Shortest Path

A fundamental problem in graphs is finding the shortest path from vertex A to vertex B. Fortunately there are several simple (and efficient algorithms for doing this).

There are different techniques to find the shortest path between vertex A to vertex B. one of the most popular shortest path algorithm is **Dijkestra Algorithm**.

Dijkestra Algorithm

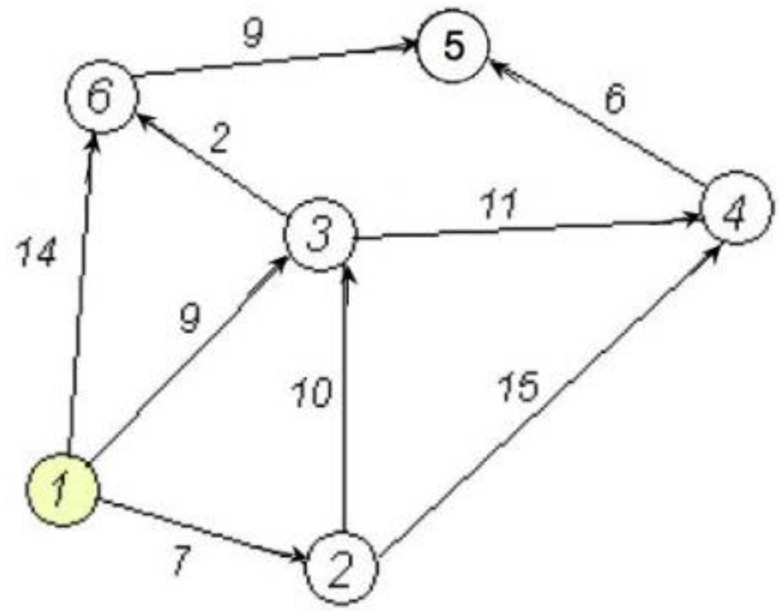
- Dijkstra's algorithm is applied to automatically find directions between physical locations, such as driving directions on Google Maps.
- In a networking or telecommunication applications, Dijkstra's algorithm has been used for solving the min-delay path problem (which is the shortest path problem). For example in data network routing, the goal is to find the path for data packets to go through a switching network with minimal delay.
- • It is also used for solving a variety of shortest path problems arising in plant and facility layout, robotics, transportation, and VLSI∗ design.

Dijkestra Algorithm Working

Suppose we want to find a shortest path from a given node **S** to other nodes in a network (one-to-all shortest path problem)

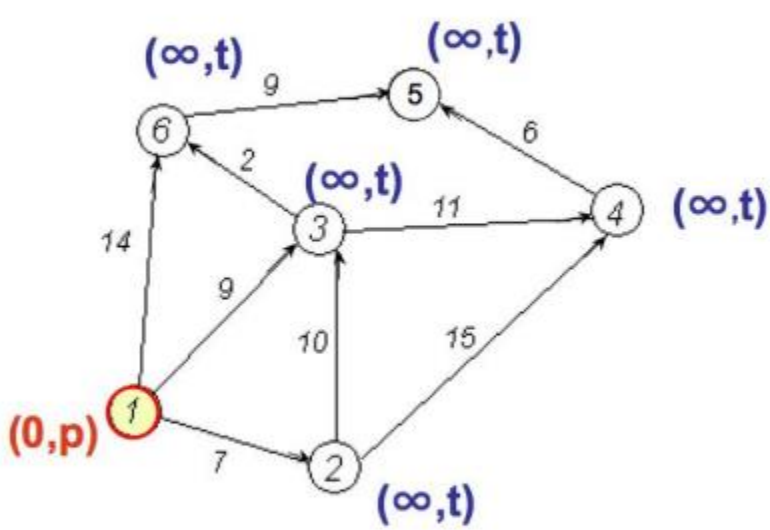
- It finds the shortest path from a given node s to all other nodes in the network
- Node **S** is called a starting node or an initial node
- How is the algorithm achieving this?
- Dijkstra's algorithm starts by assigning some initial values for the distances from node **S** and to every other node in the network
- It operates in steps, where at each step the algorithm improves the distance values.
- At each step, the shortest distance from node s to another node is determined

Dijkestra Algorithm Example



We want to find the shortest path from node 1 to all other nodes using Dijkstra's algorithm.

Initialization - Step 1

Node 1 is designated as the current node The state of node 1 is (0, p) Every other node has state (∞, t)	

Step 2

<table>
<tr>
<td>Nodes 2, 3,and 6 can be reached from the current node 1

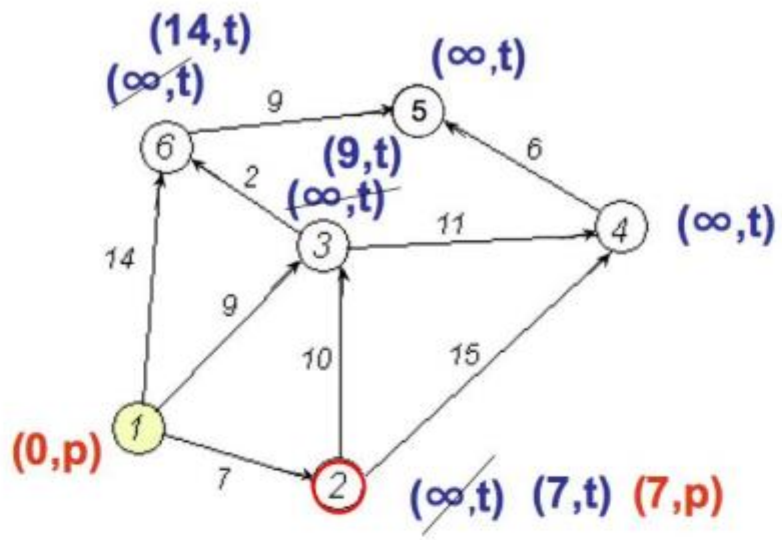
Update distance values for these nodes

d2 = min{∞, 0 + 7} = 7
d3 = min{∞, 0 + 9} = 9
d6 = min{∞, 0 + 14} = 14

Now, among the nodes 2, 3, and 6, node 2 has the smallest distance value

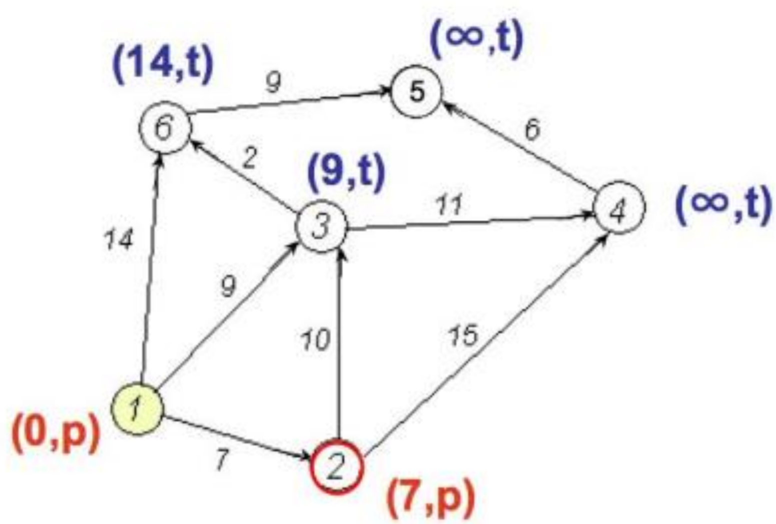
The status label of node 2 changes to permanent, so its state is (7, p),

while the status of 3 and 6 remains temporary

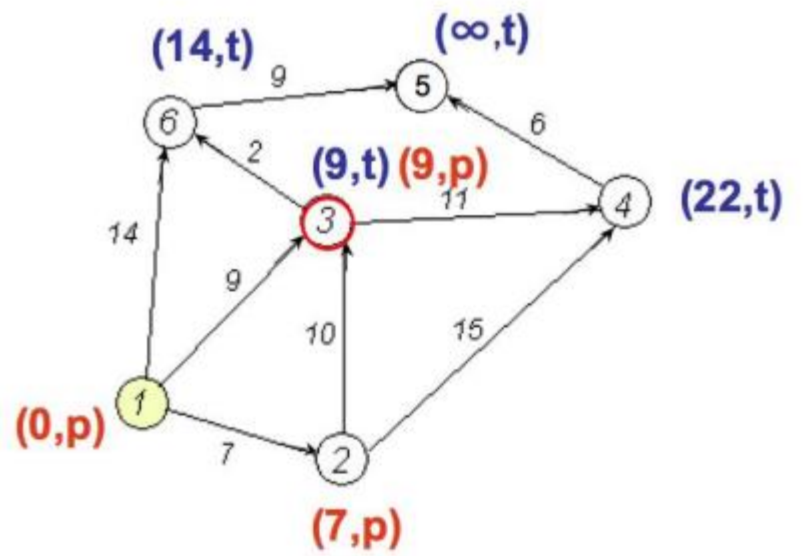
Node 2 becomes the current node</td>
<td>
</td>
</tr>
</table>

Step 3

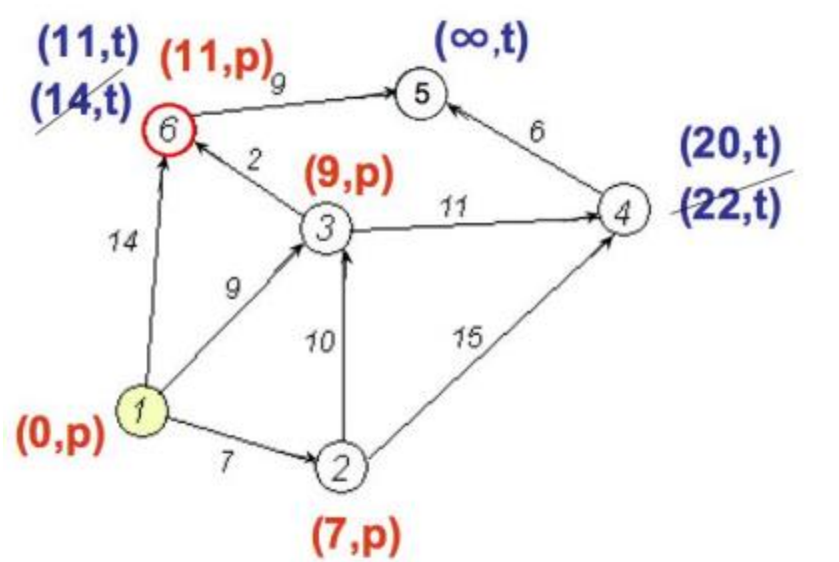
<table>
<tr>
<td>Graph at the end of Step 2

We are not done, not all nodes have been reached from node 1, so we perform another iteration (back to Step 2)</td>
<td>
</td>
</tr>
</table>

Another Implementation of Step 2

Nodes 3 and 4 can be reached from the current node 2 Update distance values for these nodes d3 = min{9, 7 + 10} = 9 d6 = min{∞, 7 + 15} = 22 Now, between the nodes 3 and 4 node 3 has the smallest distance value The status label of node 3 changes to permanent, while the status of 6 remains temporary Node 3 becomes the current node. We are not done (Step 3 fails), so we perform another Step 2	

Another Implementation of Step 2

Nodes 6 and 4 can be reached from the current node 3 Update distance values for them d4 = min{22, 9 + 11} = 20 d6 = min{14, 9 + 2} = 11 Now, between the nodes 6 and 4 node 6 has the smallest distance value The status label of node 6 changes to permanent, while the status of 4 remains temporary Node 6 becomes the current node. We are not done (Step 3 fails), so we perform another Step 2	

Another Step 2

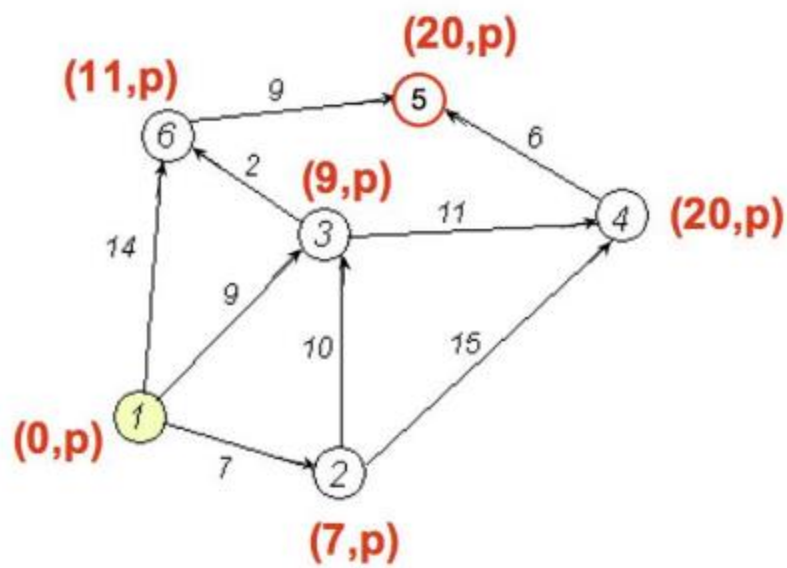
<table>
<tr>
<td>Node 5 can be reached from the current node 6

Update distance value for node 5
d5 = min{∞, 11 + 9} = 20

Now, node 5 is the only candidate, so its status changes to permanent

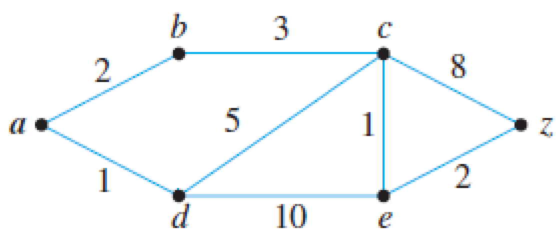
Node 5 becomes the current node

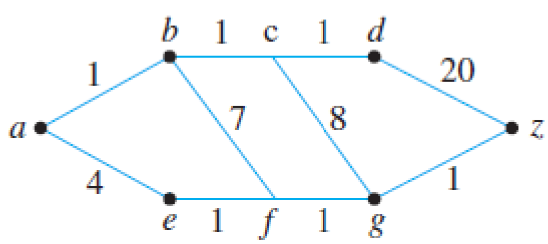
From node 5 we cannot reach any other node.

Hence, node 4 gets permanently labeled and we are done.</td>
<td>
</td>
</tr>
</table>

Exercise Questions

Q 1: Use Dijkestra Algorithm to find the shortest path from a-z for each of the given graph. (10)





Q 2: (4, 6)

a. A pipeline is to be built that will link six cities. The cost (in hundreds of millions of dollars) of constructing each potential link depends on distance and terrain and is shown in the weighted graph below. Find a system of pipelines to connect all the cities and yet minimize the total cost.