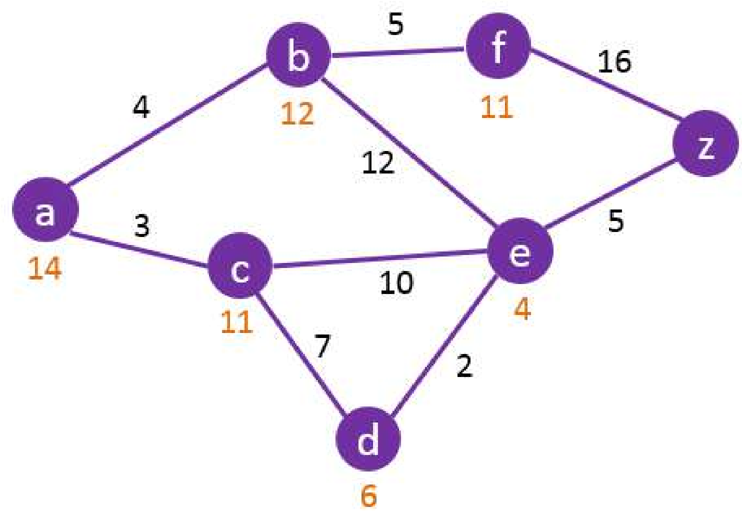
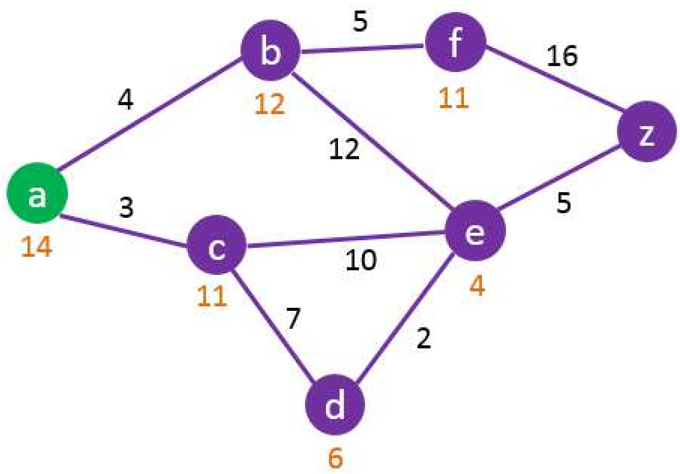


A* Search Algorithm

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

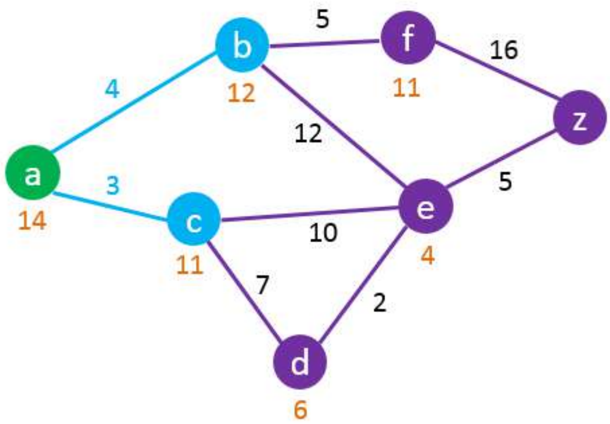
Numbers in orange are the heuristic values, distances in a straight line (as the crow flies) from a node to node Z.



Node	Status	Shortest Distance From A	Heuristic Distance to Z	Total Distance*	Previous Node
A	Current	0	14	14	
B		∞	12		
C		∞	11		
D		∞	6		
E		∞	4		
F		∞	11		
Z		∞	0		

* Total Distance = Shortest Distance from A + Heuristic Distance to Z

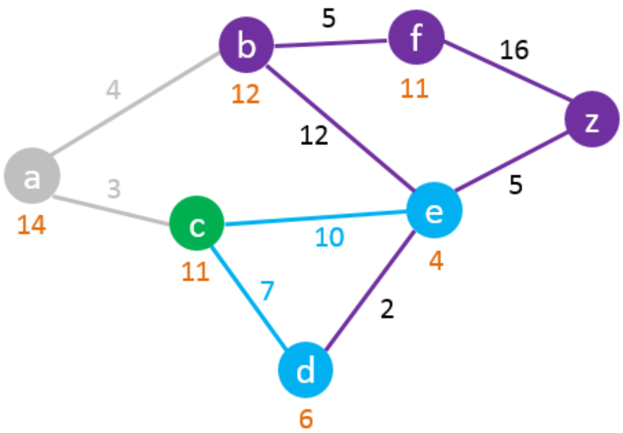
Start by setting the starting node (A) as the current node.



Node	Status	Shortest Distance From A	Heuristic Distance to Z	Total Distance*	Previous Node
A	Current	0	14	14	
B		~~∞~~ 4	12	16	A
C		~~∞~~ 3	11	14	A
D		∞	6		
E		∞	4		
F		∞	11		
Z		∞	0		

* Total Distance = Shortest Distance from A + Heuristic Distance to Z

Check all the nodes connected to A and update their **"Shortest Distance from A"** and set their **"previous node"** to "A".
Update their total distance by adding the shortest distance from A and the heuristic distance to Z.

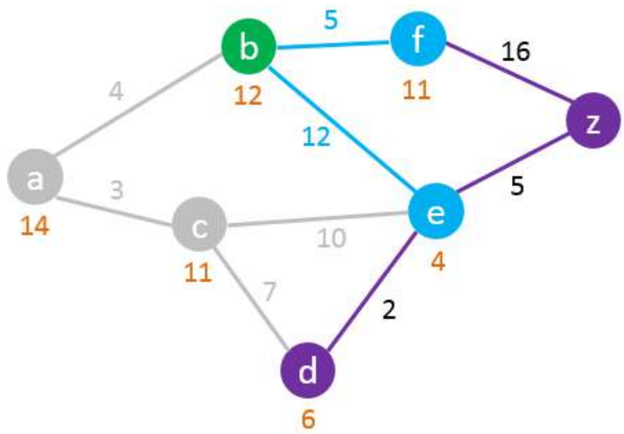


Node	Status	Shortest Distance From A	Heuristic Distance to Z	Total Distance*	Previous Node
A	Visited	0	14	14	
B		4	12	16	A
C	Current	3	11	14	A
D		~~∞~~ 3+7=10	6	16	C
E		~~∞~~ 3+10=13	4	17	C
F		∞	11		
Z		∞	0		
* Total Distance = Shortest Distance from A + Heuristic Distance to Z					

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

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.

The next current node (unvisited node with the shortest total distance) could be either node B or node D. Let's use node B.



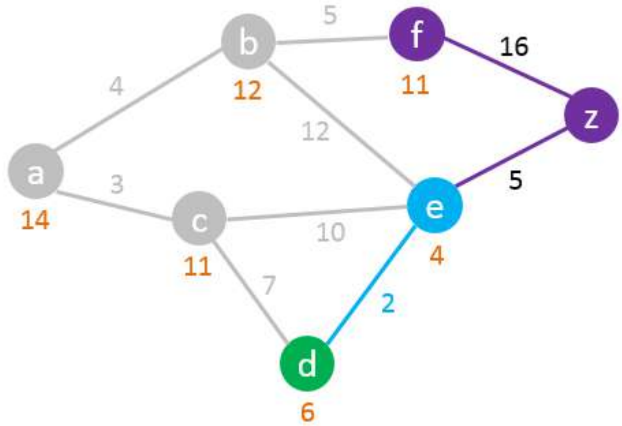
Node	Status	Shortest Distance From A	Heuristic Distance to Z	Total Distance*	Previous Node
A	Visited	0	14	14	
B	Current	4	12	16	A
C	Visited	3	11	14	A
D		10	6	16	C
E		13 4+12=~~16~~	4	17	C
F		~~∞~~ 4+5=9	11	20	B
Z		∞	0		

* Total Distance = Shortest Distance from A + Heuristic Distance to Z

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

B -> E: 4 + 12 = 16 > 13 – Do not change Node E
B -> F: 4 + 5 = 9 < ∞ – Change Node F

The next current node (unvisited node with the shortest total distance) is D.

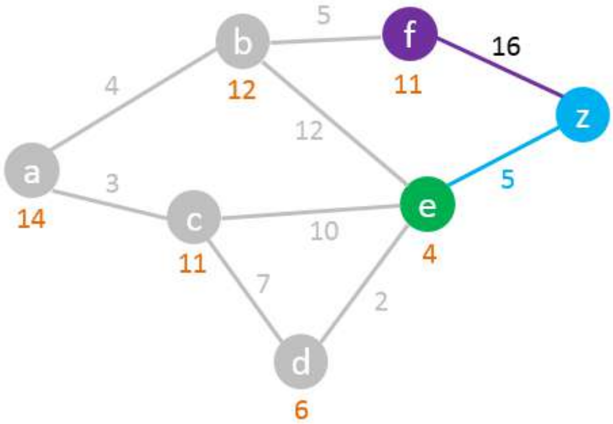


Node	Status	Shortest Distance From A	Heuristic Distance to Z	Total Distance*	Previous Node
A	Visited	0	14	14	
B	Visited	4	12	16	A
C	Visited	3	11	14	A
D	Current	10	6	16	C
E		~~13~~ 10+2=12	4	16	D
F		9	11	20	B
Z		∞	0		
* Total Distance = Shortest Distance from A + Heuristic Distance to Z					

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

D -> E: 10 + 2 = 12 < 13 – Change Node E

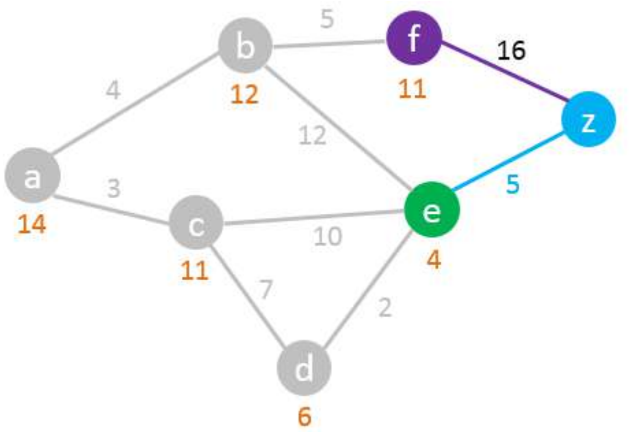
The next current node (unvisited node with the shortest total distance) is E.



Node	Status	Shortest Distance From A	Heuristic Distance to Z	Total Distance*	Previous Node
A	Visited	0	14	14	
B	Visited	4	12	16	A
C	Visited	3	11	14	A
D	Visited	10	6	16	C
E	Current	12	4	16	D
F		9	11	20	B
Z		∞ 12+5=17	0	17	E
* Total Distance = Shortest Distance from A + Heuristic Distance to Z					

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

E -> Z: 12 + 5 = 17 < ∞ – Change Node Z



Node	Status	Shortest Distance From A	Heuristic Distance to Z	Total Distance*	Previous Node
A	Visited	0	14	14	
B	Visited	4	12	16	A
C	Visited	3	11	14	A
D	Visited	10	6	16	C
E	Visited	12	4	16	D
F		9	11	20	B
Z	Current	17	0	17	E

* Total Distance = Shortest Distance from A + Heuristic Distance to Z

We found a path from A to Z, but is it the shortest one?

Check all unvisited nodes. In this example, there is only one unvisited node (F). However its total distance (20) is already greater than the distance we have from A to Z (17) so there is no need to visit node F as it will not lead to a shorter path.

We found the shortest path from A to Z.
Read the path from Z to A using the previous node column:
Z > E > D > C > A

So the Shortest Path is:
A – C – D – E – Z with a length of 17