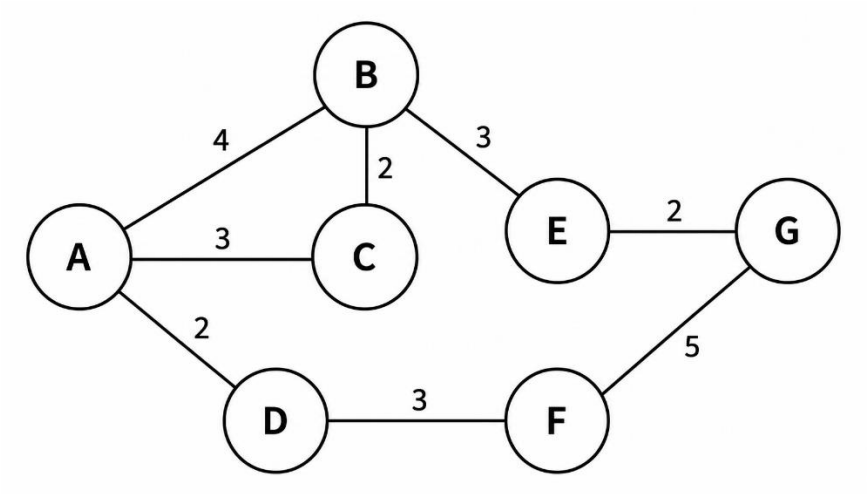
4.3.6 Optimisation Algorithms

Dijkstra's Algorithm

Dijkstra's algorithm calculates **the shortest path in a graph between a start node and every other node in the graph.**
It can be used with **directed or undirected weighted graphs**, provided the edge weights are non-negative.

The Process

Performing Dijkstra's on the graph below using A as the start node.



Set the distance from the start to zero for the start node and to infinity for all other nodes.

Node	Distance	Previous Node
A	**0**	
B	∞	
C	∞	
D	∞	
E	∞	
F	∞	
G	∞	

Visit **node A** and update the table with the distances to each of its neighbours.

Node	Distance	Previous Node
A	**0**	
B	4	A
C	3	A
D	2	A
E	∞	
F	∞	
G	∞	

Choose an unvisited node with the smallest value in the Distance column. In this case **node D** is chosen. Then update the neighbours of D but only if the distance is less than the distance previously written in the table.

Node	Distance	Previous Node
A	**0**	
B	4	A
C	3	A
D	2	A
E	∞	
F	5	D
G	∞	

Next **node C** is chosen as it is an unvisited node with the smallest Distance value. No change is made to B as the distance of 5 through C is larger than the distance 4 in the table.

Node	Distance	Previous Node
A	**0**	
B	4	A
C	3	A
D	2	A
E	∞	
F	5	D
G	∞	

Next **node B** is chosen

Node	Distance	Previous Node
A	**0**	
B	4	A
C	3	A
D	2	A
E	7	B
F	5	D
G	∞	

Next **node F** is chosen

Node	Distance	Previous Node
A	**0**	
B	4	A
C	3	A
D	2	A
E	7	B
F	5	D
G	10	F

Next **node E** is chosen. The distance to G is shorter then the value recorded so far to it is updated.

Node	Distance	Previous Node
A	**0**	
B	4	A
C	3	A
D	2	A
E	7	B
F	5	D
G	9	E

Once every node is visited the algorithm has finish and the table shows the shortest distance between Node A and every other node in the graph.

Node	Distance	Previous Node
A	**0**	
B	4	A
C	3	A
D	2	A
E	7	B
F	5	D
G	9	E