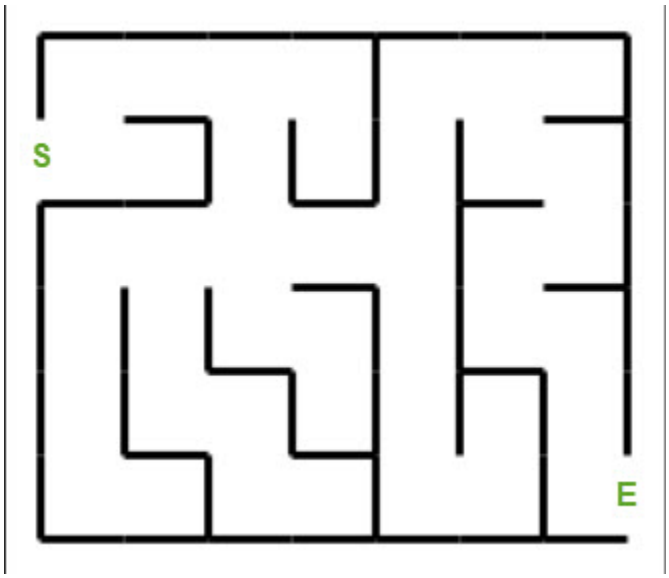
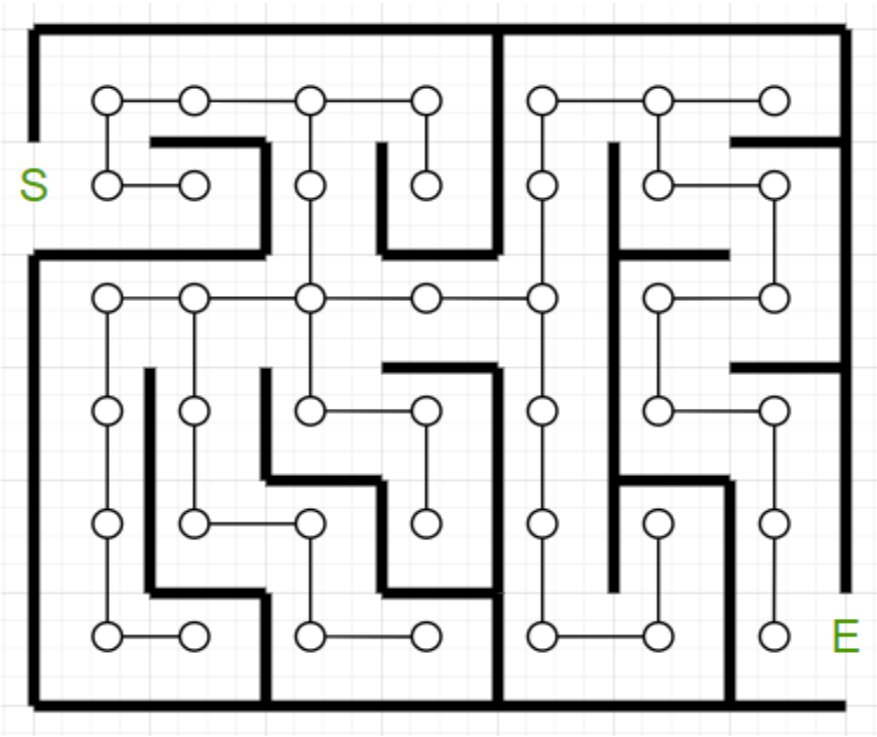
Week 12 Homework 2:

Q24. Project: Use Bellman Ford's Algorithm to find the shortest path of a maze.

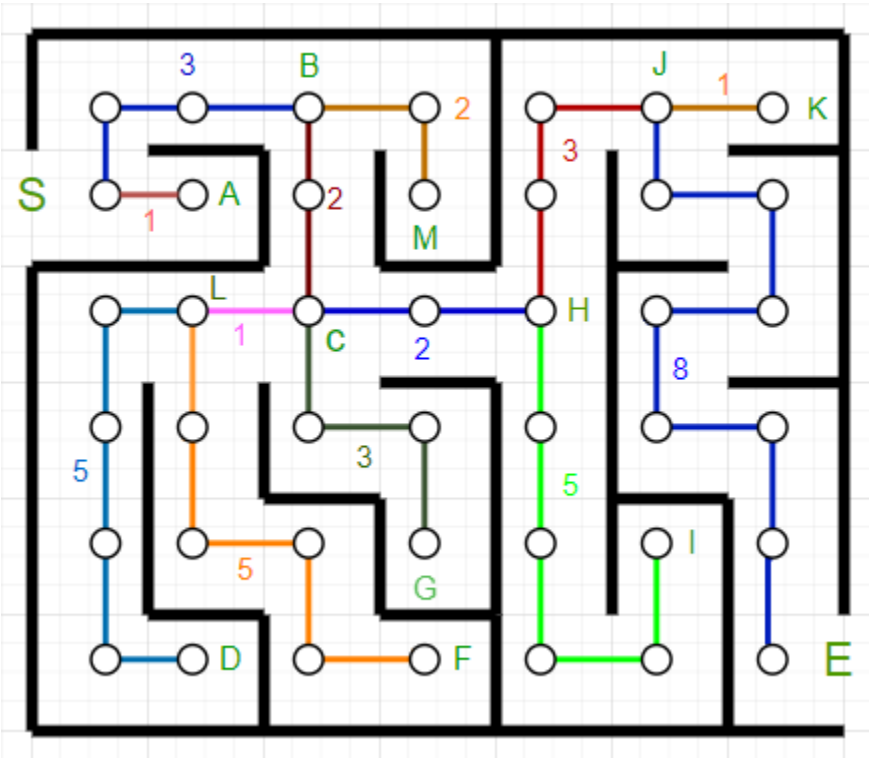


Answer:

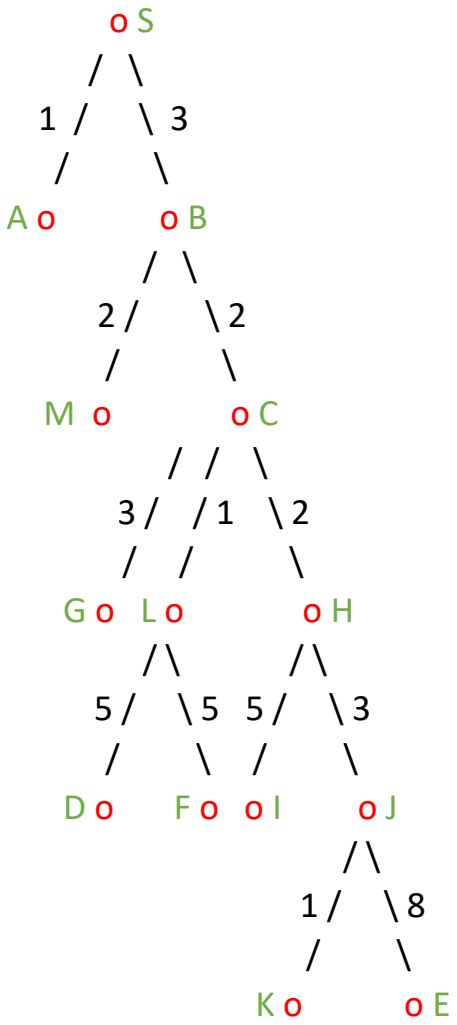
Step 1.



Step 2.



Step 3. Convert the maze into tree



Step 4. Find the distance of each vertex.

The Bellman-Ford algorithm propagates correct distance estimate |V -1| (V: Vertices)

So, Maze has total 14 vertices = |14 - 1| = 13 iterations

∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
S	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**

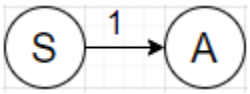
Cycle 1

- **S**

Starts from S, so S's value is 0.

0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
S	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**

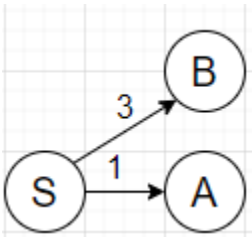
- **A**



Since 0 + 1 = 1 < ∞, which means A value is changed.

0	1	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
S	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**

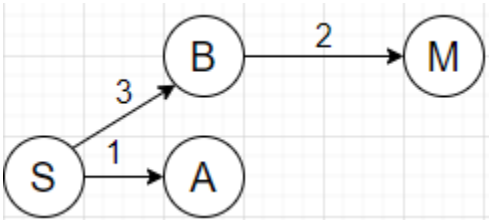
- **B**



Since 0 + 3 = 3 < ∞, B's value is changed.

0	1	3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
S	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**

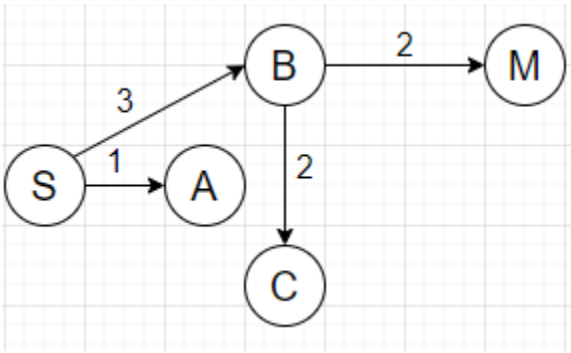
➢ **M**



Since 3 + 2 = 5, M's value is changed.

0	1	3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	5
S	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**

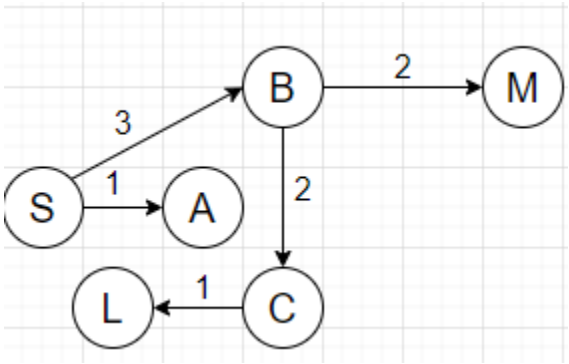
➢ **C**



Since 3 + 2 = 5 < ∞, C's value is changed.

0	1	3	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	5
S	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**

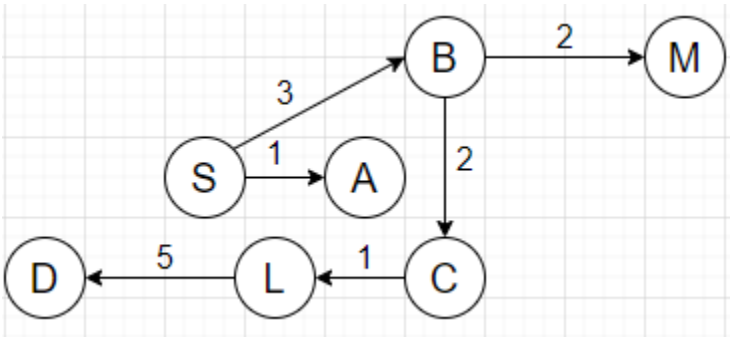
➢ **L**



Since 5 + 1 = 6 < ∞, L's value is changed.

0	1	3	5	∞	∞	∞	∞	∞	∞	∞	∞	6	5
S	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**

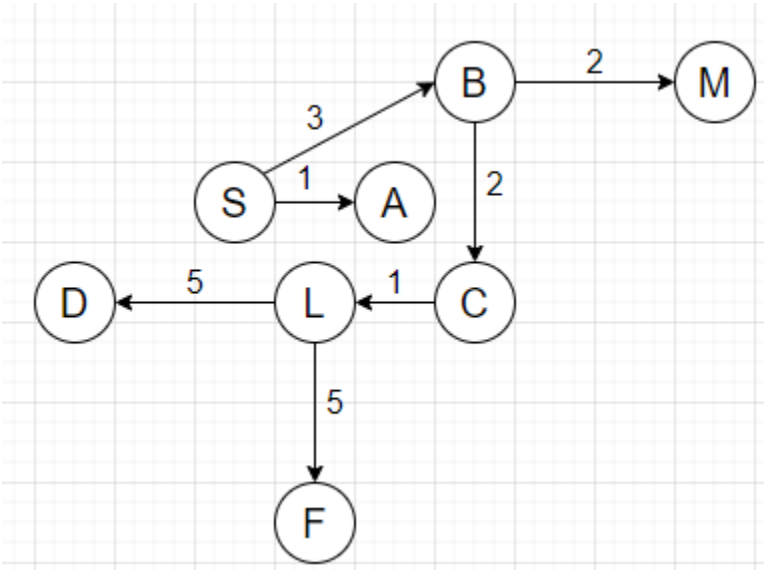
- **D**



Since 6 + 5 = 11 < ∞, D's value is changed.

0	1	3	5	11	∞	∞	∞	∞	∞	∞	∞	6	5
S	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**

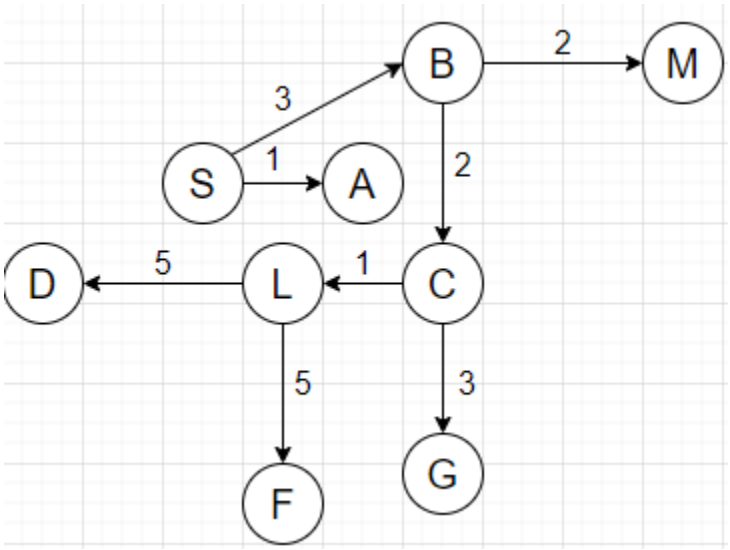
- **F**



Since 6 + 5 =11 < ∞, F's value is changed.

0	1	3	5	11	∞	11	∞	∞	∞	∞	∞	6	5
S	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**

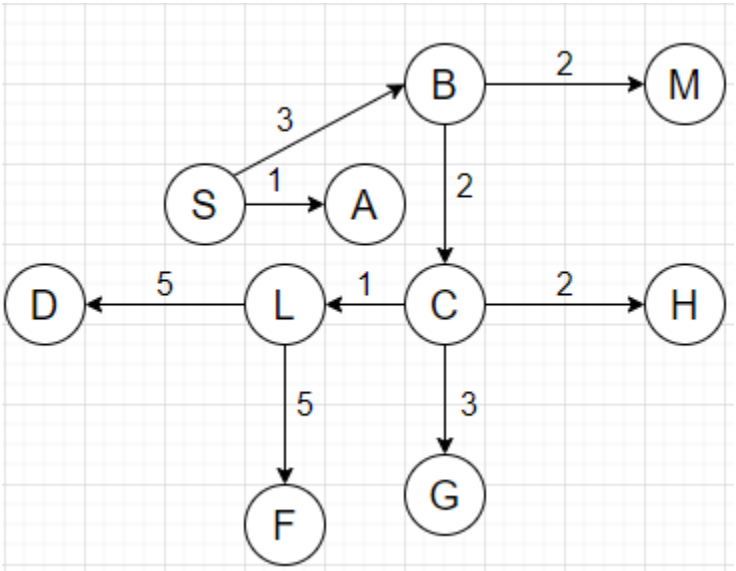
- **G**



Since 5 + 3 = 8 < ∞, G's value is changed.

0	1	3	5	11	∞	11	8	∞	∞	∞	∞	6	5
S	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**

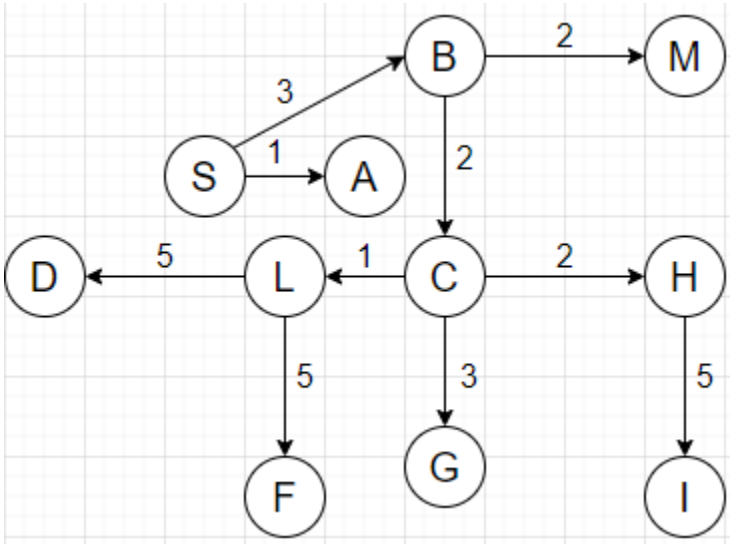
- **H**



Since 5 + 2 = 7 < ∞, H's value is changed.

0	1	3	5	11	∞	11	8	7	∞	∞	∞	6	5
S	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**

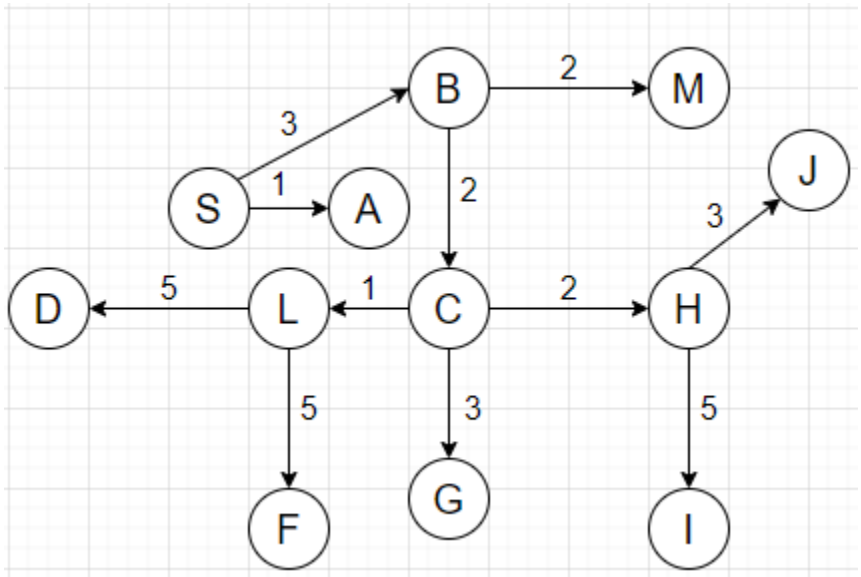
- **I**



Since 7 + 5 = 12 < ∞, I's value is changed.

0	1	3	5	11	∞	11	8	7	12	∞	∞	6	5
S	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**

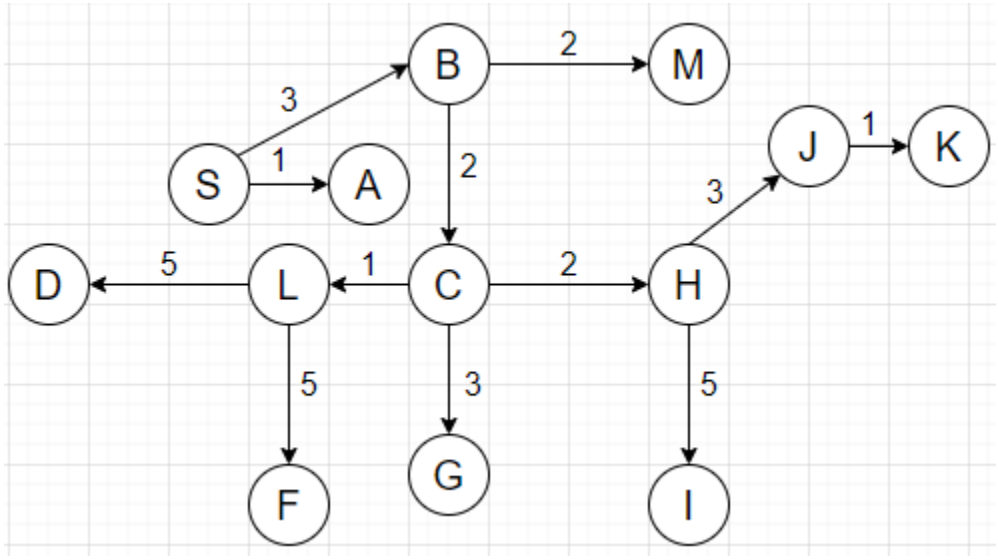
➢ **J**



Since 7 + 3 = 10 < ∞, J's value is changed.

0	1	3	5	11	∞	11	8	7	12	10	∞	6	5
S	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**

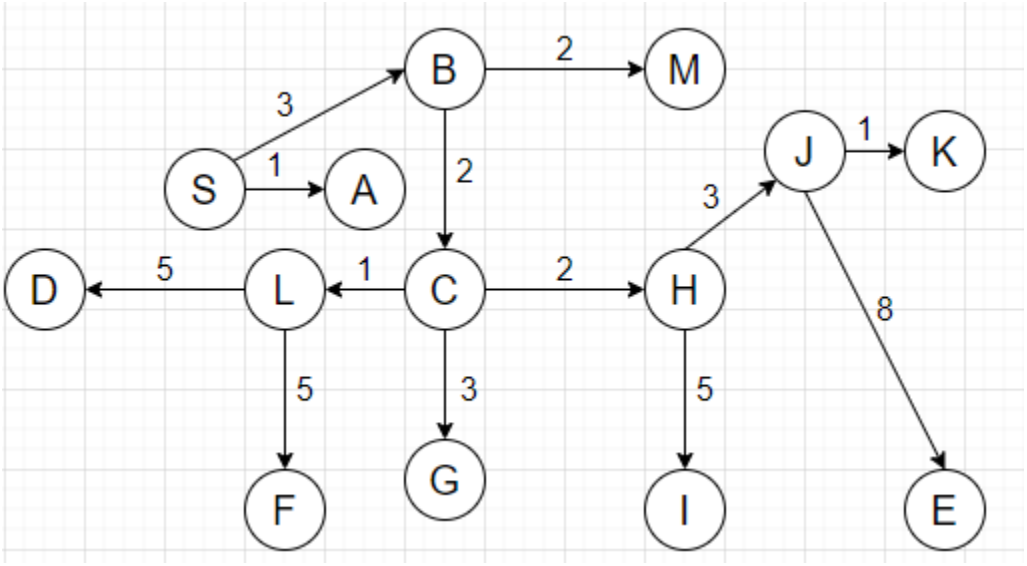
➢ **K**



Since 10 + 1 = 11 < ∞, K's value is changed.

0	1	3	5	11	∞	11	8	7	12	10	11	6	5
S	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**

- **E**



Since 10 + 8 = 18 < ∞, E's value is changed.

0	1	3	5	11	**18**	11	8	7	12	10	11	6	5
S	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**

The process ends at cycle one as there are no vertices to change and also |V -1| = |14 - 1| = 13 iteration reached.

Step 5: Find the minimum distance between two vertices.

Hence, the minimum distance between vertex S and vertex E is 18.

Step 6: Find the shortest path of maze

Based on the predecessor infor..mation, the path is **S -> B -> C -> H -> J -> E**

Step 7. Find the Big-O of shortest path of S and E vertices (Bellman Ford's Algorithm).

Runs in O(V * E), Where V is Total vertices and E is number of edges.

Time complexity = O(V * E)

= O(13 * 14)

= O(182)

= 182

Step 8. Compare performance of Dijkstra's Algorithm and Bellman Ford Algorithm.

Big-O comparison of Dijkstra's and Bellman Ford's Algorithm.

The Time complexity of Dijkstra's Algorithm is 15.

The Time complexity of Bellman Ford's Algorithm is 182.

Comparing how many steps are required to find a graph that has the shortest path?

The Dijkstra's algorithm takes seven steps.

The Bellman Ford's algorithm takes thirteen steps.

Conclusion:

- To find the shortest path Dijkstra is efficient than Bellman Ford's Algorithm.
- Dijkstra's Algorithm helps to find shortest path of every node.