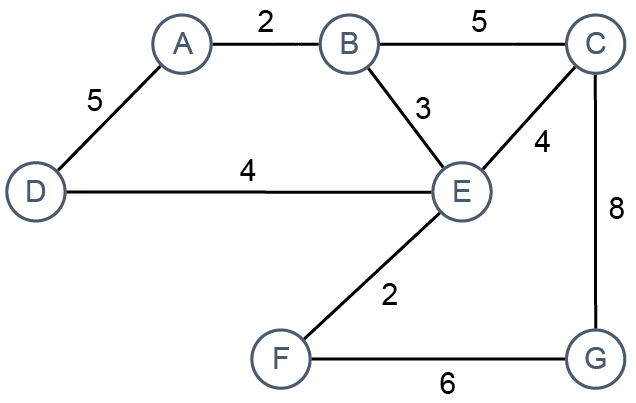
# Homework 6 Optimisation algorithms

# 1. The stations in an underground network are represented as a graph, with nodes representing the stations and weighted edges representing the connections between them. The weights represent the time taken to travel between two stations.



(a) Show how the graph may be represented as an adjacency list. [7]

|  |  |  |  |
| --- | --- | --- | --- |
| A |  |  | [B:2, D:5] |
| B |  |  | [“A:2”, “C:5”, “E:3”] |
| C |  |  | [“B:5”, “E:4”, “G:8”] |
| D |  |  | [“A:5”, “E:4”] |
| E |  |  | [“B:3”, “C:4”, “D:4”, “F:2”] |
| F |  |  | [“E:2”, “G:6”] |
| G |  |  | [“C:8”, “F:6”] |

# 

(b) There is a well-known optimisation algorithm for finding the shortest distance between a start node and all other nodes. What is the name of the algorithm? [1]

Dijsktra

(c) A **priority queue** may be used as a supporting data structure in the implementation of the algorithm. What are the main features of a priority queue?

First in first oust in which items of higher priority are held nearer the front of the queue than items of lower priority [2]

Items in the queue can skip positions and rearrange themselves based on what priority they have in the queue.

(d) The start node is A. Describe the first **two** steps in the algorithm. Do not include the use and operation of the priority queue in your answer.

First A is visited and added to the visited list to show that we have already visited A and don’t need to revisit A. Then we check A’s neighbours which are B and D, B’s distance is only 2 units away whereas D takes 5 time units. So we visit B first because it has a shorter time required. Then we check B’s neighbours and add on the time taken to reach each of B’s neighbours. So for B to C from A, we would need 2 + 5 = 7 time, for E, we need 2 + 3 = 5. Now we add B to visited and now have the choice to go between D and E since both are 5 time units away so equivalent. Before we move onto next, we edit B so that the “path” is A because it tells us that to get from B we came from A. Now we go to D and check it’s neighbours (we go from A), since A is where we came from the only available neighbour is E, so we check the distance which is 5 + 9 = 13, compared to B’s path of just 5, so we go from B to E instead of D to E. Add D to visited, continue algorithm from B.

[5]

(e) Use the diagram to trace through the algorithm and calculate final distances.

What are the shortest distances from A to E, F and G? [3]

|  |  |  |  |
| --- | --- | --- | --- |
| Node | Visited | Distance (time taken to reach) | Path |
| A | X | A = 0 | ------------------------- |
| B | X | B = inf -> 2 | A |
| D | X | D = inf -> 5 | A |
| E | X | E = inf -> 5 | B |
| C | X | C = inf -> 7 | B |
| F | X | F = inf -> 7 | E |
| G | X | G = inf -> 15 -> 13 | C -> F |

A to E:

E <- B <- A, so ABE, 5

A to F:

F <- E <- B <- A, so ABEF, 7

A to G:

G <- F <- E <- B <- A, so ABEFG, 13

(f) State two other applications of the algorithm. [2]

Navigation systems for finding the shortest path to a place you want to go to.

Designing pathfinding systems for AI and computers in video games

[Total 20 Marks]