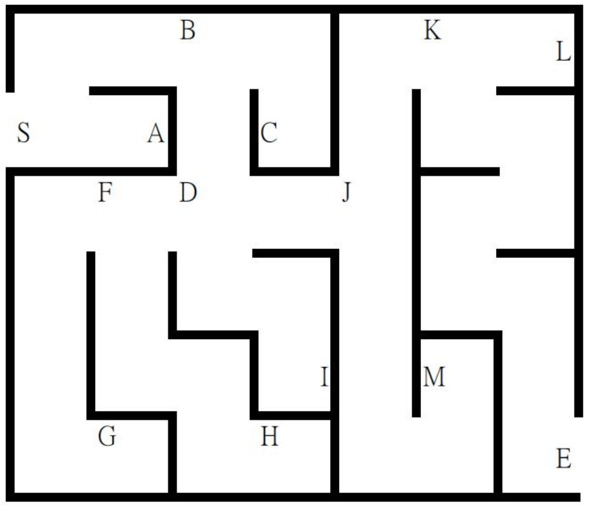
GET OUT OF A MAZE

Use Dijkstra's Algorithm to find the shortest path

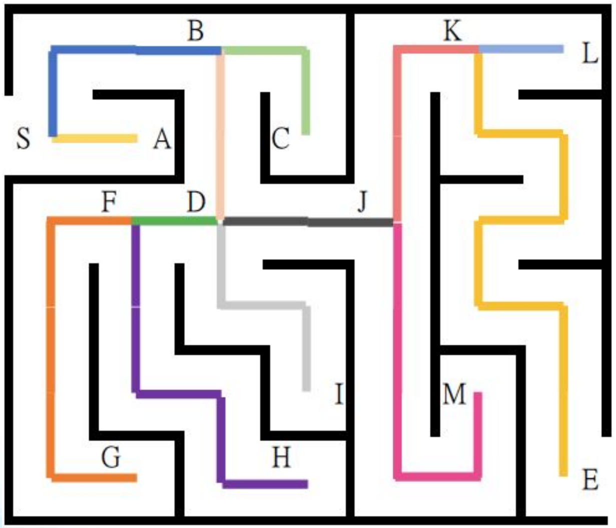
What's a maze

- A maze must includes at least one entry and one exit(goal).
- A maze can be find a path or paths from entries to exits.
- Look at the diagram in the right side, it is a maze with Entry-S and Exit-E.



How many path in the maze?

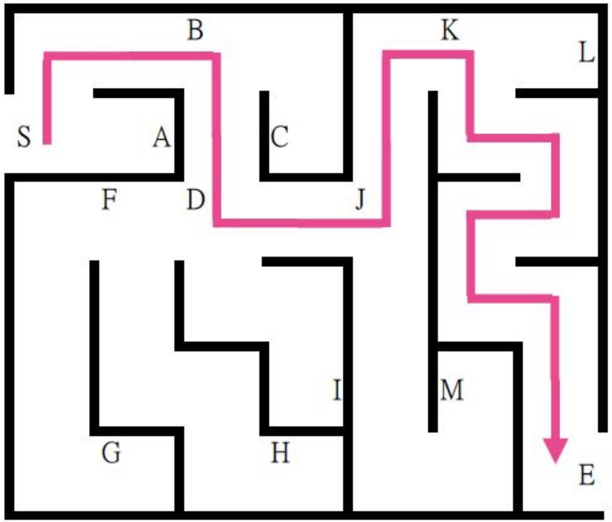
Take the maze the maze below for example:



This maze creates 13 paths which are SA, SB, BC, BD, DF, DI, DJ, FG, FH, JK, JM, KL, KE.

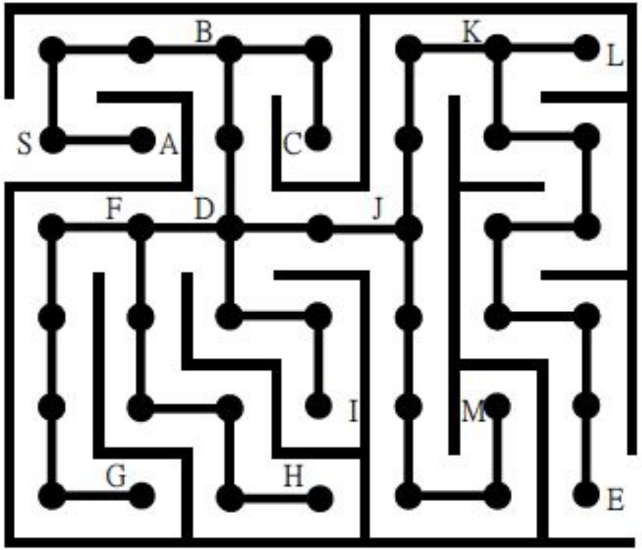
How to go from S to E?

- Take the paths SB, BD, DJ, JK, KE is the way from S to E.
- This simple example is easy for humans but if it's a complicated maze we will need computer to help.

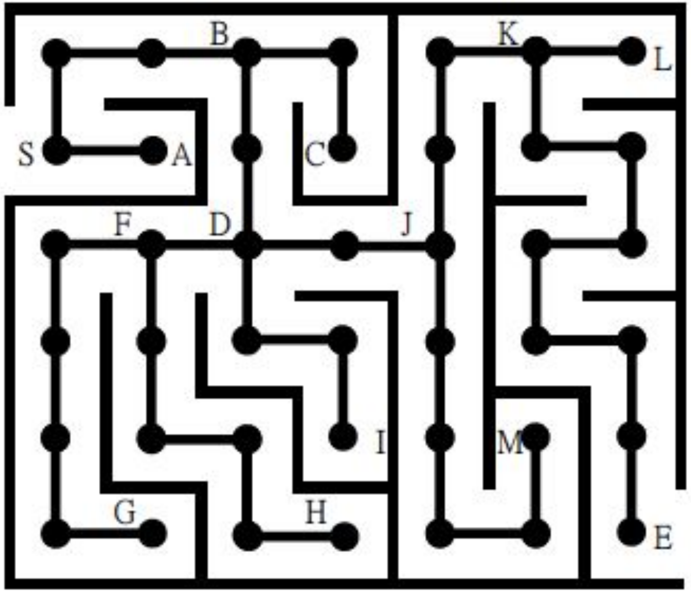


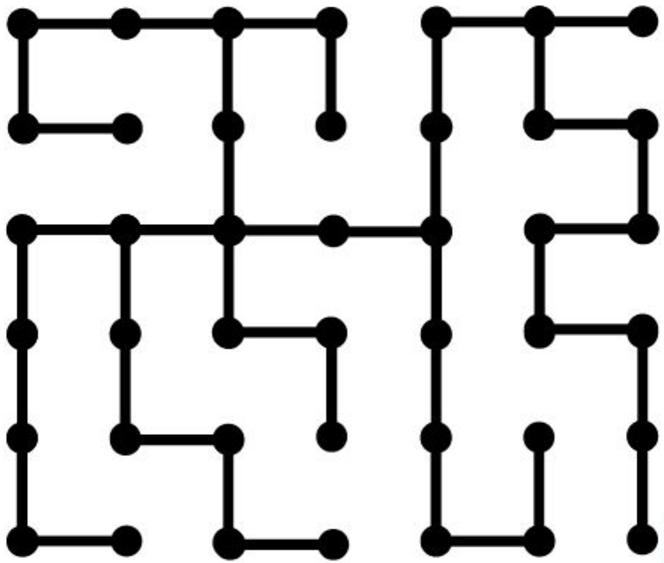
Find the node

❖ First step is to find out the node of this maze, there are 42 nodes in this maze.

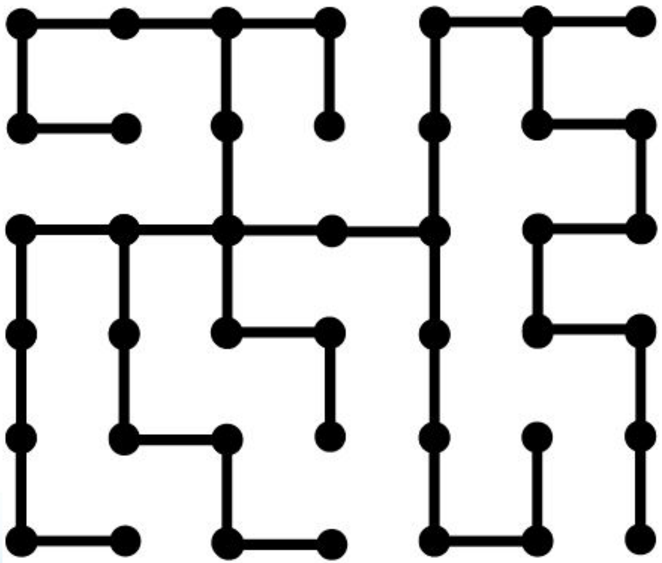


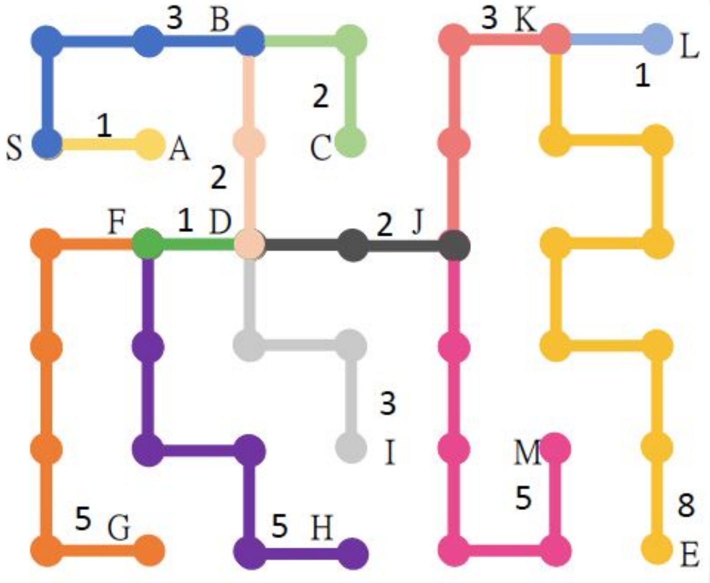
Simple the maze into graphs



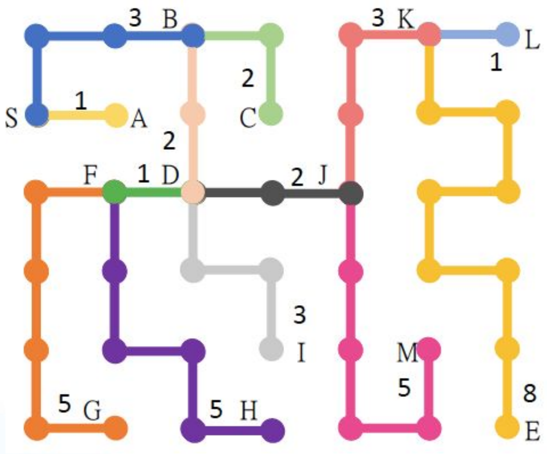


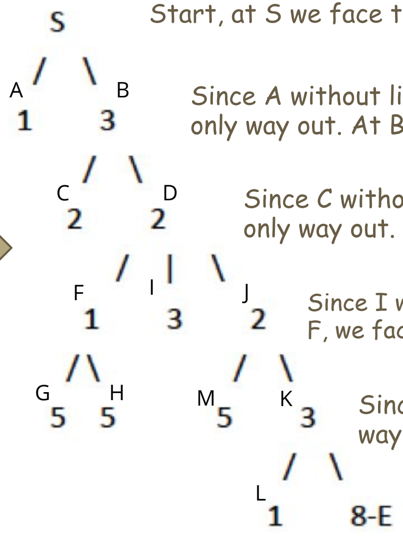
Find out the distance





Create the tree





Start, at S we face two paths connet to A and B.

Since A without linking to other path except A, B is the only way out. At B, we face two paths connect to C and D.

Since C without linking to other path except B, D is the only way out. At D, we face 3 paths connect to F, I and J.

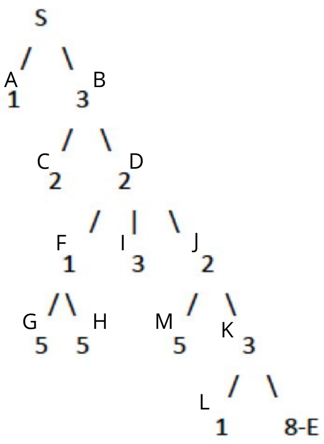
Since I without linking to other path. F and J could be the way out. At F, we face 2 paths to G and H. At J, we face 2 paths to K and M.

Since G, H, M without linking to other path, K is the only way out. At K, we face 2 paths connect to L and E.

E is our distanation.

Dijkstra’s Algorithm-1

Dijkstra’s algorithm solves the single-source shortest-paths problem on a directed weighted graph G = (V, E), where all the edges are non-negative (i.e., w(u, v) ≥ 0 for each edge (u, v) Є E).



	Initial	Step1	Step2	Step3	Step4	Step5	Step6	Step7	Step8	Step9	Step10	Step11	Step12	Step13
		S	A	B	C	D	F	J	I	K	G	H	L	M
	Next	Next	Next	Next	Next	Next	Next	Next	Next	Next	Next	Next	Next	Next
	S	A	B	C	D	F	J	I	K	G	H	L	M	E
S	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
B	∞	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
D	∞	∞	∞	5	5	5	5	5	5	5	5	5	5	5
F	∞	∞	∞	∞	∞	6	6	6	6	6	6	6	6	6
G	∞	∞	∞	∞	∞	∞	11	11	11	11	11	11	11	11
H	∞	∞	∞	∞	∞	∞	11	11	11	11	11	11	11	11
I	∞	∞	∞	∞	∞	8	8	8	8	8	8	8	8	8
J	∞	∞	∞	∞	∞	7	7	7	7	7	7	7	7	7
K	∞	∞	∞	∞	∞	∞	∞	10	10	10	10	10	10	10
L	∞	∞	∞	∞	∞	∞	∞	∞	∞	11	11	11	11	11
M	∞	∞	∞	∞	∞	∞	∞	12	12	12	12	12	12	12
E	∞	∞	∞	∞	∞	∞	∞	∞	∞	18	18	18	18	18

Dijkstra’s Algorithm-2

	Initial	Step1	Step2	Step3	Step4	Step5	Step6	Step7	Step8	Step9	Step10	Step11	Step12	Step13
		S	A	B	C	D	F	J	I	K	G	H	L	M
	Next	Next	Next	Next	Next	Next	Next	Next	Next	Next	Next	Next	Next	Next
	S	A	B	C	D	F	J	I	K	G	H	L	M	E
S	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
B	∞	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
D	∞	∞	∞	5	5	5	5	5	5	5	5	5	5	5
F	∞	∞	∞	∞	∞	6	6	6	6	6	6	6	6	6
G	∞	∞	∞	∞	∞	∞	11	11	11	11	11	11	11	11
H	∞	∞	∞	∞	∞	∞	11	11	11	11	11	11	11	11
I	∞	∞	∞	∞	∞	8	8	8	8	8	8	8	8	8
J	∞	∞	∞	∞	∞	7	7	7	7	7	7	7	7	7
K	∞	∞	∞	∞	∞	∞	∞	10	10	10	10	10	10	10
L	∞	∞	∞	∞	∞	∞	∞	∞	∞	11	11	11	11	11
M	∞	∞	∞	∞	∞	∞	∞	12	12	12	12	12	12	12
E	∞	∞	∞	∞	∞	∞	∞	∞	∞	18	18	18	18	18

Rule 1. The total distance is the distance from start to current stop add distance from current stop to target stop, and it needs to be compared to the distance it already had which one is smaller.

Rule 2. Choose the smallest distance spot as the next stop.

Rule 3. Next stop can’t be the one which have been choosed.

1. Start from S, therefore, distance from S to S 0.
2. There are 2 paths connect to S, pass one node to A or pass three nodes to B, choose the nearest one, then A is the next stop.
3. From B, there are 2 paths to go, 2 nodes to C and 2 nodes to D,from S to B is 3 nodes, therefore from S to C or D is 3+2=5 nodes.

Shortest path to the exit

	Initial	Step1	Step2	Step3	Step4	Step5	Step6	Step7	Step8	Step9	Step10	Step11	Step12	Step13
		S	A	B	C	D	F	J	I	K	G	H	L	M
	Next	Next	Next	Next	Next	Next	Next	Next	Next	Next	Next	Next	Next	Next
	S	A	B	C	D	F	J	I	K	G	H	L	M	E
S	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
B	∞	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
D	∞	∞	∞	5	5	5	5	5	5	5	5	5	5	5
F	∞	∞	∞	∞	∞	6	6	6	6	6	6	6	6	6
G	∞	∞	∞	∞	∞	∞	11	11	11	11	11	11	11	11
H	∞	∞	∞	∞	∞	∞	11	11	11	11	11	11	11	11
I	∞	∞	∞	∞	∞	8	8	8	8	8	8	8	8	8
J	∞	∞	∞	∞	∞	7	7	7	7	7	7	7	7	7
K	∞	∞	∞	∞	∞	∞	∞	10	10	10	10	10	10	10
L	∞	∞	∞	∞	∞	∞	∞	∞	∞	11	11	11	11	11
M	∞	∞	∞	∞	∞	∞	∞	12	12	12	12	12	12	12
E	∞	∞	∞	∞	∞	∞	∞	∞	∞	18	18	18	18	18

- Calculate by Dijkstra's Algorithm, the smallest path from S to E needs to pass 18 nodes.

Thank you!