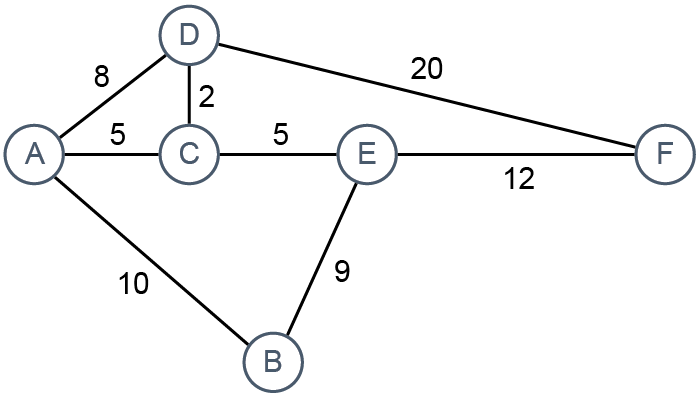
# Worksheet 5 Optimisation algorithms

1. The questions below refer to the following weighted graph.

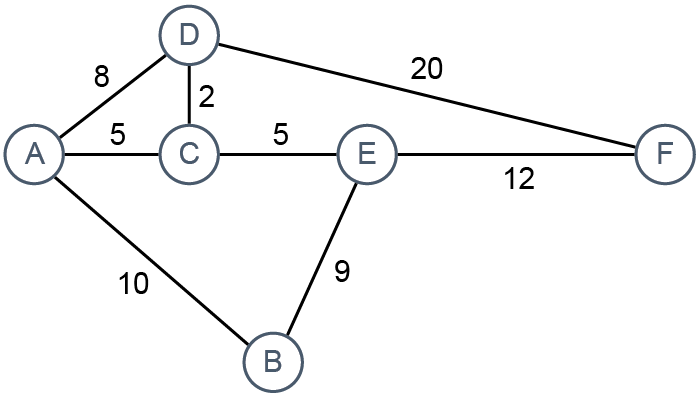
(a) The priority queue at the start is shown below. Mark these “costs” at each vertex.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A = | B = | C = | D = | E = | F = |

(b) Djikstra’s algorithm is used to find the shortest distance form the start node A to every other node.

Show the temporary distances assigned to each node, and the state of the priority queue after A and C have been visited.



Priority queue:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

Once these distances have been added to the priority queue, the algorithm proceeds as follows:

While the priority queue is not empty:

Remove the node at the front of the queue. This is the current node.

For each neighbour, compute new distance by adding together the temporary distance at the current node and the length of the edge going to that neighbour.

If the new distance is less than the neighbour’s current distance, replace the neighbour’s distance by the new distance.

(c) Which is the next node to be visited? What will be the state of the priority queue, and the temporary distance at F, after this node has been visited?

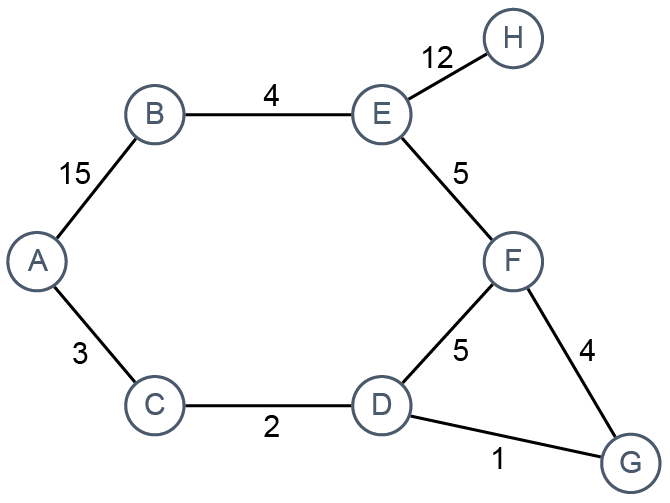
Priority queue:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

(d) Will any further changes be made to temporary distances after this step? Explain.

2. Use Djikstra’s algorithm to find the shortest distance from A to every other node. Colour each node as it is completed or visited (dequeued) and enter the temporary distances on the graph, changing them if and when required to end up with the shortest distances.

Show the state of the priority queue as each node is visited.



**Priority queue**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A = 0 | B = ∞ | C = ∞ | D = ∞ | E = ∞ | F = ∞ | G = ∞ | H = ∞ |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |