Nama: M Irfan S

NRP: 171111051

Tugas 7

Graph.java

```
2. * To change this license header, choose License Headers in Project Properties.
    * To change this template file, choose Tools | Templates
3.
4. * and open the template in the editor.
5.
6.
7.
8. import java.util.ArrayList;
9.
10./**
11. *
12. * @author yohan
13. */
14. public class Graph {
15.
        ArrayList<GraphNode> nodes;
        ArrayList<GraphEdge> edges;
16.
17.
18.
        /* set this.nodes into new Arraylist<GraphNode>
19.
        set this.edges into new Arraylist<GraphEdge>
20.
        */
21.
        public Graph() {
22.
          this.nodes = new ArrayList<GraphNode>();
23.
          this.edges = new ArrayList<GraphEdge>();
24.
25.
        }
26.
27.
        void add_node(GraphNode new_node) {
28.
            this.nodes.add(new_node);
29.
        }
30.
31.
        void add_edge(GraphEdge new_edge) {
32.
            this.edges.add(new_edge);
33.
34.
35.
        void remove_node(GraphNode deleted_node) {
            this.nodes.remove(deleted_node);
36.
37.
            int i = 0;
38.
            while (i < this.edges.size()) {</pre>
39.
                GraphEdge edge = edges.get(i);
40.
                if (edge.src == deleted_node || edge.dst == deleted_node) {
41.
                    this.edges.remove(edge);
42.
                } else {
43.
                    i++;
44.
                }
45.
            }
46.
47.
48.
        void remove_edge(GraphEdge deleted_edge) {
49.
            this.edges.remove(deleted_edge);
50.
51.
52.
        ArrayList<GraphEdge> get_edges_by_source_node(GraphNode node) {
```

```
53.
            ArrayList<GraphEdge> node_edges = new ArrayList<GraphEdge>();
            for (int i = 0; i < this.edges.size(); i++) {</pre>
54.
55.
                GraphEdge edge = this.edges.get(i);
56.
                if (edge.src == node || edge.dst == node) {
57.
                    node_edges.add(edge);
58.
59.
60.
            return node edges;
61.
62.
63.
        GraphNode get node by data(int data) {
64.
            for (int i = 0; i < this.nodes.size(); i++) {</pre>
65.
                GraphNode node = this.nodes.get(i);
66.
                if (node.data == data) {
67.
                    return node;
68.
69.
70.
            return null;
71.
        }
72.
73.
        Tree to_tree(int root_data) {
            TreeNode first_tree_node = new TreeNode(root_data);
74.
75.
            first_tree_node = this.completing_tree_node(first_tree_node);
76.
            Tree t = new Tree(first tree node);
77.
            return t;
78.
79.
80.
        TreeNode completing tree node(TreeNode tree node) {
81.
            int data = tree_node.data;
82.
            GraphNode graph_node = this.get_node_by_data(data);
83.
            ArrayList<GraphEdge> edges = this.get_edges_by_source_node(graph_node);
84.
            for (int i = 0; i < edges.size(); i++) {</pre>
85.
                GraphEdge edge = edges.get(i);
                if (edge.src == graph_node) {
86.
                    int new_data = edge.dst.data;
87.
88.
                    boolean should add new data = true;
89.
                    TreeNode current_tree_node = tree_node;
90.
                    while (current_tree_node != null) {
91.
                         if (current_tree_node.data == new_data) {
92.
                             should add new data = false;
93.
                             break;
94.
                         }
95.
                         current_tree_node = current_tree_node.parent;
96.
97.
                    if (should_add_new_data) {
98.
                        TreeNode new_tree_node = new TreeNode(new_data);
99.
                         tree_node.add_child(new_tree_node, edge.distance);
100.
                                int last index = tree node.children.size() - 1;
101.
                                tree_node.children.set(last_index, this.completing_tree_
   node(new_tree_node));
102.
103.
                        }
104.
105.
                    return tree_node;
106.
107.
           }
```

GraphEdge.java

```
    /*
    * To change this license header, choose License Headers in Project Properties.
```

```
3. * To change this template file, choose Tools | Templates
4. * and open the template in the editor.
5.
6.
7.
8. /**
9.
10. * @author yohan
11. */
12. public class GraphEdge {
13.
14.
       GraphNode src;
15.
       GraphNode dst;
16.
      double distance;
17.
18.
      /* set this.src into new_src
19.
       * set this.dst into new_dst
20.
       * set this.distance into new_distance
21.
       */
22.
       public GraphEdge(GraphNode new_src, GraphNode new_dst, double new_distance) {
23.
        this.src = new_src ;
24.
         this.dst = new_dst ;
25.
26. }
          this.distance = new_distance;
27. }
```

GraphNode.java

```
    /*
    * To change this license header, choose License Headers in Project Properties.
    * To change this template file, choose Tools | Templates

4. * and open the template in the editor.
5. */
6.
7.
8. /**
9.
10. * @author yohan
11. */
12. public class GraphNode {
13.
14. int data;
15.
16.
         public GraphNode(int new_data) {
17.
18.
             this.data = new_data;
19. }
```

Modul7.java

```
    /*
    * To change this license header, choose License Headers in Project Properties.
    * To change this template file, choose Tools | Templates
    * and open the template in the editor.
    */
    /**
    /**
    *
```

```
9. * @author yohan
10. */
11. public class Modul7 {
12.
13.
        public static void main(String[] args) {
14.
            Graph g = new Graph();
15.
            GraphNode[] graph_node_list = {
                new GraphNode(∅),
16.
                new GraphNode(1),
17.
18.
                new GraphNode(2),
19.
                new GraphNode(3),
20.
                new GraphNode(4),};
21.
22.
            for (GraphNode graph_node : graph_node_list) {
23.
                g.add_node(graph_node);
24.
25.
26.
            int[][] path_list = {{0, 1, 1},
27.
            \{0, 2, 1\},\
28.
            {1, 3, 1},
29.
            \{2, 3, 1\},\
30.
            {3, 4, 2}
31.
            };
32.
33.
            for (int[] path : path_list) {
34.
                GraphNode first_node = graph_node_list[path[0]];
35.
                GraphNode second_node = graph_node_list[path[1]];
36.
                double distance = path[2];
                g.add_edge(new GraphEdge(first_node, second_node, distance));
37.
38.
                g.add_edge(new GraphEdge(second_node, first_node, distance));
39.
40.
            g.to tree(∅).print();
41.
        }
42.
43.}
```

Tree.java

```
    * To change this license header, choose License Headers in Project Properties.

3. * To change this template file, choose Tools | Templates
4. * and open the template in the editor.
    */
5.
6.
7.
8. /**
9.
10. * @author yohan
11. */
12. public class Tree {
13.
       TreeNode root;
14.
       public Tree() {
15.
16.
           this.root = null;
17.
18.
       public Tree(TreeNode root) {
19.
20.
           this.root = root;
21.
       }
22.
23.
       void print() {
```

TreeNode.java

```
    /*
    * To change this license header, choose License Headers in Project Properties.

    * To change this template file, choose Tools | Templates
4. * and open the template in the editor.
5.
6.
7.
import java.util.ArrayList;
9.
10. /**
11. *
12. * @author yohans
13. */
14. public class TreeNode {
15.
       TreeNode parent;
16.
       double distance;
17.
       ArrayList<TreeNode> children;
18.
       int data;
19.
20.
       public TreeNode(int new_data) {
21.
            this.data = new data;
22.
           this.parent = null;
23.
            this.distance = 0.0;
24.
           this.children = new ArrayList<TreeNode>();
25.
       }
26.
27.
        /* set this node's parent into new parent
       * set this node's distance into distance
28.
       * if this node's parent is not nul, then add this as parent's child
29.
30.
       */
31.
       void set parent(TreeNode new parent, double distance) {
32.
           this.parent = new_parent;
33.
            this.distance = distance;
34.
35.
            if (this.parent != null) {
            parent.children.add(this);
36.
37.
            }
38.
39.
40.
       // alias to set_parent(new_parent, 0)
41.
       void set_parent(TreeNode new_parent) {
42.
           this.set_parent(new_parent, 0);
43.
       }
44.
45.
        /* call new_child.set_parent. The new parent of new_child should be this
46.
       * The distance of new_child should be set to distance
       */
47.
48.
       void add_child(TreeNode new_child, double distance) {
49.
       new_child.set_parent(this);
```

```
50. new_child.distance = distance;
51.
52.
        /* Simply remove child from this node's children */
53.
54.
        void remove_child(TreeNode child) {
            child.set_parent(this);
55.
56.
            distance = child.distance;
57.
            this.children.remove(child);
58.
59.
     /* print this node's data, this node's distance, and distance + this node's dis
60.
  tance
       ^{st} for each of this node's children, recursively call child's print method
61.
       */
62.
63.
        void print(String spaces, double distance) {
64.
            System.out.println(data+ " distance from parent : "+this.distance+" te Dist
  ance from initial node : "+(distance+this.distance));
           for (int i = 0; i < this.children.size(); i++) {</pre>
65.
66.
            this.children.get(i).print(" " ,this.distance+distance);
67.
68.
69.
       void print() {
70.
            this.print(" ", 0);
71.
72.
73.}
```

Output

```
Command Prompt
                                                                                                                                                                                                                                                 1.176 Modul7.java
7.225 Modul7.rar
6/11/2018 16.41
8/11/2018 12.21
                                                              544 Tree.class
556 Tree.java
1.677 TreeNode.class
6/11/2018
                        16.45
6/11/2018 16.42
                           7.27 2.169 TreeNode.java
13 File(s) 21.623 bytes
2 Dir(s) 28.615.733.248 bytes free
6/11/2018
:\Kuliah\Semester3\Praktikum Progdas 2\Modul7>javac modul7.java
:\Kuliah\Semester3\Praktikum Progdas 2\Modul7>java modul7
rror: Could not find or load main class modul7
 :\Kuliah\Semester3\Praktikum Progdas 2\Modul7>javac Modul7.java
:\Kuliah\Semester3\Praktikum Progdas Z\Modul7>java Modul7
distance from parent : 0.0 to Distance from initial node :
distance from parent : 1.0 to Distance from initial node :
distance from parent : 1.0 to Distance from initial node :
distance from parent : 1.0 to Distance from initial node :
distance from parent : 1.0 to Distance from initial node :
                                              : 2.0 to Distance from initial node
: 2.0 to Distance from initial node
: 1.8 to Distance from initial node
: 1.0 to Distance from initial node
: 1.0 to Distance from initial node
: 2.0 to Distance from Initial node
 distance from parent
distance from parent
distance from parent
                                                                                                                                 1.0
  distance from parent
  distance from parent
   \Kuliah\Semester3\Praktikum Progdas 2\Modul7>
```