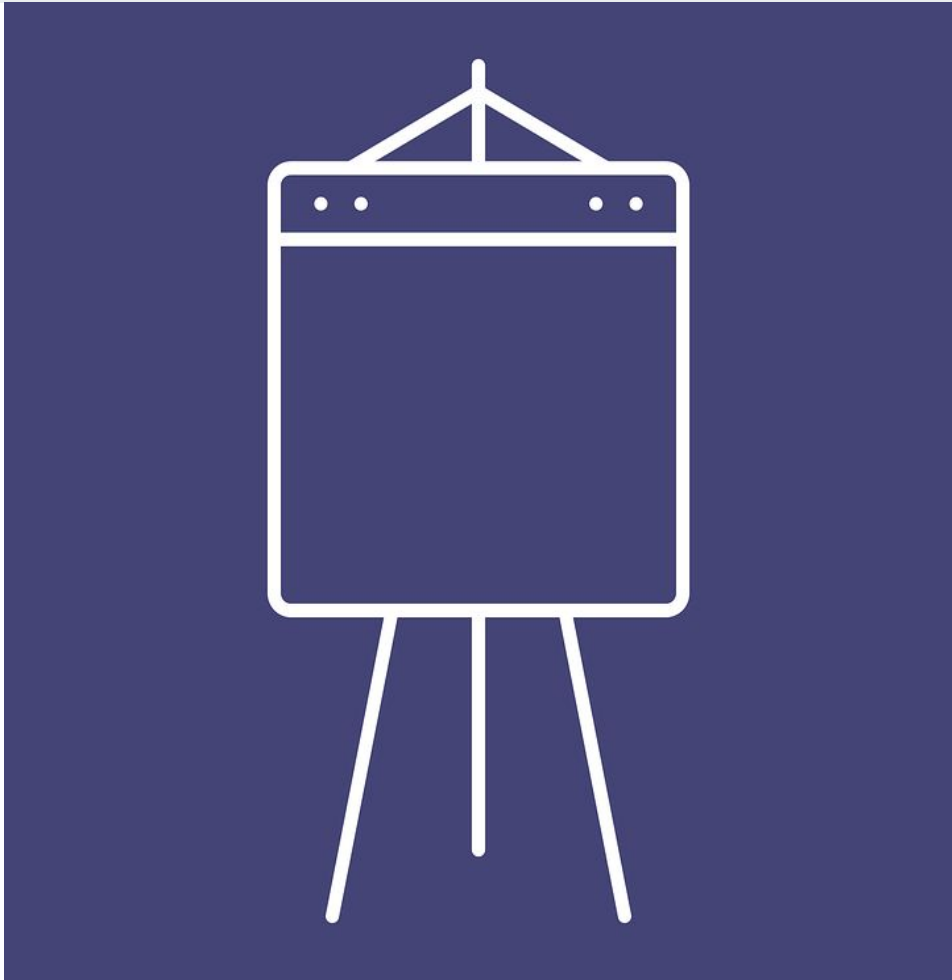
MAZE
---Shortest Path

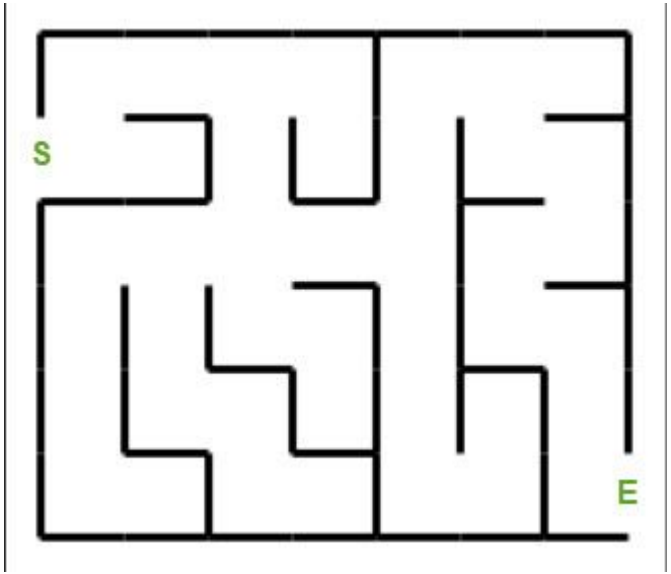
By Yixin Cao
19536 CS 455

Table of Content



Introduction

This project is to find the shortest path of the maze on the right using Dijkstra’s Algorithm.



Design

1. Understand the problem:

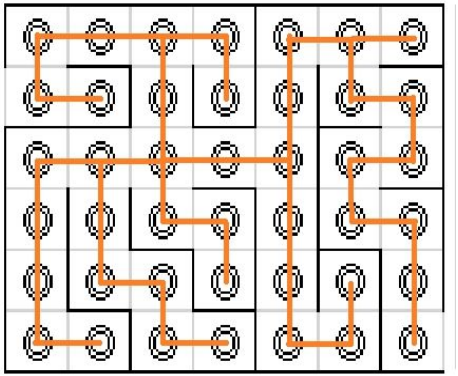
   This project is to find the shortest path from S to E in the maze.

2. Investigate to find possible solutions which may include your own solution:

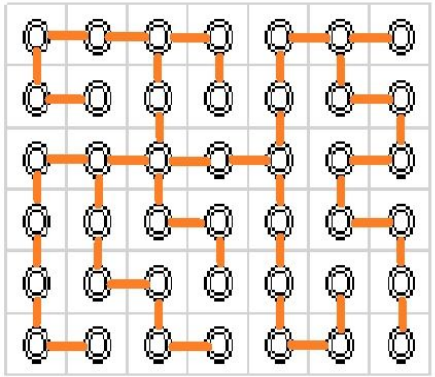
   Convert the maze to a tree of nodes and use Dijkstra’s Algorithm, Bellman Ford’s Algorithm and spinning tree to find the shortest path.

Implement

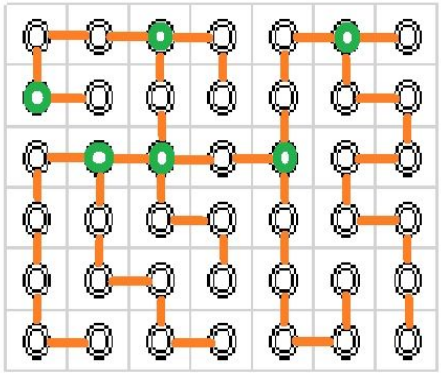
Step 1:



Step 2:

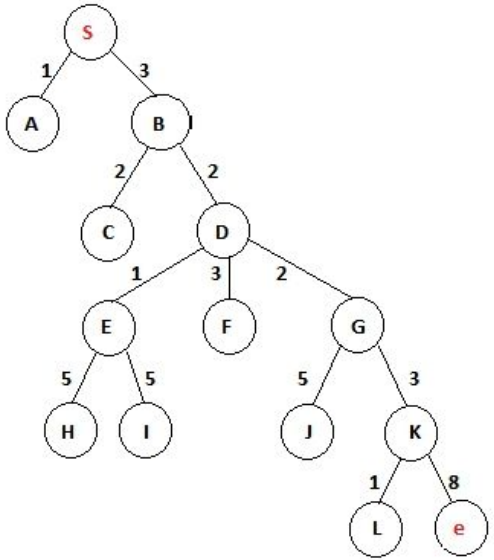


Step 3:



Implement (cont.)

Convert to tree



Shortest Path:

S -> B -> D -> G -> K -> E

Implement (cont.) ---Dijkstra's Algorithm

	Initial()	(S)	(S, A)	(S, A, B)	(S, A, B, C)	(S, A, B, C, D)	(S, A, B, C, D, E)	(S, A, B, C, D, E, H)	(S, A, B, C, D, E, H, I)	(S, A, B, C, D, E, H, I, G)	(S, A, B, C, D, E, H, I, G, F)	(S, A, B, C, D, E, H, I, G, F, K)	(S, A, B, C, D, E, H, I, G, F, K, L)	(S, A, B, C, D, E, H, I, G, F, K, L, J)	(S, A, B, C, D, E, H, I, G, F, K, L, J, e)
Next	S	A	B	C	D	E	H	I	G	F	K	L	J	e	
S	0	0	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~
A	∞	1	1	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~
B	∞	3	3	3	~~3~~	~~3~~	~~3~~	~~3~~	~~3~~	~~3~~	~~3~~	~~3~~	~~3~~	~~3~~	~~3~~
C	∞	∞	∞	5	5	~~5~~	~~5~~	~~5~~	~~5~~	~~5~~	~~5~~	~~5~~	~~5~~	~~5~~	~~5~~
D	∞	∞	∞	5	5	5	~~5~~	~~5~~	~~5~~	~~5~~	~~5~~	~~5~~	~~5~~	~~5~~	~~5~~
E	∞	∞	∞	∞	∞	6	6	~~6~~	~~6~~	~~6~~	~~6~~	~~6~~	~~6~~	~~6~~	~~6~~
F	∞	∞	∞	∞	∞	8	8	8	8	8	8	~~8~~	~~8~~	~~8~~	~~8~~
G	∞	∞	∞	∞	∞	7	7	7	7	7	~~7~~	~~7~~	~~7~~	~~7~~	~~7~~
H	∞	∞	∞	∞	∞	∞	11	11	~~11~~	~~11~~	~~11~~	~~11~~	~~11~~	~~11~~	~~11~~
I	∞	∞	∞	∞	∞	∞	11	11	11	~~11~~	~~11~~	~~11~~	~~11~~	~~11~~	~~11~~
J	∞	∞	∞	∞	∞	∞	∞	∞	∞	12	12	12	12	12	~~12~~
K	∞	∞	∞	∞	∞	∞	∞	∞	∞	10	10	10	~~10~~	~~10~~	~~10~~
L	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	11	11	~~11~~	~~11~~
e	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	18	18	18	18

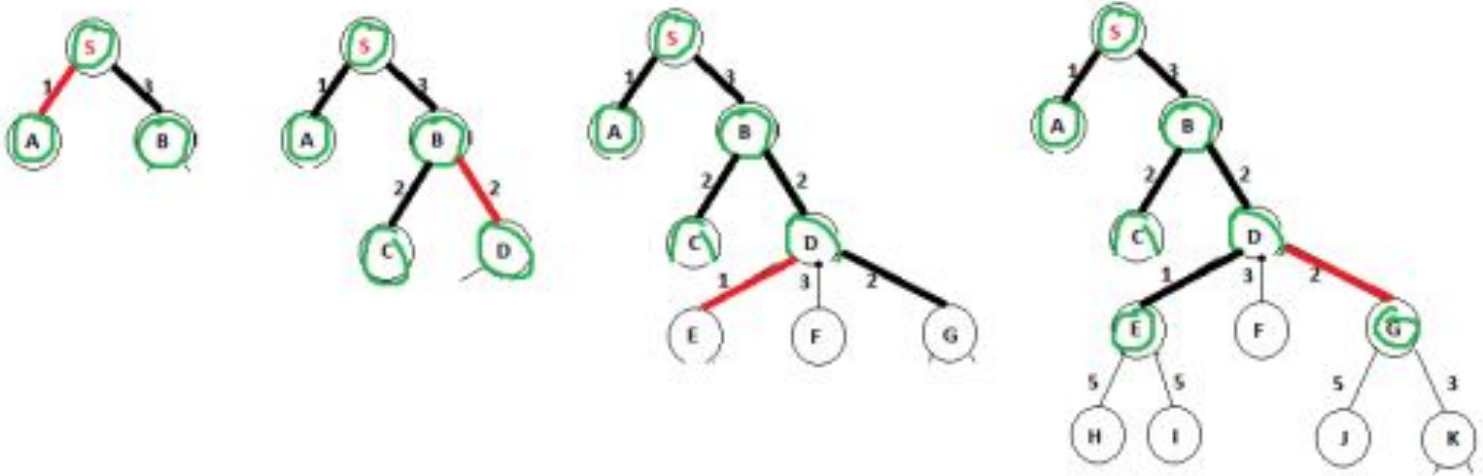
Implement (cont.) ---Bellman Ford's Algorithm

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

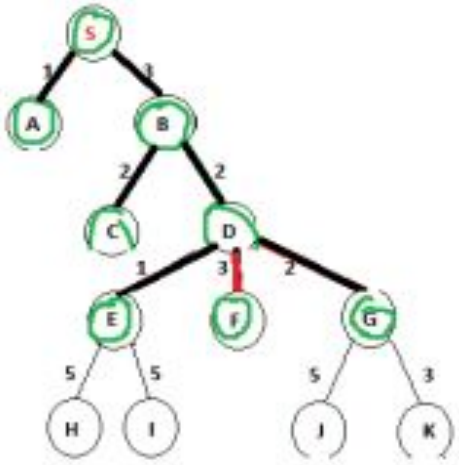
2nd iteration													
S	A	B	C	D	E	F	G	H	I	J	K	L	e
0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
0	1	3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
0	1	3	5	5	∞	∞	∞	∞	∞	∞	∞	∞	∞
0	1	3	5	5	6	8	7	∞	∞	∞	∞	∞	∞
0	1	3	5	5	6	8	7	∞	∞	12	10	∞	∞
0	1	3	5	5	6	8	7	∞	∞	12	10	11	18
0	1	3	5	5	6	8	7	11	11	12	10	11	18

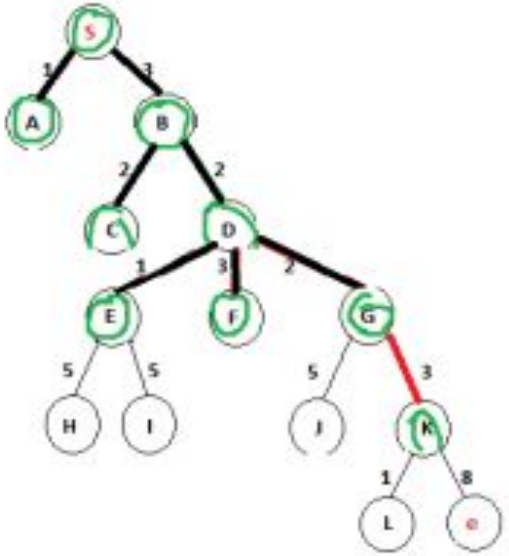
Shortest path is 18.

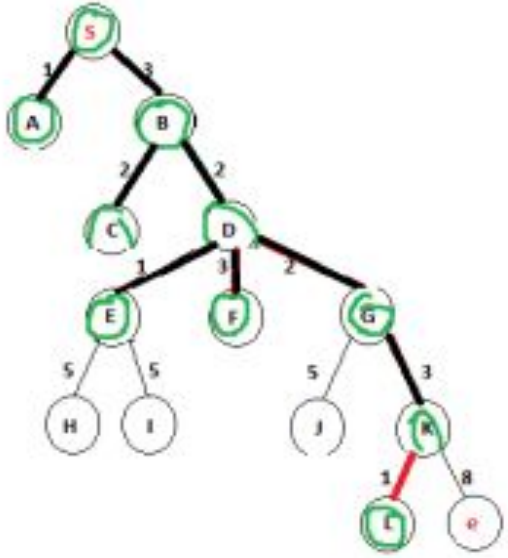
Implement (cont.) ---Prim's Minimum Spanning Tree



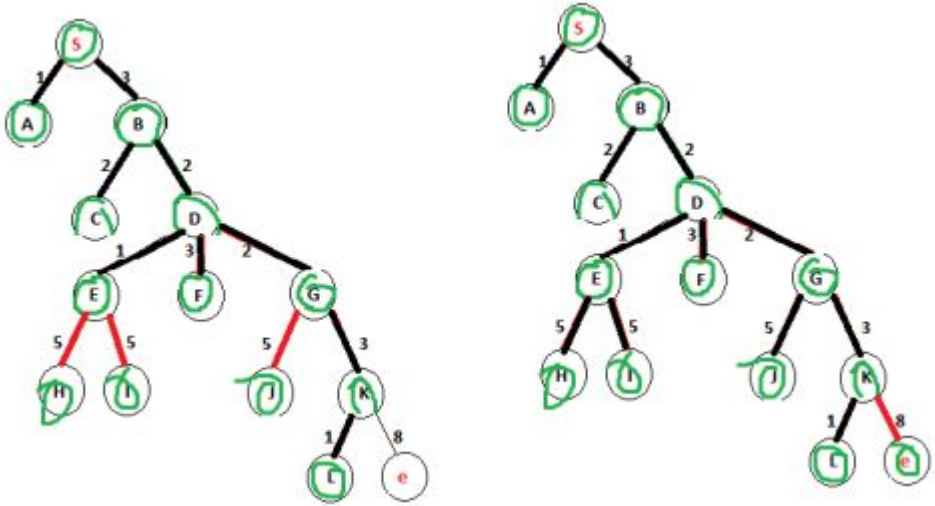
Implement (cont.) ---Prim's Minimum Spanning Tree







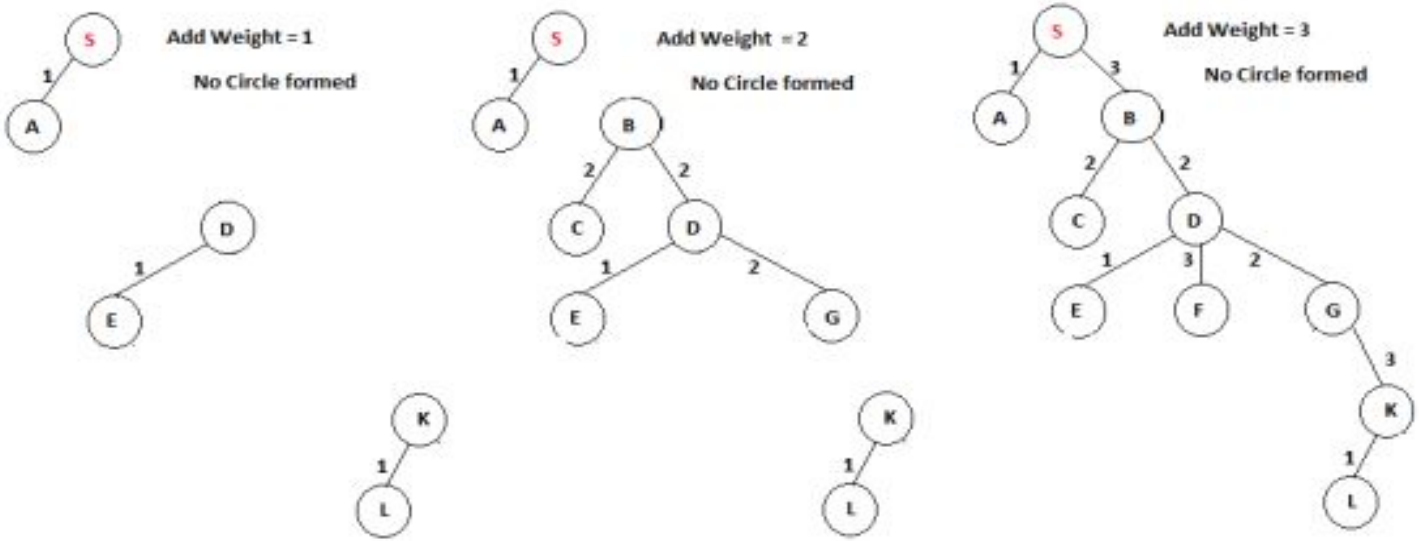
Implement (cont.) ---Prim’s Minimum Spanning Tree



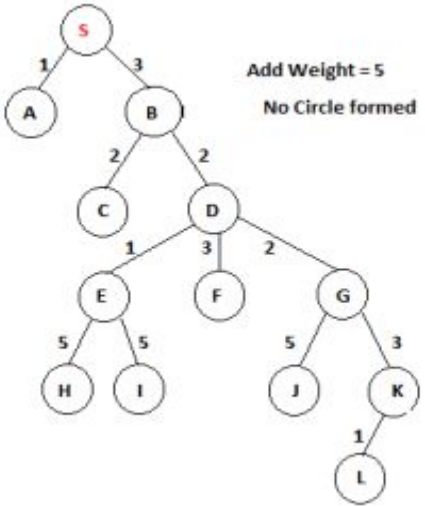
Implement (cont.) ---Kruskal's Minimum Spanning Tree

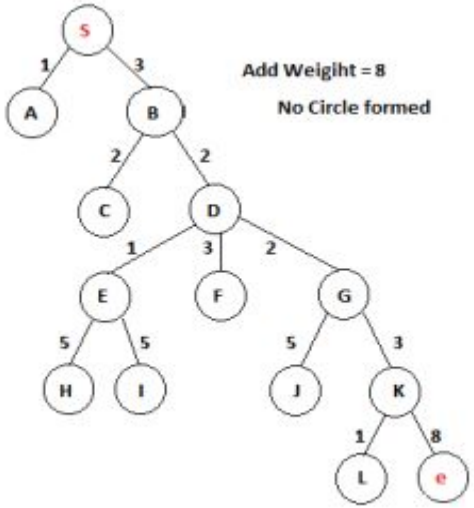
Weight	Src	Dest
1	S	A
1	D	E
1	K	L
2	B	C
2	B	D
2	D	G
3	S	B
3	D	F
3	G	K
5	E	H
5	E	I
5	G	J
8	K	e

Implement (cont.) ---Kruskal's Minimum Spanning Tree

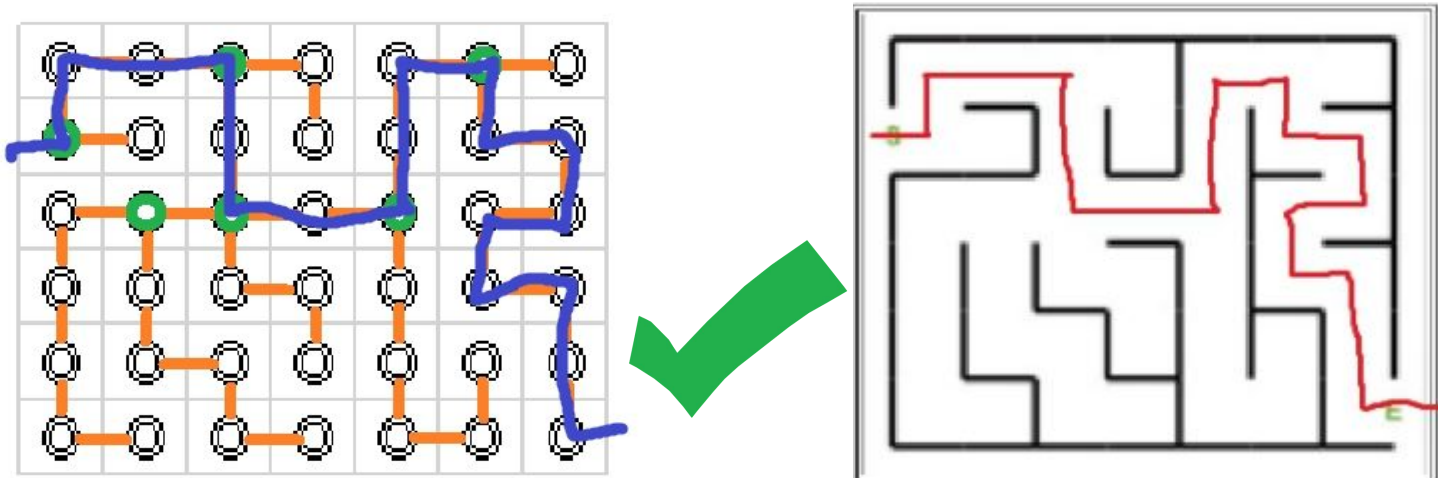


Implement (cont.) ---Kruskal's Minimum Spanning Tree





Test



Enhancement Ideas --- Compare algorithms

	Time Complexity	Steps
Dijkstra's Algorithm	O(V^2+E)	15 Steps
Bellman Ford's Algorithm	O(VE)	14 Steps * 2 iterations
Prim's Minimum Spanning Tree	O((v + E)logV)	NA
Kruskal's Minimum Spanning Tree	O(E * logV)	NA

Enhancement Ideas

For very complicated maze problem, we can also convert the maze into x by y squares, and convert it to tree with nodes. Using either Dijkstra's Algorithm or Bellman Ford's Algorithm to find out the shortest path and tested with the other. Spinning tree is another way to find the shortest path.

We can also think the other way to create different mazes.

This shortest path project can be imply on real world such as finding shortest route on map etc..

Conclusion

The shortest path is successfully found out with Dijkstra’s Algorithm, Bellman Ford’s Algorithm and spinning tree. By comparison of Big O, Kruskal’s Minimum Spanning Tree is the fastest. All methods are tested with no issue.

References

https://npu85.npu.edu/~henry/npu/classes/algorithm/tutorialpoints_daa/slide/shortest_paths.html

https://npu85.npu.edu/~henry/npu/classes/algorithm/graph_alg/slide/exercise_graph_alg.html

https://npu85.npu.edu/~henry/npu/classes/capstone/job/slide/portofolio.html#step_2

https://towardsdatascience.com/solving-mazes-with-python-f7a412f2493f

https://daemianmack.org/posts/2019/12/mazes-for-programmers-dijkstras-algorithm.html