PATH PLANNING FOR A ROBOT

LET THE SEARCH BEGIN !!!

DIJKSTRA'S ALGORITHM

- Dijkstra's algorithm is a path finding algorithm for finding the shortest path from a starting node to a target node in a weighted graph.
- A graph is a like a map where we need to figure out the shortest path.
- The algorithm creates a tree of shortest paths from the starting vertex, the source, to all other points in the graph.

Below are the detailed steps used in Dijkstra's algorithm to find the shortest path from a single source vertex to all other vertices in the given graph.

Algorithm

1) Create a set named *Set* (shortest path tree set) that keeps track of vertices included in shortest path tree, i.e., whose minimum distance from source is calculated and finalized. Initially, this set is empty.

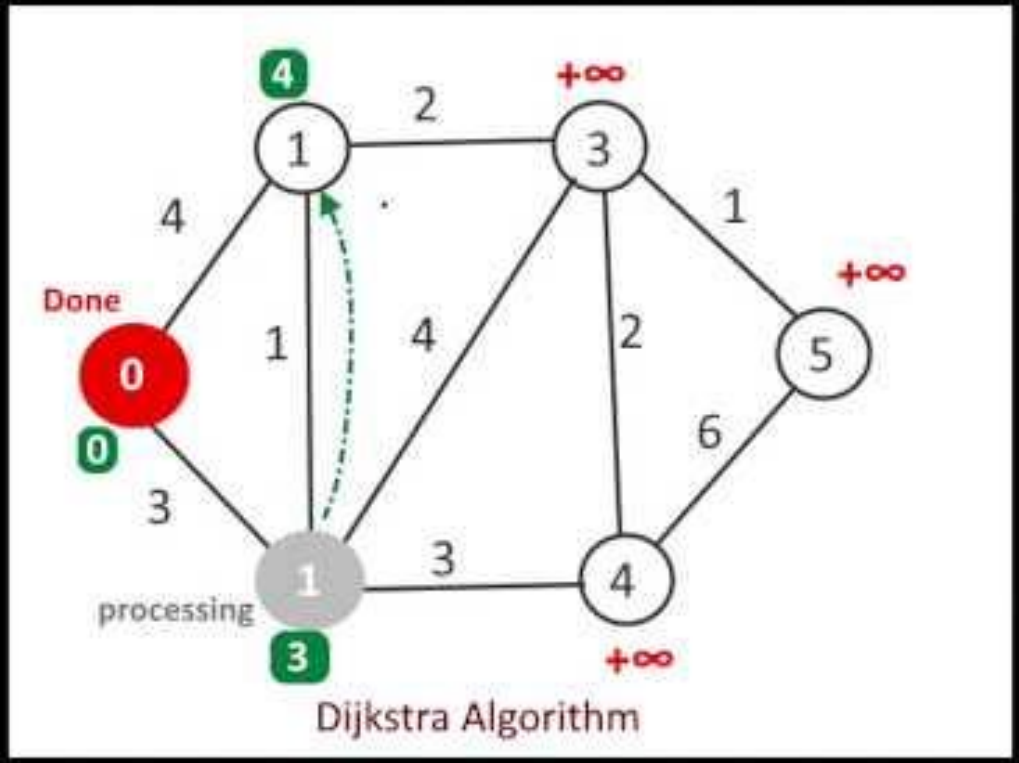
2) Assign a distance value to all vertices in the input graph. Initialize all distance values as INFINITE. Assign distance value as 0 for the source vertex so that it is picked first.

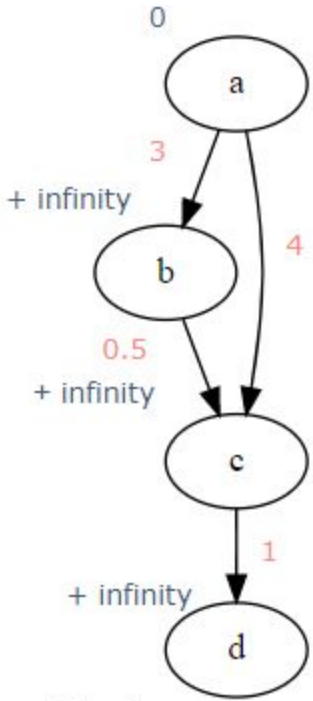
3) While *Set* doesn't include all vertices

….**a)** Pick a vertex u which is not there in *Set* and has minimum distance value.

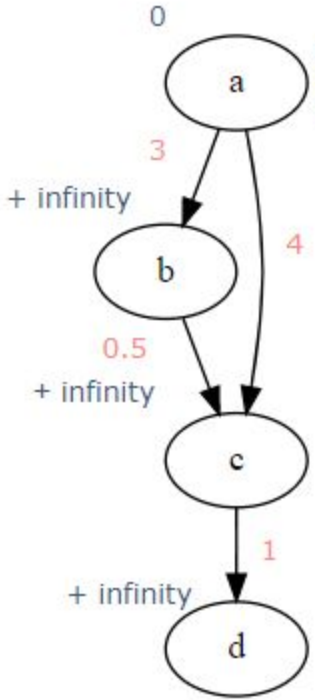
….**b)** Include u to *Set*.

….**c)** Update distance value of all adjacent vertices of u. To update the distance values, iterate through all adjacent vertices. For every adjacent vertex v, if sum of distance value of u (from source) and weight of edge u-v, is less than the distance value of v, then update the distance value of v.


4
+∞
2
1
3
4
1
+∞
Done
1
4
2
5
0
6
0
3
3
1
4
processing
3
+∞
Dijkstra Algorithm

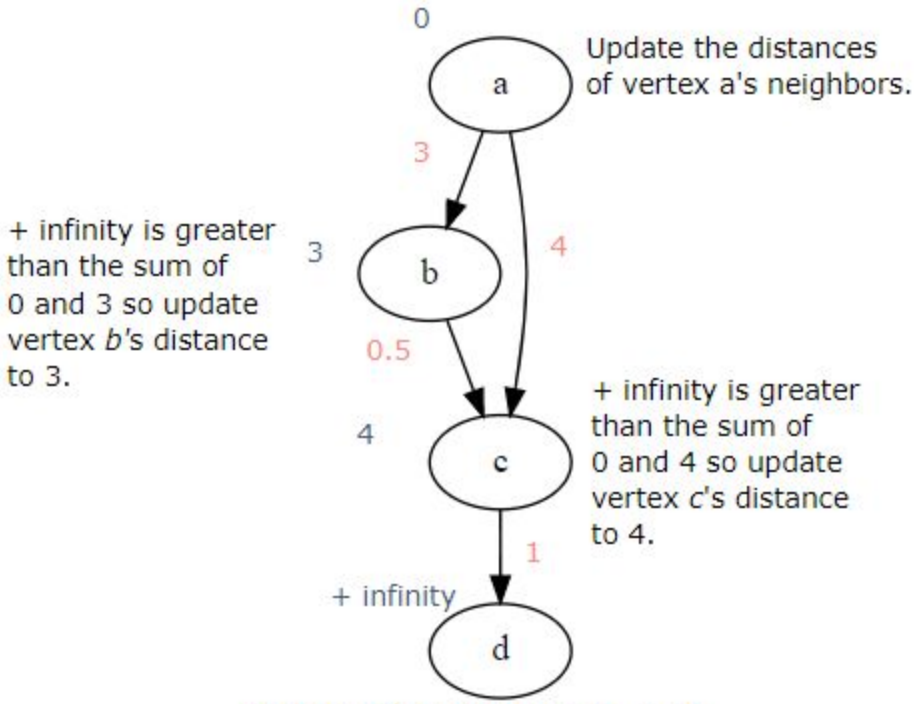


Unvisited vertices : a, b, c, d

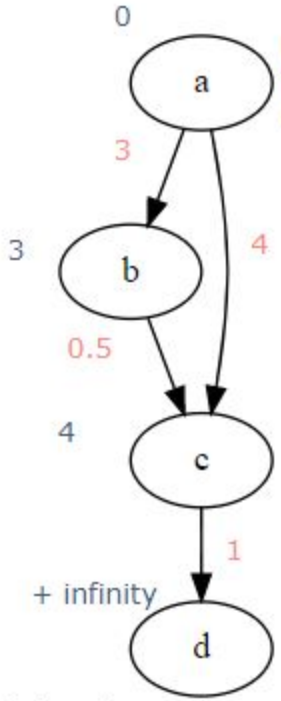


From the unvisited vertices, vertex *a* has the minimum distance so visit it.

Unvisited vertices : a, b, c, d

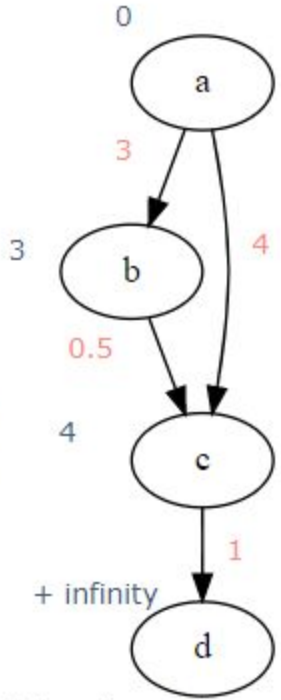


Unvisited vertices : b, c, d

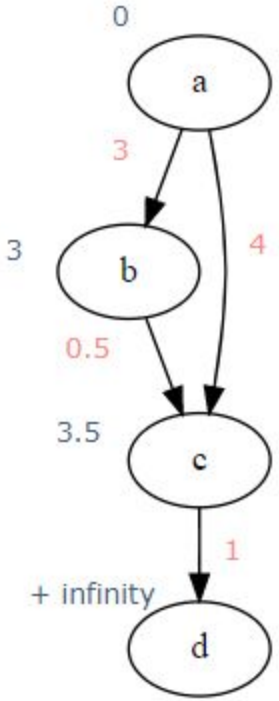


From the unvisited vertices, vertex *b* has the minimum distance so visit it.

Unvisited vertices : b, c, d

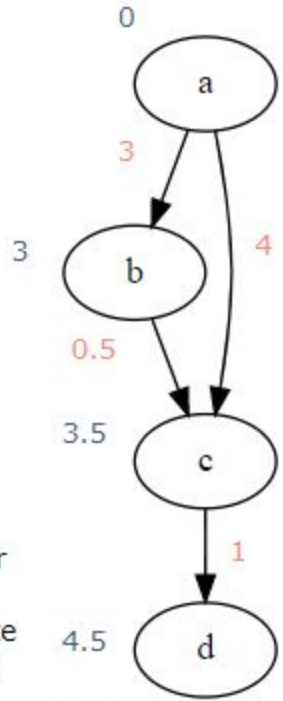


Unvisited vertices : c, d



From the unvisited vertices, vertex *c* has the minimum distance so visit it.

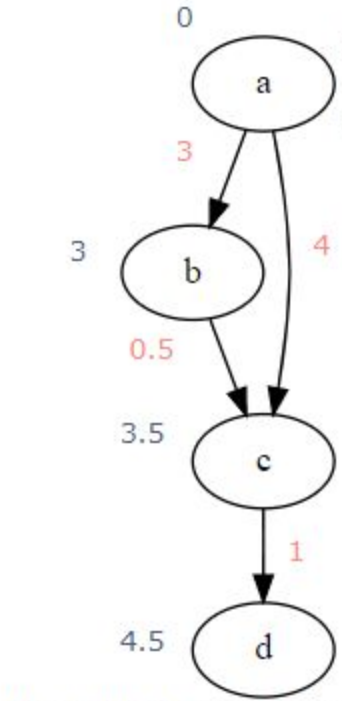
Unvisited vertices : c, d



Update the distances of vertex *c*'s neighbors

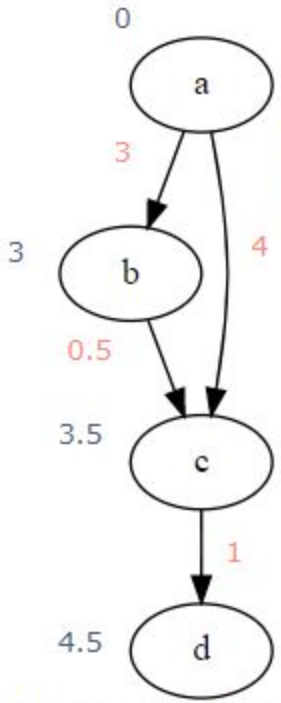
+ infinity is greater than the sum of 3.5 and 1 so update vertex *c*'s distance to 4.5.

Unvisited vertices : d



From the unvisited vertices, vertex *d* has the minimum distance so visit it.

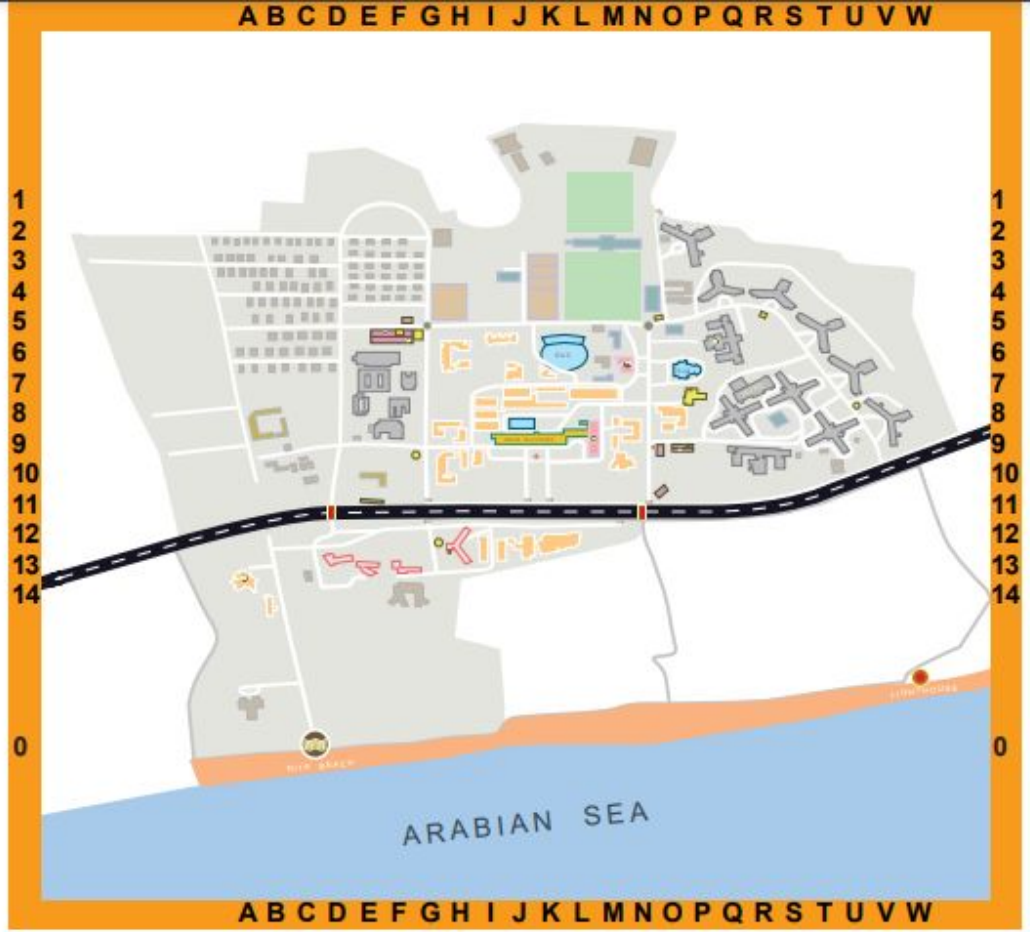
Unvisited vertices : d



Vertex *d* has no neighbors and all the vertices in the graph have been visited so the algorithm terminates.

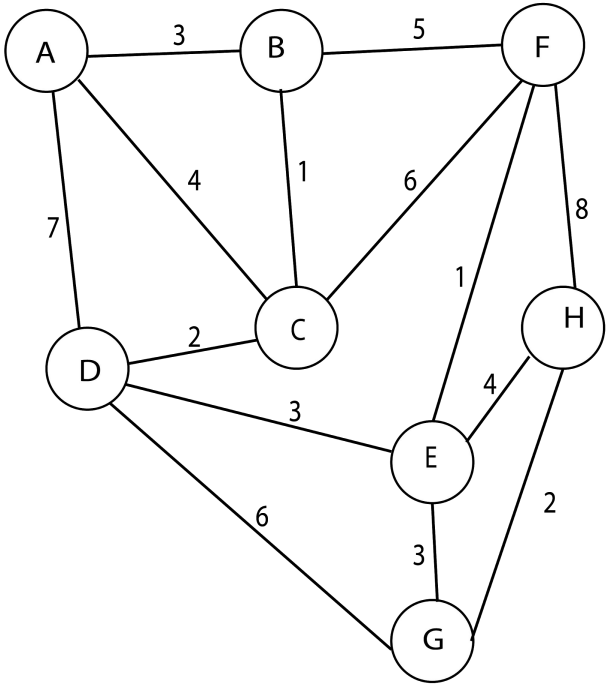
All the vertices have been visited.

LET’S TRY OUT !!!


A B C D E F G H I J K L M N O P Q R S T U V W
1 2 3 4 5 6 7 8 9 10 11 12 13 14
0
ARABIAN SEA
HOSTELS
GUEST HOUSE
ACADEMIC BLOCKS
EATING JOINTS
UNDERPASS
GATES

The Institution of Engineers,
NITK Chapter

NITK MAP



A- NITK HOSTEL

B- NITK MAIN BUILDING

C- NESCAFE

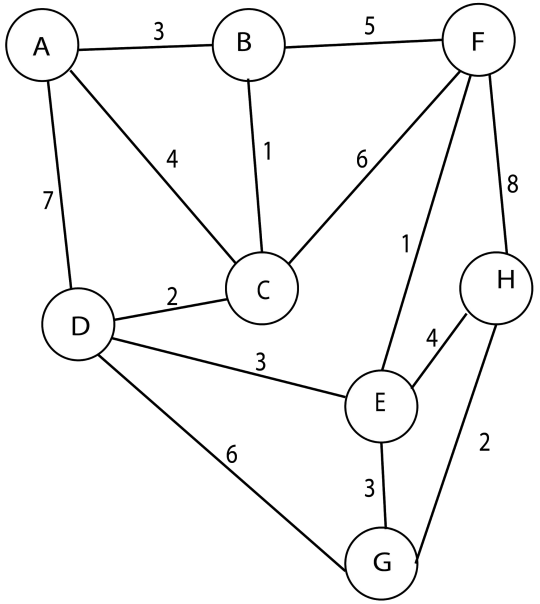
D-MAIN LECTURE HALL

E- MAIN GROUND

F- SWIMMING POOL

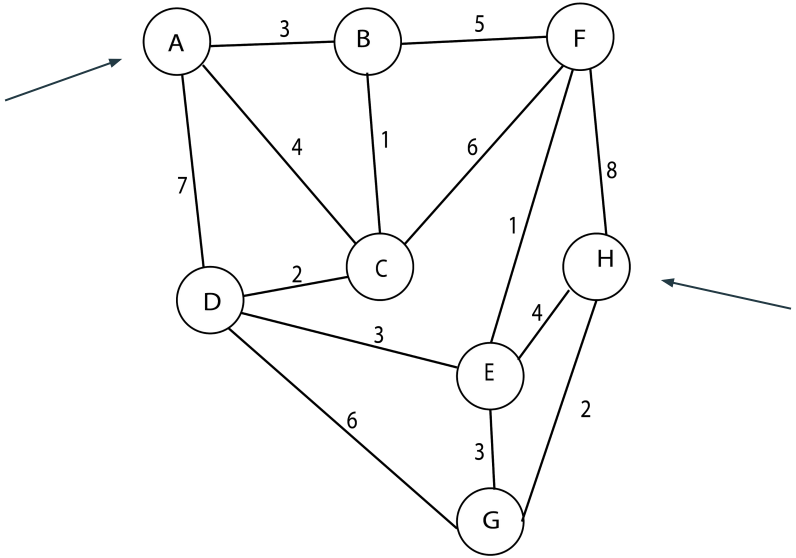
G-CENTRAL LIBRARY

H- SILVER JUBILEE AUDITORIUM



The Institution of Engineers,
NITK Chapter

DISTANCE BETWEEN NITK HOSTEL AND SILVER JUBILEE AUDITORIUM



The Institution of Engineers,
NITK Chapter

THANKYOU !!!