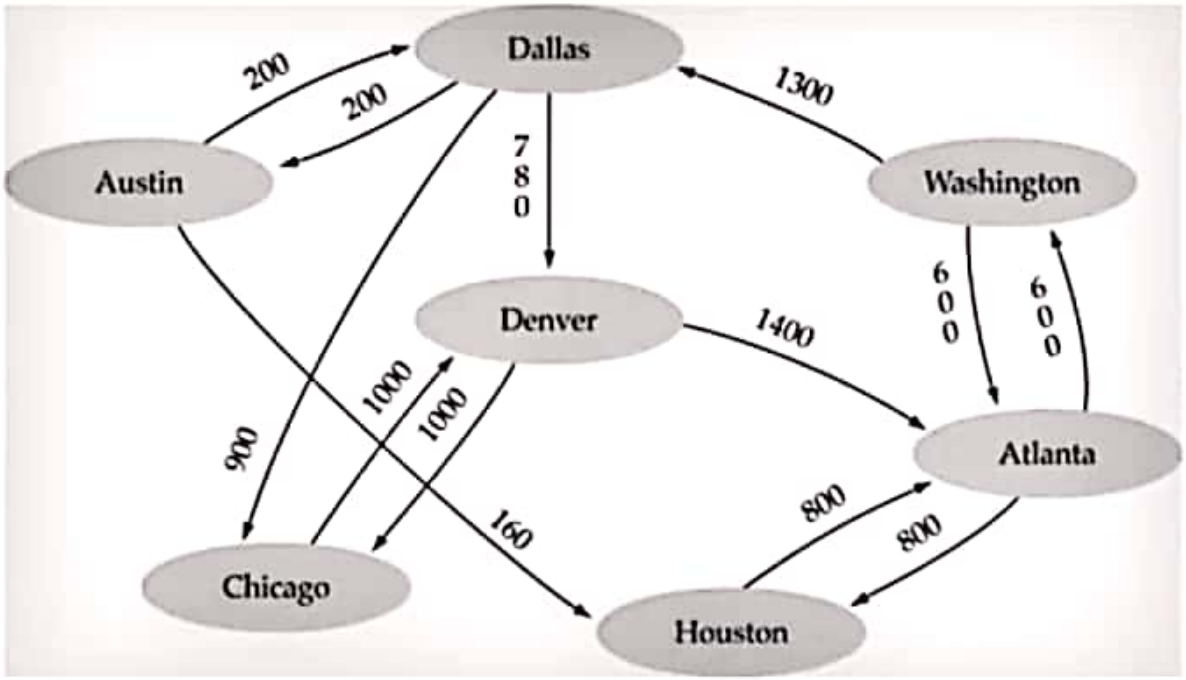
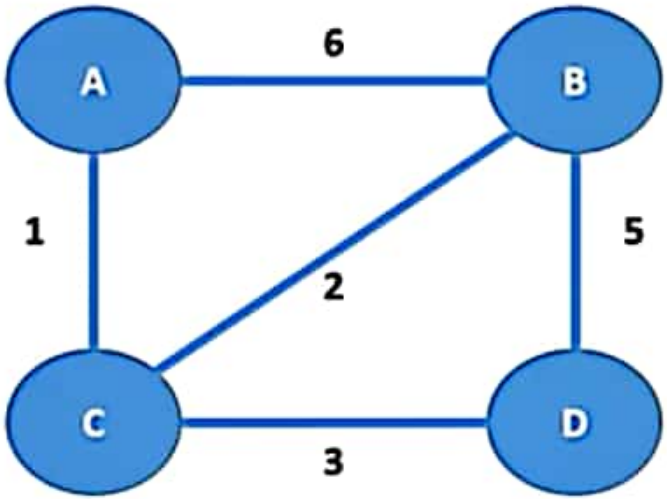
Dijkstra's Shortest Path Algorithm

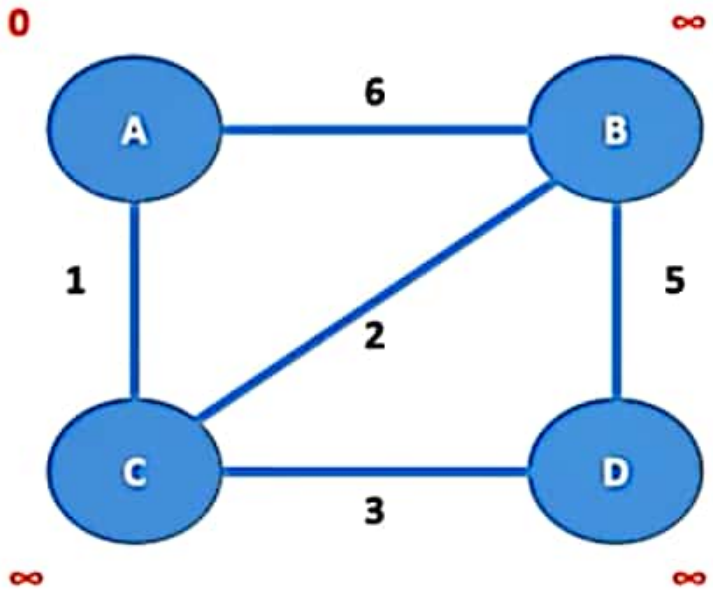


Suppose you have given a graph where source node is ‘A’



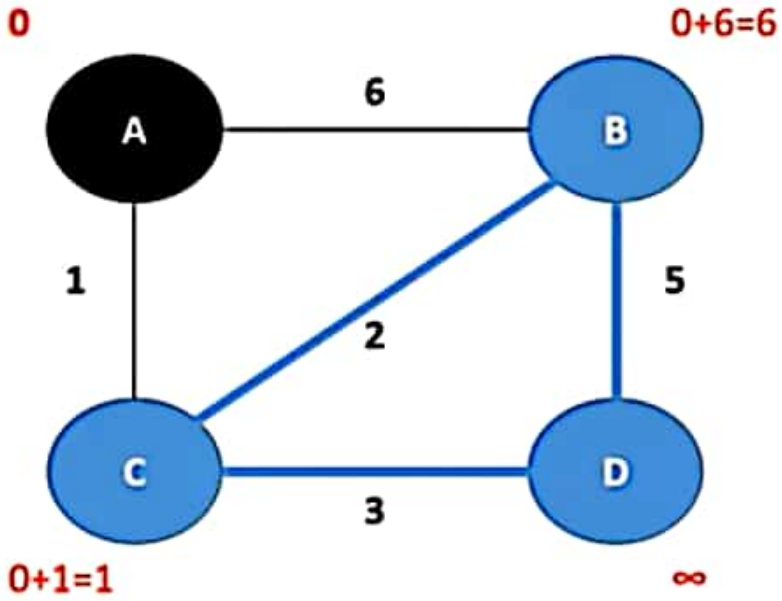
Make a table where,

1. All the distance from ‘A’ is infinity
2. Distance from ‘A’ to ‘A’ is zero



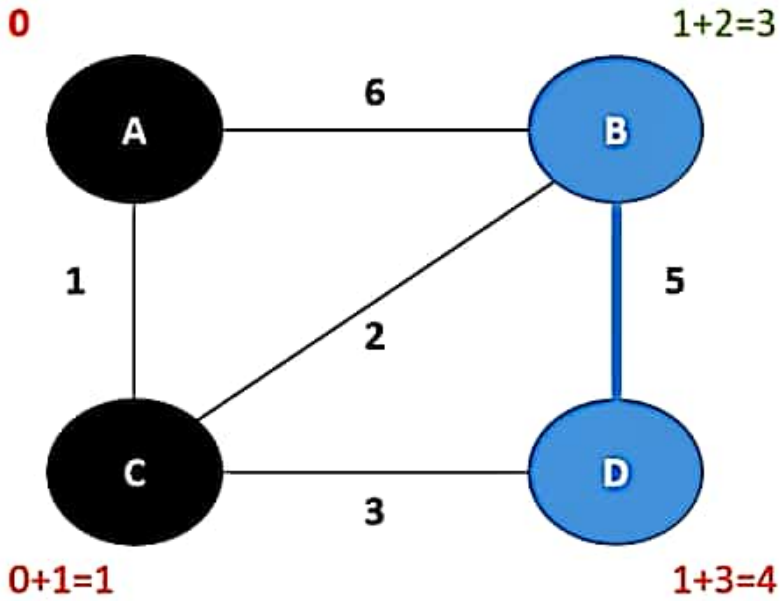
Node	Shortest dis from source	Previous Node
A	0	A
B	∞	*
C	∞	*
D	∞	*

1. Make ‘A’(source) visited
2. Update **the cost** and **previous node** of adjunct nodes



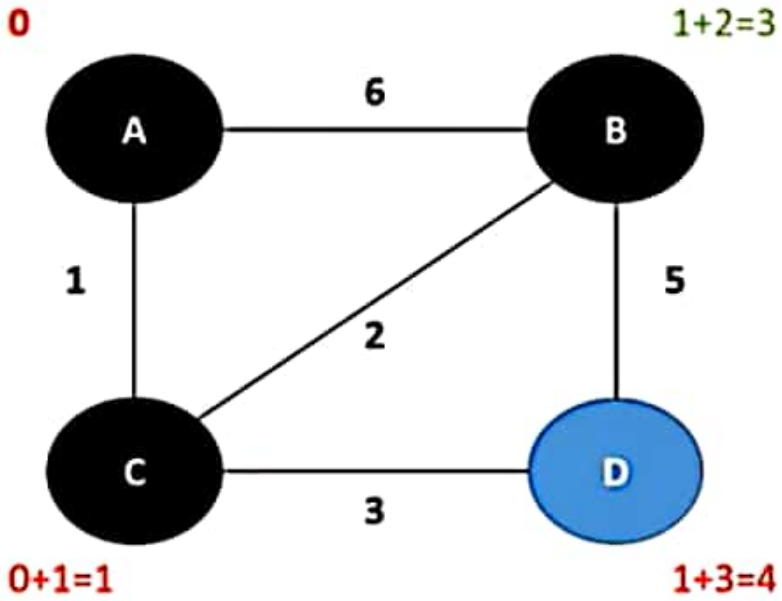
Node	Shortest dis from source	Previous Node
~~A~~	~~0~~	~~A~~
B	6	A
C	1	A
D	∞	*

S1. Pick the shortest distance node (Here C)
S2. Make **it visited** and update **the cost** (where necessary) and **previous node** of adjunct nodes



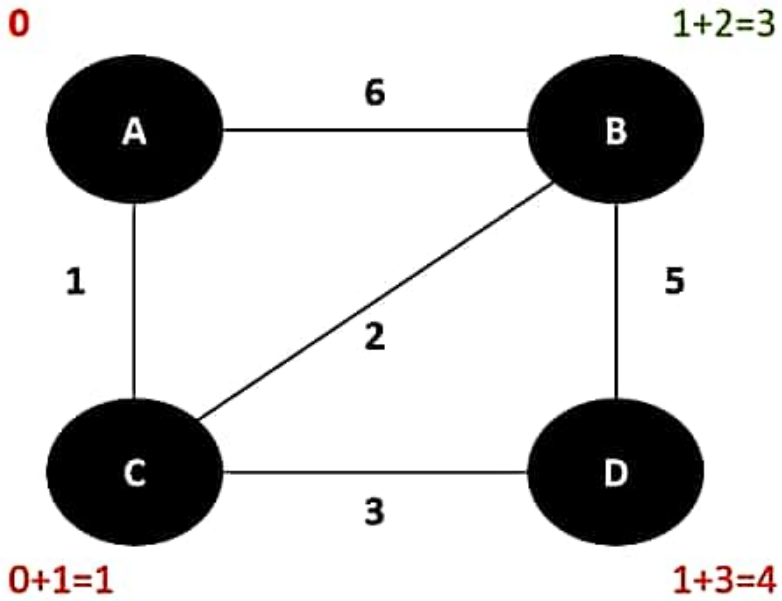
Node	Shortest dis from source	Previous Node
~~A~~	~~0~~	~~A~~
B	3	C
~~C~~	~~1~~	~~A~~
D	4	C

Do the same steps (S1 & S2) again and again until all the nodes become visited



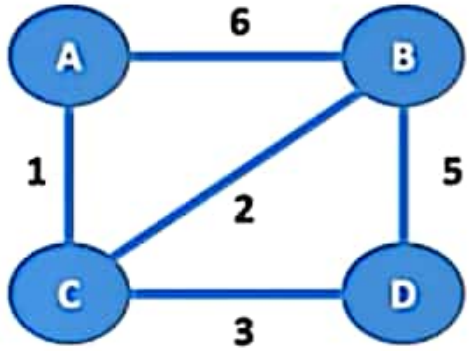
Node	Shortest dis from source	Previous Node
~~A~~	~~0~~	~~A~~
~~B~~	~~3~~	~~C~~
~~C~~	~~1~~	~~A~~
D	4	C

Do the same steps (S1 & S2) again and again until all the nodes become visited



Node	Shortest dis from source	Previous Node
A	0	A
B	3	C
C	1	A
D	4	C

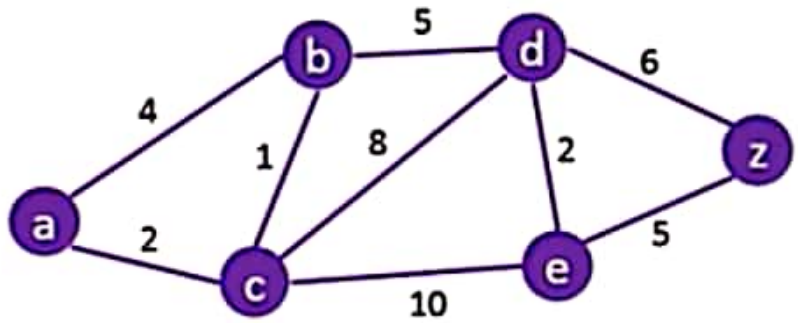
Here we have got our final table. Lets generate the shortest path



Node	Shortest dis from source	Previous Node
A	0	A
B	3	C
C	1	A
D	4	C

Path	Cost
A -> A	0
B -> C-> A	3
C -> A	1
D -> C -> A	4

Practice problem



Dijkstra's Algorithm

What is the shortest path to travel from A to Z?