POS-HW4-Problem2

Zejie Zhou

October 2018

1 Graph.java Test Cases

• **Constructor Test**

strategy: create a new graph using constructor and iterator over the constructed graph.

expected output: iterator.hasNext()==false, the iterator does not have the next element

• **addNode Test**

strategy: add four different nodes into the graph, then add duplicate nodes into the graph

expected output: distinct nodes will be added into the graph, duplicate nodes will not be added into the graph

• **addEdge Test**

strategy: add edges with different end nodes, edges with different labels, edges with different start nodes into the graph, and duplicate edges into the graph

expected output: duplicate edges will not be added into the graph, other edges will be added into the graph.

• **getEdge Test**

strategy: get edges of nodes that are in the graph and nodes that are not in the graph

expected output: nodes that are in the graph will return a tree set of edges that start from that nodes. Nodes that are not in the graph will return an empty tree set.

• **getAllNodes Test**

strategy: add 4 distinct nodes into the graph

expected output: return a tree set containing all 4 distinct nodes in the graph

- **listNodes Test**

strategy: add 4 distinct nodes and 3 edges(one of which containing a new node)into the graph

expected output: return an iterator of a tree set containing 5 distinct nodes in the graph in sorted order

- **childNodes Test**

strategy: add 5 distinct edges containing 5 distinct nodes into the graph, then add a duplicate edge into the graph

expected output: return an iterator of a tree set containing child nodes of the parent node in sorted order, adding duplicate edge will not change the output

2 GraphWrapper.java Test Cases

• **Constructor Test**

strategy: create a new Graph Wrapper and iterate over it

expected output: the size of the graph will be 0 and iterator does not have the next element.

• **addNode Test**

strategy: add 4 distinct nodes and 2 duplicate nodes into the graph

expected output: distinct nodes will be added into the graph, duplicate nodes will not be added into the graph.

• **addEdge Test**

strategy: add edges with different start nodes, edges with different end nodes, edges with different labels, and duplicate edges into the graph

expected output: duplicate edges will be added into the graph, other edges will not be added into the graph.

• **listNodes Test**

strategy: add 4 distinct nodes and 3 edges(one of which containing a new node) into the graph.

expected output: return an iterator of a tree set containing 5 distinct nodes in sorted order.

• **childNodes Test**

strategy: add 5 distinct edges containing 5 distinct nodes and a duplicate edge into the graph

expected output: return an iterator of a tree set containing 5 distinct child nodes of the parent node in sorted order.

3 Heuristic

Use test cases of all equivalence class(based on paths in specification), and run with input values at boundaries of these classes