**Shri Vile Parle Kelavani Mandal’s**

**Institute of Technology, Dhule**

****

**Design and analysis of algorithm lab assignment**

Name : Nikita Ramkrushna Patil

Roll No : 42

PRN : 2254491246043

Class : Second Year (IT)

**Bellman Ford Algorithm**

Theory :

Dijkstra’s algorithm works properly only for non negative weights. Dijkstra’s algorithm fails if the graph consists of negative weights.This is so because in Dijkstra’s algorithm once a vertex is made permanent we don’t relabel it. i.e. shortest path to it is finalized.In Dijkstra’s vertex made permanent at each step and shortest distance to a vertex is finalized at each step but in Bellman Ford algorithm the shortest distances are not finalized till the end of the algorithm . Thus in Bellman-Ford algorithm, we drop the concept of making vertices permanent . This is why Dijkstra’s algorithm is known as label setting algorithm and Bellman Ford algorithm is known as label correcting algorithm.

Algorithm:

Step 1:

Initialize the pathlength of all vertices to infinity and predecessor of all vertices to NIL.

Step 2:

Make the pathlength of source vertex equal to 0 and insert it into the queue.

Step 3:

Delete a vertex from the front of the queue and make it the current vertex.

Step 4:

Examine all the vertices adjacent to the current vertex. Check the condition of minimum weight for these vertices and do relabeling if required, and the relabeling is done by using the formula ,

if pathlength (current ) + weight (current,v) < pathlength(v)

pathlength(v) = pathlength (current ) + weight (current,v)

predecessor(v) = current

If pathlength of destinathion vertex is less than the pathlength (current ) + weight (current,v) then no relabeling is done.

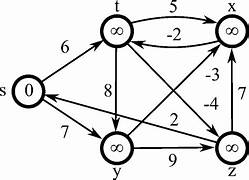
Step 5:

Each vertex that is relabeled is inserted into the queue provided it is not already present in the queue.

Step 6:

Repeat the steps (2),(3) and (5) till the queue becomes empty.

**Problem :** Find the shortest path using Bellman Ford algorithm for the following graph . Note that vertex z is source vertex.



**Solution:**

|  |  |  |  |
| --- | --- | --- | --- |
| Current | Adjacent | Pathlength Redefination | Queue |
| z | s,x | **s** 0+2 <∞ pathlen(s)=2 , pred(s)=z, Enqueue s  **x** 0+7 <∞ pathlen(x)=7 , pred(x)=z, Enqueue x | s,x |
| s | t,y | **t** 2+6 <∞ pathlen(t)=8 , pred(t)=s, Enqueue t  **y** 2+7 <∞ pathlen(y)=9 , pred(y)=s, Enqueue y | x,t,y |
| x | t | **t 7**+(-2) <8 pathlen(t)=5 , pred(t)=x, t already in  queue | t,y |
| t | x,y,z | **x** 5+5 > 7 Not relabel  **y** 5+8 > 9 Not relabel  **z** 5+(-4) > 0 Not relabel | y |
| y | z | **z** 9+9 >0 Not relabel | Queue is empty |

The shortest paths in this way computed by Bellman Ford algorithm for source vertex z are given below-

|  |  |  |  |
| --- | --- | --- | --- |
| Source | Destination | Shortest Path | Length |
| z | t | z-x-t | 5 |
| z | x | z-x | 7 |
| z | y | z-y | 9 |
| z | s | z-s | 2 |