Aim: Dynamic Programming Approach: Multistage graph

A multistage graph G=(V,E) which is a directed graph. In this graph all the vertices are partitioned into the k stages where k>=2.

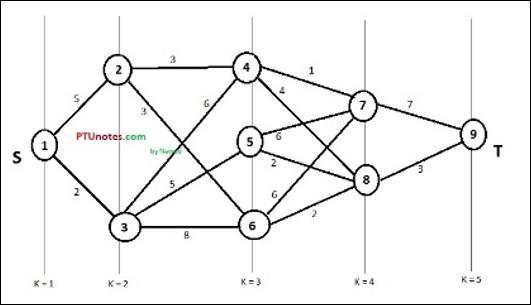
In multistage graph problem we have to find the shortest path from source to sink.There is set of vertices in each stage. The multistage graph can be solved using forward and backward approach.

***G*** is usually assumed to be a weighted graph. In this graph, cost of an edge *(i, j)* is represented by *c(i, j)*. Hence, the cost of path from source ***s*** to sink ***t*** is the sum of costs of each edges in this path.

The multistage graph problem is finding the path with minimum cost from source ***s*** to sink ***t***.

## Example

Consider the following example to understand the concept of multistage graph.



According to the formula, we have to calculate the cost **(i, j)** using the following steps

### Step-1: Cost (K-2, j)

In this step, three nodes (node 4, 5. 6) are selected as **j**. Hence, we have three options to choose the minimum cost at this step.

*Cost(3, 4) = min {c(4, 7) + Cost(7, 9),c(4, 8) + Cost(8, 9)} = 7*

*Cost(3, 5) = min {c(5, 7) + Cost(7, 9),c(5, 8) + Cost(8, 9)} = 5*

*Cost(3, 6) = min {c(6, 7) + Cost(7, 9),c(6, 8) + Cost(8, 9)} = 5*

### Step-2: Cost (K-3, j)

Two nodes are selected as j because at stage k - 3 = 2 there are two nodes, 2 and 3. So, the value i = 2 and j = 2 and 3.

*Cost(2, 2) = min {c(2, 4) + Cost(4, 8) + Cost(8, 9),c(2, 6) +*

*Cost(6, 8) + Cost(8, 9)} = 8*

*Cost(2, 3) = min {c(3, 4) + Cost(4, 9) + Cost(8, 9),c(3, 5) + Cost(8, 9)} = 7*

### Step-3: Cost (K-4, j)

*Cost (1, 1) = min {c(1, 2) + Cost(2, 6) + Cost(6, 8) + Cost(8, 9),*

*c(1, 3) + Cost(3, 6) + Cost(6, 8 + Cost(8, 9))} = 13*

Hence, the path having the minimum cost is **1→ 2→ 6→ 8→ 9**.

Algorithm:

Dynamic Programming solution:

Let **path(i,j)** be some specification of the minimal path from vertex **j** in set **i** to vertex **t**; **C(i,j)** is the cost of this path; **c(j,t)** is the weight of the edge from **j** to **t**.

**C(i,j) = min { c(j,l) + C(i+1,l) }**

**l in Vi+1**

**(j,l) in E**

To write a simple algorithm, assign numbers to the vertices so those in stage **Vi** have lower number those in stage **Vi+1**.

**int[] MStageForward(Graph G)**

**{**

**// returns vector of vertices to follow through the graph**

**// let c[i][j] be the cost matrix of G**

**int n = G.n (number of nodes);**

**int k = G.k (number of stages);**

**float[] C = new float[n];**

**int[] D = new int[n];**

**int[] P = new int[k];**

**for (i = 1 to n) C[i] = 0.0;**

**for j = n-1 to 1 by -1 {**

**r = vertex such that (j,r) in G.E and c(j,r)+C(r) is minimum**

**C[j] = c(j,r)+C(r);**

**D[j] = r;**

**}**

**P[1] = 1; P[k] = n;**

**for j = 2 to k-1 {**

**P[j] = D[P[j-1]];**

**}**

**return P;**

**}**

Using dynamic approach programming strategy, the multistage graph problem is solved. This is because in multistage graph problem we obtain the minimum path at each current stage by considering the path length of each vertex obtained in earlier stage.