Bellman-Ford Algorithm:

The Bellman-Ford Algorithm operates on an input graph G, with |V| vertices and |E| edges. A single source vertex S, must be provided as well, as the Bellman-Ford algorithm is a single-source shortest path algorithm. No destination vertex needs to be supplied, however, because Bellman-Ford calculates the shortest distance to all vertices in the graph from the source vertex.

The Bellman-Ford algorithm, ~~operates~~ like Dijkstra's alg. uses the principle of relaxation to find increasingly accurate path length. Bellman-Ford though tackles two main issues with this process.

1. If there are negative weight cycles, the search for a shortest path will go on forever.
2. Choosing a bad ordering for relaxations leads to exponential relaxations.

The detection of negative cycles is important, but the main contribution of this algorithm is in its ordering of relaxations. Dijkstra's algorithm is a greedy alg. that selects the nearest vertex that has not been processed. Bellman-Ford on the other hand, relaxes all the edges.

Bellman-Ford labels the edges for a graph G as

$$e_1, e_2, \ldots e_m,$$

and that set of edges is relaxed exactly $|V|-1$ times, where $|V|$ is the number of vertices in the graph.

Algorithm Psuedo-code :

```
for v in V :
    v.distance = infinity
    v.p = None
source.distance = 0
for i from 1 to |V| - 1 :
    for (u,v) in E :
        relax (u,v)
for (u,v) in E :
    if v.distance > u.distance + weight (u,v) :
        print "A negative weight cycle exists"
```

complexity :
$O(VE)$

```
relax (u,v) :
    if v.distance > u.distance + weight (u,v) :
        v.distance = u.distance + weight (u,v)
        v.p = u.
```

→ First for loop sets the distance to each vertex in the graph to infinity and later changed for the source vertex to zero and the p value for each vertex is set to nothing. This value is a pointer to a predecessor (parent) vertex so that we can create a path later.

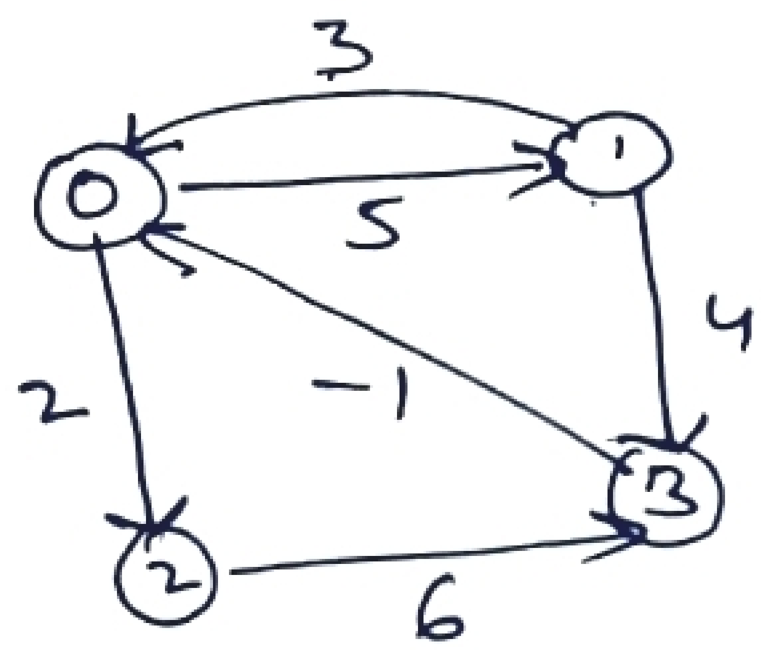
→ Next for loop simply goes through each edge (u,v) in E and relaxes it. This process is done |V|-1 times.

Relaxation is the most important step in Bellman-Ford. It increases the accuracy of the distance to any given vertex. Relaxation works by continuously shortening the calculated distance between vertices comparing that distance with other known distances.

A very short and simple addition to the Bellman-Ford algorithm can allow it to detect negative cycles, which is very important because it disallows shortest-path finding together.

A negative cycle in a weighted graph is a cycle whose total weight is negative.

Ex:-



Find the shortest path from vertex 0 to all other vertices using Bellman-ford algorithm.

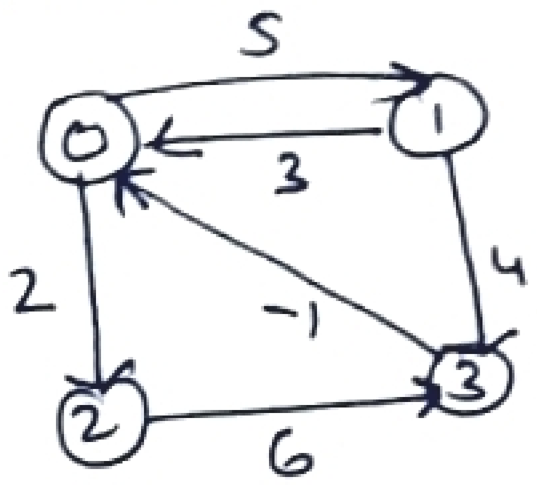
Sol

Edge	weight
$0 \to 1$	5
$0 \to 2$	2
$1 \to 0$	3
$1 \to 3$	4
$2 \to 3$	6
$3 \to 0$	-1

4

	0	1	2	3
initialization d	0	∞	∞	∞
iteration 1	0	5	2	9
iteration 2	0	5	2	~~9~~ 8
iteration 3	0	5	2	8

	0	1	2	3
initialization p	-	-	-	-
iteration 1	-	0	0	1
iteration 2	-	0	0	~~1~~ 2
iteration 3	-	0	0	2



edge	weight
$0 \to 1$	5
$0 \to 2$	2
$1 \to 0$	3
$1 \to 3$	4
$2 \to 3$	6
$3 \to 0$	-1

iteration 1

1) edge $0 \to 1$

relax (0, 1)

$u = 0,\ v = 1$

$v.d = \infty,\ u.d = 0$

$w(u, v) = 5$

relax (0, 1)

$1.d > 0.d + w(0, 1)$

$\infty > 0 + 5$ (T)

$1.d = 0 + 5 = 5$

$1.p = u = 0$

2) edge $0 \to 2$

$u = 0,$

$v = 2,$

$v.d = \infty,$

$u.d = 0$

$w(u, v) = 2$

relax (0, 2)

$2.d > 0.d + w(0, 2)$

$\infty > 0 + 2$ (T)

$2.d = 0 + 2 = 2$

$2.p = u = 0$

3) edge $1 \to 0$

$u = 1,$

$v = 0$

$v.d = 0$

$u.d = 5$

$w(1, 0) = 3$

relax (1, 0)

$0.d > 1.d + w(1, 0)$

$0 > 5 + 3$ (F)

4) edge $1 \to 3$

$u = 1,$

$v = 3,$

$3.d = \infty$

$1.d = 5$

$w(1, 3) = 4$

relax (1, 3)

$3.d > 1.d + w(1, 3)$

$\infty > 5 + 4 \Rightarrow 9$

$3.d = 9$

$3.p = 1$

5) edge $2 \to 3$ (5)

$u = 2$,
$v = 3$
$3.d = 9$
$2.d = 2$
$w(2,3) = 6$

relax(2,3)
$3.d > 2.d + w(2,3)$
$9 > 2 + 6$ (F)

6) $3 \to 0$

$u = 3$
$v = 0$
$3.d = 9$
$0.d = 0$
$w(3,0) = -1$

relax(3,0)
$0.d > 3.d + w(3,0)$
$0 > 9 + (-1)$ (F)

---

iteration 2

(1) ~~end~~ edge $0 \to 1$

$u = 0$
$v = 1$
$1.d = 5$
$0.d = 0$
$w(0,1) = 5$

relax(0,1)
$1.d > 0.d + w(0,1)$
$5 > 0 + 5$ (F)

(2) edge $0 \to 2$

$u = 0$
$v = 2$
$2.d = 2$
$0.d = 0$
$w(0,2) = 2$

relax(0,2)
$2.d > 0.d + w(0,2)$
$2 > 0 + 2$ (F)

(3) edge $1 \to 0$

$u = 1$
$v = 0$
$u.d = 1.d = 5$
$v.d = 0.d = 0$
$w(1,0) = 3$

relax(1,0)
$0.d > 1.d + w(1,0)$
$0 > 5 + 3$ (F)

(4) edge $1 \to 3$

$u = 1$
$v = 3$
$u.d = 1.d = 5$
$v.d = 3.d = 9$
$w(1,3) = 4$

relax(1,3)
$3.d > 1.d + w(1,3)$
$9 > 5 + 4$ (F)

6

5) edge $2 \to 3$

$u = 2$
$v = 3$
$u.d = 2.d = 2$
$v.d = 3.d = 9$
$w(2,3) = 6$

relax (2, 3)

$3.d > 2.d + w(2,3)$
$9 > 2 + 6$
$9 > 8$ (T)
$3.d = 8$
$3.p = 2$

6) edge $3 \to 0$

$u = 3$
$v = 0$
$u.d = 3.d = 8$
$v.d = 0.d = 0$
$w(3,0) = -1$

relax (3, 0)

$0.d > 3.d + w(3,0)$
$0 > 8 - 1$ (F)

---

iteration 3

1) edge $0 \to 1$

$u = 0$
$v = 1$
$u.d = 0.d = 0$
$v.d = 1.d = 5$
$w(0,1) = 5$

relax (0, 1)

$1.d > 0.d + w(0,1)$
$5 > 0 + 5$ (F)

2) edge $0 \to 2$

$u = 0$
$v = 2$
$u.d = 0.d = 0$
$v.d = 2.d = 2$
$w(0,2) = 2$

relax (0, 2)

$2.d > 0.d + w(0,2)$
$2 > 0 + 2$ (F)

3) edge $1 \to 0$

$u = 1$
$v = 0$
$u.d = 1.d = 5$
$0.d = 0$
$w(1,0) = 3$

relax (1, 0)

$0.d > 1.d + w(1,0)$
$0 > 5 + 3$ (F)

4) edge $1 \to 3$

$u = 1$
$v = 3$
$u.d = 1.d = 5$
$v.d = 3.d = 8$
$w(1,3) = 4$

relax (1, 3)

$3.d > 1.d + w(1,3)$
$8 > 5 + 4$ (F)

5) edge $2 \to 3$ relax $(2,3)$ 7

$u = 2$

$v = 3$

$2 \cdot d = 2$

$3 \cdot d = 8$

$w(2,3) = 6$

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

$8 > 2 + 6$ (F)

6) edge $3 \to 0$ relax $(3,0)$

$u = 3$

$v = 0$

$3 \cdot d = 8$

$0 \cdot d = 0$

$w(3,0) = -1$

$0 \cdot d > 3 \cdot d + w(3,0)$

$0 > 8 - 1$ (F)

After 3 iterations ~~repeate the edge~~ check for detecting the negative cycle.

① edge $0 \to 1$

$v \cdot d > u \cdot d + w(u,v)$

$1 \cdot d > 0 \cdot d + w(0,1)$

$5 > 0 + 5$ (F)

② edge $0 \to 2$

$2 \cdot d > 0 \cdot d + w(0,2)$

$2 > 0 + 2$ (F)

③ edge $1 \to 0$

$0 \cdot d > 1 \cdot d + w(1,0)$

$0 > 5 + 3$ (F)

④ edge $1 \to 3$

$3 \cdot d > 1 \cdot d + w(1,3)$

$8 > 5 + 4$ (F)

⑤ edge $2 \to 3$

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

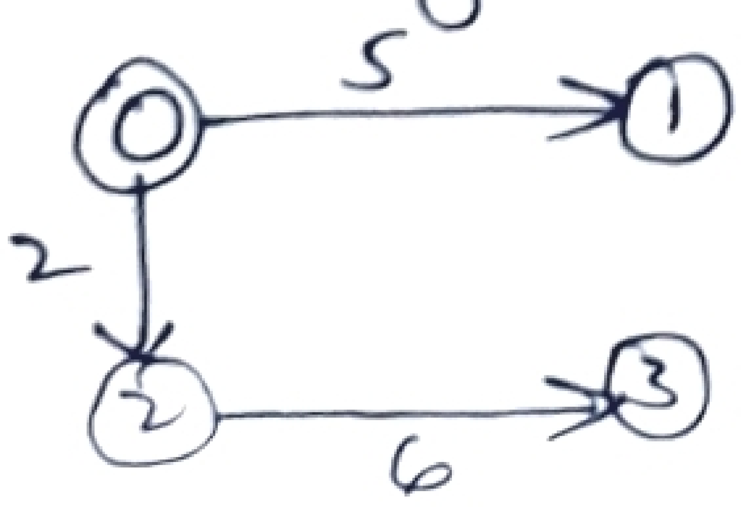
$8 > 2 + 6$ (F)

⑥ edge $3 \to 0$

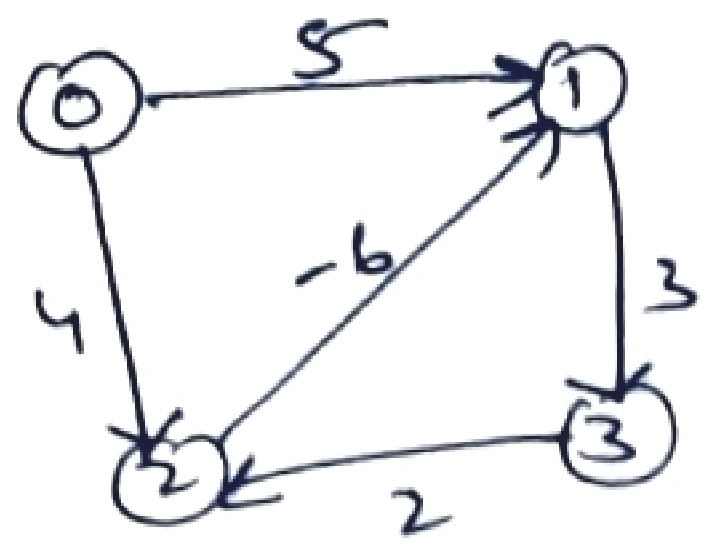
$0 \cdot d > 3 \cdot d + w(3,0)$ (F)

$0 > 8 - 1$

Since all the conditions were false there are no negative cycles in the graph and Iteration 3 gives the shortest ~~path~~ path from 0 to other vertices.



Ex 2



edge	w (u,v)
0 → 1	5
0 → 2	4
1 → 3	3
2 → 1	-6
3 → 2	2

	0	1	2	3
d	0	∞	∞	∞
iteration 1	0	$\cancel{5}$ 2	4	8
iteration 2	0	-2	$\cancel{4}$ 3	1
iteration 3	0	$\cancel{-2}$ -3	3	1

	0	1	2	3
p	-	-	-	-
it 1	-	$\cancel{0}$ 2	0	1
it 2	-	2	$\cancel{0}$ 3	1
it 3	-	2	3	1

iteration 1

① edge 0 → 1

u = 0
v = 1
v.d = ∞
u.d = 0
w(0,1) = 5

relax (0,1)
1.d > 0.d + w(0,1)
∞ > 0 + 5 (T)

1.d = 5
1.p = 0

② edge 0 → 2

u = 0
v = 2
2.d = $\cancel{5}$ ∞
0.d = 0
w(0,2) = 4

relax (0,2)
2.d > 0.d + w(0,1)
∞ > 0 + 4 (T)

2.d = 4
2.p = 0

③ edge 1 → 3

u = 1
v = 3
1.d = 5
3.d = ∞
w(1,3) = 3

relax (1,3)
3.d > 1.d + w(1,3)
∞ > 5 + 3 (T)

3.d = 8
3.p = 1

④ 2 → 1

u = 2
v = 1
2.d = 4
1.d = 5
w(2,1) = -6

relax (2,1)
1.d > 2.d + w(2,1)
5 > 4 + (-6) (T)

1.d = -2
1.p = 2

5) edge $3 \to 2$

$u = 3$

$v = 2$

$3.d = 8$

$2.d = 4$

$w(3,2) = 2$

relax (3,2)

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

$4 > 8 + 2$ (F)

---

iteration 2

① edge $0 \to 1$

$u = 0$

$v = 1$

$0.d = 0$

$1.d = -2$

$w(0,1) = 5$

relax (0,1)

$1.d > 0.d + w(0,1)$

$-2 > 0 + 5$ (F)

② edge $0 \to 2$

$u = 0$

$v = 2$

$0.d = 0$

$2.d = 4$

$w(0,2) = 4$

relax (0,2)

$2.d > 0.d + w(0,2)$

$4 > 0 + 4$ (F)

③ edge $1 \to 3$

$u = 1$

$v = 3$

$1.d = -2$

$3.d = 8$

$w(1,3) = 3$

relax (1,3)

$3.d > 1.d + w(1,3)$

$8 > -2 + 3$

$8 > 1$ (T)

$3.d = 1$

$3.p = 1$

④ edge $2 \to 1$

$u = 2$

$v = 1$

$1.d = -2$

$2.d = 4$

$w(2,1) = -6$

relax (2,1)

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

$-2 > 4 + -6$ (F)

⑤ edge $3 \to 2$

$u = 3$

$v = 2$

$3.d = 1$

$2.d = 4$

$w(3,2) = 2$

relax (3,2)

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

$4 > 1 + 2$ (T)

$2.d = 3$

$2.p = 3$

iteration 3

1. edge $0 \to 1$
   $u = 0$
   $v = 1$
   $0.d = 0$
   $1.d = -2$
   $w(0,1) = 5$

relax $(0,1)$
$1.d > 0.d + w(0,1)$
$-2 > 0 + 5$ (F)

2. edge $0 \to 2$
   $u = 0$
   $v = 2$
   $0.d = 0$
   $2.d = 3$
   $w(0,2) = 4$

relax $(0,2)$
$2.d > 0.d + w(0,2)$
$3 > 0 + 4$ (F)

3. edge $1 \to 3$
   $u = 1$
   $v = 3$
   $1.d = -2$
   $3.d = 1$
   $w(1,3) = 3$

relax $(1,3)$
$3.d > 1.d + w(1,3)$
$1 > -2 + 3$ (F)

4. edge $2 \to 1$
   $u = 2$
   $v = 1$
   $2.d = 3$
   $1.d = -2$
   $w(2,1) = -6$

relax $(2,1)$
$1.d > 2.d + w(2,1)$
$-2 > 3 + (-6)$ (T)
$1.d = -3$
$1.p = 2$

5. edge $3 \to 2$
   $u = 3$
   $v = 2$
   $3.d = 1$
   $2.d = -3$
   $w(3,2) = 2$

relax $(3,2)$
$2.d > 3.d + w(3,2)$
$-3 > 1 + 2$ (F)

Detecting -ve cycle

1. edge $0 \rightarrow 1$

   $1.d > 0.d + w(0,1)$

   $-3 > 0 + 5$ (F)

2. edge $0 \rightarrow 2$

   $2.d > 0.d + w(0,2)$

   $3 > 0 + 4$ (F)

3. edge $1 \rightarrow 3$

   $3.d > 1.d + w(1,3)$

   $1 > -3 + 3$

   $1 > 0$ (T)

$\Rightarrow$ negative cycle exists

$\Rightarrow$ NO shortest path.