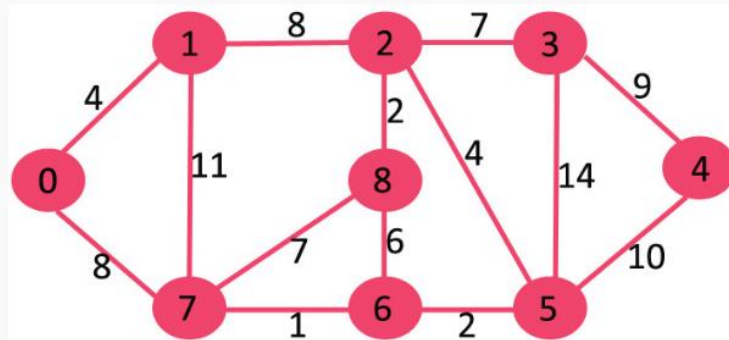


## DIJKSTRA ALGORITHM

**Problem Statement:** Given a graph and a source vertex in the graph, find the shortest paths from the source to all vertices in the given graph.

**Examples:**

*Input: src = 0, the graph is shown below.*



**Output:** 0 4 12 19 21 11 9 8 14

**Explanation:** The distance from 0 to 1 = 4.

The minimum distance from 0 to 2 = 12. 0->1->2

The minimum distance from 0 to 3 = 19. 0->1->2->3

The minimum distance from 0 to 4 = 21. 0->7->6->5->4

The minimum distance from 0 to 5 = 11. 0->7->6->5

The minimum distance from 0 to 6 = 9. 0->7->6

The minimum distance from 0 to 7 = 8. 0->7

The minimum distance from 0 to 8 = 14. 0->1->2->8

Follow the steps below to solve the problem:

- Create a set sptSet (shortest path tree set) that keeps track of vertices included in the shortest-path tree, i.e., whose minimum distance from the source is calculated and finalized. Initially, this set is empty.
- Assign a distance value to all vertices in the input graph. Initialize all distance values as INFINITE. Assign the distance value as 0 for the source vertex so that it is picked first.
- While sptSet doesn't include all vertices
  - Pick a vertex u which is not there in sptSet and has a minimum distance value.
  - Include u to sptSet.
  - Then 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 the sum of the 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  $v$ .

Note: We use a boolean array `sptSet[]` to represent the set of vertices included in SPT. If a value `sptSet[v]` is true, then vertex  $v$  is included in SPT, otherwise not. Array `dist[]` is used to store the shortest distance values of all vertices.

Explanation: [https://www.youtube.com/watch?v=Gd92jSu\\_cZk](https://www.youtube.com/watch?v=Gd92jSu_cZk)