Report Assignment 3

Amelie Löwe, Håkan Johansson
November 14, 2019

Exercise 1

The isAcyclic() method:

In both the directed and undirected graph we used the Depth First Search(DFS) approach to solve the traversing problem. By using a boolean array we could keep track of which path was already visited, (those we set to true) and which ones still where untouched(false). This was done to avoid traversing graph multiple times. If a Vertice was visited before and the path led back to it again a cycle was found and the isAcyclic method returns false .

Time complexity analysis:

The time complexity of the Depth First Search(DFS) search algorithm is :
O(V+E) where V is the number of vertices in the graph and E is the number of edges in the graph. This would be the worst case for our acyclic methods since it means that we have to traverse each node of the graph.

The isConnected() method:

Time complexity analysis:
Worst case for the undirected is the linear time complexity O(E + V). For the Directed we also have O(
Undirected graph:
Directed graph:

The connectedComponents() method:

Time complexity analysis:
Time complexity for the undirected graph is in worst case O(V + E). That is due to we visit all the nodes and at most using all the edges.

In the case for the directed graph we will get time complexity O(N^2). Which is based on the fact that we need to transpose the graph which runs in O(N^2), and the rest is O(V + E).

Undirected graph:

Directed graph:

Exercise 2

The hasEulerPath() method:

Time complexity analysis:

The eulerPath() method:

This method uses the hasEulerPath to check if it
Time complexity analysis:

Exercise 3

For all of the methods in the third exercise the social network used a Breadth First Traversal (or Search) to solve the searching of the different friendships

The time complexity for a BFS algorithm is O(V+E) where V is the number of vertices in the graph and E is the number of edges in the graph.

The numberOfPeopleAtFriendshipDistance() method:

Time complexity analysis:

The furthestDistanceInFriendshipRelationships() method:

Time complexity analysis:

The possibleFriends() method:

Time complexity analysis: