SHORTEST PATH FINDING ALGORITHM

It is the shortest distance path from one source to another vertices

- All pair shortest path
- Single source

To find the shortest path in a graph. There are 2 types of algorithm to find shortest path.

1. All pair shortest path

   To find the shortest path from each vertex to every other vertex using the algorithm Floyd warshall's Algorithm.

2. Single source shortest path

To find the shortest path from a single source vertex 'u' to a destination vertex 'v' using the algorithm Dijkstra's Algorithm.

Dijkstra's Algorithm

Step 1: Assign every node in tentative distance infinite.

Step 2: set initial node as current and other nodes as unvisited.

Step 3: For current node consider all other unvisited nodes and calculate tentative distance. compare current distance with calculated distance and assign the smaller value.

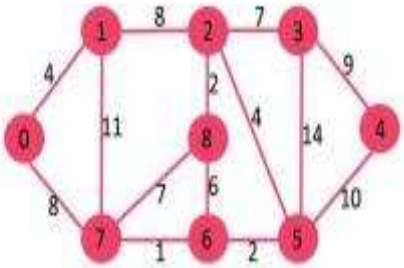
$d(u) + c(u, v) < d(v)$

$d(v) = d(u) + c(u, v)$

Step 4: When all the neighbors are considered of the current node make it visited.

Step 5: When the destination node is visited then we can stop the process.

Let us understand with the following example:

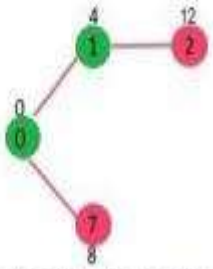


Zero is the source vertex.

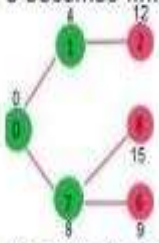
Assign a tentative distance.

The set *sptSet* is initially empty and distances assigned to vertices are {0, INF, INF, INF, INF, INF, INF, INF} where INF indicates infinite. Now pick the vertex with minimum distance value. The vertex 0 is picked, include it in *sptSet*. So *sptSet* becomes {0}. After including 0 to *sptSet*, update distance values of its adjacent vertices. Adjacent vertices of 0 are 1 and 7. The distance values of 1 and 7 are updated as 4 and 8. Following subgraph shows vertices and their distance values, only the vertices with finite distance values are shown. The vertices included in SPT are shown in green colour.

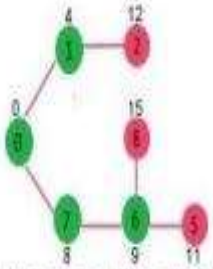
Pick the vertex with minimum distance value and not already included in SPT (not in sptSET). The vertex 1 is picked and added to sptSet. So sptSet now becomes {0, 1}. Update the distance values of adjacent vertices of 1. The distance value of vertex 2 becomes 12.



Pick the vertex with minimum distance value and not already included in SPT (not in sptSET). Vertex 7 is picked. So sptSet now becomes {0, 1, 7}. Update the distance values of adjacent vertices of 7. The distance value of vertex 6 and 8 becomes finite (15 and 9 respectively).



Pick the vertex with minimum distance value and not already included in SPT (not in sptSET). Vertex 6 is picked. So sptSet now becomes {0, 1, 7, 6}. Update the distance values of adjacent vertices of 6. The distance value of vertex 5 and 8 are updated.



We repeat the above steps until *sptSet* does include all vertices of given graph. Finally, we get the following Shortest Path Tree (SPT).