Yatish Kelkar TE IT 8001

DAA Assignment 3

* Aim : Write a program to implement Bellman-Ford algorithm using Dynamic programming & verify time complexity.

* Theory :

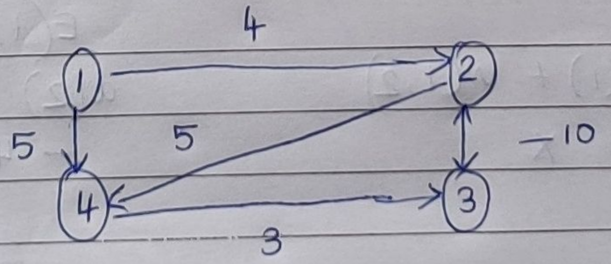
- The Bellman-Ford algorithm serves the single source shortest path problems in which edge weights may be negative.

- Algorithm returns a boolean value indicating whether or not there is a negative weight cycle that is reachable from the source.

- The algorithm relaxes edges, progressively decreasing an estimate $V; \delta$ on the weight of the shortest path from source S to each vertex $v \in V$ until it achieves actual shortest path weight $\delta(S, v)$

- Returns true if graph contains no negative weight cycles that are reachable from source.

- Example :



Edges		Node	π	δ
1 2	$\rightarrow$ 4	1	N	0
1 4	$\rightarrow$ 5	2	N	$\cancel{\infty}$ $\cancel{4}$ $\cancel{-2}$ $\cancel{-4}$ -6
2 4	$\rightarrow$ 5	3	N	$\cancel{\infty}$ $\cancel{8}$ $\cancel{6}$ 4
4 3	$\rightarrow$ 3	4	N	$\cancel{\infty}$ $\cancel{5}$ $\cancel{3}$ 1
3 2	$\rightarrow$ -10			

1st path (4, 3)

E(1,2)

$\delta(2) > d(1) + w(1,2)$

$\infty > 4$

$d(3) > d(4) + w(4,3)$

$\infty > 8$

(1,4)

$d(4) > d(1) + w(1,4)$

$\infty > 0 + 5$

$\infty > 5$

(3,2)

$d(2) > d(3) + w(3,2)$

$4 > 8 - 10$

$4 > -2$

(2,4)

$d(4) > d(2) + w(2,4)$

$5 > 4 + 5$

$5 > 9$

2nd path

E(1,2)

$d(2) > d(1) + w(1,2)$

$-2 > 4$ X

3rd path

E(1,2)

$d(2) > d(1) + w(1,2)$

$-4 > 4$ X

$E(1,4)$
$d(4) > d(1) + w(1,4)$
$5 > 5$ ×

$E(1,4)$
$d(4) > d(1) + w(1,4)$
$3 > 5$ ×

$E(2,4)$
$d(4) > d(2) + w(2,4)$
$5 > 3$ ✓

$E(2,4)$
$d(4) > d(2) + w(2,4)$
$3 > 1$

$E(4,3)$
$d(3) > d(4) + w(4,3)$
$8 > 6$ ✓

$E(4,3)$
$d(3) > d(4) + w(4,3)$
$6 > 4$ ✓

$E(3,2)$
$d(2) > d(3) + w(3,2)$
$-2 > -4$
$d(2) \rightarrow -4$

$E(3,2)$
$d(2) > d(3) + w(3,2)$
$-4 > -6$
$d(2) \rightarrow -6$

Consider 1^{st} iteration
$d(2) > d(1) + w(1,2)$
$-6 > 4$ ×

$d(4) > d(1) + w(1,4)$
$1 > 0+5$ ×

$d(4) > d(2) + w(2,4)$
$1 > -1$ ✓

Returns false. Graph contains –ve cycle.

* Algorithm

i) Initialize single source (G,S)
ii) For i = 1 to |G, V| - 1
iii) For each edge (U, V) ∈ G, E
iv) Relax (U, V, W)
v) For each edge (U,V) ∈ E
vi) if (V,d) < (U,d) + w(U,V)
return False
vii) return true

* Time Complexity : Bellman Ford algorithm runs in time O(V.E) ti
takes O(E) & worst case will be θ |V| |E|

* Conclusion : Thus we have implemented Bellman Ford algorithm