Single Source Shortest Path(Bellman-Ford Algorithm)

we make the following observations:

1. If the shortest path from v to u with at most k, k > 1,edges has no more than

k -1 edges, then $dist^k[u] = dist^{k-1}[u]$.

2. If the shortest path from v to u with at most k, k > 1,edges has exactly k edges, then it is made up of a shortest path from v to some vertex j followed by the edge

{j, u}.The path from v to j has k – 1 edges, and its length is $dist^{k-1}[j]$. All vertices i such that the edge (i, u) is in the graph are candidates for j. Since we are interested in a shortest path, the i that minimizes $dist^{k-1}[i] + cost[i, u]$ is the correct value for j.

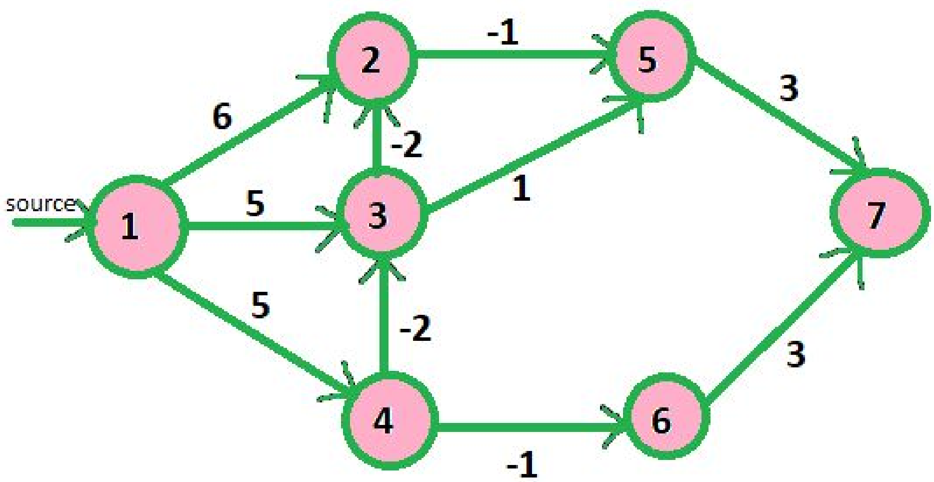
These observations result in the following recurrence for dist:

$$\mathbf{dist^k[u] = \min \{dist^{k-1}[u], \min_i\{dist^{k-1}[i] + cost[i, u]\}\}}$$

This recurrence can be used to compute $dist^k$ from $dist^{k-1}$ for, k = 2,3,...,n -1.

Example

The following figure gives a seven-vertex graph, together with the arrays $dist^k$, k = 1,.,6...These arrays were computed using the above equation.



For instance, $dist^k[1]$= 0 for all k since 1 is the source node. Also, $dist^1[2]$ = 6, $dist^1[3]$= 5, and $dist^1[4]$ = 5, since there are edges from 1 to these nodes. The distance $dist^i[]$ is ∞ for the nodes 5, 6, and 7 since there are no edges to these from 1.

Adjacency matrix or **cost matrix** of the above graph is following

	1	2	3	4	5	6	7
1	0	6	5	5	∞	∞	∞
2	∞	0	∞	∞	-1	∞	∞
3	∞	-2	0	∞	1	∞	∞
4	∞	∞	-2	0	∞	-1	∞
5	∞	∞	∞	∞	0	∞	3
6	∞	∞	∞	∞	∞	0	3
7	∞	∞	∞	∞	∞	∞	0

$dist^2[2] = \min\{dist^1[2], \min_i\{dist^1[i] + cost[i, 2]\}$

$= \min\{6, \min\{0+6, 5 – 2, 5 + \infty, \infty + \infty, \infty + \infty, \infty + \infty\}\}$

$= \min\{6, 3\}$

$= 3$

Here the terms $0 +6$, $5 - 2$, $5 + \infty$, $\infty + \infty$, $\infty + \infty$, $\infty + \infty$ correspond to a choice of i = 1,3,4,5,6,and 7 respectively.

$dist^2[3] = \min\{dist^1[3], \min_i\{dist^1[i] + cost[i, 3]\}$

$= \min\{5, \min\{0+5, 6+\infty, 5-2, \infty+\infty, \infty+\infty, \infty+\infty\}$

$= \min\{5, 3\}$

$= 3$

here i = 1,2,4,5,6,7 respectively. Similarly you solve

$dist^2[4]$,.....,$dist^2[7]$,$dist^3[2]$,.....,$dist^3[7]$,$dist^4[2]$,........,$dist^4[7]$,$dist^5[2]$,.....,$dist^5[7]$,$dist^6[2]$, ...$dist^6[7]$.

k	$Dist^k[1....7]$						
	1	2	3	4	5	6	7
1	0	6	5	5	∞	∞	∞
2	0	3	3	5	5	4	∞
3	0	1	3	5	2	4	7
4	0	1	3	5	0	4	5
5	0	1	3	5	0	4	3
6	0	1	3	5	0	4	3

Algorithm

```
Algorithm BellmanFord(v, cost, dist, n)
// Single-source/all-destination shortest paths with negative edge costs
{
    for i =1 to n do                    // Initialize dist.
        dist[i] = cost[v, i];
    for k = 2 to n – 1 do
    for each u such that u ≠ v and u has at least one incoming edge do
    for each(i, u) in the graph do
        if i ≠ u do
        {
            if dist[u]>dist[i] + cost[i, u] then
            dist[u] =dist[i] + cost[i, u];
        }
}
```

Time Complexity

Time complexity of Bellman-Ford algorithm is $\Theta(|V||E|)$ where $|V|$ is number of vertices and $|E|$ is number of edges.

If the graph is complete, the value of

$$|E| = |V|(|V|-1)/2 \approx \Theta(|V|^2).$$

So overall time complexity becomes $\Theta(|V|^3)$.