(4) Implement Bellman-Ford and Dijkstra's algorithm for shortest path finding algorithm.

⊙

shortest path problem

Bellman-Ford Algorithm

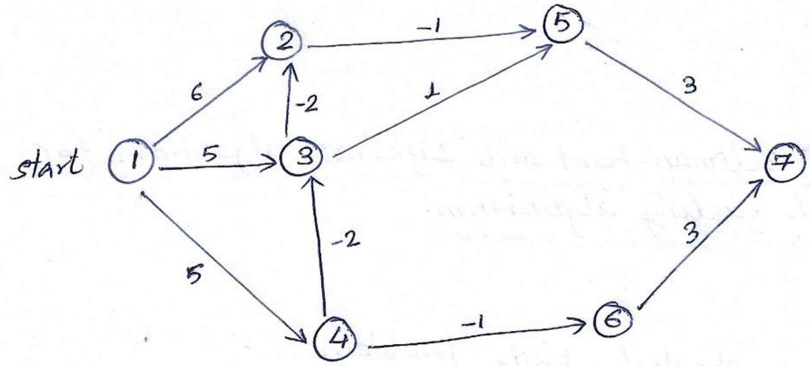
Dijkstra's Algorithm

Both will solve the single source shortest path problems.

⊙ Bellman-Ford Algorithm -

This is a single source shortest path.

We need to select a source vertex and find the shortest path to all other nodes. This allows to use negative weighted edges. This algorithm follows Dynamic programming strategy. This says that go on relaxing all the edges (n-1) times where n = no. of vertices.



so here n = no. of vertices = 7, so our algorithm will take 6 iterations.

Path length K ↓ / vertices →	1	2	3	4	5	6	7
0	0	∞	∞	∞	∞	∞	∞
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

so we get below distances from source - 1

Vertex	Distance from Vertex-1
1	0
2	1
3	3
4	5
5	0
6	4
7	3

Time complexity:

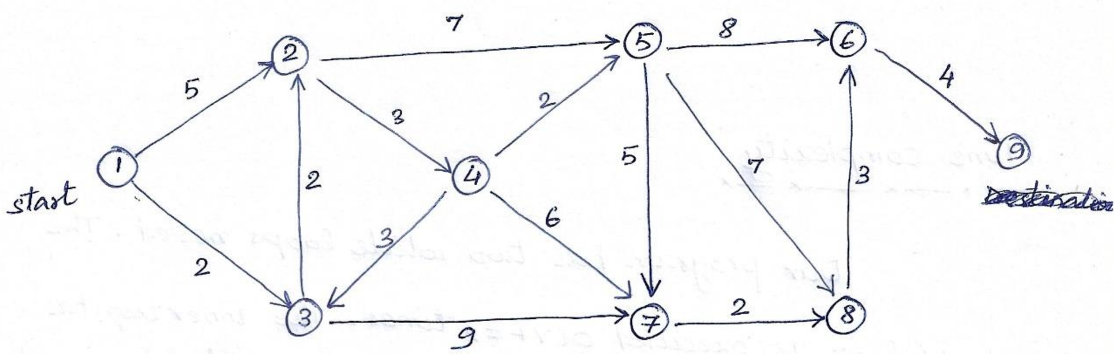
Since Dijkstra algorithm does not work for Graph with negative weights, Bellman-Ford works for such graphs. Bellman-ford is also simpler than Dijkstra and suites well for distributed systems. But time complexity of Bellman-ford algorithm is $O(VE)$

which is more than Dijkstra.

Dijkstra's Algorithm :-

This works on weighted graph $G(V, E)$ where all edges are non-negative.

Given graph



Vertex	1	2	3	4	5	6	7	8	9
Initial	0	∞	∞	∞	∞	∞	∞	∞	∞
step-1 V_1	0	5	2	∞	∞	∞	∞	∞	∞
step-2 $\underline{V_3}$	0	4	2	∞	∞	∞	11	∞	∞
step-3 V_2	0	4	2	7	11	∞	11	∞	∞
step-4 V_4	0	4	2	7	9	∞	11	∞	∞
step-5 V_5	0	4	2	7	9	17	11	16	∞
step-6 V_7	0	4	2	7	9	17	11	13	∞
step-7 V_8	0	4	2	7	9	16	11	13	∞
step-8 V_6	0	4	2	7	9	16	11	13	20

Hence minimum (shortest) distance of vertex 9 from vertex-1 is 20 and path is

$1 \xrightarrow{2} 3 \xrightarrow{9} 7 \xrightarrow{2} 8 \xrightarrow{3} 6 \xrightarrow{4} 9$

Time complexity

Our program has two while loops nested. The inner loop is executed $O(V+E)$ times. The inner loop has decreaskey() which takes $O(\log V)$ time. So overall time complexity

$= O((E+V) * \log V)$

$= O(E \log V)$