Nama: M Irfan S

NRP : 171111051

**Tugas 7**

Graph.java

1. /\*
2. \* 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. \*/

8. **import** java.util.ArrayList;
10. /\*\*
11. \*
12. \* @author yohan
13. \*/
14. **public** **class** Graph {
15. ArrayList<GraphNode> nodes;
16. ArrayList<GraphEdge> edges;
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>();
25. }
27. **void** add\_node(GraphNode new\_node) {
28. **this**.nodes.add(new\_node);
29. }
31. **void** add\_edge(GraphEdge new\_edge) {
32. **this**.edges.add(new\_edge);
33. }
35. **void** remove\_node(GraphNode deleted\_node) {
36. **this**.nodes.remove(deleted\_node);
37. **int** i = 0;
38. **while** (i < **this**.edges.size()) {
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. }
48. **void** remove\_edge(GraphEdge deleted\_edge) {
49. **this**.edges.remove(deleted\_edge);
50. }
52. ArrayList<GraphEdge> get\_edges\_by\_source\_node(GraphNode node) {
53. ArrayList<GraphEdge> node\_edges = **new** ArrayList<GraphEdge>();
54. **for** (**int** i = 0; i < **this**.edges.size(); i++) {
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. }
63. GraphNode get\_node\_by\_data(**int** data) {
64. **for** (**int** i = 0; i < **this**.nodes.size(); i++) {
65. GraphNode node = **this**.nodes.get(i);
66. **if** (node.data == data) {
67. **return** node;
68. }
69. }
70. **return** **null**;
71. }
73. Tree to\_tree(**int** root\_data) {
74. TreeNode first\_tree\_node = **new** TreeNode(root\_data);
75. first\_tree\_node = **this**.completing\_tree\_node(first\_tree\_node);
76. Tree t = **new** Tree(first\_tree\_node);
77. **return** t;
78. }
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++) {
85. GraphEdge edge = edges.get(i);
86. **if** (edge.src == graph\_node) {
87. **int** new\_data = edge.dst.data;
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

1. /\*
2. \* 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. \*/

8. /\*\*
9. \*
10. \* @author yohan
11. \*/
12. **public** **class** GraphEdge {
14. GraphNode src;
15. GraphNode dst;
16. **double** distance;
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. **this**.distance = new\_distance;
26. }
27. }

GraphNode.java

1. /\*
2. \* 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. \*/

8. /\*\*
9. \*
10. \* @author yohan
11. \*/
12. **public** **class** GraphNode {
14. **int** data;
16. **public** GraphNode(**int** new\_data) {
17. **this**.data = new\_data;
18. }
19. }

Modul7.java

1. /\*
2. \* 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. \*/
7. /\*\*
8. \*
9. \* @author yohan
10. \*/
11. **public** **class** Modul7 {
13. **public** **static** **void** main(String[] args) {
14. Graph g = **new** Graph();
15. GraphNode[] graph\_node\_list = {
16. **new** GraphNode(0),
17. **new** GraphNode(1),
18. **new** GraphNode(2),
19. **new** GraphNode(3),
20. **new** GraphNode(4),};
22. **for** (GraphNode graph\_node : graph\_node\_list) {
23. g.add\_node(graph\_node);
24. }
26. **int**[][] path\_list = {{0, 1, 1},
27. {0, 2, 1},
28. {1, 3, 1},
29. {2, 3, 1},
30. {3, 4, 2}
31. };
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];
37. g.add\_edge(**new** GraphEdge(first\_node, second\_node, distance));
38. g.add\_edge(**new** GraphEdge(second\_node, first\_node, distance));
39. }
40. g.to\_tree(0).print();
41. }
43. }

Tree.java

1. /\*
2. \* 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. \*/

8. /\*\*
9. \*
10. \* @author yohan
11. \*/
12. **public** **class** Tree {
13. TreeNode root;
15. **public** Tree() {
16. **this**.root = **null**;
17. }
19. **public** Tree(TreeNode root) {
20. **this**.root = root;
21. }
23. **void** print() {
24. **if** (**this**.root == **null**) {
25. System.out.println();
26. } **else** {
27. **this**.root.print();
28. }
29. }
30. }

TreeNode.java

1. /\*
2. \* 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. \*/

8. **import** java.util.ArrayList;
10. /\*\*
11. \*
12. \* @author yohans
13. \*/
14. **public** **class** TreeNode {
15. TreeNode parent;
16. **double** distance;
17. ArrayList<TreeNode> children;
18. **int** data;
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. }
27. /\* set this node's parent into new parent
28. \* set this node's distance into distance
29. \* if this node's parent is not nul, then add this as parent's child
30. \*/
31. **void** set\_parent(TreeNode new\_parent, **double** distance) {
32. **this**.parent = new\_parent;
33. **this**.distance = distance;
35. **if** (**this**.parent != **null**) {
36. parent.children.add(**this**);
37. }
38. }
40. // alias to set\_parent(new\_parent, 0)
41. **void** set\_parent(TreeNode new\_parent) {
42. **this**.set\_parent(new\_parent, 0);
43. }
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. }
53. /\* Simply remove child from this node's children \*/
54. **void** remove\_child(TreeNode child) {
55. child.set\_parent(**this**);
56. distance = child.distance;
57. **this**.children.remove(child);
58. }
60. /\* print this node's data, this node's distance, and distance + this node's distance
61. \* for each of this node's children, recursively call child's print method
62. \*/
63. **void** print(String spaces, **double** distance) {
64. System.out.println(data+ " distance from parent : "+**this**.distance+" te Distance from initial node : "+(distance+**this**.distance));
65. **for** (**int** i = 0; i < **this**.children.size(); i++) {
66. **this**.children.get(i).print(" " ,**this**.distance+distance);
67. }
68. }
70. **void** print() {
71. **this**.print(" ", 0);
72. }
73. }

Output

