Panji Iman Baskoro
171111023

Aktivitas Pertemuan 7

Modul7.java

```java
public class Modul7 {

  public static void main(String[] args) {
    Graph g = new Graph();
    GraphNode[] graph_node_list = {
      new GraphNode(0),
      new GraphNode(1),
      new GraphNode(2),
      new GraphNode(3),
      new GraphNode(4),
    };

    for (GraphNode graph_node : graph_node_list) {
      g.add_node(graph_node);
    }

    int[][] path_list = {{0, 1, 1},
    {0, 2, 1},
    {1, 3, 1},
    {2, 3, 1},
    {3, 4, 2},
    {1, 1, 5},
    {3, 4, 2},
    {1, 1, 5},
    };

    for (int[] path : path_list) {
      GraphNode first_node = graph_node_list[path[0]];
      GraphNode second_node = graph_node_list[path[1]];
      double distance = path[2];
      g.add_edge(new GraphEdge(first_node, second_node, distance));
      g.add_edge(new GraphEdge(second_node, first_node, distance));
    }
    g.to_tree(0).print();
  }

}
```

Tree.java

```java
public class Tree {
  TreeNode root;

  public Tree() {
    this.root = null;
  }

  public Tree(TreeNode root) {
    this.root = root;
  }

  void print() {
    if (this.root == null) {
      System.out.println();
    } else {
      this.root.print();
    }
  }
}
```

TreeNode.java

```java
import java.util.ArrayList;

public class TreeNode {
  TreeNode parent;
  double distance;
  ArrayList<TreeNode> children;
  int data;

  public TreeNode(int new_data) {
    this.data = new_data;
    this.parent = null;
    this.distance = 0.0;
    this.children = new ArrayList<TreeNode>();
  }

  void set_parent(TreeNode new_parent, double distance) {
    this.parent = new_parent;
    this.distance = distance;
    if (this.parent != null) {
      parent.children.add(this);
    }
  }

  void set_parent(TreeNode new_parent) {
    this.set_parent(new_parent, 0);
  }

  void add_child(TreeNode new_child, double distance) {
    new_child.set_parent(this);
    new_child.distance = distance;

  }

  void remove_child(TreeNode child) {
    child.set_parent(this);
    distance = child.distance;
    this.children.remove(child);
  }

  void print(String spaces, double distance) {
    System.out.println(data+" Distance from Parent "+this.distance+ " distance from initial node : "+(distance+this.distance));
    for (int i = 0; i < this.children.size(); i++) {
      this.children.get(i).print(" ", (distance+this.distance));
    }
  }

  void print() {
    this.print("", 0);
  }
}
```

GraphNode.java

```java
public class GraphNode {

  int data;

  public GraphNode(int new_data) {
    this.data = new_data;
  }
}
```

GraphEdge.java

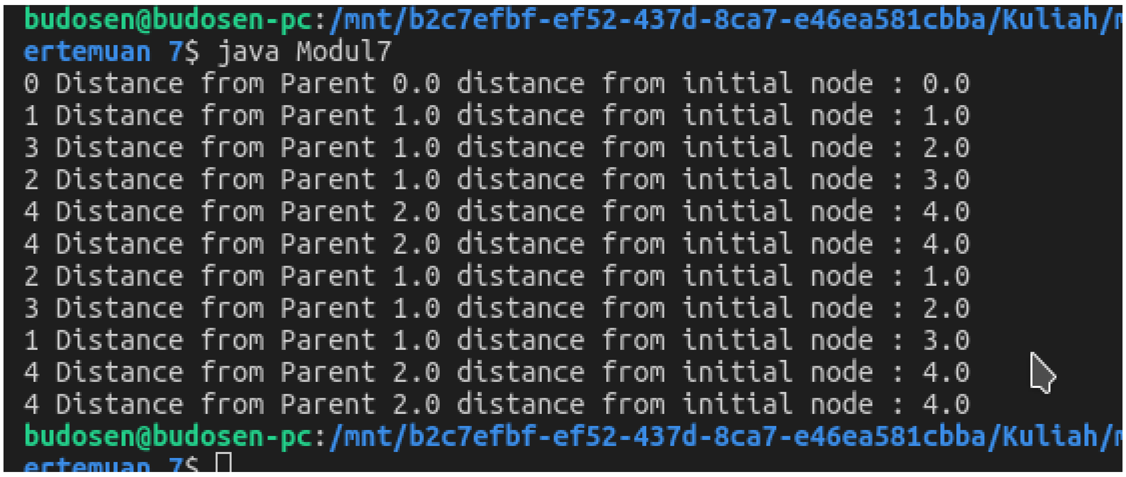
```java
public class GraphEdge {

  GraphNode src;
  GraphNode dst;
  double distance;

  public GraphEdge(GraphNode new_src, GraphNode new_dst, double new_distance) {

    this.src = new_src;
    this.dst = new_dst;
    this.distance = new_distance;
  }
}
```

Graph.java

```java
import java.util.ArrayList;
```

```java
public class Graph {
  ArrayList<GraphNode> nodes;
  ArrayList<GraphEdge> edges;

  public Graph() {
    this.nodes = new ArrayList<GraphNode>();
    this.edges = new ArrayList<GraphEdge>();
  }

  void add_node(GraphNode new_node) {
    this.nodes.add(new_node);
  }

  void add_edge(GraphEdge new_edge) {
    this.edges.add(new_edge);
  }

  void remove_node(GraphNode deleted_node) {
    this.nodes.remove(deleted_node);
    int i = 0;
    while (i < this.edges.size()) {
      GraphEdge edge = edges.get(i);
      if (edge.src == deleted_node || edge.dst == deleted_node) {
        this.edges.remove(edge);
      } else {
        i++;
      }
    }
  }

  void remove_edge(GraphEdge deleted_edge) {
    this.edges.remove(deleted_edge);
  }

  ArrayList<GraphEdge> get_edges_by_source_node(GraphNode node) {
    ArrayList<GraphEdge> node_edges = new ArrayList<GraphEdge>();
    for (int i = 0; i < this.edges.size(); i++) {
      GraphEdge edge = this.edges.get(i);
      if (edge.src == node || edge.dst == node) {
        node_edges.add(edge);
      }
    }
    return node_edges;
  }

GraphNode get_node_by_data(int data) {
  for (int i = 0; i < this.nodes.size(); i++) {
    GraphNode node = this.nodes.get(i);
    if (node.data == data) {
      return node;
    }
  }
  return null;
}

Tree to_tree(int root_data) {
  TreeNode first_tree_node = new TreeNode(root_data);
  first_tree_node = this.completing_tree_node(first_tree_node);
  Tree t = new Tree(first_tree_node);
  return t;
}

TreeNode completing_tree_node(TreeNode tree_node) {
  int data = tree_node.data;
  GraphNode graph_node = this.get_node_by_data(data);
  ArrayList<GraphEdge> edges = this.get_edges_by_source_node(graph_node);
  for (int i = 0; i < edges.size(); i++) {
    GraphEdge edge = edges.get(i);
    if (edge.src == graph_node) {
      int new_data = edge.dst.data;
      boolean should_add_new_data = true;
      TreeNode current_tree_node = tree_node;
      while (current_tree_node != null) {
        if (current_tree_node.data == new_data) {
          should_add_new_data = false;
          break;
        }
        current_tree_node = current_tree_node.parent;
      }
      if (should_add_new_data) {
        TreeNode new_tree_node = new TreeNode(new_data);
        tree_node.add_child(new_tree_node, edge.distance);
        int last_index = tree_node.children.size() - 1;
        tree_node.children.set(last_index, this.completing_tree_node(new_tree_node));
      }
    }
  }
  return tree_node;
}
}
```

Output :



```
budosen@budosen-pc:/mnt/b2c7efbf-ef52-437d-8ca7-e46ea581cbba/Kuliah/r
ertemuan 7$ java Modul7
0 Distance from Parent 0.0 distance from initial node : 0.0
1 Distance from Parent 1.0 distance from initial node : 1.0
3 Distance from Parent 1.0 distance from initial node : 2.0
2 Distance from Parent 1.0 distance from initial node : 3.0
4 Distance from Parent 2.0 distance from initial node : 4.0
4 Distance from Parent 2.0 distance from initial node : 4.0
2 Distance from Parent 1.0 distance from initial node : 1.0
3 Distance from Parent 1.0 distance from initial node : 2.0
1 Distance from Parent 1.0 distance from initial node : 3.0
4 Distance from Parent 2.0 distance from initial node : 4.0
4 Distance from Parent 2.0 distance from initial node : 4.0
budosen@budosen-pc:/mnt/b2c7efbf-ef52-437d-8ca7-e46ea581cbba/Kuliah/r
ertemuan 7$
```