1. Dijkstra versus Floyd-Warshall:

   (a) Describe the main differences between Dijkstra’s algorithm and the Floyd-Warshall algorithm for solving the single source shortest path algorithm. Give the advantages and disadvantages of each algorithm.

   (b) What is the most general condition for Dijkstra’s algorithm to fail on an instance that the Floyd-Warshall algorithm does not fail on.

   (c) Design an efficient algorithm that will take a directed weighted graph and a vertex in that graph determine whether or not Dijkstra’s algorithm will return the correct answer for the single source shortest path algorithm.

   4 marks

2. Hamiltonian Paths. A Hamiltonian path is a path that visits every vertex of a directed graph exactly once. Whether a graph has a Hamiltonian path is a surprisingly difficult property to determine. This problem is known to be NP-complete.

   (a) Carefully describe what it means for a problem to be NP-complete.

   (b) Propose an algorithm to determine whether a graph has a Hamiltonian path, describe how it works and give its complexity. The algorithm should be as efficient as you can possibly make it (but no polynomial solutions please).

   4 marks

3. A* is a shortest path algorithm that uses a priority first search. The priority is the distance from the source to the current node (as in Dijkstra’s algorithm) added to an estimate of the distance from the current node to the goal. Do some research ion this topic and explain:

   (a) how A* can improve the efficiency of searching for a shortest path.

   (b) what are some of the complications that using distance estimates can introduce, and how may we avoid these complications.

   2 marks

These tutorial exercises count for 4% of your final assessment. Ensure that you carry out the work yourself.