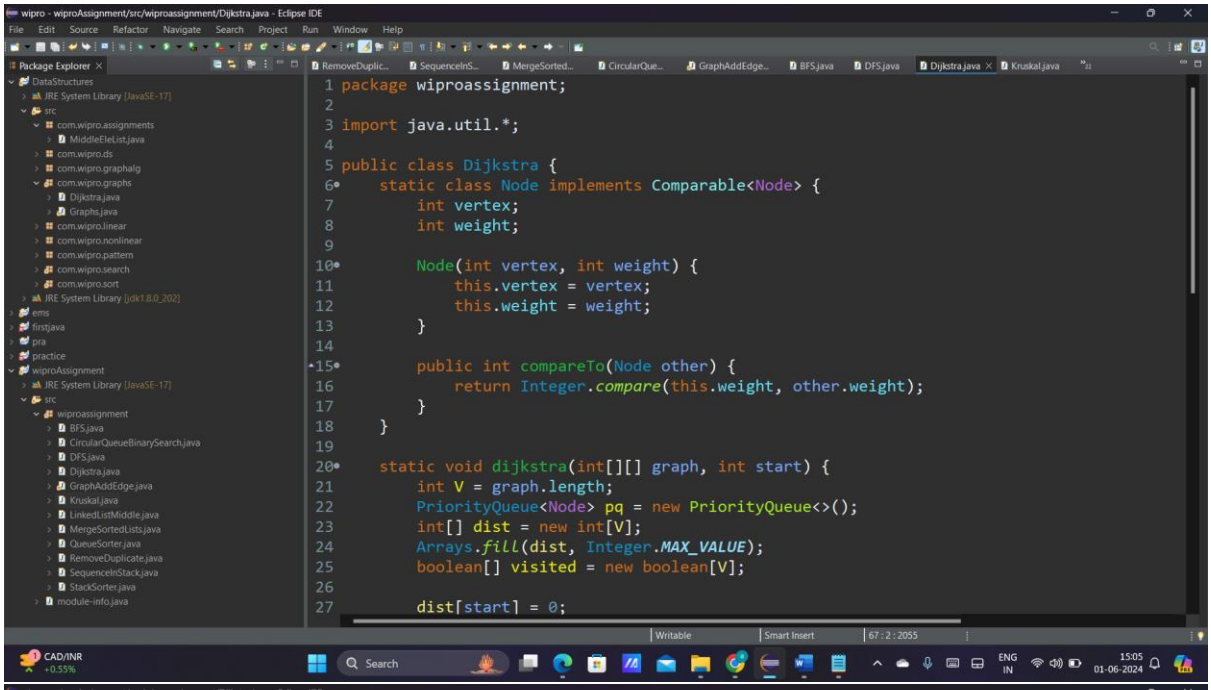

wipro - wiproAssignment/src/wiproassignment/DFS.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
DataStructures
JRE System Library [JavaSE-17]
src
com.wipro.assignments
MiddleEleList.java
com.wipro.ds
com.wipro.graphalg
com.wipro.graphs
com.wipro.linear
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

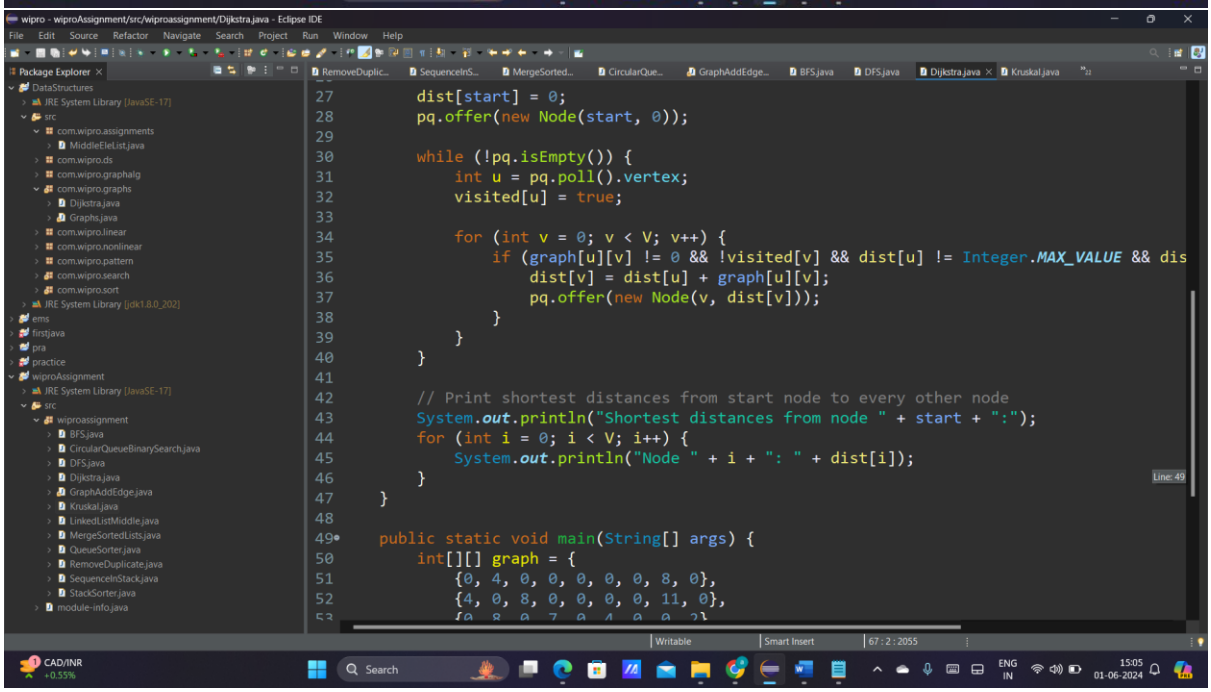

wipro - wiproAssignment/src/wiproassignment/DFS.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
DataStructures
JRE System Library [JavaSE-17]
src
com.wipro.assignments
MiddleEleList.java
com.wipro.ds
com.wipro.graphalg
com.wipro.graphs
com.wipro.linear
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

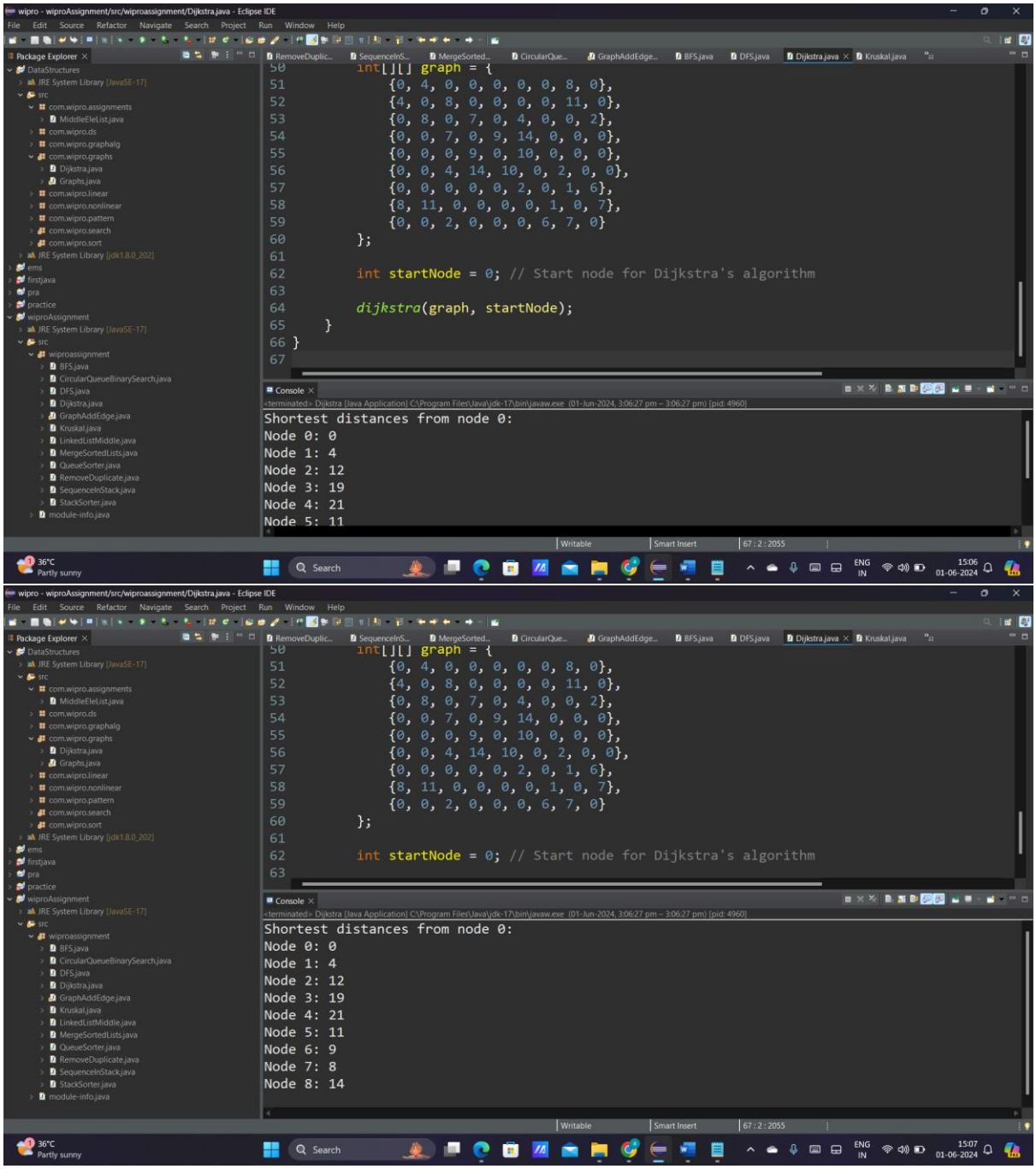



Task 1: Dijkstra’s Shortest Path Finder

Code Dijkstra’s algorithm to find the shortest path from a start node to every other node in a weighted graph with positive weights.


wipro - wiproAssignment/src/wiproassignment/Dijkstra.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
Dijkstra.java
Graphs.java
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
Dijkstra.java
GraphAddEdge.java
Kruskal.java
LinkedListMiddle.java
MergeSortedLists.java
QueueSorter.java
RemoveDuplicate.java
SequenceInStack.java
StackSorter.java
module-info.java
RemoveDuplic...
SequenceInS...
MergeSorted...
CircularQue...
GraphAddEdge...
BFS.java
DFS.java
Dijkstra.java
Kruskal.java
1 package wiproassignment;
2
3 import java.util.*;
4
5 public class Dijkstra {
6 static class Node implements Comparable<Node> {
7 int vertex;
8 int weight;
9
10 Node(int vertex, int weight) {
11 this.vertex = vertex;
12 this.weight = weight;
13 }
14
15 public int compareTo(Node other) {
16 return Integer.compare(this.weight, other.weight);
17 }
18 }
19
20 static void dijkstra(int[][] graph, int start) {
21 int V = graph.length;
22 PriorityQueue<Node> pq = new PriorityQueue<>();
23 int[] dist = new int[V];
24 Arrays.fill(dist, Integer.MAX_VALUE);
25 boolean[] visited = new boolean[V];
26
27 dist[start] = 0;
Writable
Smart Insert
67 : 2 : 2055
CAD/INR
+0.55%
Search
ENG
IN
15:05
01-06-2024


wipro - wiproAssignment/src/wiproassignment/Dijkstra.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
Dijkstra.java
Graphs.java
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
Dijkstra.java
GraphAddEdge.java
Kruskal.java
LinkedListMiddle.java
MergeSortedLists.java
QueueSorter.java
RemoveDuplicate.java
SequenceInStack.java
StackSorter.java
module-info.java
RemoveDuplic...
SequenceInS...
MergeSorted...
CircularQue...
GraphAddEdge...
BFS.java
DFS.java
Dijkstra.java
Kruskal.java
27 dist[start] = 0;
28 pq.offer(new Node(start, 0));
29
30 while (!pq.isEmpty()) {
31 int u = pq.poll().vertex;
32 visited[u] = true;
33
34 for (int v = 0; v < V; v++) {
35 if (graph[u][v] != 0 && !visited[v] && dist[u] != Integer.MAX_VALUE && dis
36 dist[v] = dist[u] + graph[u][v];
37 pq.offer(new Node(v, dist[v]));
38 }
39 }
40 }
41
42 // Print shortest distances from start node to every other node
43 System.out.println("Shortest distances from node " + start + ":");
44 for (int i = 0; i < V; i++) {
45 System.out.println("Node " + i + ": " + dist[i]);
46 }
47 }
48
49 public static void main(String[] args) {
50 int[][] graph = {
51 {0, 4, 0, 0, 0, 0, 0, 8, 0},
52 {4, 0, 8, 0, 0, 0, 0, 11, 0},
Line: 49
Writable
Smart Insert
67 : 2 : 2055
CAD/INR
+0.55%
Search
ENG
IN
15:05
01-06-2024

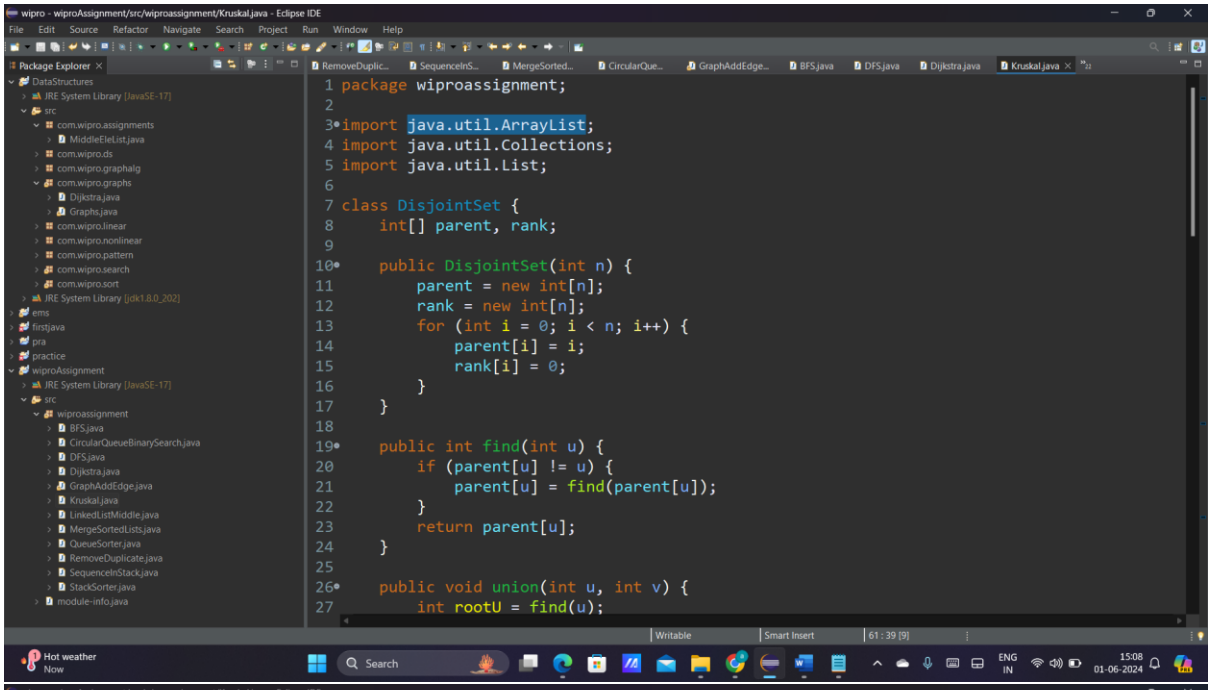


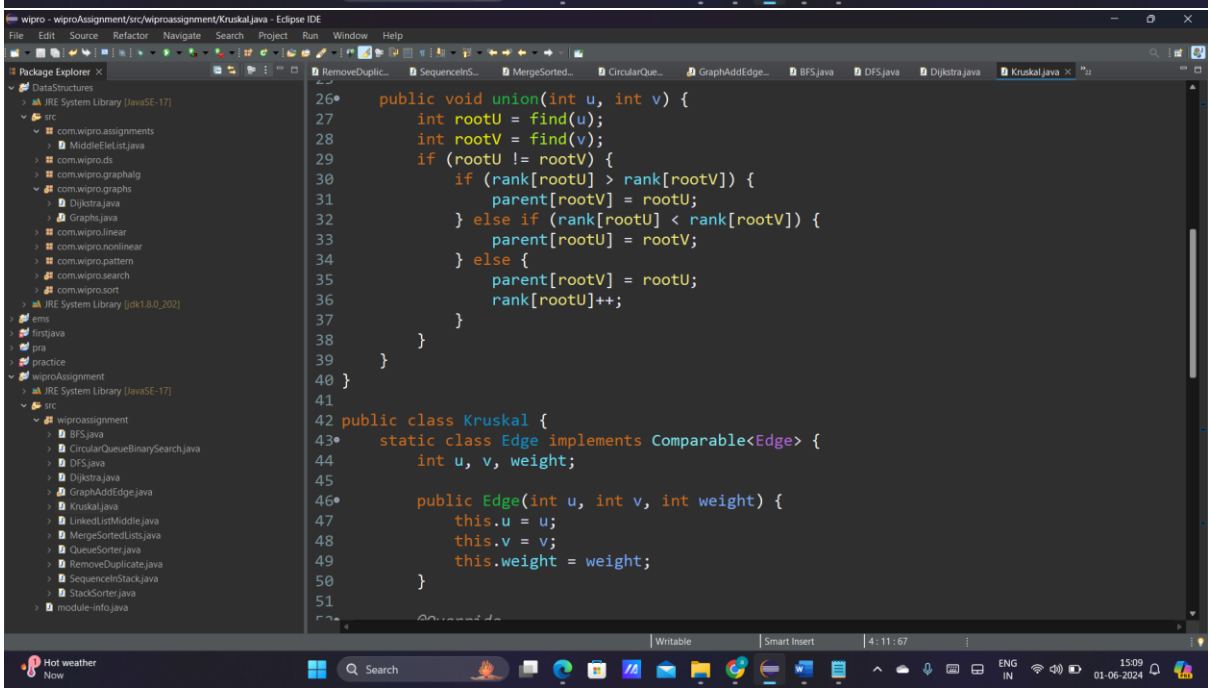


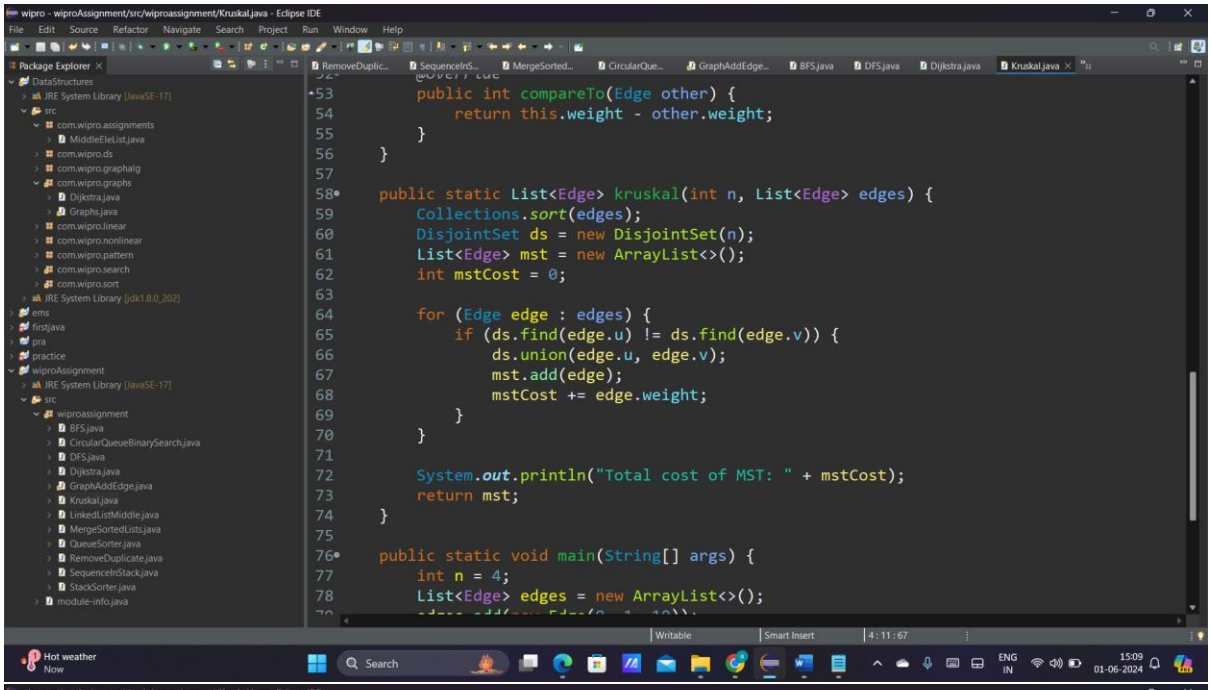


Task 2: Kruskal's Algorithm for MST

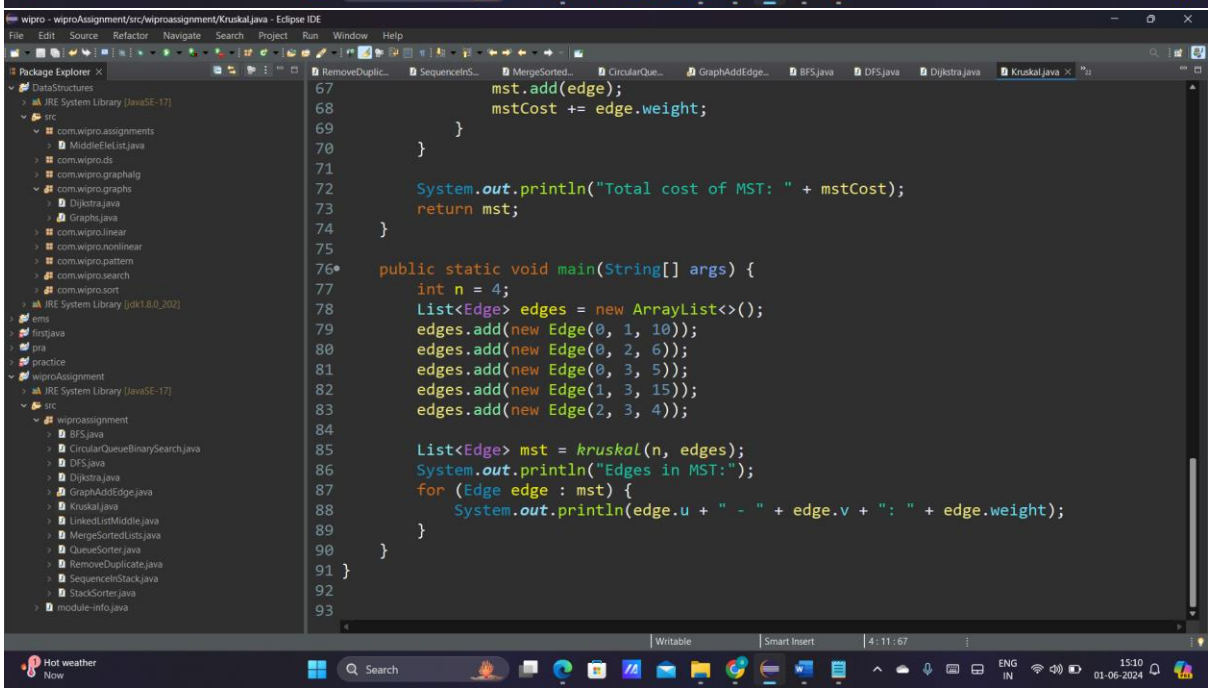
Implement Kruskal's algorithm to find the minimum spanning tree of a given connected, undirected graph with non-negative edge weights.


wipro - wiproAssignment/src/wiproassignment/Kruskal.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
Dijkstra.java
Graphs.java
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
Dijkstra.java
GraphAddEdge.java
Kruskal.java
LinkedListMiddle.java
MergeSortedLists.java
QueueSorter.java
RemoveDuplicate.java
SequenceInStack.java
StackSorter.java
module-info.java
RemoveDuplic...
SequenceInS...
MergeSorted...
CircularQue...
GraphAddEdge...
BFS.java
DFS.java
Dijkstra.java
Kruskal.java
1 package wiproassignment;
2
3 import java.util.ArrayList;
4 import java.util.Collections;
5 import java.util.List;
6
7 class DisjointSet {
8 int[] parent, rank;
9
10 public DisjointSet(int n) {
11 parent = new int[n];
12 rank = new int[n];
13 for (int i = 0; i < n; i++) {
14 parent[i] = i;
15 rank[i] = 0;
16 }
17 }
18
19 public int find(int u) {
20 if (parent[u] != u) {
21 parent[u] = find(parent[u]);
22 }
23 return parent[u];
24 }
25
26 public void union(int u, int v) {
27 int rootU = find(u);
Writable
Smart Insert
61 : 39 [9]
Hot weather
Now
Search
ENG
IN
15:08
01-06-2024

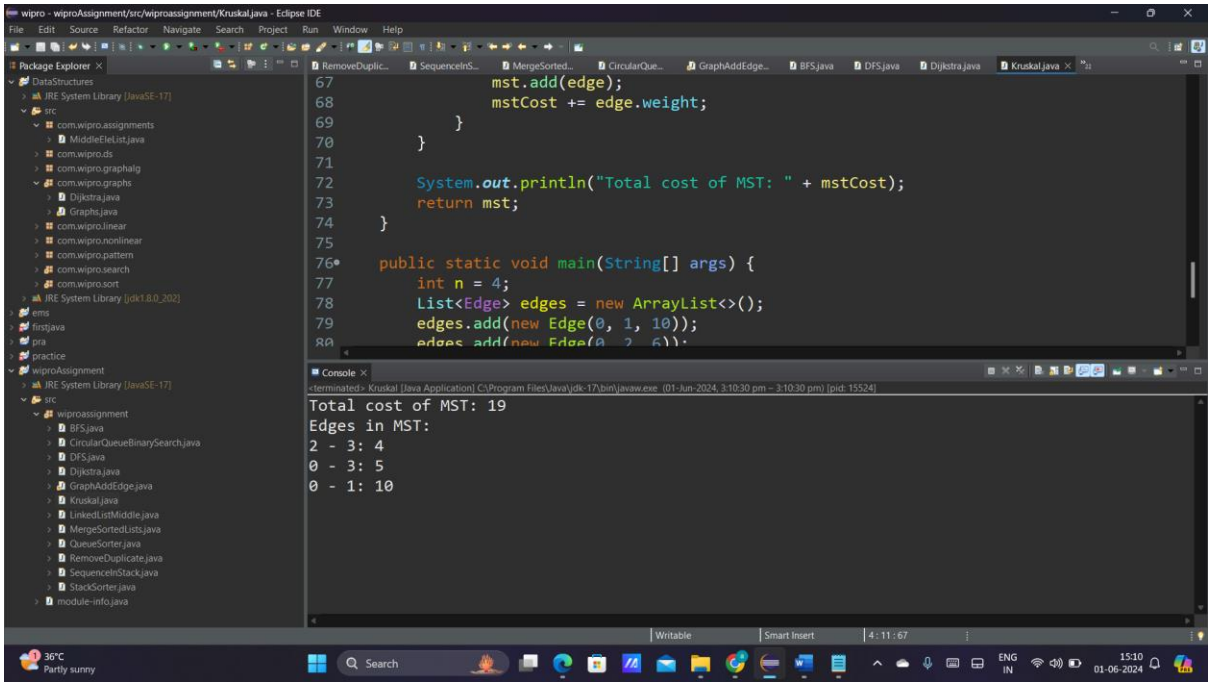

wipro - wiproAssignment/src/wiproassignment/Kruskal.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
DataStructures
JRE System Library [JavaSE-17]
src
com.wipro.assignments
MiddleEleList.java
com.wipro.ds
com.wipro.graphalg
com.wipro.graphs
Dijkstra.java
Graphs.java
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
Dijkstra.java
GraphAddEdge.java
Kruskal.java
LinkedListMiddle.java
MergeSortedLists.java
QueueSorter.java
RemoveDuplicate.java
SequenceInStack.java
StackSorter.java
module-info.java
RemoveDuplic...
SequenceInS...
MergeSorted...
CircularQue...
GraphAddEdge...
BFS.java
DFS.java
Dijkstra.java
Kruskal.java
26 public void union(int u, int v) {
27 int rootU = find(u);
28 int rootV = find(v);
29 if (rootU != rootV) {
30 if (rank[rootU] > rank[rootV]) {
31 parent[rootV] = rootU;
32 } else if (rank[rootU] < rank[rootV]) {
33 parent[rootU] = rootV;
34 } else {
35 parent[rootV] = rootU;
36 rank[rootU]++;
37 }
38 }
39 }
40 }
41
42 public class Kruskal {
43 static class Edge implements Comparable<Edge> {
44 int u, v, weight;
45
46 public Edge(int u, int v, int weight) {
47 this.u = u;
48 this.v = v;
49 this.weight = weight;
50 }
51
Writable
Smart Insert
4 : 11 : 67
Hot weather
Now
Search
ENG
IN
15:09
01-06-2024



```
        public int compareTo(Edge other) {
            return this.weight - other.weight;
        }
    }

    public static List<Edge> kruskal(int n, List<Edge> edges) {
        Collections.sort(edges);
        DisjointSet ds = new DisjointSet(n);
        List<Edge> mst = new ArrayList<>();
        int mstCost = 0;

        for (Edge edge : edges) {
            if (ds.find(edge.u) != ds.find(edge.v)) {
                ds.union(edge.u, edge.v);
                mst.add(edge);
                mstCost += edge.weight;
            }
        }

        System.out.println("Total cost of MST: " + mstCost);
        return mst;
    }

    public static void main(String[] args) {
        int n = 4;
        List<Edge> edges = new ArrayList<>();
```

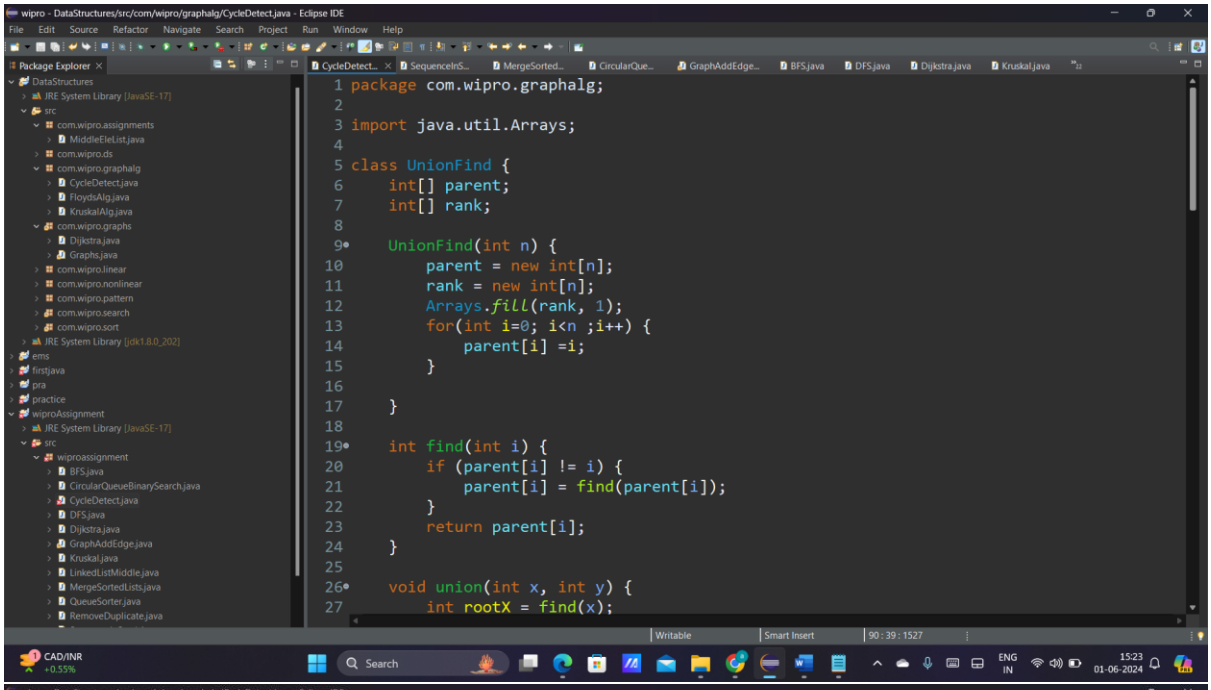


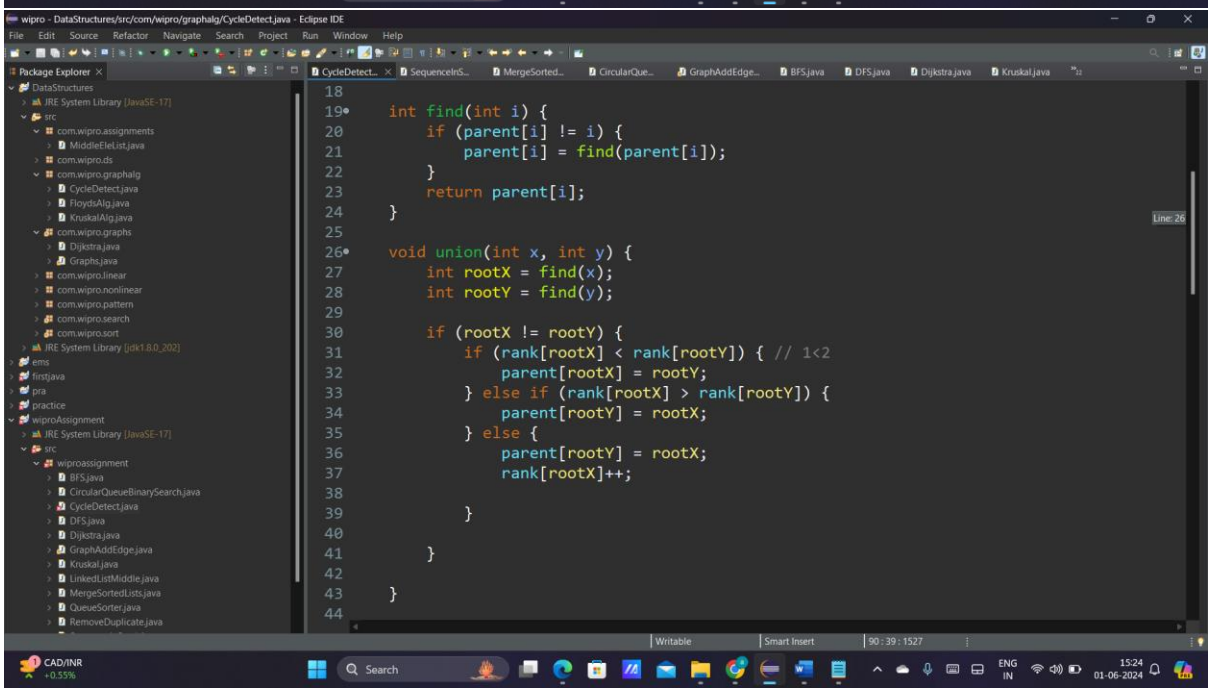
```
                mst.add(edge);
                mstCost += edge.weight;
            }
        }

        System.out.println("Total cost of MST: " + mstCost);
        return mst;
    }

    public static void main(String[] args) {
        int n = 4;
        List<Edge> edges = new ArrayList<>();
        edges.add(new Edge(0, 1, 10));
        edges.add(new Edge(0, 2, 6));
        edges.add(new Edge(0, 3, 5));
        edges.add(new Edge(1, 3, 15));
        edges.add(new Edge(2, 3, 4));

        List<Edge> mst = kruskal(n, edges);
        System.out.println("Edges in MST:");
        for (Edge edge : mst) {
            System.out.println(edge.u + " - " + edge.v + ": " + edge.weight);
        }
    }
}
```

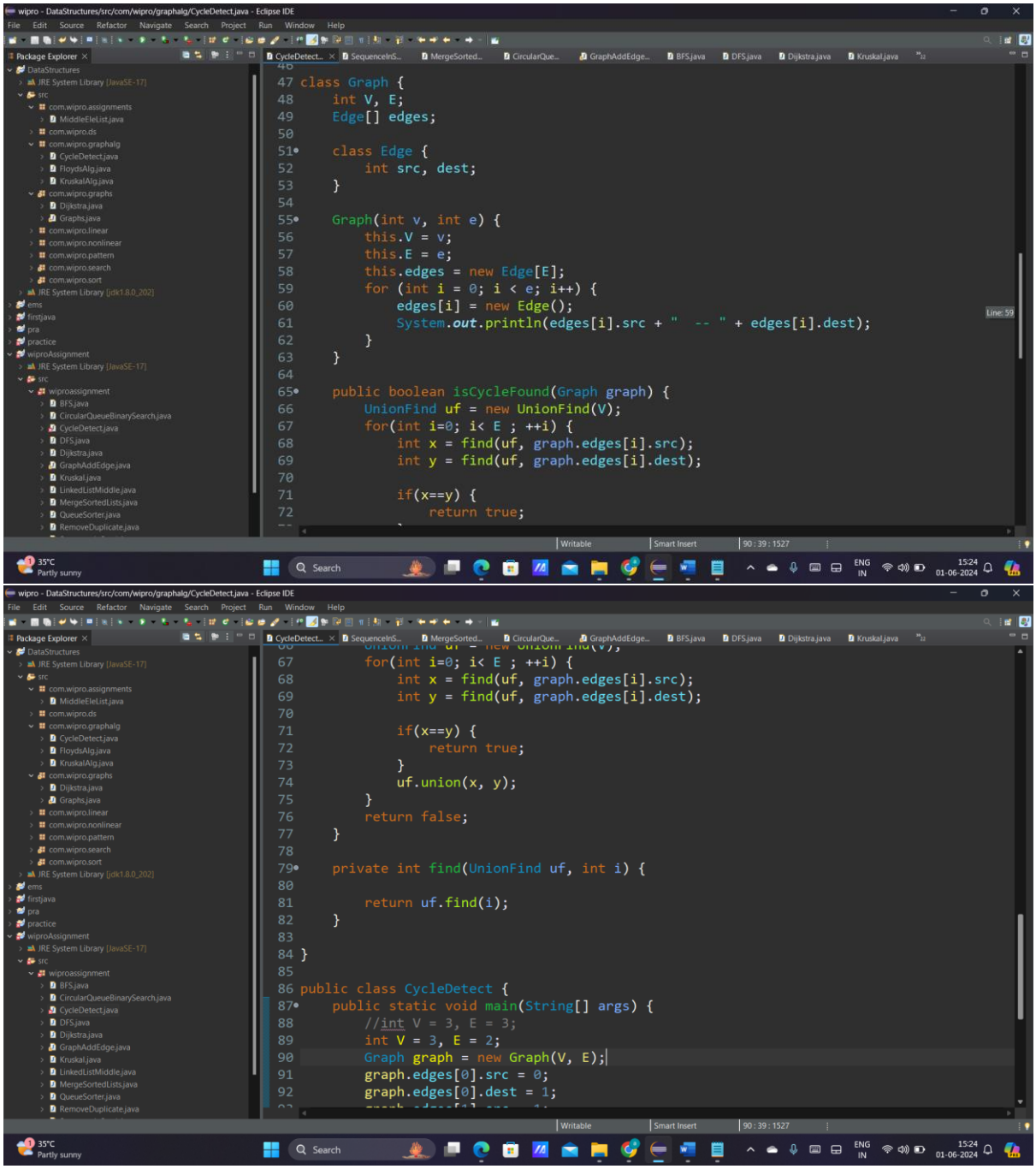



Task 3: Union-Find for Cycle Detection

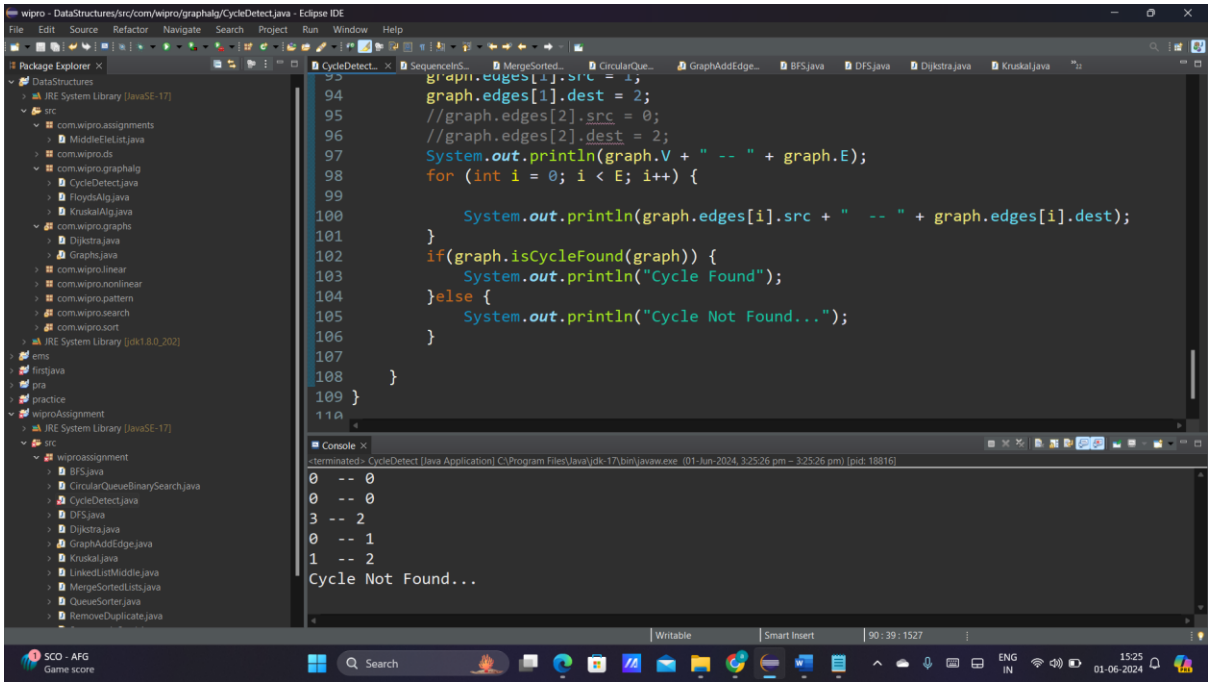
Write a Union-Find data structure with path compression. Use this data structure to detect a cycle in an undirected graph.


wipro - DataStructures/src/com/wipro/graphalg/CycleDetect.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
CycleDetect.java
FloydsAlg.java
KruskalAlg.java
com.wipro.graphs
Dijkstra.java
Graphs.java
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
CycleDetect.java
DFS.java
Dijkstra.java
GraphAddEdge.java
Kruskal.java
LinkedListMiddle.java
MergeSortedLists.java
QueueSorter.java
RemoveDuplicate.java
CycleDetect... SequenceInS... MergeSorted... CircularQue... GraphAddEdge... BFS.java DFS.java Dijkstra.java Kruskal.java
1 package com.wipro.graphalg;
2
3 import java.util.Arrays;
4
5 class UnionFind {
6 int[] parent;
7 int[] rank;
8
9 UnionFind(int n) {
10 parent = new int[n];
11 rank = new int[n];
12 Arrays.fill(rank, 1);
13 for(int i=0; i<n ;i++) {
14 parent[i] =i;
15 }
16
17 }
18
19 int find(int i) {
20 if (parent[i] != i) {
21 parent[i] = find(parent[i]);
22 }
23 return parent[i];
24 }
25
26 void union(int x, int y) {
27 int rootX = find(x);
Writable Smart Insert 90 : 39 : 1527
CAD/INR +0.55%
Search
ENG IN
15:23 01-06-2024


wipro - DataStructures/src/com/wipro/graphalg/CycleDetect.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
CycleDetect... SequenceInS... MergeSorted... CircularQue... GraphAddEdge... BFS.java DFS.java Dijkstra.java Kruskal.java
18
19 int find(int i) {
20 if (parent[i] != i) {
21 parent[i] = find(parent[i]);
22 }
23 return parent[i];
24 }
25
26 void union(int x, int y) {
27 int rootX = find(x);
28 int rootY = find(y);
29
30 if (rootX != rootY) {
31 if (rank[rootX] < rank[rootY]) { // 1<2
32 parent[rootX] = rootY;
33 } else if (rank[rootX] > rank[rootY]) {
34 parent[rootY] = rootX;
35 } else {
36 parent[rootY] = rootX;
37 rank[rootX]++;
38
39 }
40
41 }
42
43 }
44
Line: 26
Writable Smart Insert 90 : 39 : 1527
CAD/INR +0.55%
Search
ENG IN
15:24 01-06-2024

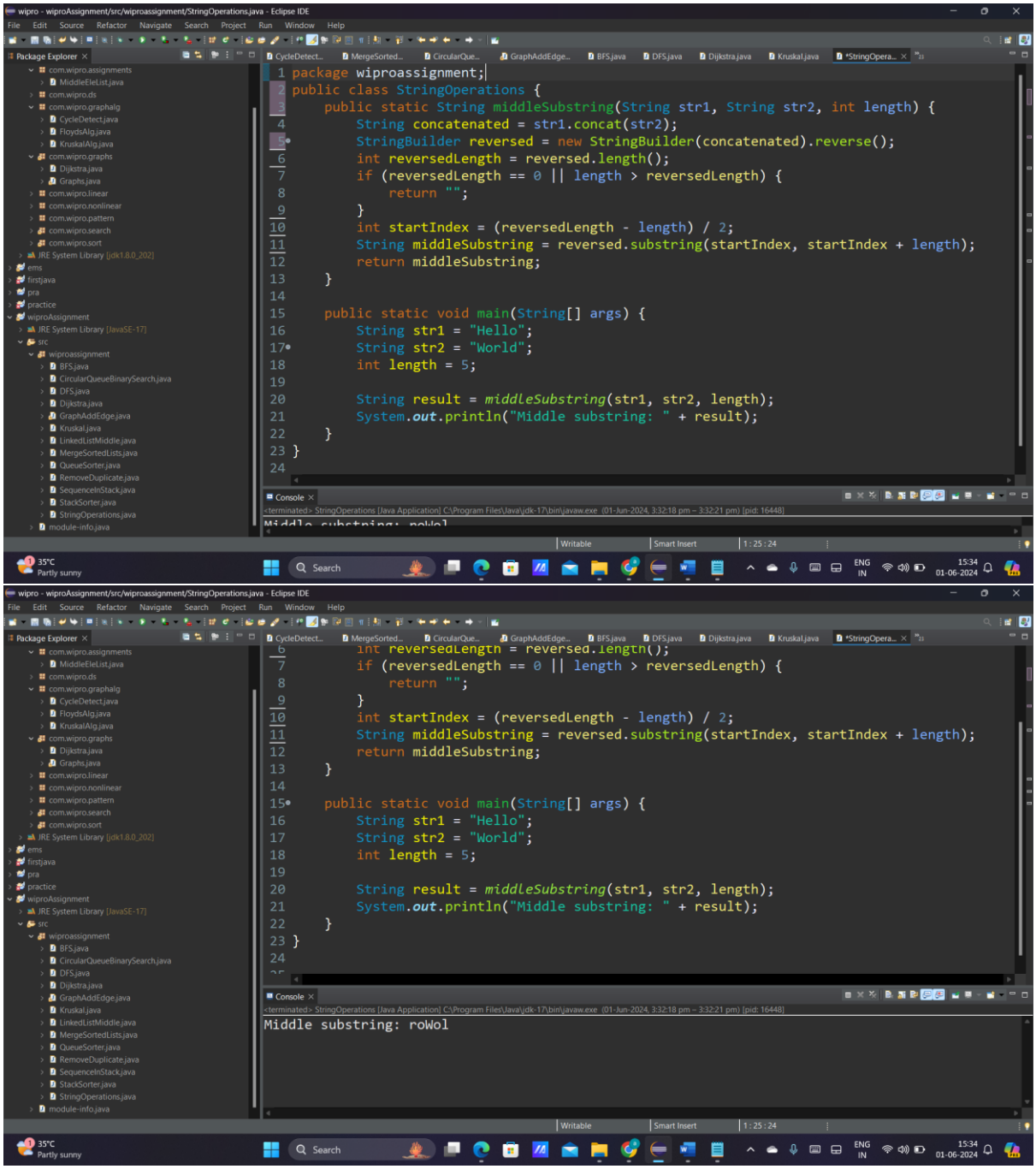

wipro - DataStructures/src/com/wipro/graphalg/CycleDetect.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
DataStructures
JRE System Library [JavaSE-17]
src
com.wipro.assignments
MiddleEleList.java
com.wipro.ds
com.wipro.graphalg
CycleDetect.java
FloydsAlg.java
KruskalAlg.java
com.wipro.graphs
Dijkstra.java
Graphs.java
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
CycleDetect.java
DFS.java
Dijkstra.java
GraphAddEdge.java
Kruskal.java
LinkedListMiddle.java
MergeSortedLists.java
QueueSorter.java
RemoveDuplicate.java
CycleDetect... SequenceInS... MergeSorted... CircularQue... GraphAddEdge... BFS.java DFS.java Dijkstra.java Kruskal.java
47 class Graph {
48 int V, E;
49 Edge[] edges;
50
51 class Edge {
52 int src, dest;
53 }
54
55 Graph(int v, int e) {
56 this.V = v;
57 this.E = e;
58 this.edges = new Edge[E];
59 for (int i = 0; i < e; i++) {
60 edges[i] = new Edge();
61 System.out.println(edges[i].src + " -- " + edges[i].dest);
62 }
63 }
64
65 public boolean isCycleFound(Graph graph) {
66 UnionFind uf = new UnionFind(V);
67 for(int i=0; i< E ; ++i) {
68 int x = find(uf, graph.edges[i].src);
69 int y = find(uf, graph.edges[i].dest);
70
71 if(x==y) {
72 return true;
Line: 59
Writable Smart Insert 90 : 39 : 1527
35°C Partly sunny
Search
ENG IN
15:24 01-06-2024


wipro - DataStructures/src/com/wipro/graphalg/CycleDetect.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
CycleDetect.java
FloydsAlg.java
KruskalAlg.java
com.wipro.graphs
Dijkstra.java
Graphs.java
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
CycleDetect.java
DFS.java
Dijkstra.java
GraphAddEdge.java
Kruskal.java
LinkedListMiddle.java
MergeSortedLists.java
QueueSorter.java
RemoveDuplicate.java
CycleDetect... SequenceInS... MergeSorted... CircularQue... GraphAddEdge... BFS.java DFS.java Dijkstra.java Kruskal.java
67 for(int i=0; i< E ; ++i) {
68 int x = find(uf, graph.edges[i].src);
69 int y = find(uf, graph.edges[i].dest);
70
71 if(x==y) {
72 return true;
73 }
74 uf.union(x, y);
75 }
76 return false;
77 }
78
79 private int find(UnionFind uf, int i) {
80
81 return uf.find(i);
82 }
83
84 }
85
86 public class CycleDetect {
87 public static void main(String[] args) {
88 //int V = 3, E = 3;
89 int V = 3, E = 2;
90 Graph graph = new Graph(V, E);
91 graph.edges[0].src = 0;
92 graph.edges[0].dest = 1;
Writable Smart Insert 90 : 39 : 1527
35°C Partly sunny
Search
ENG IN
15:24 01-06-2024



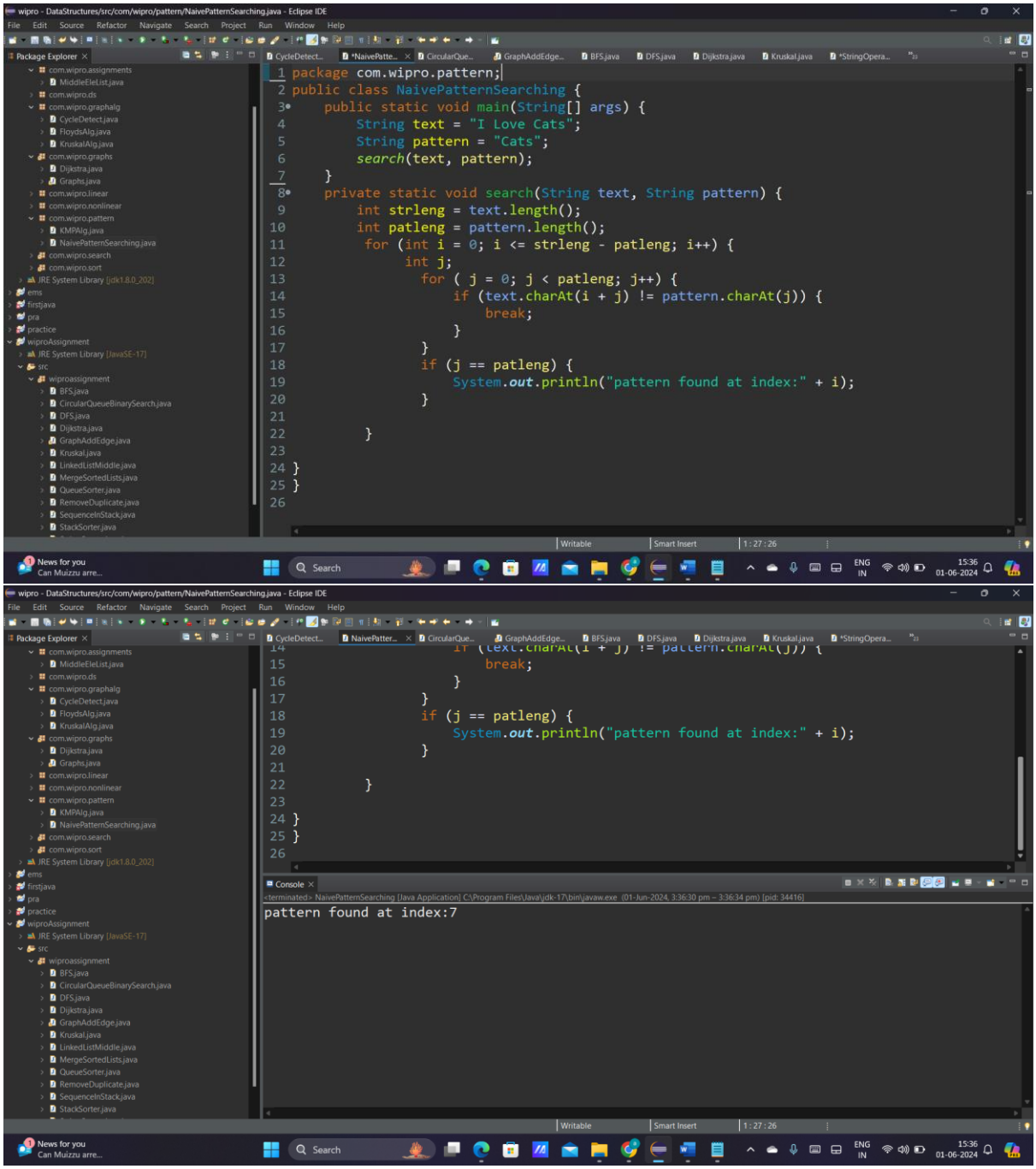

Task 1: String Operations

Write a method that takes two strings, concatenates them, reverses the result, and then extracts the middle substring of the given length. Ensure your method handles edge cases, such as an empty string or a substring length larger than the concatenated string.




Task 2: Naive Pattern Search

Implement the naive pattern searching algorithm to find all occurrences of a pattern within a given text string. Count the number of comparisons made during the search to evaluate the efficiency of the algorithm.



```java
package com.wipro.pattern;
public class NaivePatternSearching {
    public static void main(String[] args) {
        String text = "I Love Cats";
        String pattern = "Cats";
        search(text, pattern);
    }
    private static void search(String text, String pattern) {
        int strleng = text.length();
        int patleng = pattern.length();
         for (int i = 0; i <= strleng - patleng; i++) {
              int j;
                for ( j = 0; j < patleng; j++) {
                    if (text.charAt(i + j) != pattern.charAt(j)) {
                        break;
                    }
                }
                if (j == patleng) {
                    System.out.println("pattern found at index:" + i);
                }

          }

}
}
```



```
pattern found at index:7
```