

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:

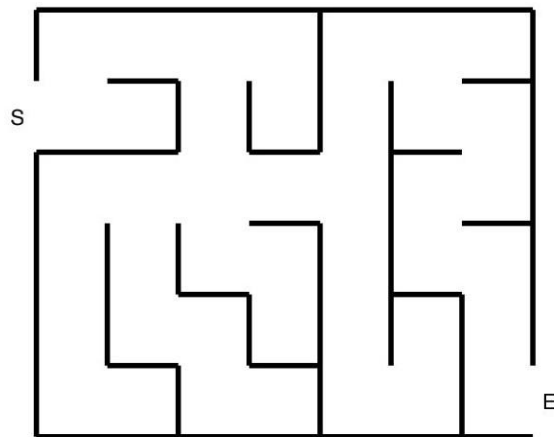
A step is defined as either comparing two numbers or replacing a number.

You can count how many steps for Dijkstra's Algorithm on the created table.

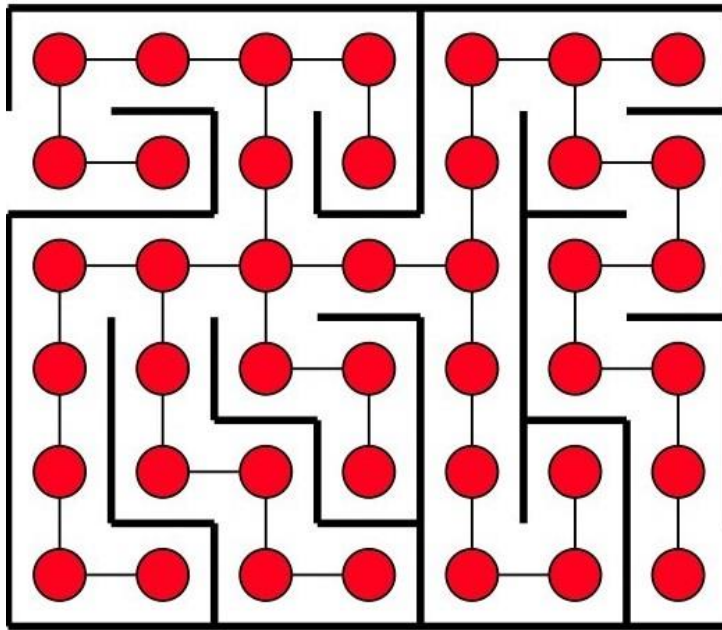
Refer this example on counting the steps for Bellman Ford's Algorithm.

Ans:

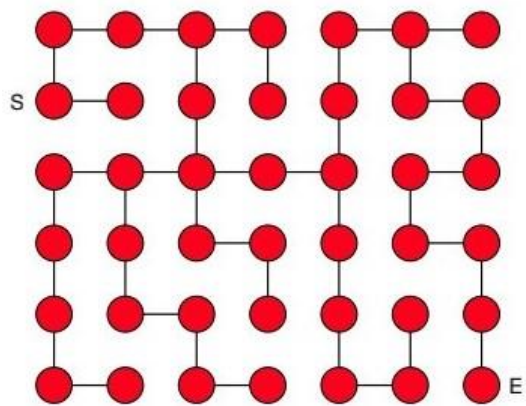
Step 1:



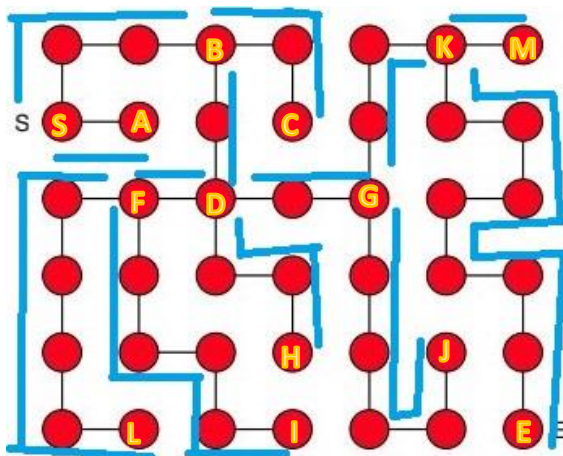
Step 2:



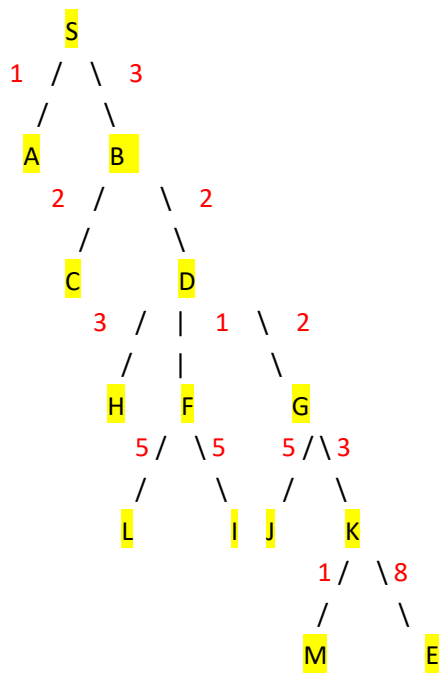
Step 3:



Step4:



Step 5:



Bellman Ford's Algorithm:

Cycle 1:

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

Cycle 2: do it one more time to check is there are any shorter path

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

So shortest path from S to E is 18

Comparing the performance of Dijkstra's Algorithm and Bellman Ford's Algorithm:

Bellman Ford's Algorithm time complexity is $O(V, E)$ and Dijkstra's Algorithm has $O(V^2 + E)$
Dijkstra's Algorithm visited vertex once, but Bellman Ford's Algorithm visited twice.