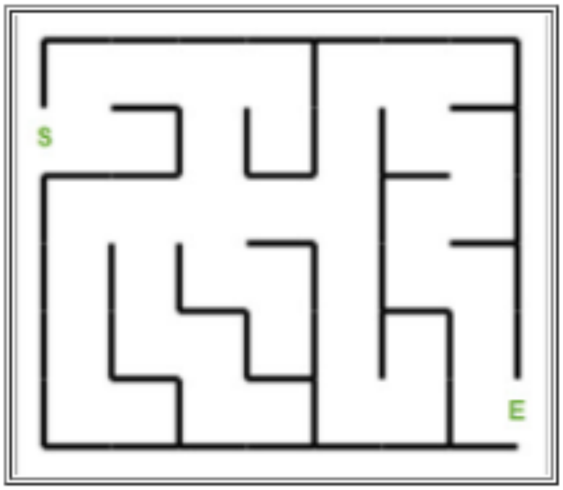
CS455 Week 11 Homework
Shoumya Singh
ID-19566

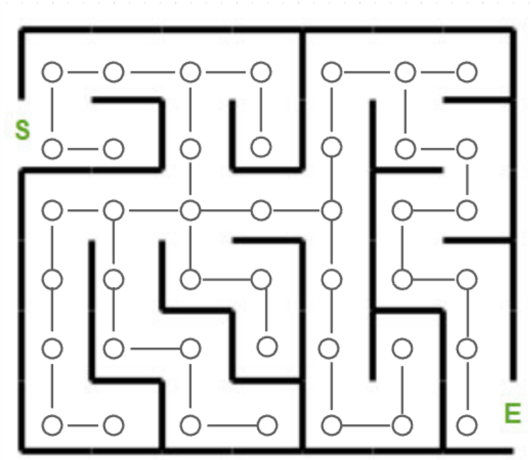
Q24 Use Bellman Ford's Algorithm to find the shortest path of a maze.

- Step 1: Similar to the previous question of finding the shortest path of the a maze. But instead of using Dijkstra's Algorithm, you will use Bellman Ford's Algorithm.
- Step 2: Comparing the performance of Dijkstra's Algorithm and Bellman Ford's Algorithm in solving this question by
  - Big-O comparison
  - comparing how many steps are required to find a graph that has the shortest path.
    - Note:
      - A step is defined as either comparing two numbers or replacing a number.
      - You can count how many steps for Dijkstra's Algorithm on the created table.
      - Refer this example on counting the steps for Bellman Ford's Algorithm.
- Step 3: Update your portofolio about the Maze project

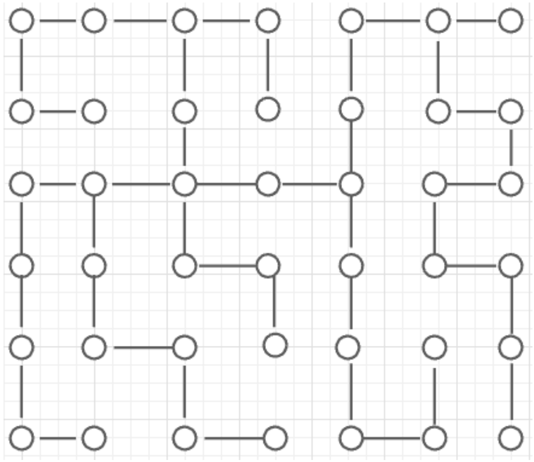
Solution:
Step 1



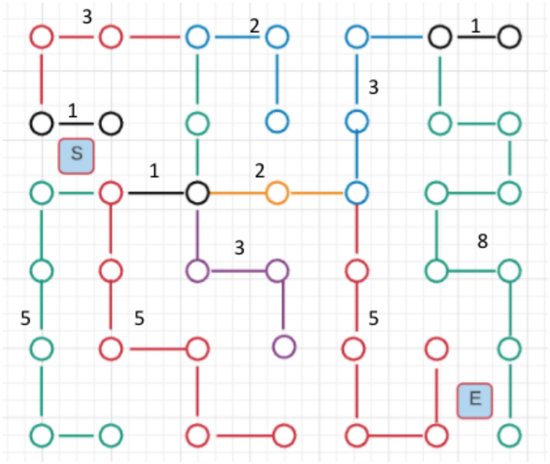
Step 2



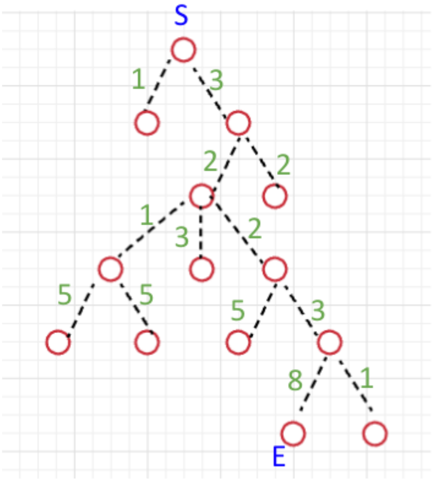
Step 3

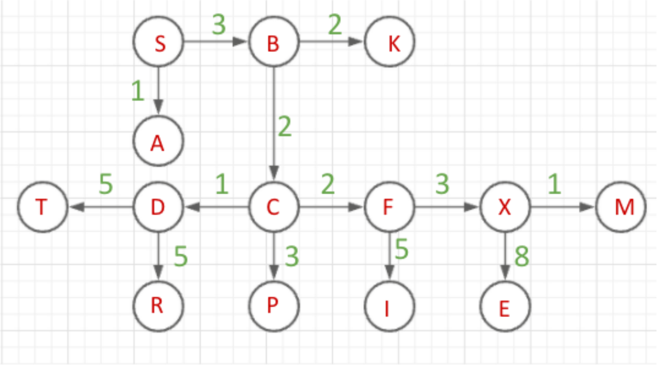


Step 4



Step 5





Step 1: Initialization

Next node to visit => B

S	A	B	C	K	P	D	F	R	T	I	X	M	E
0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

Step 2: 0+1 < ∞. Change the value of A to 1
0+3 < ∞. Change the value of B to 3

Next node to visit => B

S	A	B	C	K	P	D	F	R	T	I	X	M	E
0	1	3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

Step 3: 3+2=5 < ∞. Change the value of k to 5
3+2=5 < ∞. Change the value of C to 5

Next node to visit => C

S	A	B	C	K	P	D	F	R	T	I	X	M	E
0	1	3	5	5	∞	∞	∞	∞	∞	∞	∞	∞	∞

Step 4: 5+1=6 < ∞. Change the value of D to 6
5+2=7 < ∞. Change the value of F to 7
5+3=8 < ∞. Change the value of P to 8

Next node to visit => D

S	A	B	C	K	P	D	F	R	T	I	X	M	E
0	1	3	5	5	8	6	7	∞	∞	∞	∞	∞	∞

Step 5: 5+6=11 < ∞. Change the value of T to 11
5+6=11 < ∞. Change the value of R to 11

Next node to visit => F

S	A	B	C	K	P	D	F	R	T	I	X	M	E
0	1	3	5	5	8	6	7	11	11	∞	∞	∞	∞

Step 6: 7+5=12 < ∞. Change the value of I to 12
7+3=10 < ∞. Change the value of X to 10

Next node to visit => X

S	A	B	C	K	P	D	F	R	T	I	X	M	E
0	1	3	5	5	8	6	7	11	11	12	10	∞	∞

Step 7: 10+1=11 < ∞. Change the value of M to 11
10+8=18 < ∞. Change the value of E to 18

S	A	B	C	K	P	D	F	R	T	I	X	M	E
0	1	3	5	5	8	6	7	11	11	12	10	11	18

The process ends at cycle one as there are no vertices to change.

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

- **Bellman-Ford** algorithm is a single-source shortest path algorithm, so when we have negative edge weight then it can detect negative cycles in a graph.

- The only difference between the two is that **Bellman-Ford** is also capable of handling negative weights whereas **Dijkstra** Algorithm can only handle positives.

- The complexity of **Bellman-Ford** algorithm with respect to time is slower **than** the algorithm of **Dijkstra**.

- The first **for** loop is used for initialization, which runs in ***O(V)*** times. The next **for** loop runs |***V - 1***| passes over the edges, which takes ***O(E)*** times. Hence, Bellman-Ford algorithm runs in ***O(V, E)*** time.

- The complexity of this algorithm is fully dependent on the implementation of Extract-Min function. If extract min function is implemented using linear search, the complexity of this algorithm is $O(V^2 + E)$.