Algorithms and Data Structures

Laboratory 3

**Exercise 1 Route searching (2+2+(6+6+4)+5)**

1. Create a set of cities (as points) with coordinates x, y on a plane. The cost of going

from city A to city B is equal to the Euclidean distance between two cities, if there exists a road. You should define two scenarios:

a. There are all the direct connections (each pair of cities is connected directly)

b. There are c.a. 80% of possible roads that are available.  
You should choose the coordinates randomly from the range <-100, 100> in both directions.

1. Represent the created map as a weighted graph, where cities are the nodes and roads are the edges of the graph.
2. In the created scene, solve the traveling salesman problem: The salesman starts from a chosen city and has to visit every city exactly once before returning to the starting city. The goal is to find a path with the lowest cost.  
   In the problem, we define state as a partial or full path from the starting city and the corresponding state. You should represent the search problem in a form of state tree.
   1. Implement a full search of the tree, using BFS and DFS methods.
   2. Approximate the solution using Minimum Spanning Tree
   3. Approximate the solution using greedy search
3. Find the shortest path between two chosen cities that are not directly connected using bidirectional search.