Graph Theory

Shortest Path Algorithm

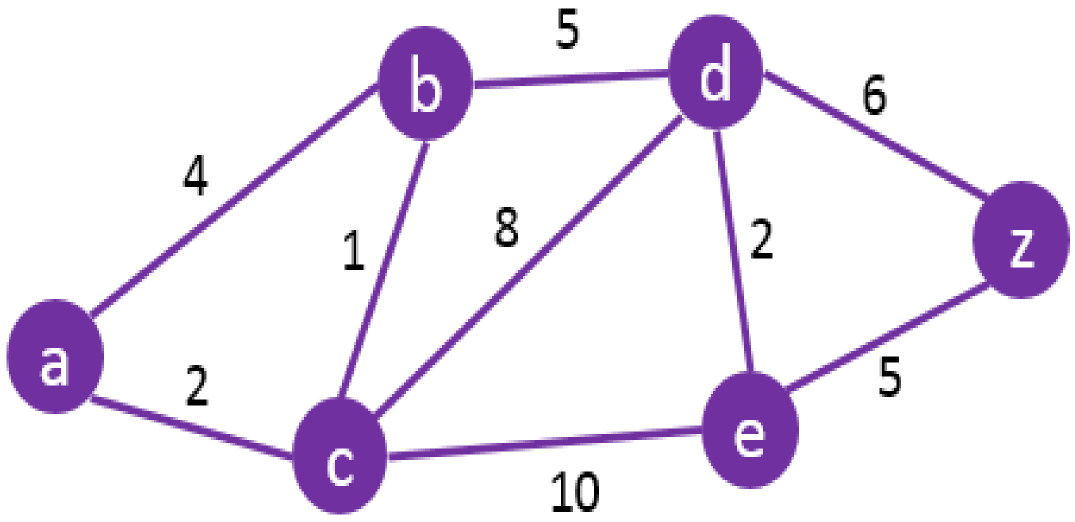
Shortest Path Algorithm: Dijkstra Algorithm

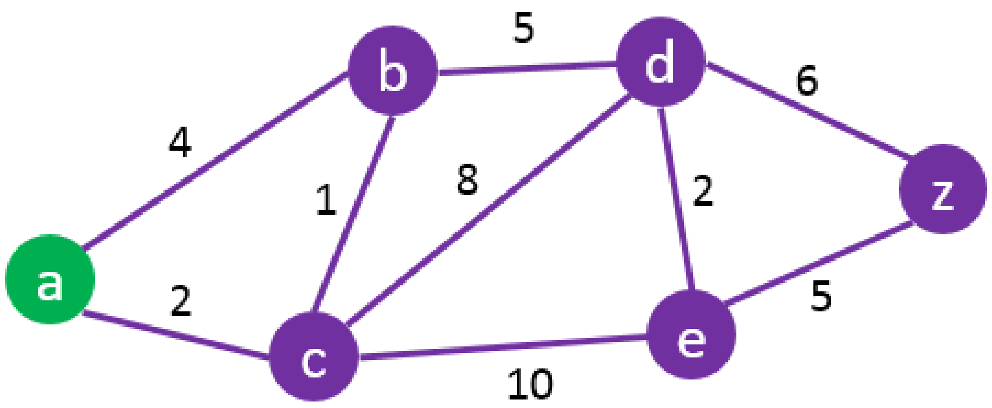
- Dijkstra’s Shortest Path Algorithm is an algorithm used to find the shortest path between two nodes of a weighted graph.
- It is a single-source shortest path algorithm. Here, single-source means that only one source is given, and we have to find the shortest path from the source to all the nodes.

Steps:

- mark all nodes as unvisited
- picked the starting node with a current distance of 0 and the rest nodes with infinity
- Now, fix the starting node as the current node,
- For the current node, analyze all of its unvisited neighbors and measure their distances by adding the current distance of the current node to the weight of the edge that connects the neighbour node and current node
- Compare the recently measured distance with the current distance assigned to the neighbouring node and make it as the new current distance of the neighbouring node,
- After that, consider all of the unvisited neighbours of the current node, mark the current node as visited,
- If the destination node has been marked visited then stop, an algorithm has ended, and
- Else, choose the unvisited node that is marked with the least distance, fix it as the new current node, and repeat the process again from step 4.

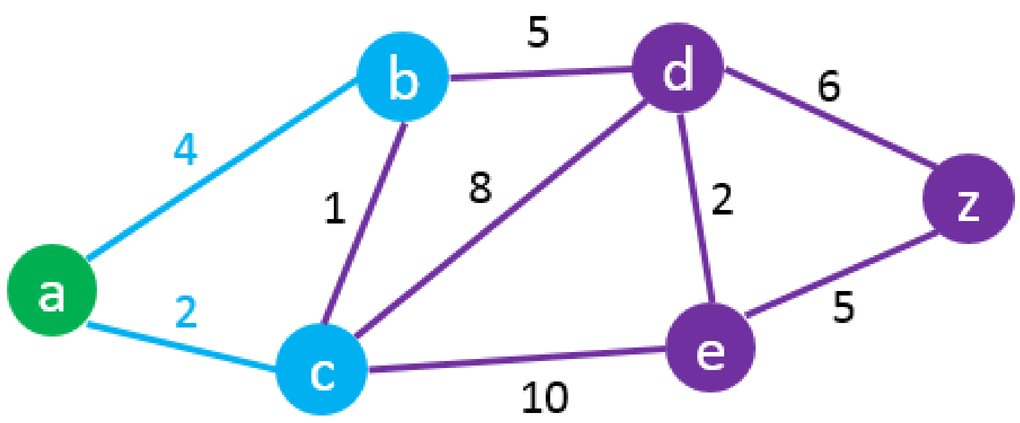
Find the shortest path from A to Z.





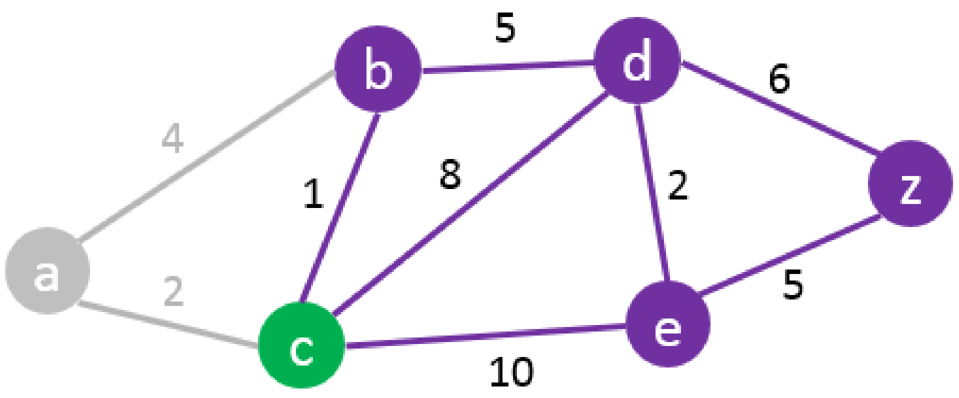
Node	Status	Shortest Distance From A	Previous Node
A	Current Node	0	
B		∞	
C		∞	
D		∞	
E		∞	
Z		∞	

Start by setting the starting node (A) as the current node.



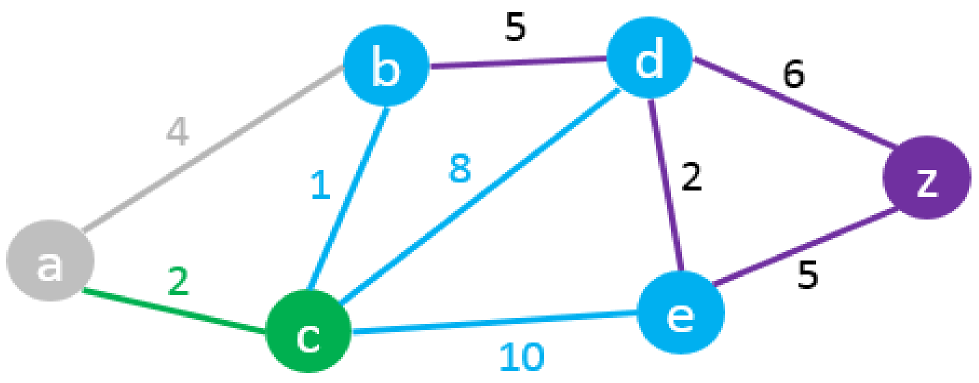
Node	Status	Shortest Distance From A	Previous Node
A	Current Node	0	
B		~~∞~~ 4	A
C		~~∞~~ 2	A
D		∞	
E		∞	
Z		∞	

Check all the nodes connected to A and update their **"Distance from A"** and set their **"previous node"** to "A".



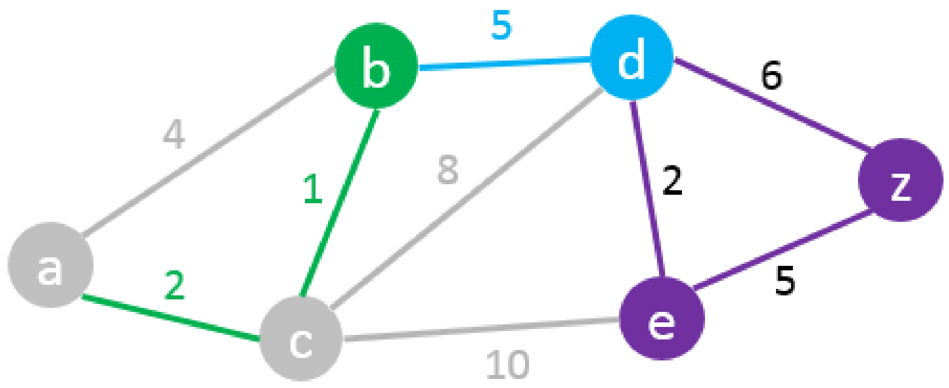
Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B		~~∞~~ 4	A
C	Current Node	~~∞~~ 2	A
D		∞	
E		∞	
Z		∞	

Set the current node (A) to **"visited"** and use the closest unvisited node to A as the **current node** (e.g. in this case: Node C).



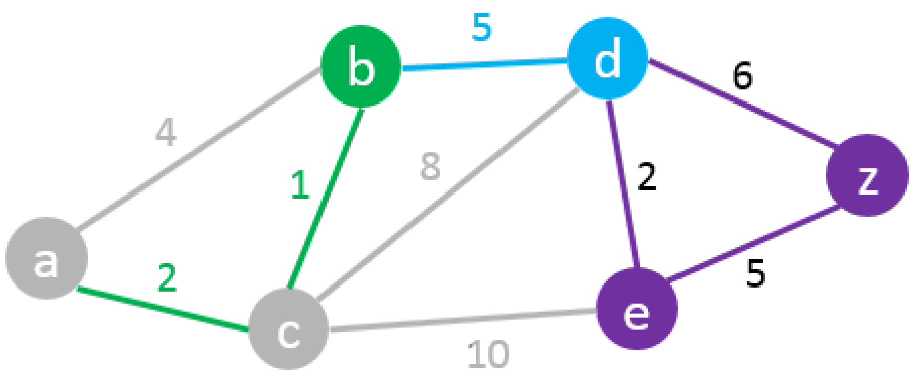
Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B		4 2+1=3	C
C	Current Node	2	A
D		∞ 2+8=10	C
E		∞ 2+10=12	C
Z		∞	

Check all unvisited nodes connected to the current node and add the distance from A to C to all distances from the connected nodes. Replace their values only if the new distance is lower than the previous one.



Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B	Current Node	3	C
C	Visited Node	2	A
D		10	C
E		12	C
Z		∞	

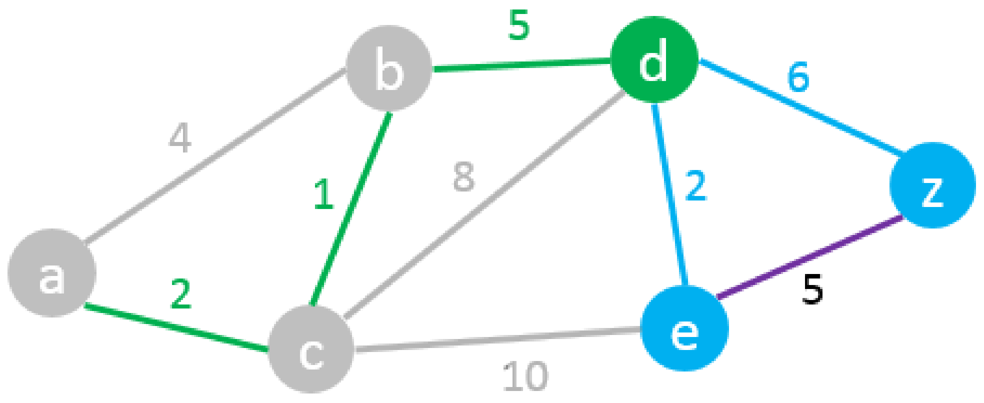
Set the current node C status to Visited.
We then repeat the same process always picking the closest unvisited node to A as the current node.
In this case node B becomes the current node.



Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B	Current Node	3	C
C	Visited Node	2	A
D		~~10~~ 3+5=8	B
E		12	C
Z		∞	

B -> D 3+5 = 8 < 10 – Change Node D

Next "Current Node" will be D as it has the shortest distance from A amongst all unvisited nodes.

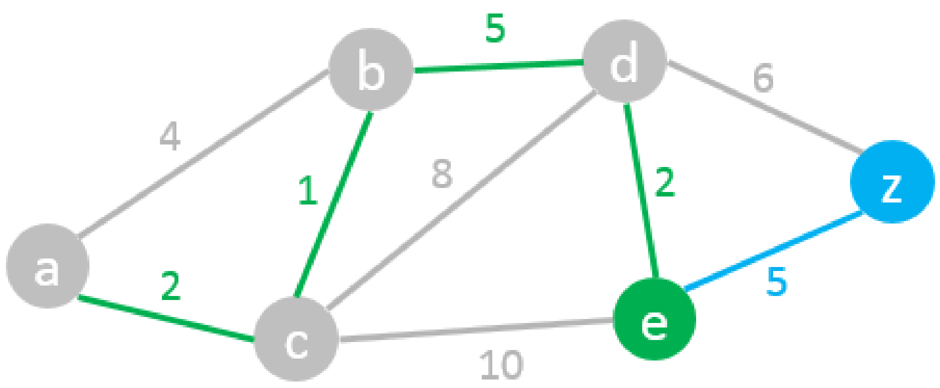


Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B	Visited Node	3	C
C	Visited Node	2	A
D	Current Node	8	B
E		~~12~~ 8 + 2 = 10	D
Z		~~∞~~ 8 + 6 = 14	D

D -> E 8+2 = 10 < 12 – Change Node E

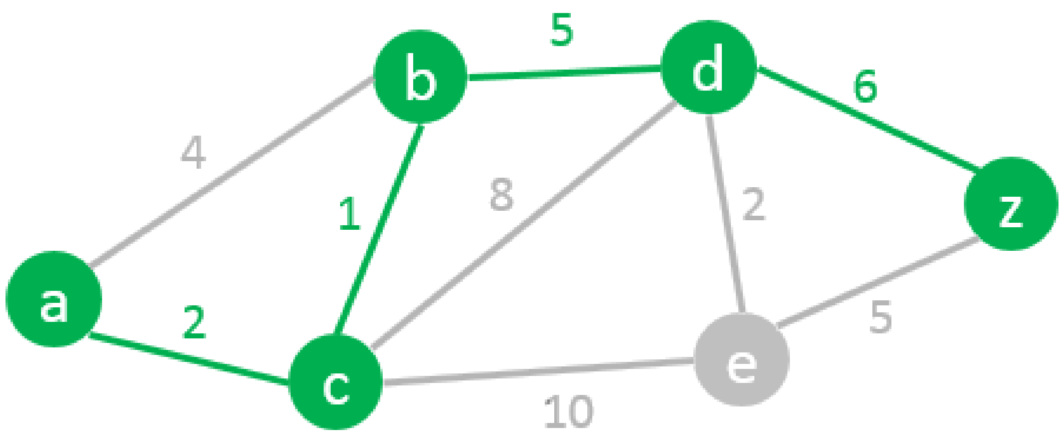
D -> Z 8+6 = 14 < ∞ – Change Node Z

- We found a path from A to Z but it may not be the shortest one yet. So we need to carry on the process.
- Next **“Current Node”**: E



Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B	Visited Node	3	C
C	Visited Node	2	A
D	Visited Node	8	B
E	Current Node	10	D
Z		14 10 + 5 = ~~15~~	D

E -> Z 10+5 = 15 > 14 – We do **not** change node Z.

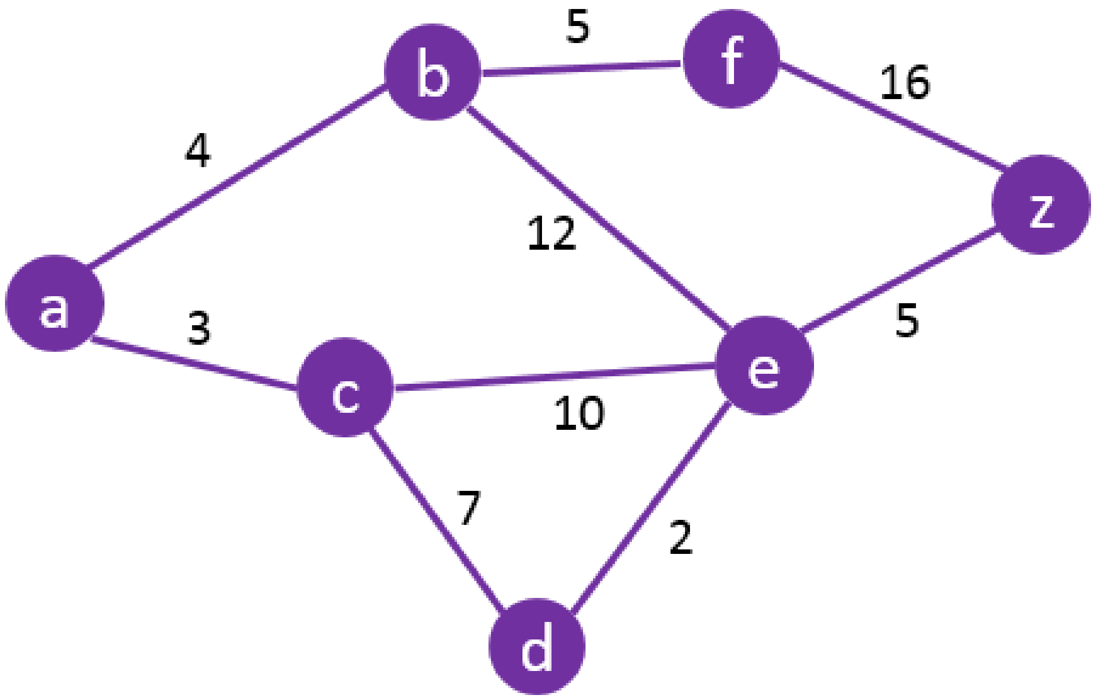


Node	Status	Shortest Distance From A	Previous Node
A	Visited Node	0	
B	Visited Node	3	C
C	Visited Node	2	A
D	Visited Node	8	B
E	Visited Node	10	D
Z	Current Node	14	D

- We found the shortest path from A to Z.
  Read the path from Z to A using the previous node column:
  Z > D > B > C > A
  So the Shortest Path is:
  A – C – B – D – Z with a length of 14

Question

Apply the steps of the Dijkstra's Algorithm to find the shortest path from A to Z using the following graph:



Question

Apply the steps of the Dijkstra's Algorithm to find the shortest path from A to Z using the following graph: