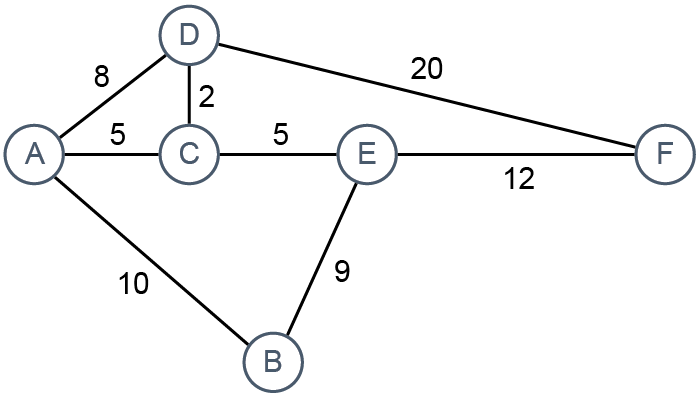
# Worksheet 6 Optimisation algorithms

**Task 1**

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

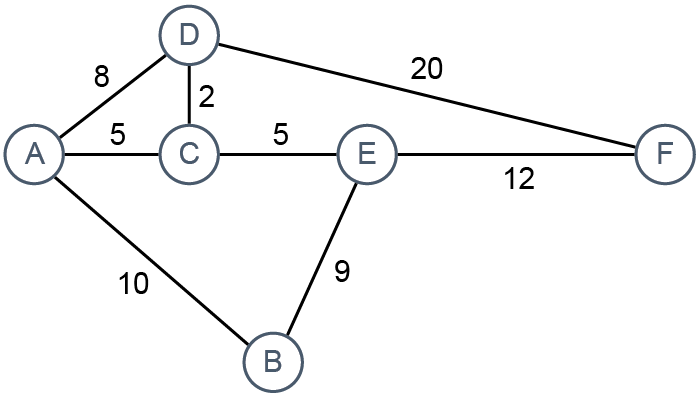
(a) The priority queue at the start is shown below. Complete the priority queue to show temporary “costs” and mark these “costs” at each vertex.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A = 0 | B = inf | C = inf | D = inf | E = inf | F = inf |

(b) Dijkstra’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:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| D | B | E |  |  |  |

Temporary Distances:

A = 0

C = 5

D = 8 -> 7

B = 10

E = 10

F = 27 -> 22

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?

D, 27

Priority queue:

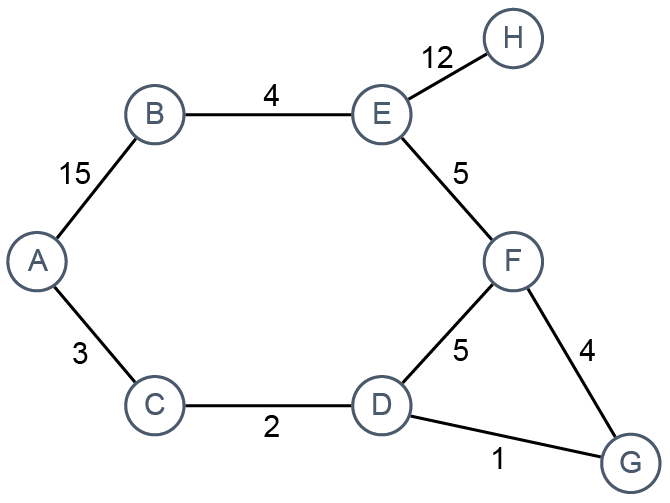
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | B=10 | E=10 | F=27 |  |  |

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

F will eventually be changed to 22 when we visit E since from E to F (ACEF) total distance is 22 which is shorter than 27 (ACDF)

2. Use Dijkstra’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 = ∞ |
| A = 0 | B = inf | C = 3 | D = inf | E = inf | F = inf | G = inf | H = inf |
| A = 0 | B = 15 | C = 3 | D = inf | E = inf | F = inf | G = inf | H = inf |
| A = 0 | B = 15 | C = 3 | D = 5 | E = inf | F = inf | G = inf | H = inf |
| A = 0 | B = 15 | C = 3 | D = 5 | E = inf | F = inf | G = 6 | H = inf |
| A = 0 | B = 15 | C = 3 | D = 5 | E = inf | F = 10 | G = 6 | H = inf |
| A = 0 | B = 15 | C = 3 | D = 5 | E = 15 | F = 10 | G = 6 | H = inf |
| A = 0 | B = 15 | C = 3 | D = 5 | E = 15 | F = 10 | G = 6 | H = 27 |
| A | C | D | G | F | B | F | H |

[remove visited from queue]

# Task 2

3. Which of the following statements are true about the Travelling Salesman problem?

(a) False The TSP is a non-computable problem

(b) True The TSP can sometimes be solved using a brute-force algorithm

(c) False Assuming a specific start city, the number of possible routes for 8 cities is double the number of routes for 4 cities

(d) False Assuming a specific start city, there are fewer than 700 possible routes for 7 cities.

4. What is a tractable problem? Include in your answer the time complexities of tractable and intractable problems, using Big-O notation.

Tractable problems are solvable problems that can be computed where the running time to solve the problem is reasonable, so with time complexities up to polynomial. For tractable problems, all problems with time complexity O(1), O(n), O(logn) etc, up to O(n^2). Inractable problems are solvable but take unreasonable amounts of time to complete as you increase value of n, for example, O(n!), and O(2^n) take a lot of time to complete for larger problems so aren’t worth it even though it can be computed.

5. (a) Complete the table below for different values of n.

|  |  |  |  |
| --- | --- | --- | --- |
| **n** | **8** | **16** | **128** |
| log2n | 3 | 4 | 7 |
| n log2n | 24 | 64 | 896 |
| n2 | 64 | 256 | 16384 |
| n3 | 512 | 4096 | 2097152 |
| 2n | 256 | 65536 | 340,282,366,920,938,463,463,374,607,431,768,211,456 (a 39-digit number) |
| n! | 40320 | 20,922,789,888,000 | A very large number! |

(b) Algorithms for problems A, B and C have time complexities O(n log2n), O(n!) and O(n4).

Which of these problems are tractable, and which are intractable?

A, C = tractable; polynomial and below time

B = intractable; factorial time

6. Choose a password of 8 lowercase alphabetic characters. Write the password below:

abcdefgh

What is the big-O notation for a brute force algorithm to crack this password?

26 x 26 = = 26 squared

26 squared x 8 = O(8n^2) => O(n^2)

Log on to a site to test the strength of the password. You could try

<https://howsecureismypassword.net/> or type into Google “How secure is my password” to find a different site.

How long will it take an average PC to crack the password?

“instantly”

How long would it take to crack a password with 16 lowercase letters?

34 thousand years

If you use a mixture of eight uppercase and lowercase letters, digits and 28 other symbols, what is the big-O notation for the time complexity of the brute force algorithm?

8(26 + 26 + 10 + 28)^2 = 518400

O(n^2) still because n is the number of available options you have for each character out of the 8.

How long will it take an average PC to crack it?

12 hours

Is cracking a password an intractable problem?

Yes it may take really really long

Is cracking a password an insoluble problem?

No you can compute it its just that its not usually worth it since it will take unreasonable amounts of time