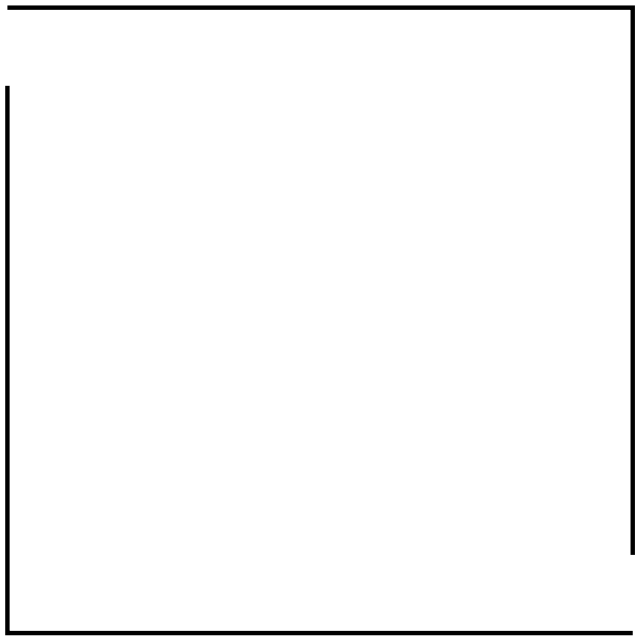
Maze-Generating Algorithms

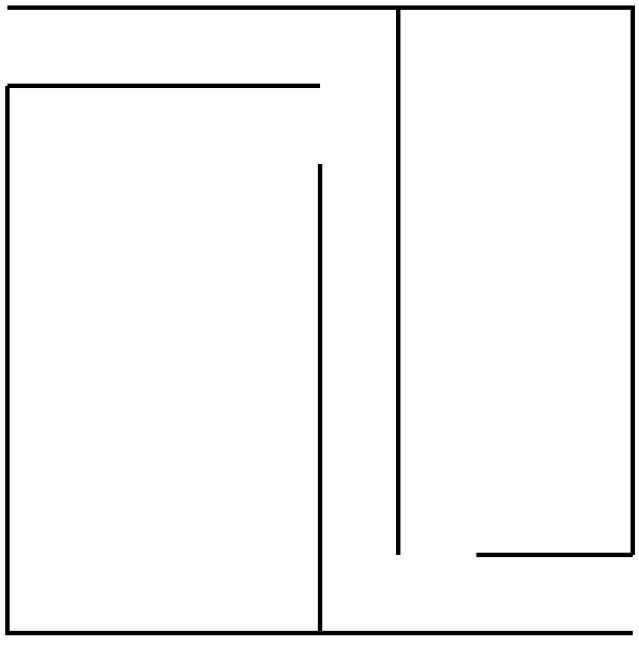
20080610 Lee, Ki Sang

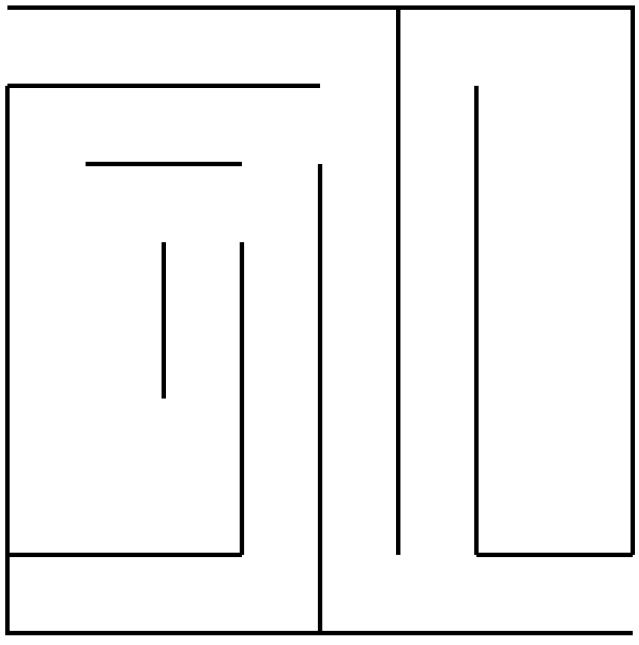
KAIST

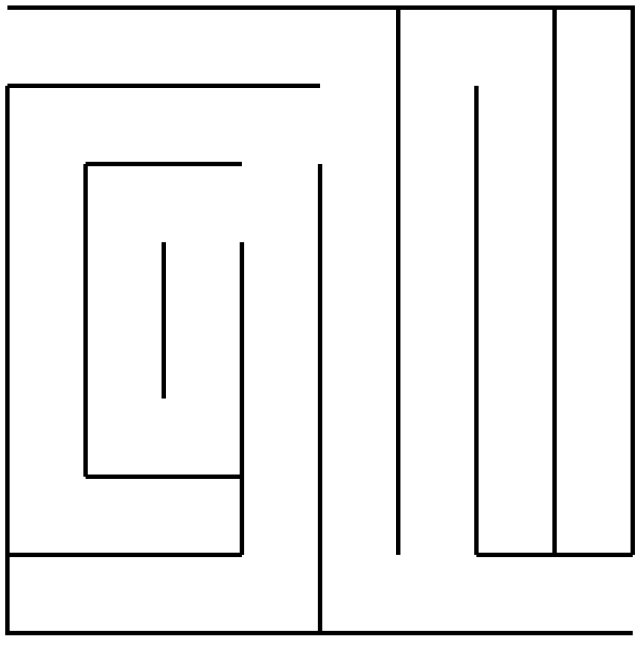
Physical Mathematics Conference 2009

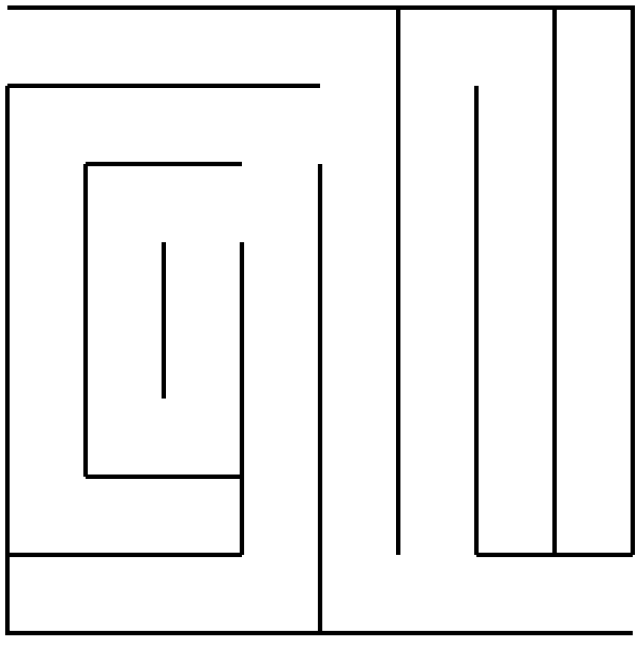
1. Maze Basics
2. General Idea of Maze Generating
3. Perfect Maze Generating
   - Depth-First Search
   - Kruskal's Algorithm



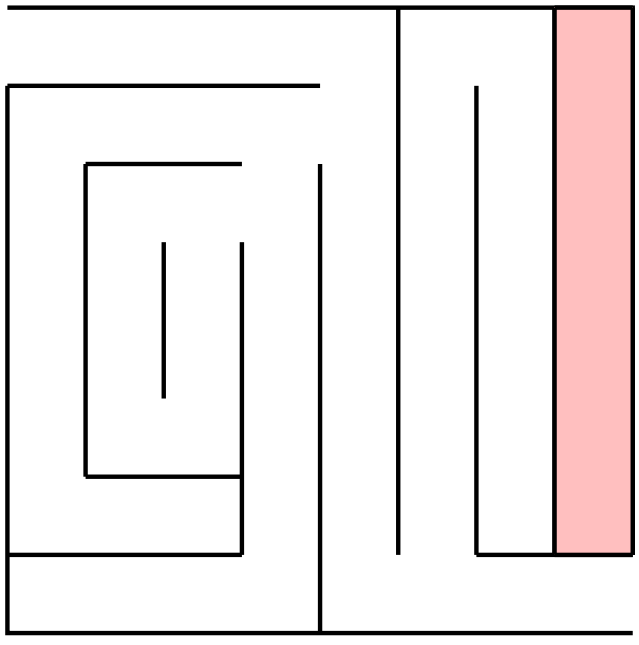




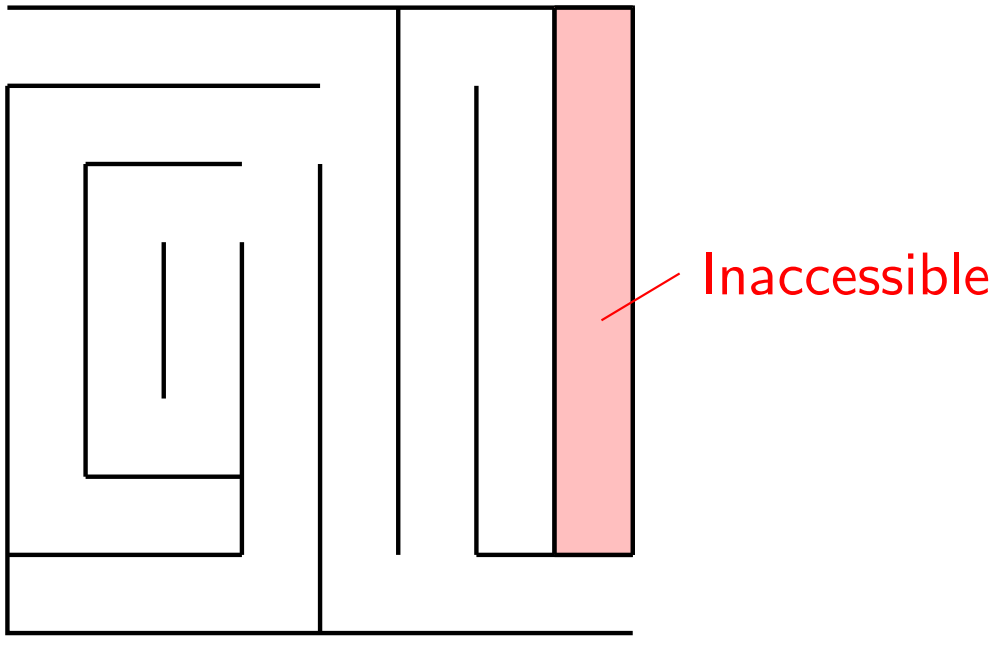




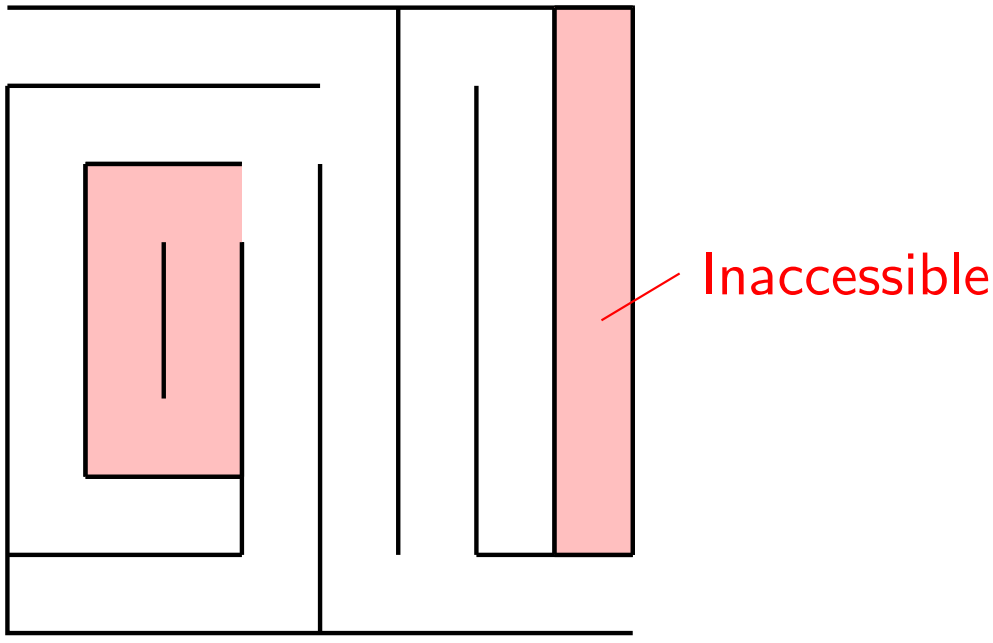
A Maze



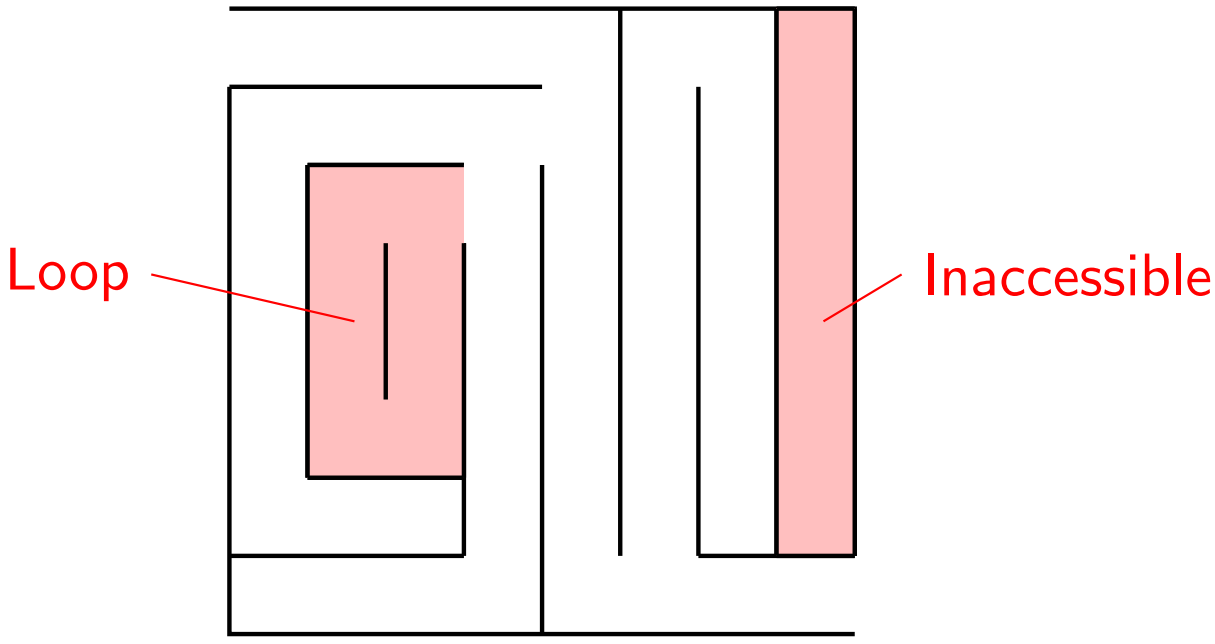
A Maze



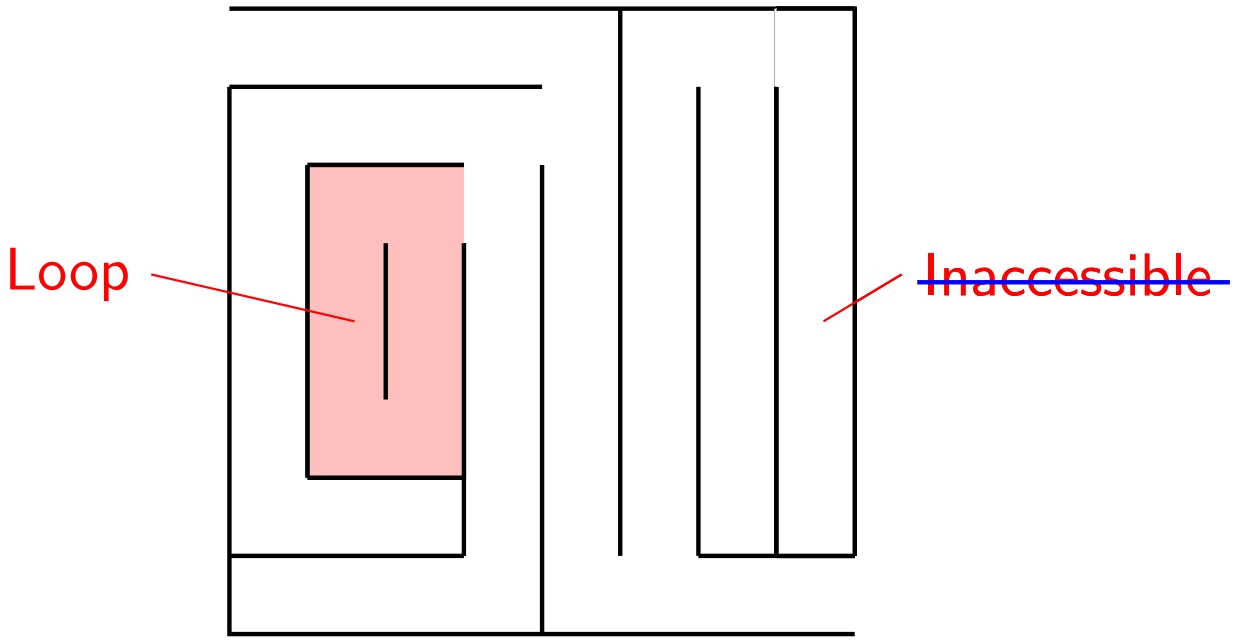
A Maze



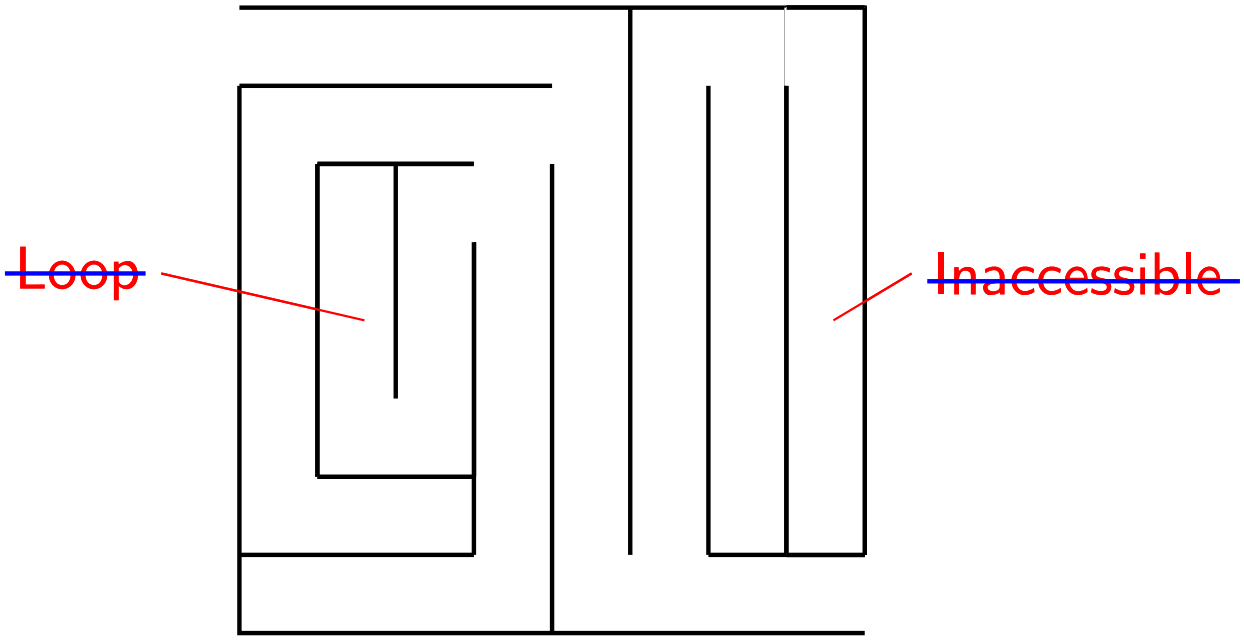
A Maze



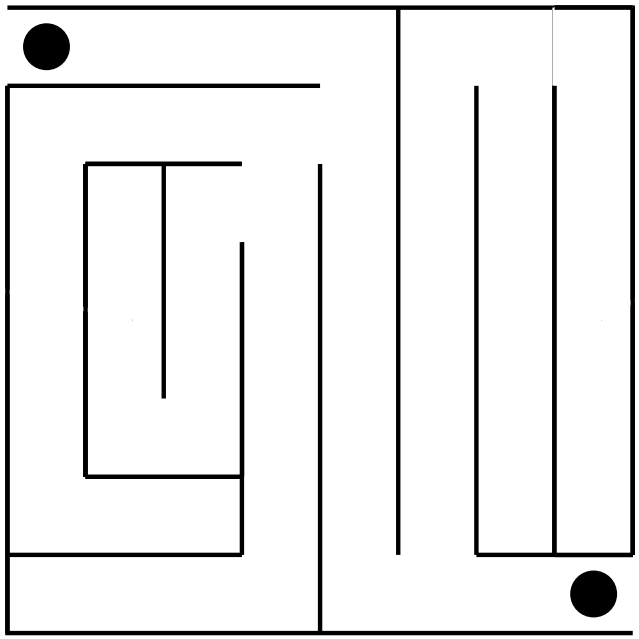
A Maze



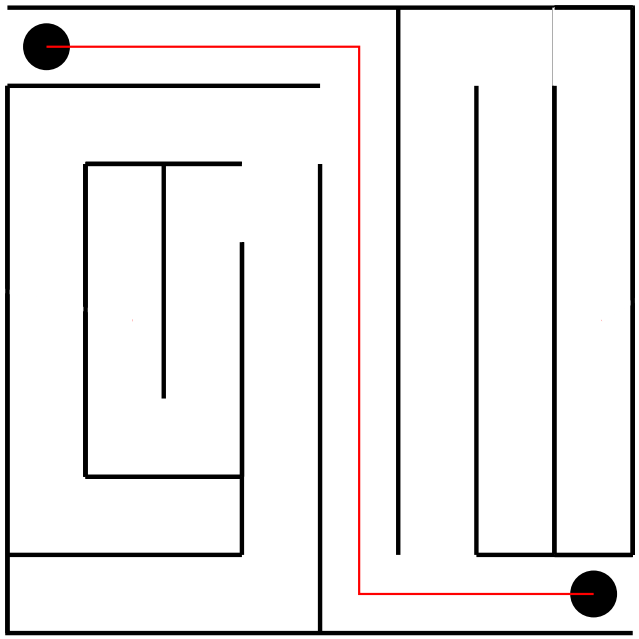
A Maze



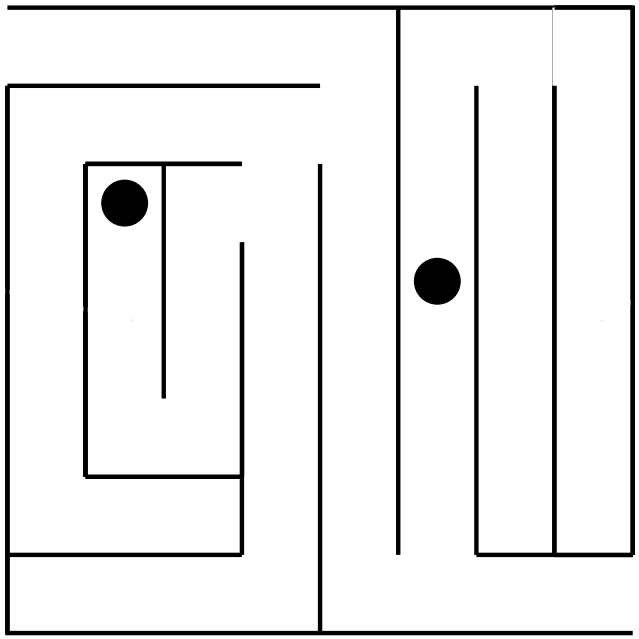
A Perfect Maze



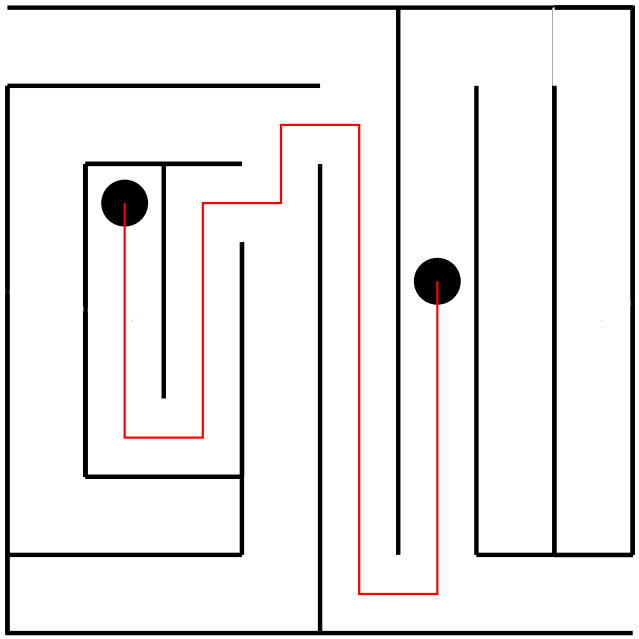
A Perfect Maze



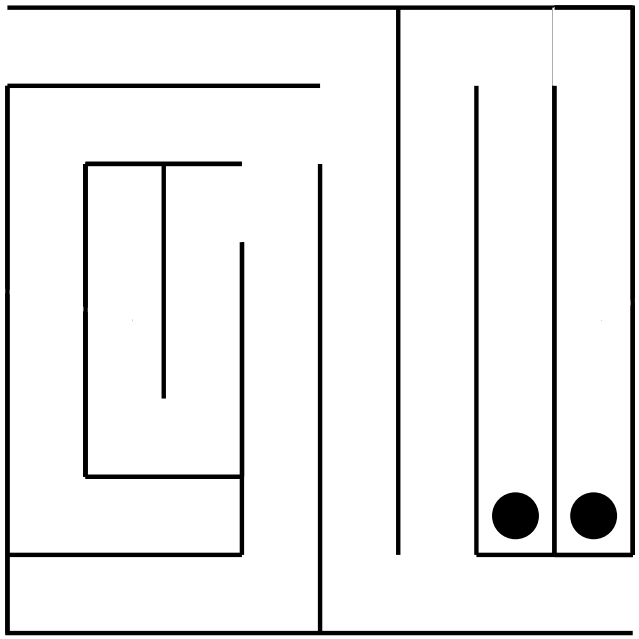
A Perfect Maze



A Perfect Maze

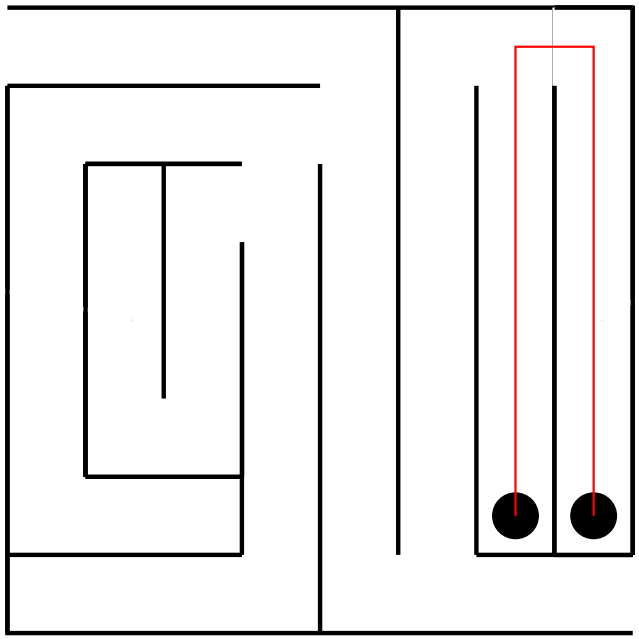


A Perfect Maze



A Perfect Maze

Between any two points,

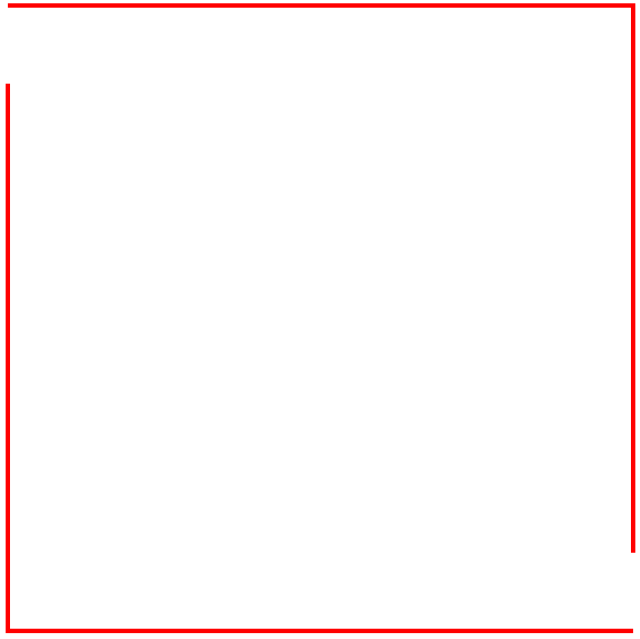


A Perfect Maze

Between any two points,
Only one path exists!

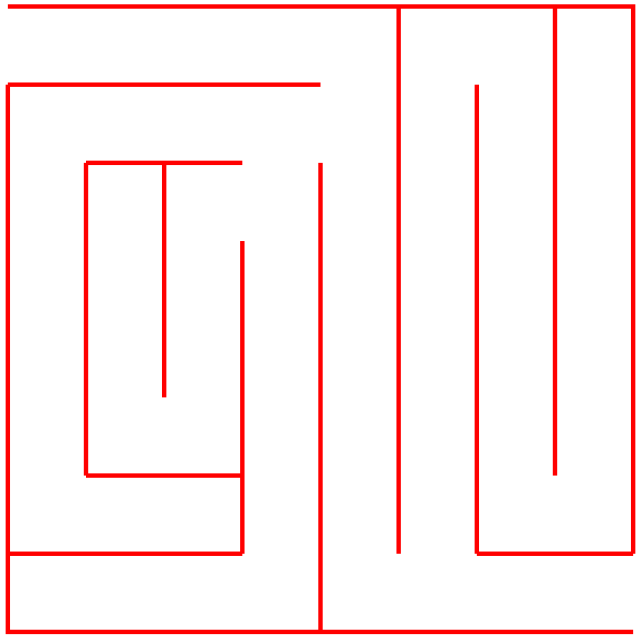
Assume mazes are

Assume mazes are



In 2-Dimensional Rectangular Grid

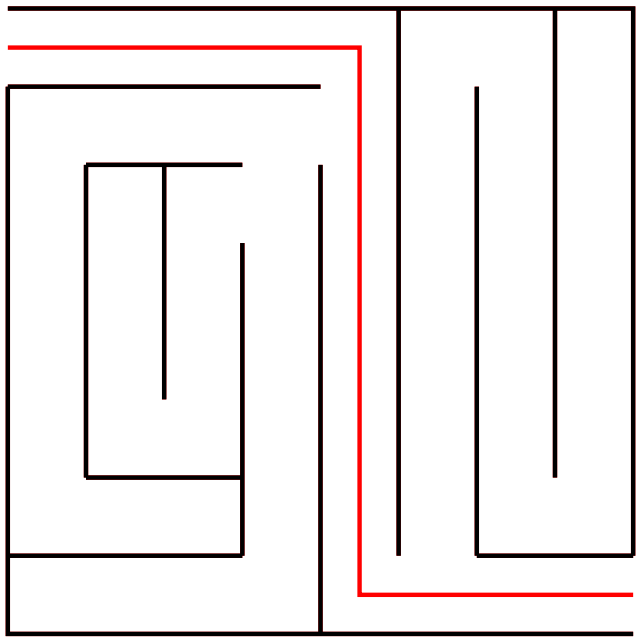
Assume mazes are



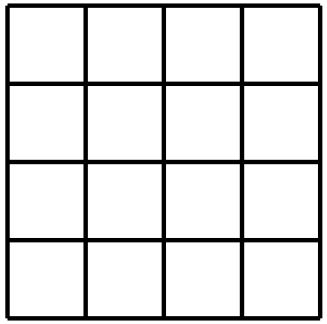
In 2-Dimensional Rectangular Grid

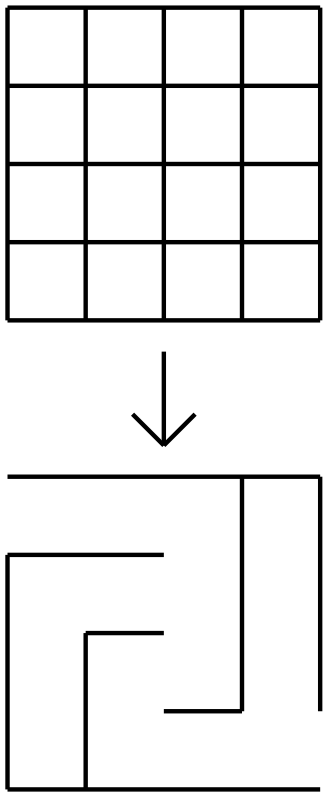
Perfect

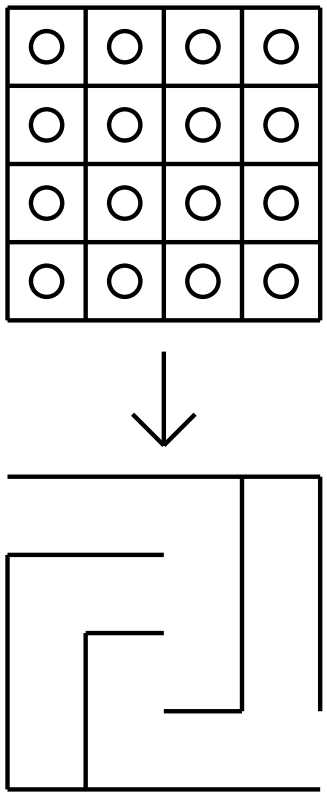
Assume mazes are

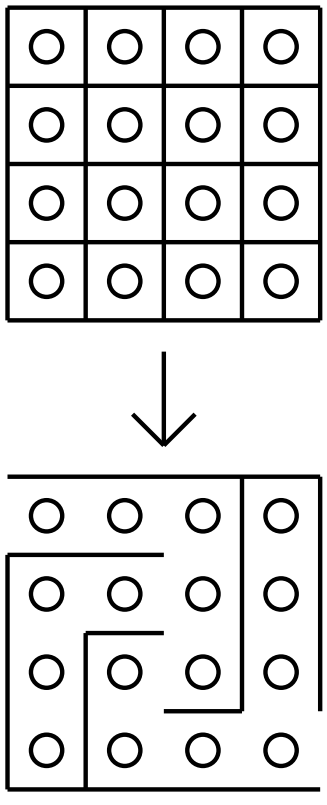


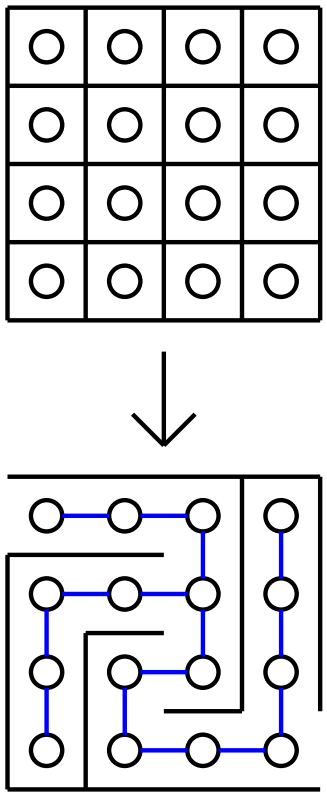
In 2-Dimensional Rectangular Grid
Perfect
Orthogonal Path

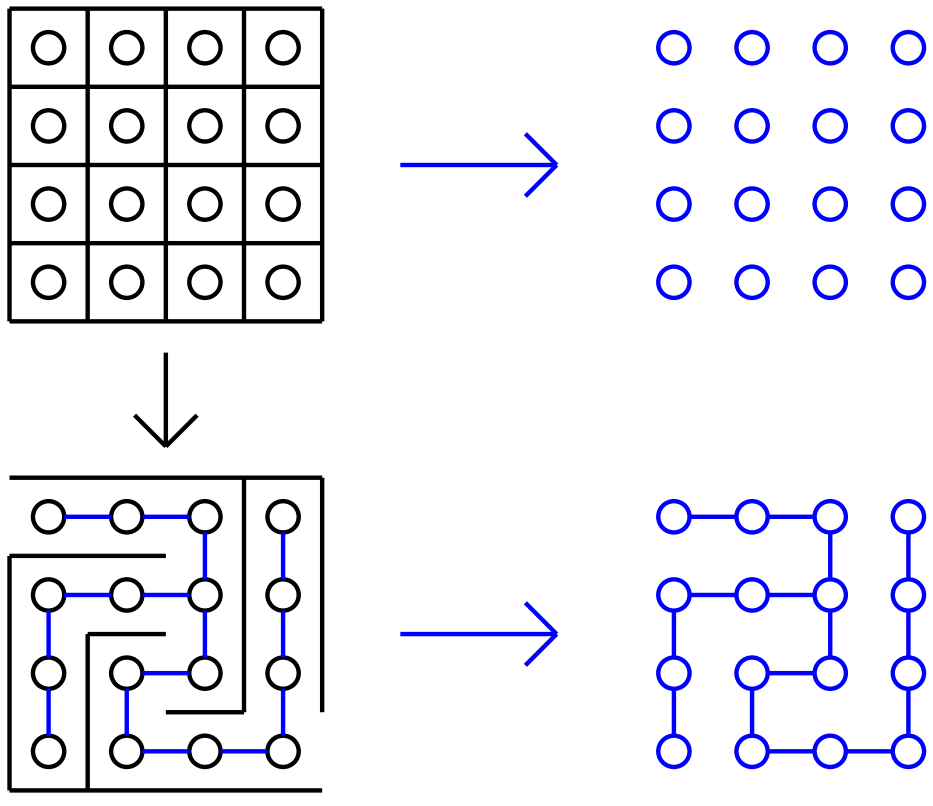


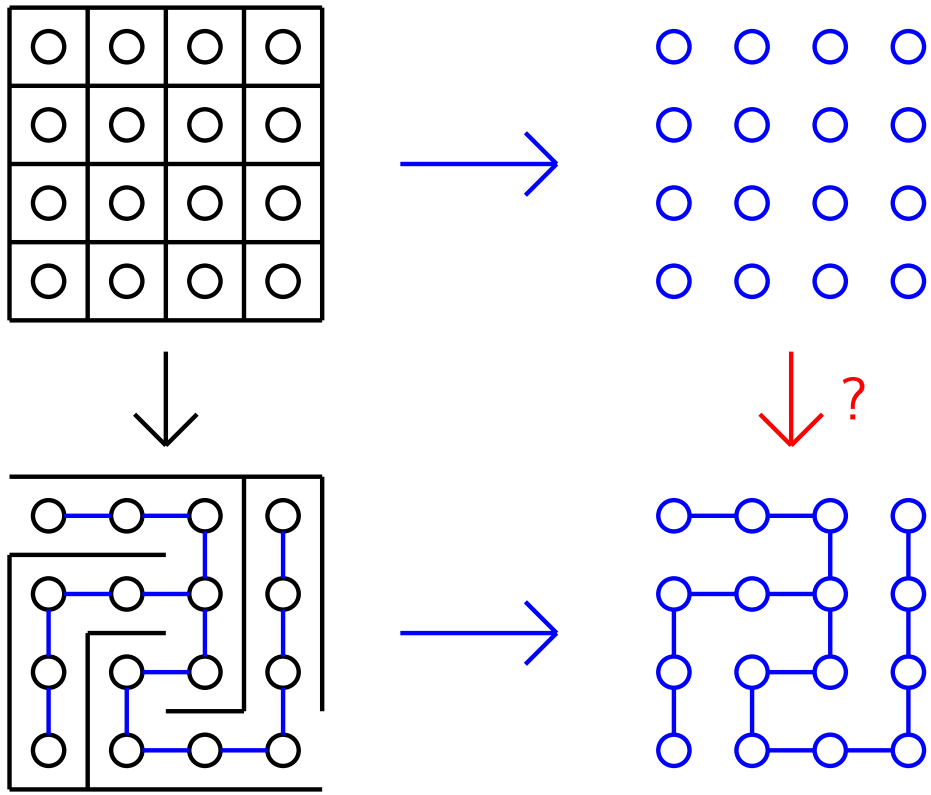






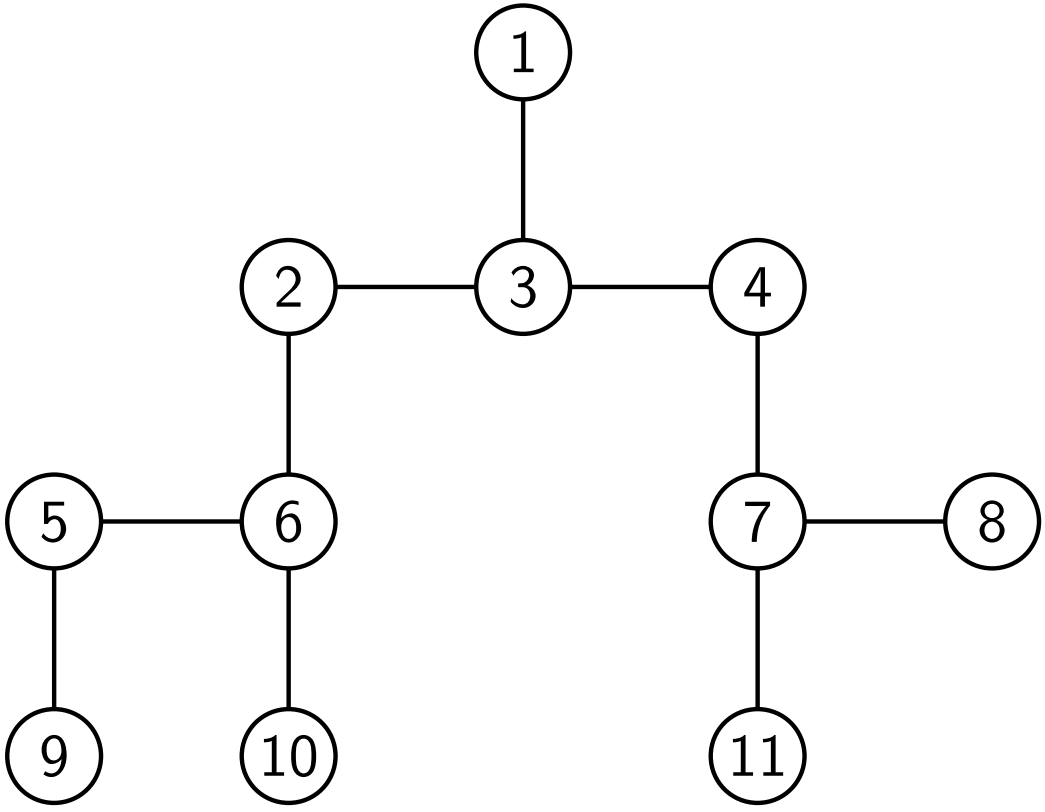




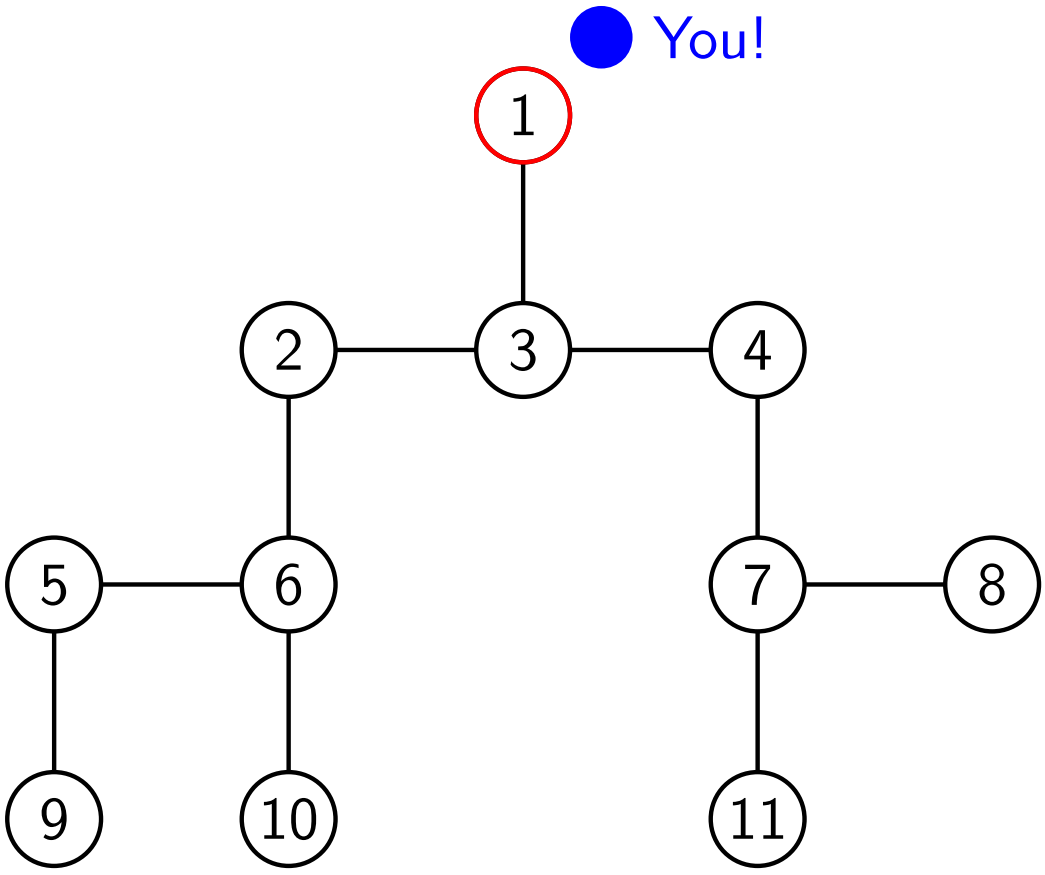


How can the 'perfect' graph be made?

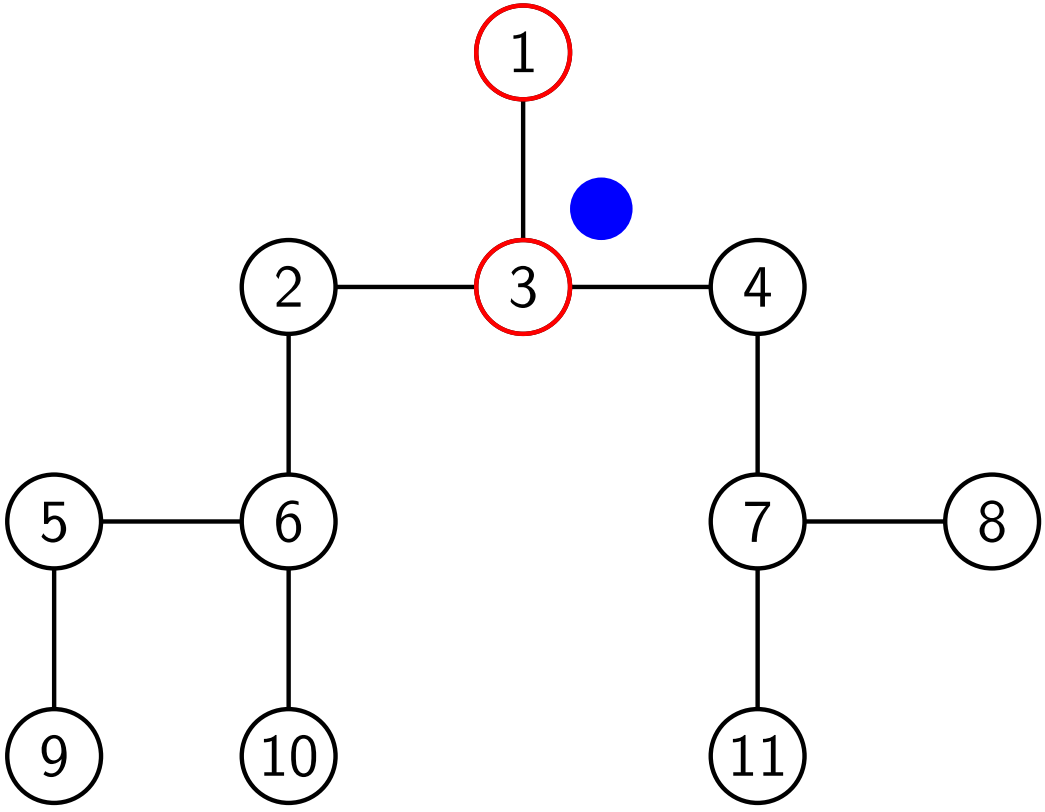
- DFS : One of the searching algorithms



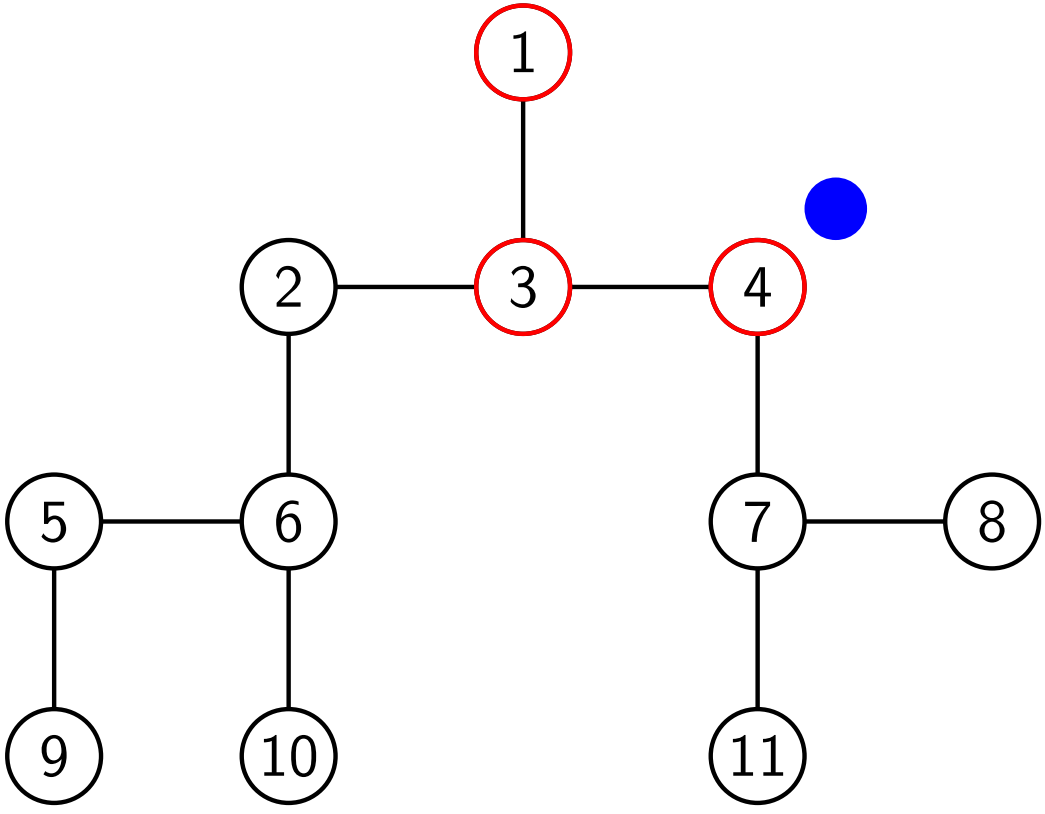
- DFS : One of the searching algorithms



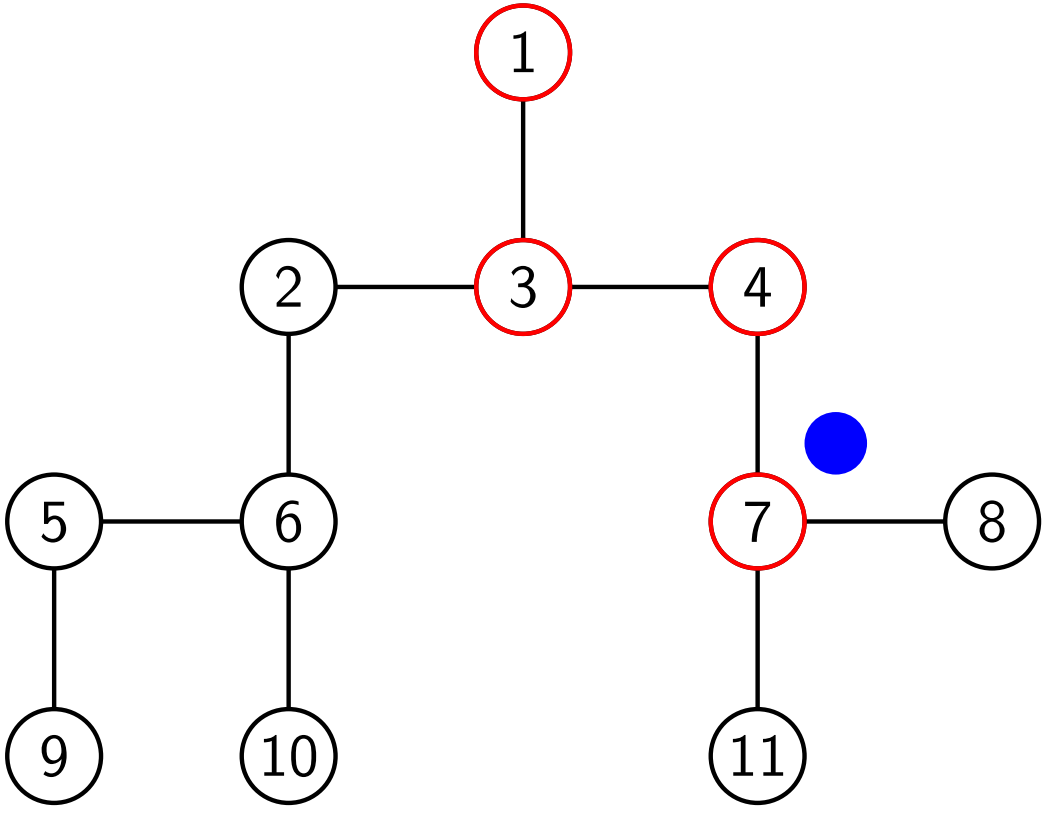
- DFS : One of the searching algorithms



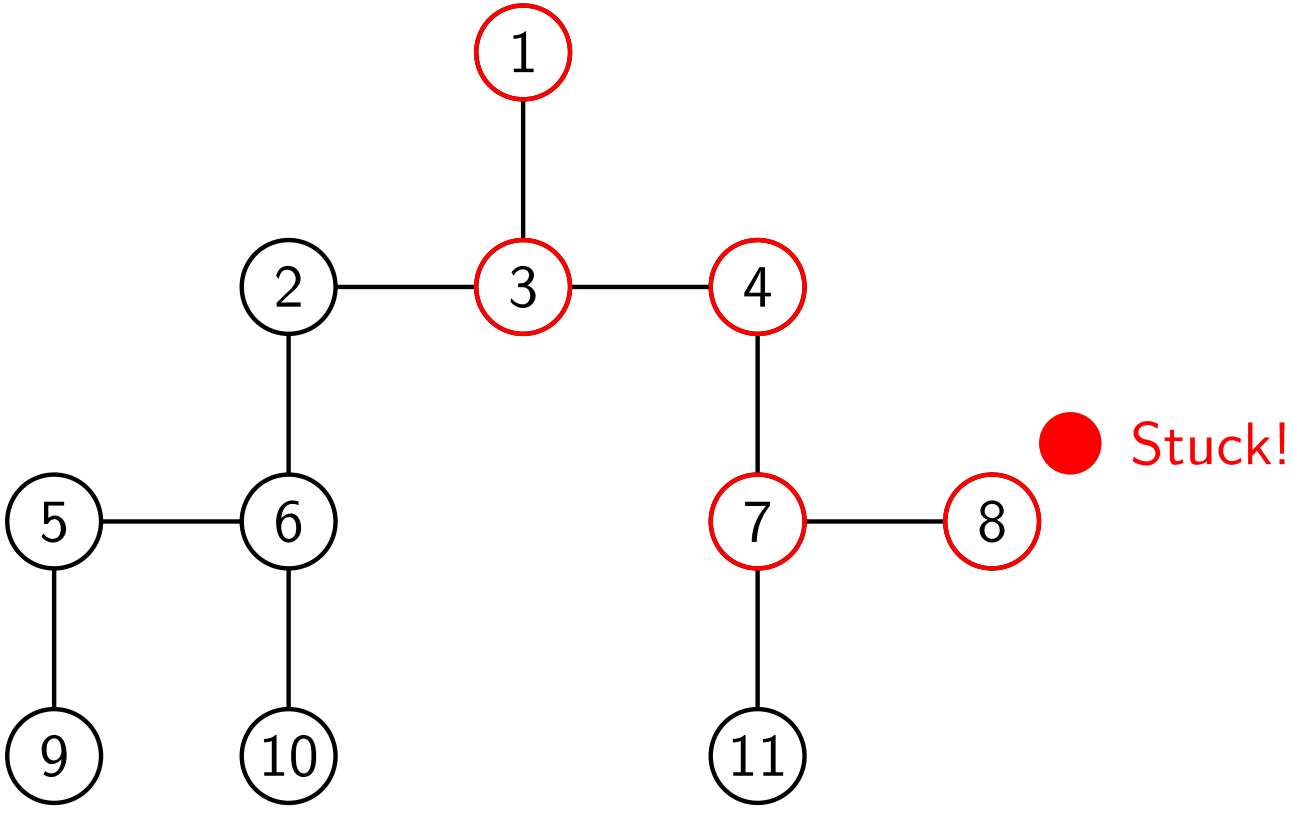
- DFS : One of the searching algorithms



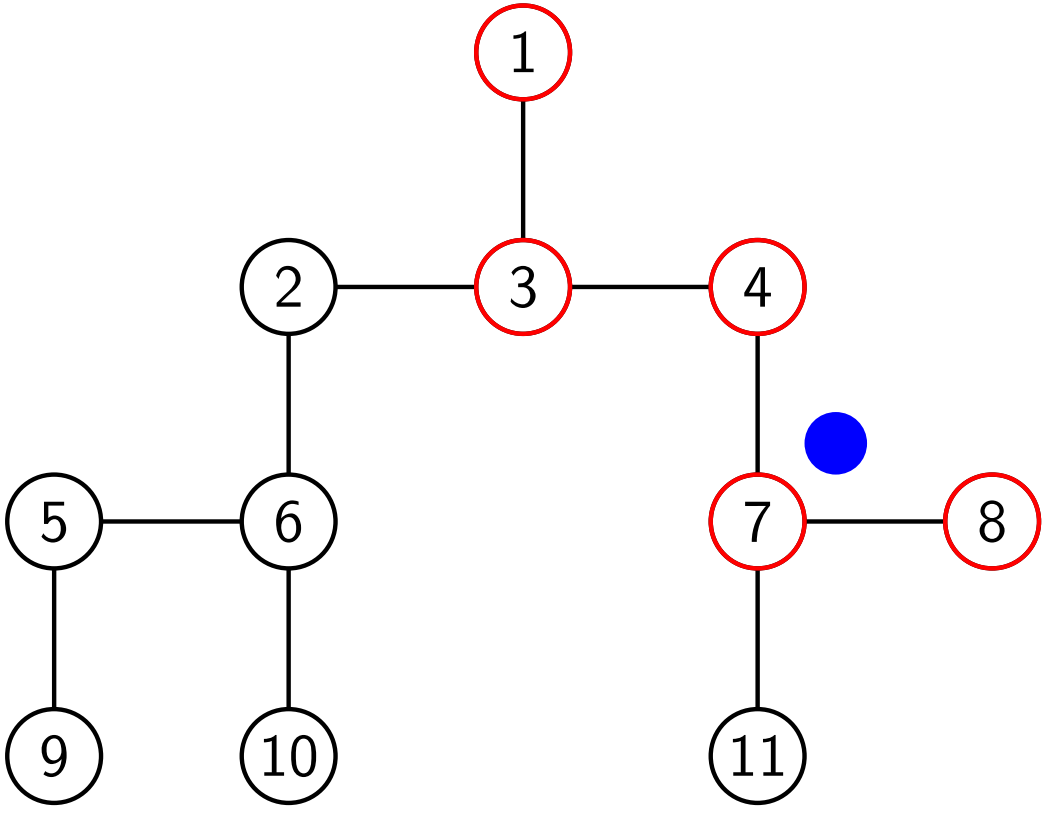
- DFS : One of the searching algorithms



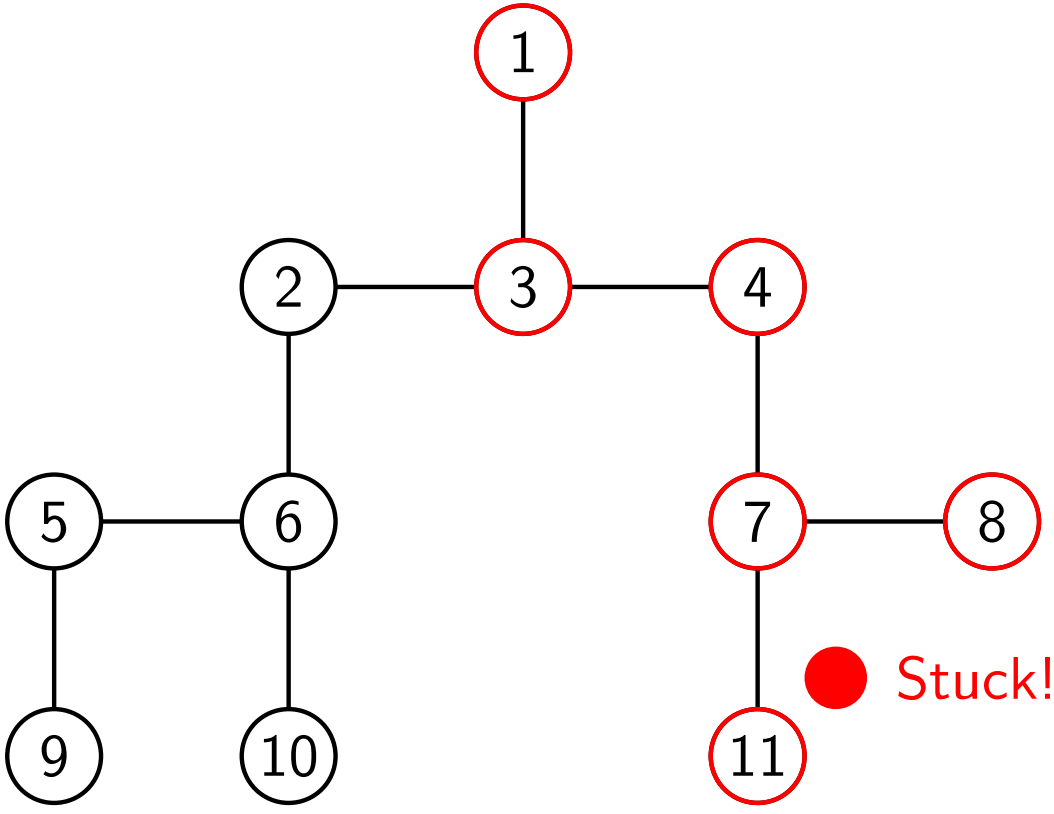
- DFS : One of the searching algorithms



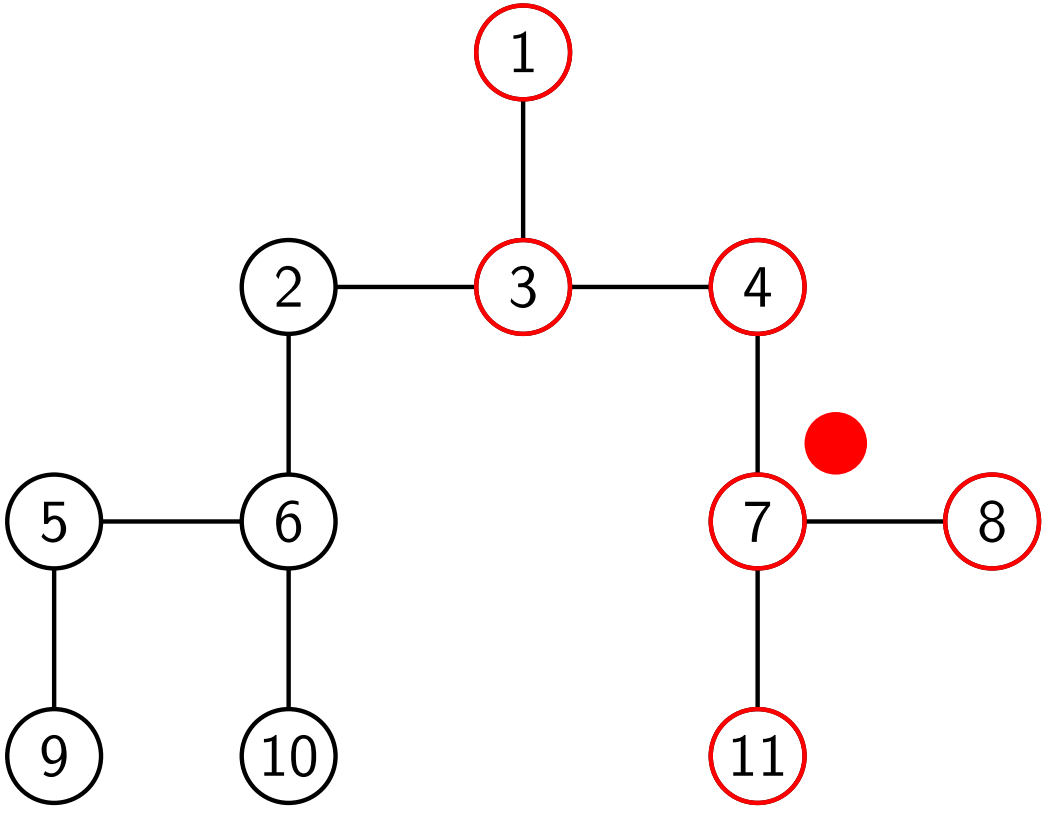
- DFS : One of the searching algorithms



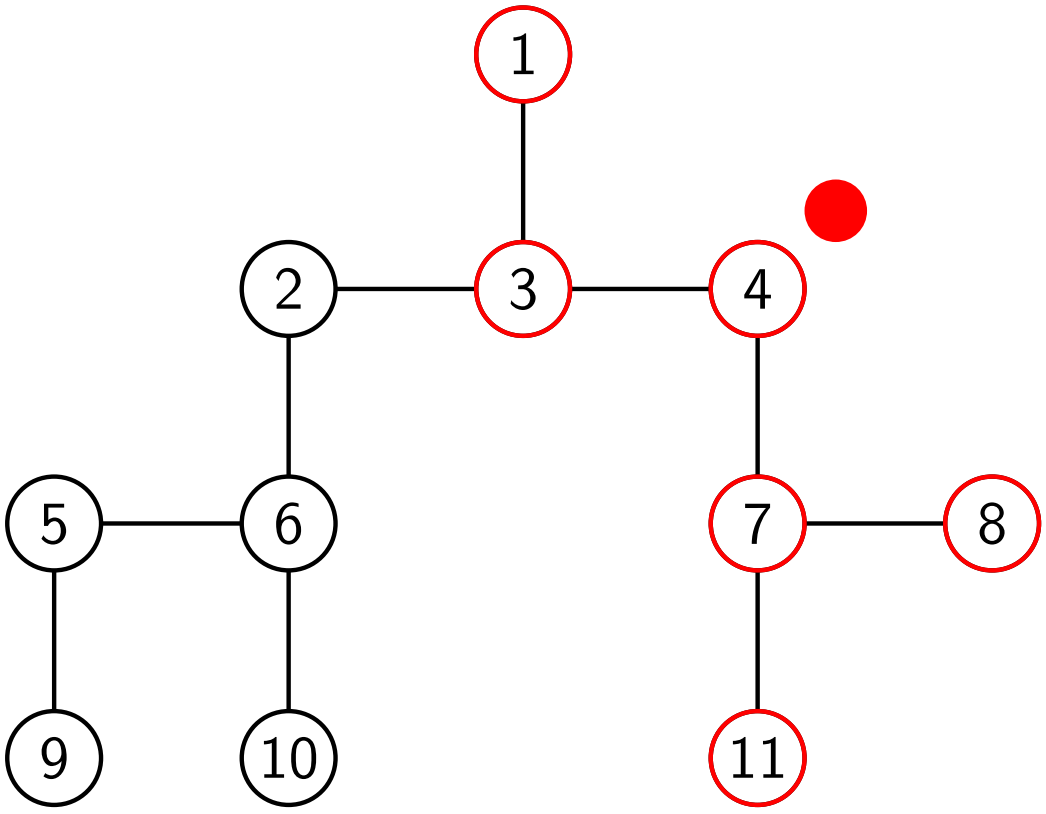
- DFS : One of the searching algorithms



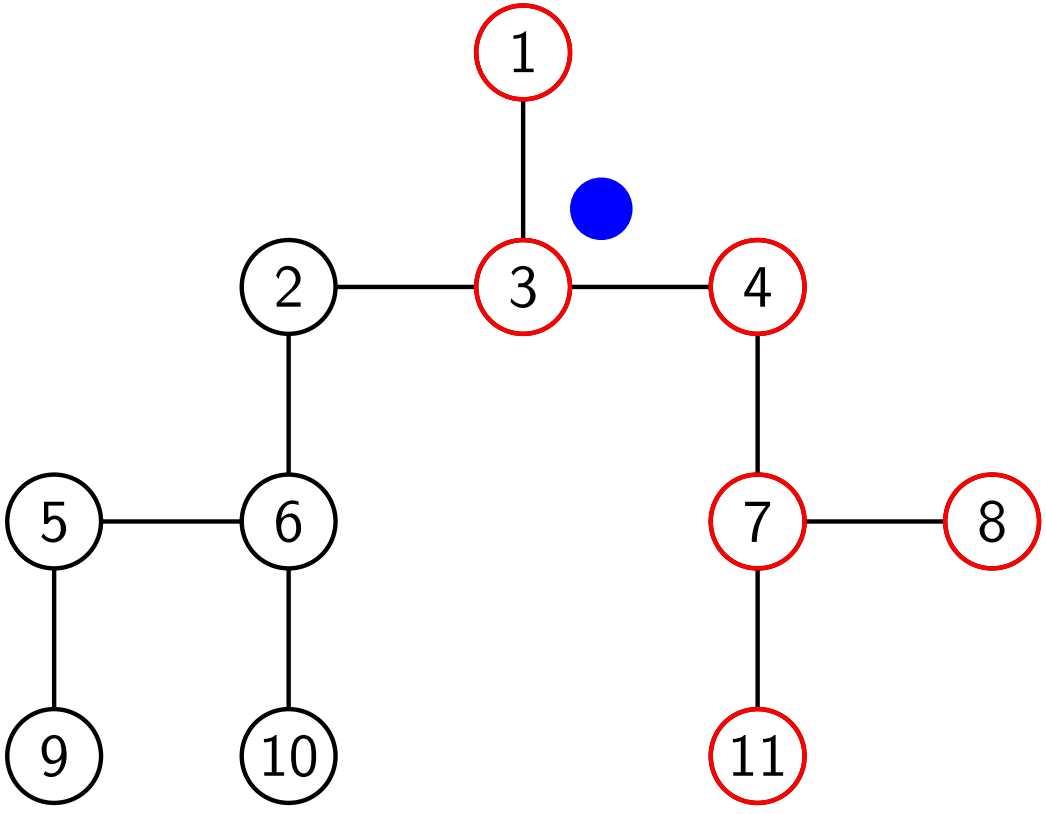
- DFS : One of the searching algorithms



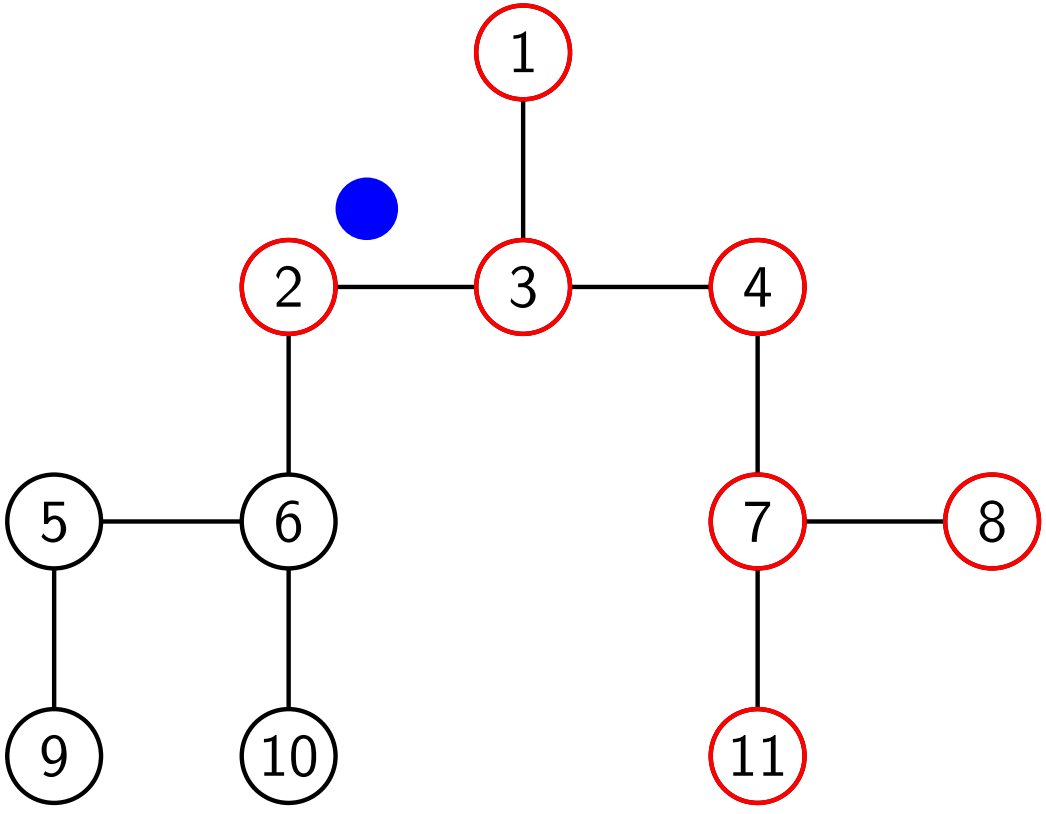
- DFS : One of the searching algorithms



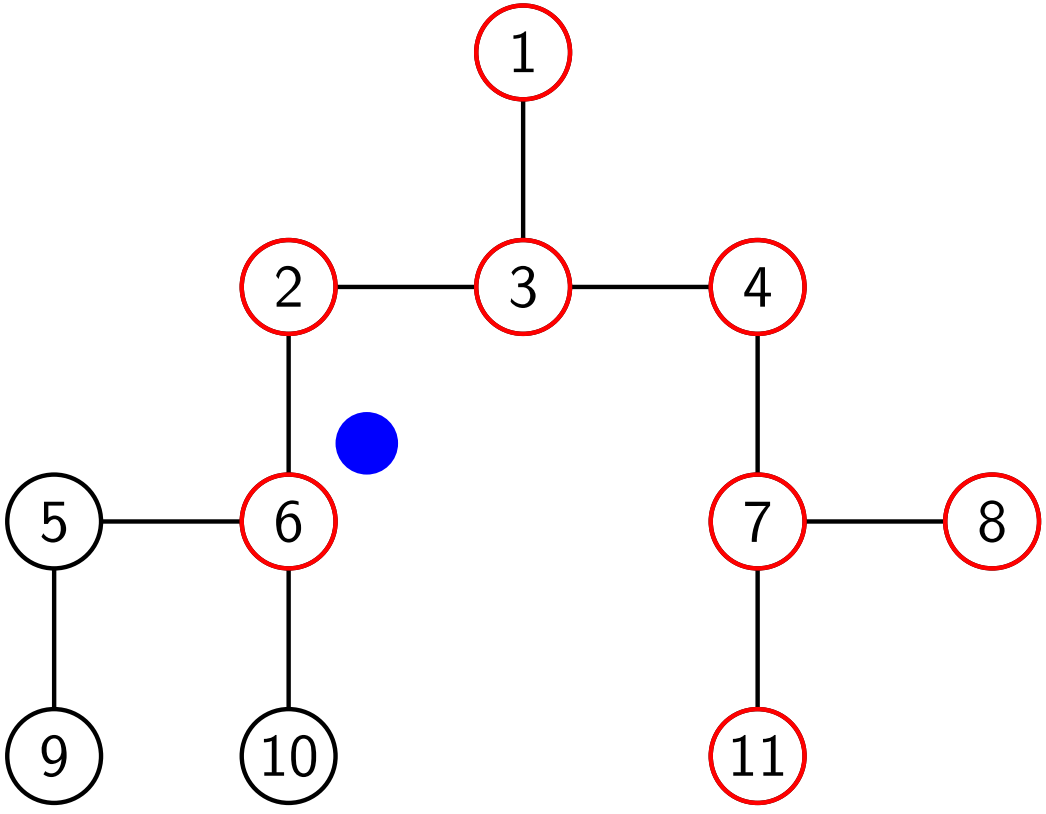
- DFS : One of the searching algorithms



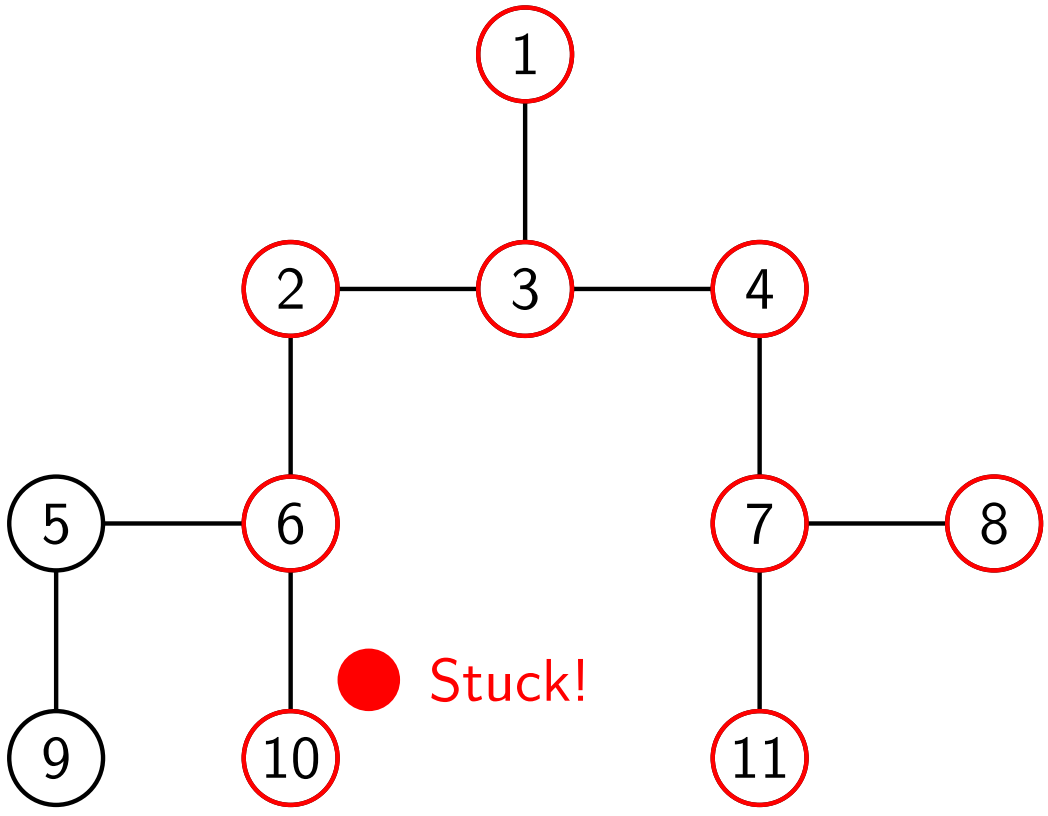
- DFS : One of the searching algorithms



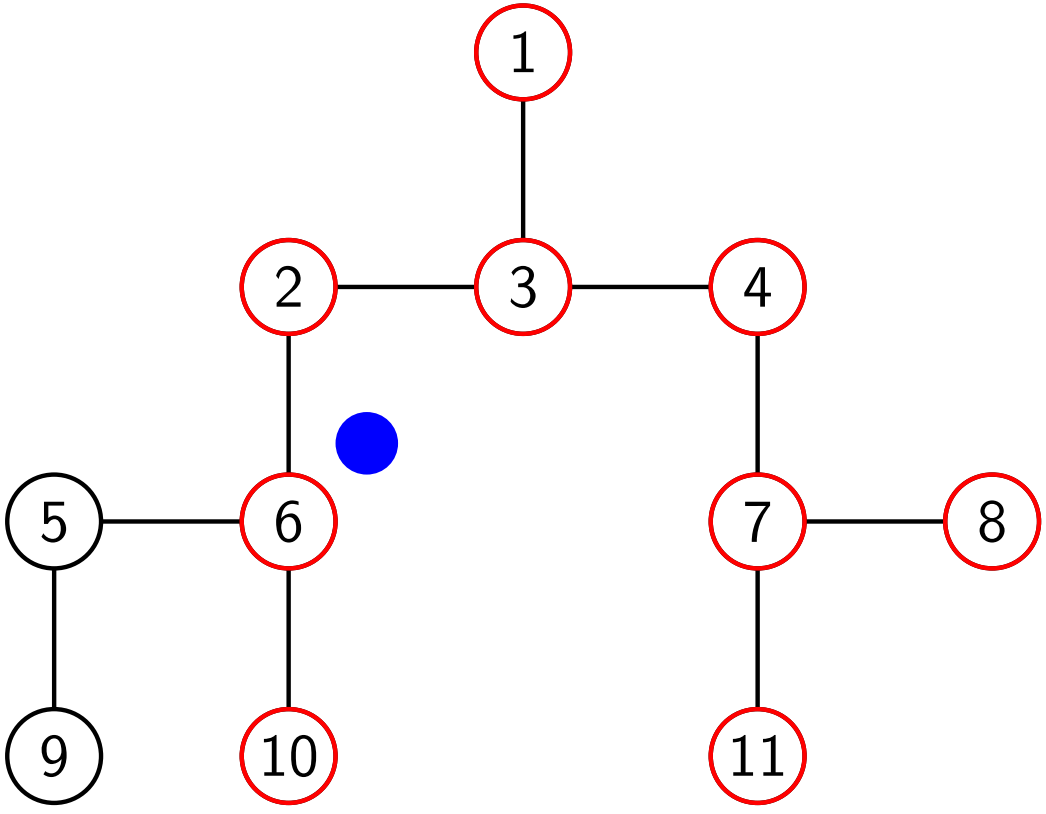
- DFS : One of the searching algorithms



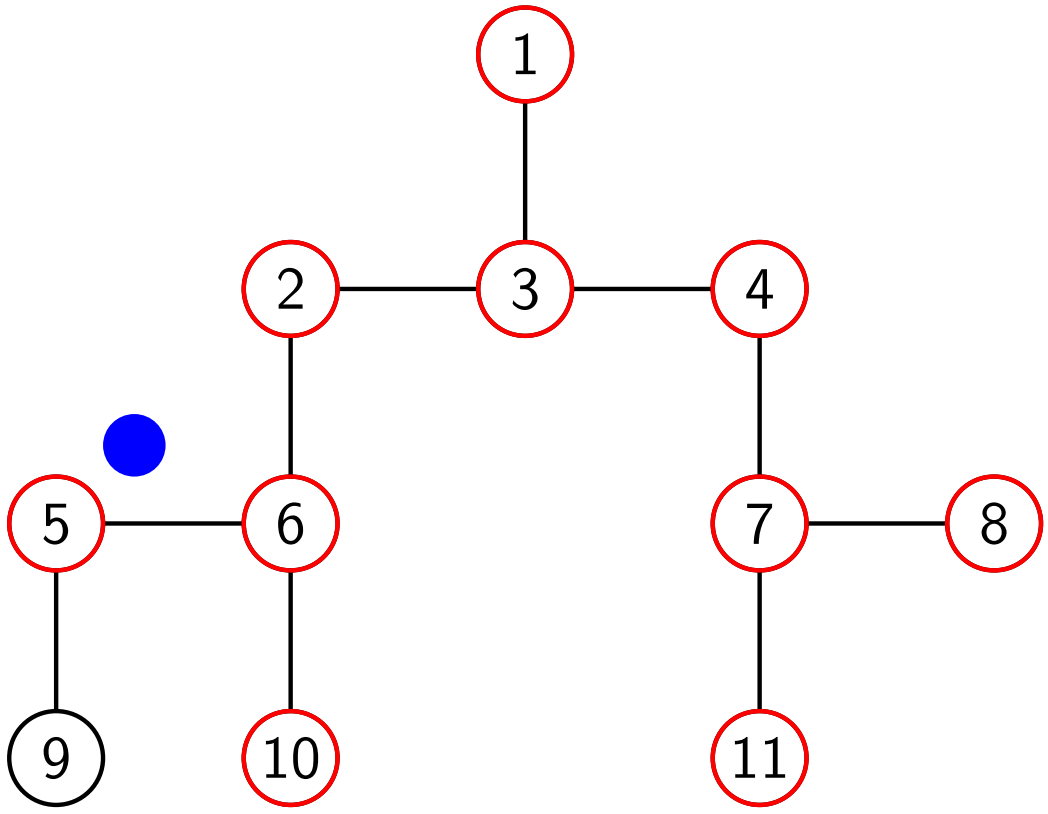
- DFS : One of the searching algorithms



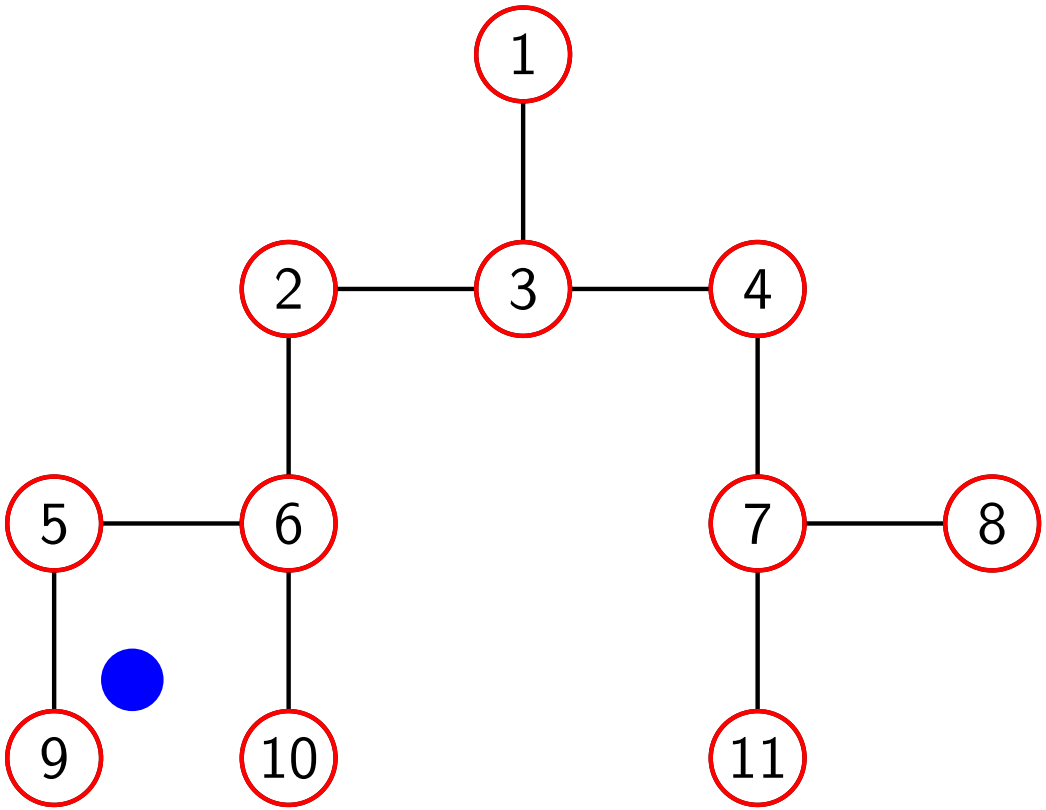
- DFS : One of the searching algorithms



- DFS : One of the searching algorithms



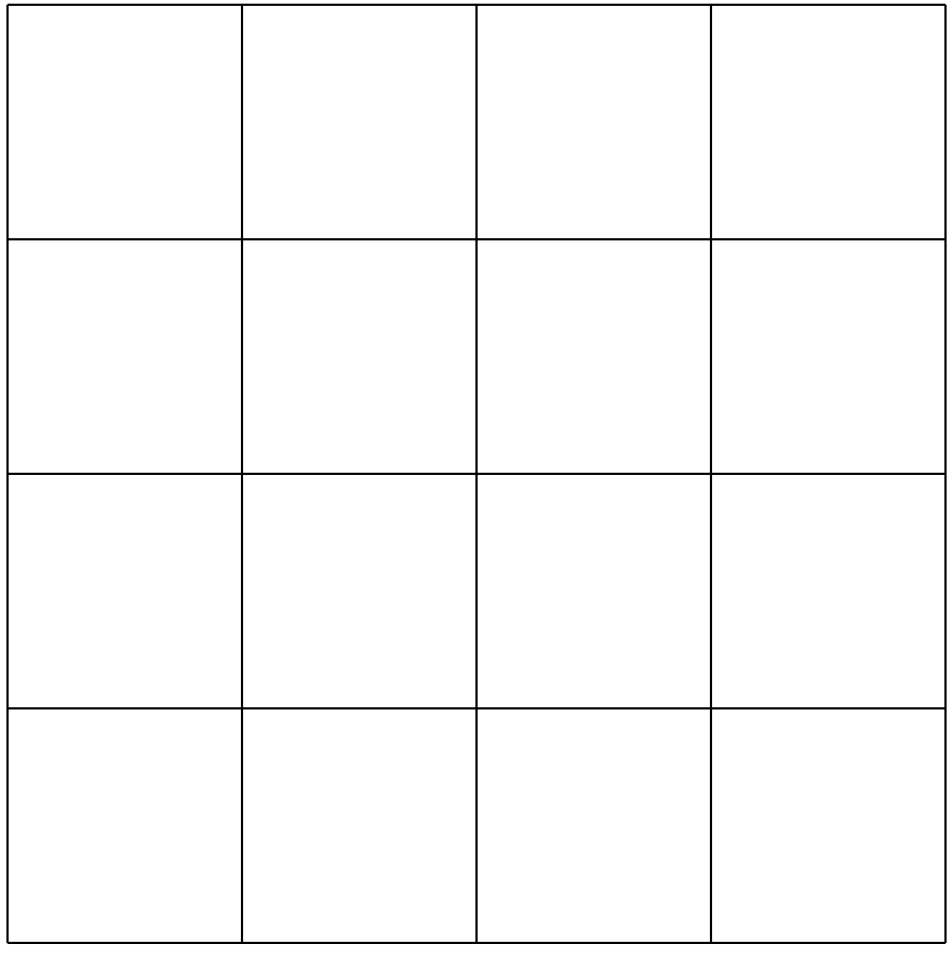
- DFS : One of the searching algorithms

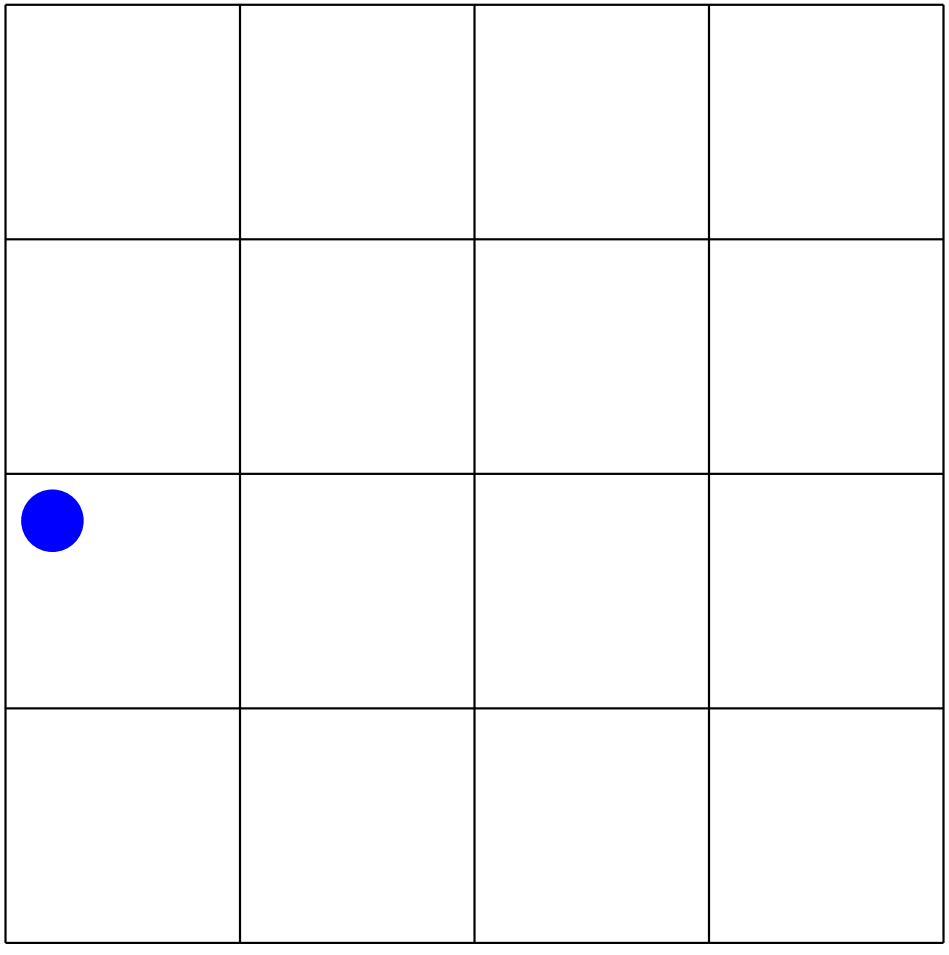


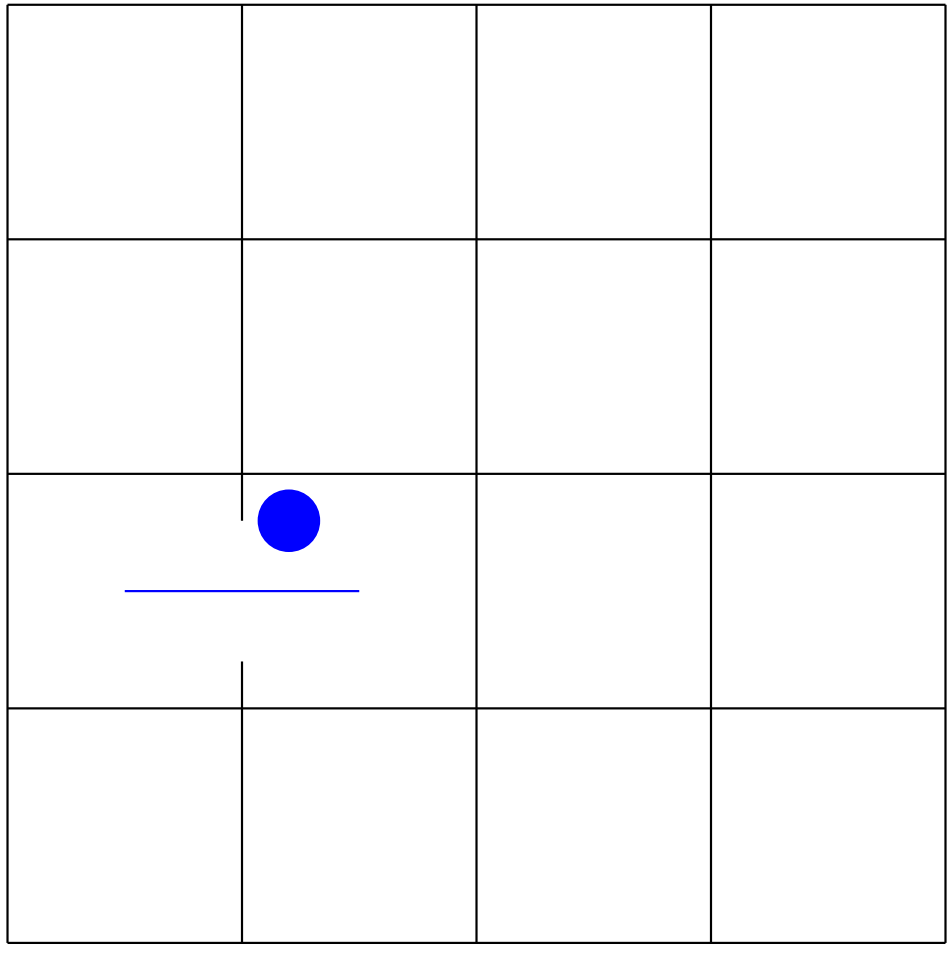
End!

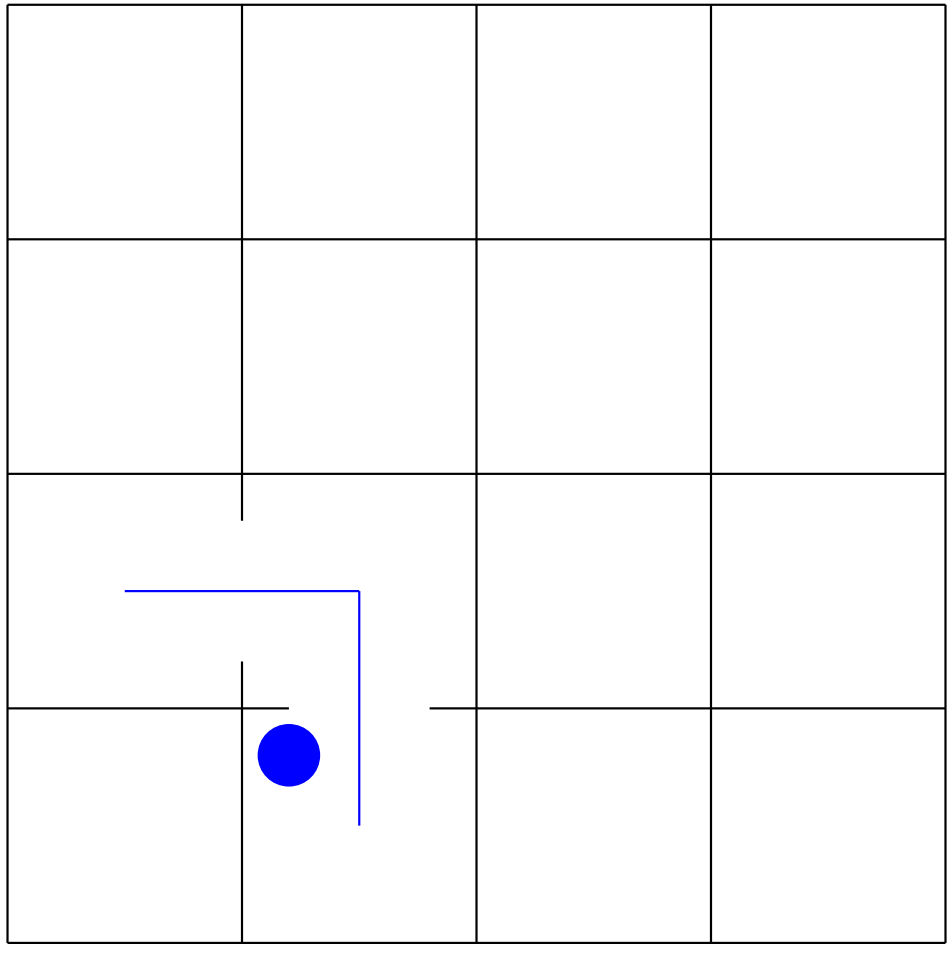
- Go deeper as you can.

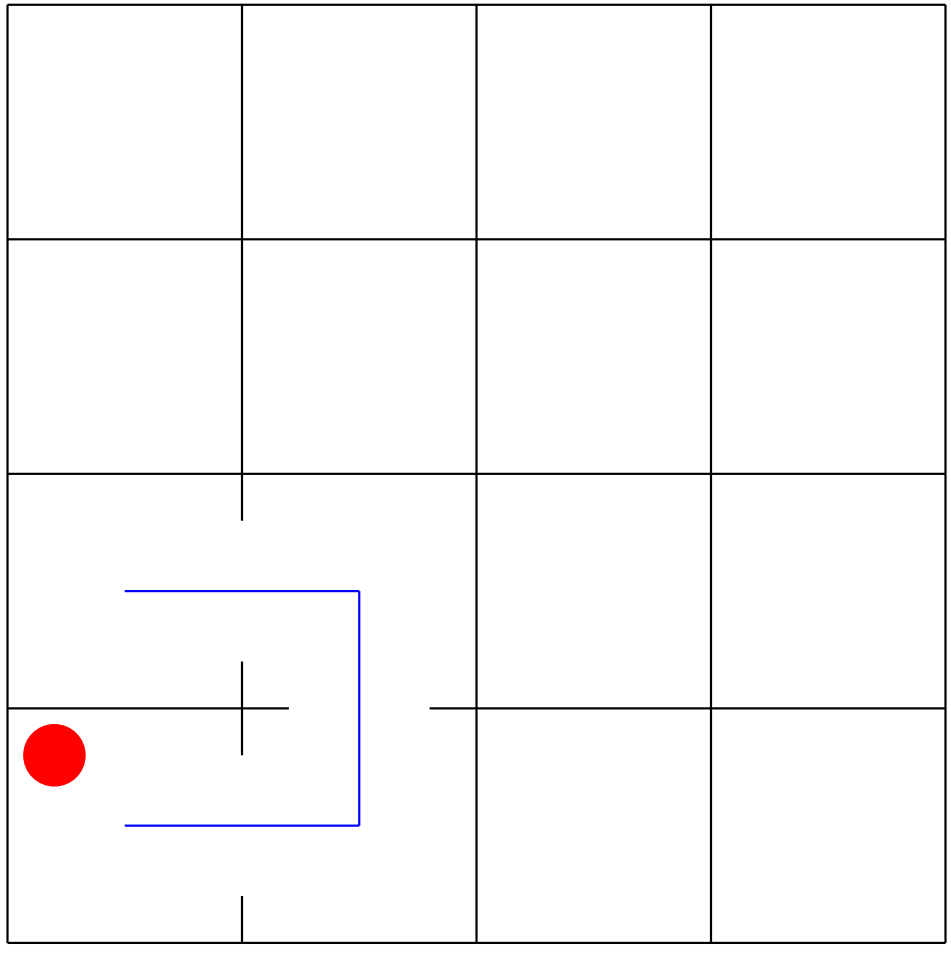
- Go deeper as you can.
- Backtrack to possible branch when you are stuck.

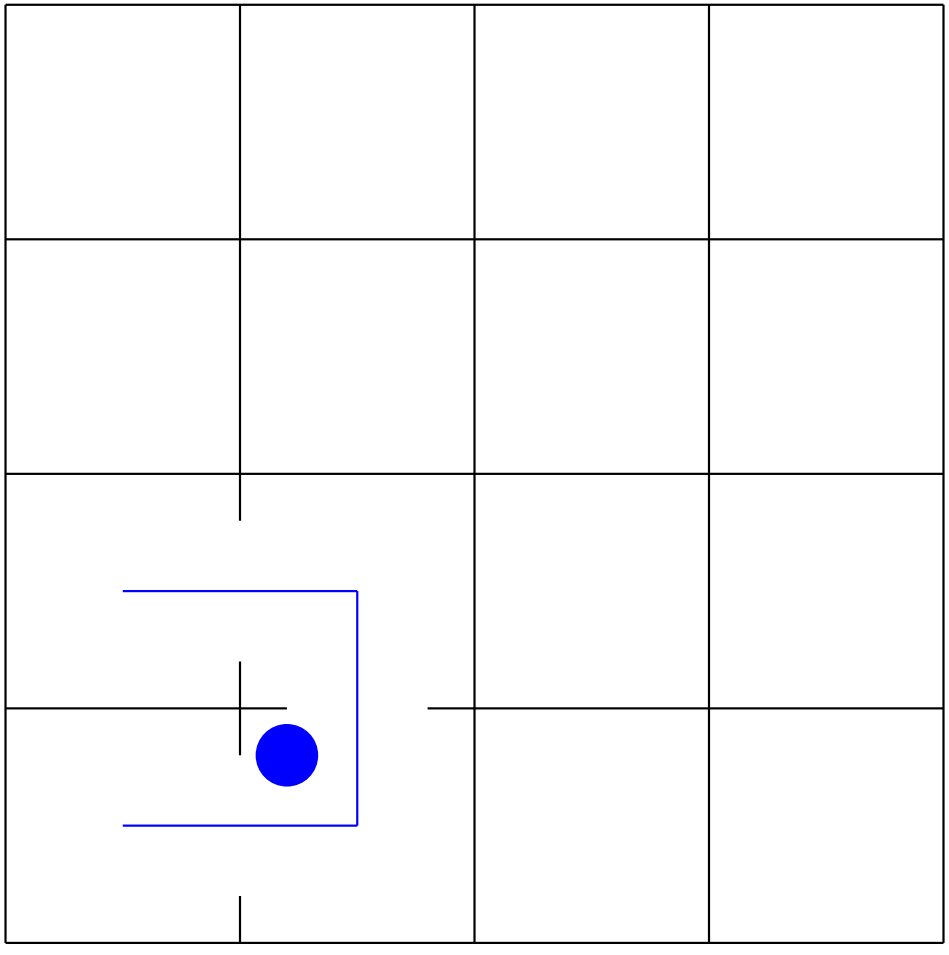


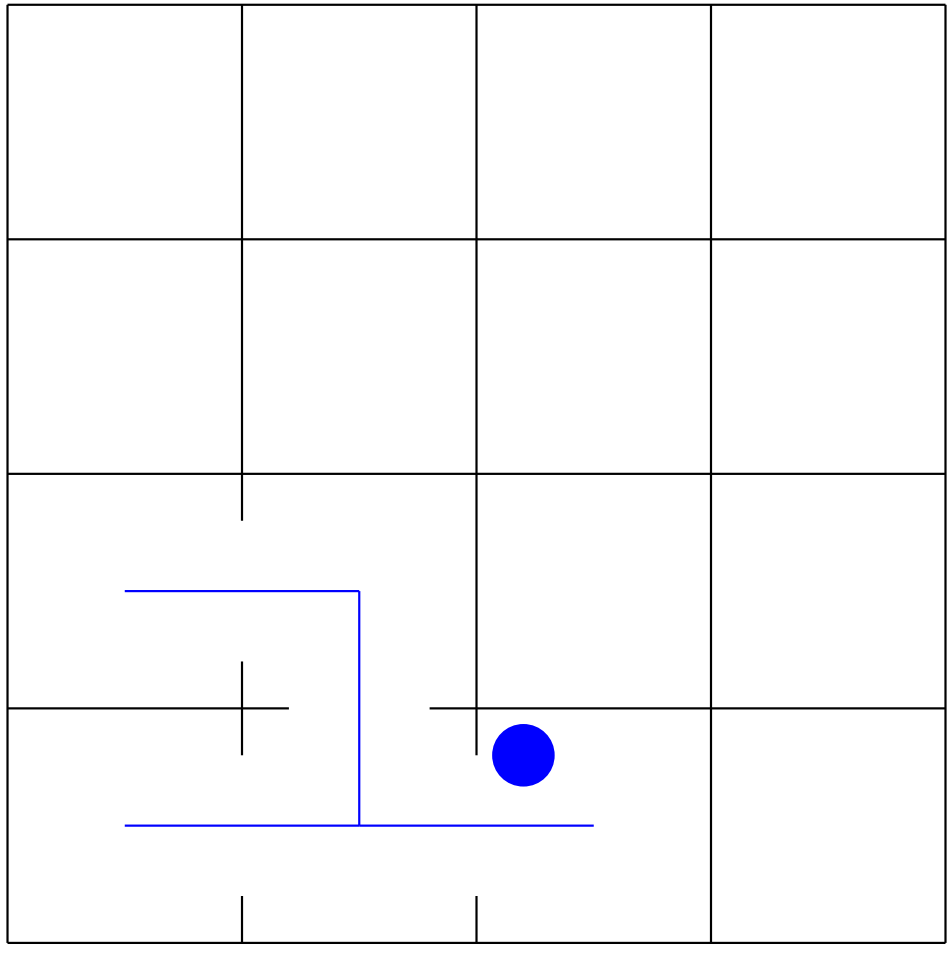


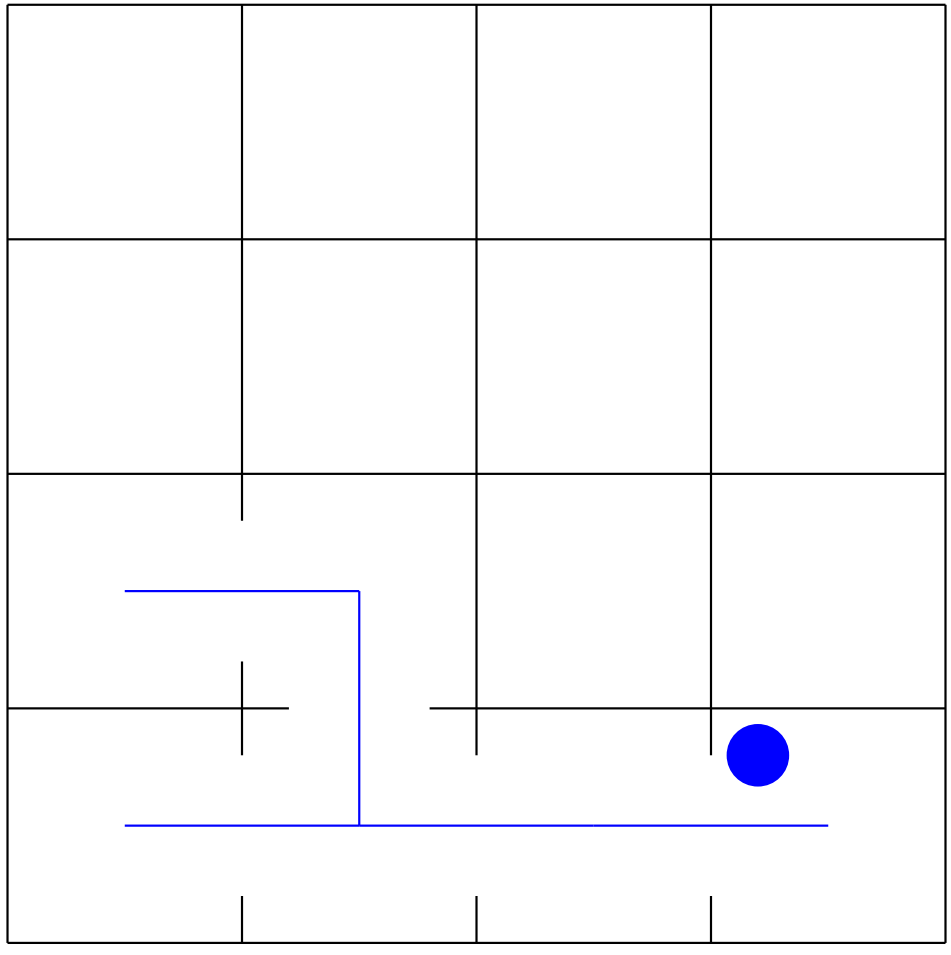


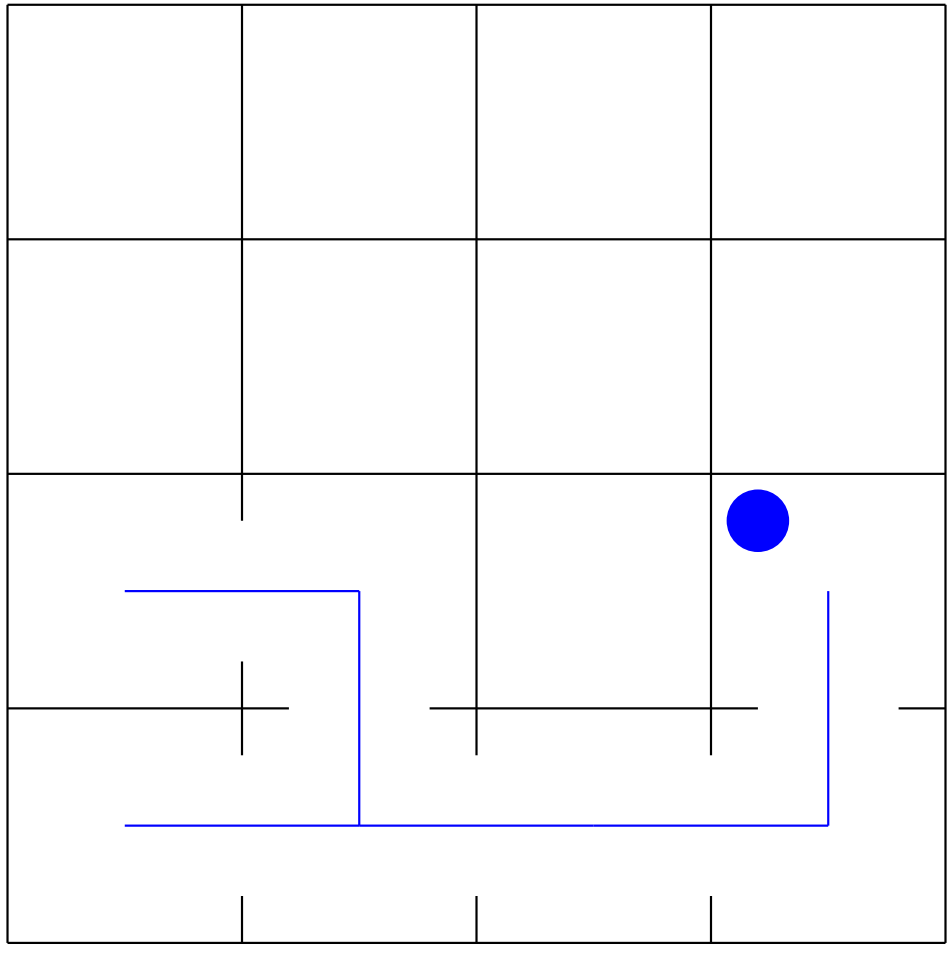


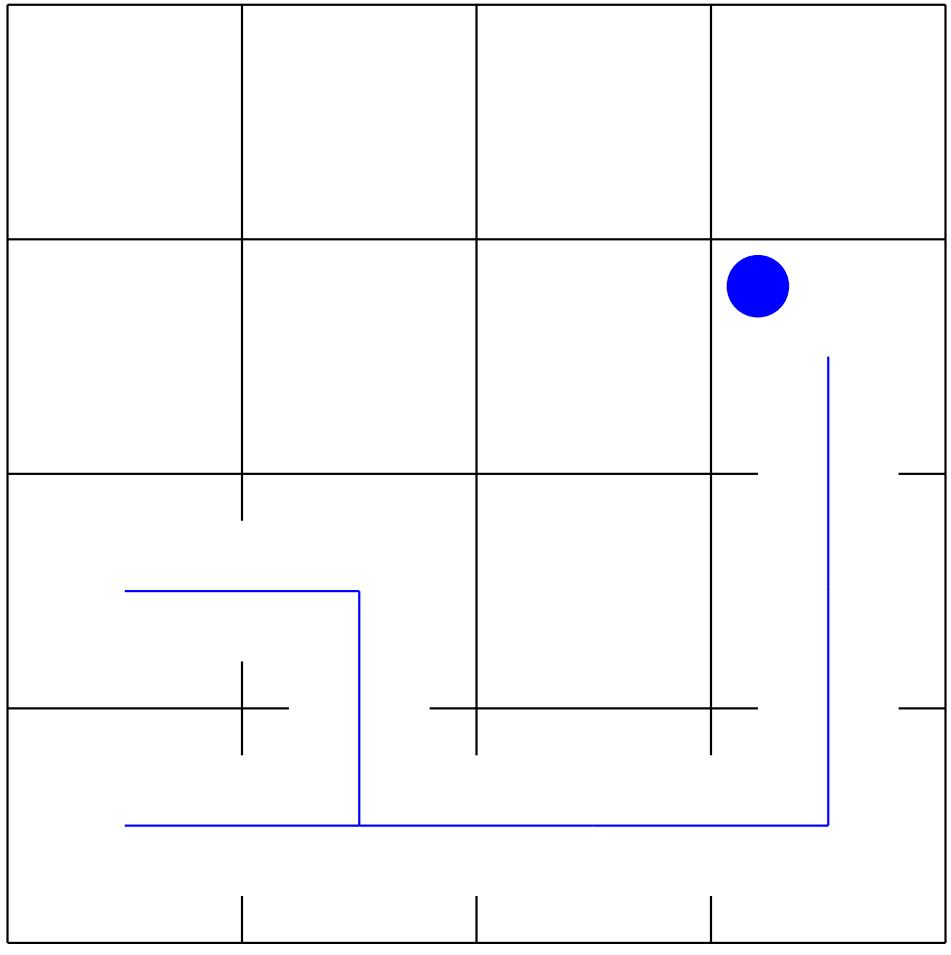


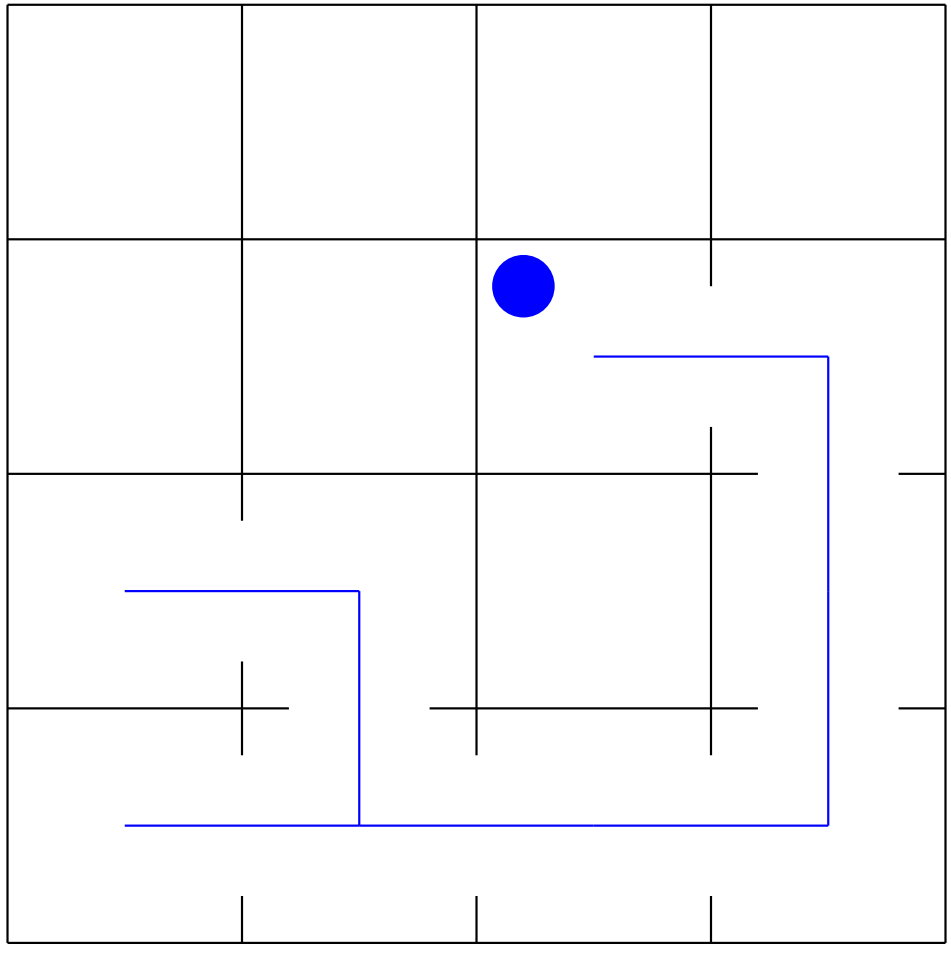


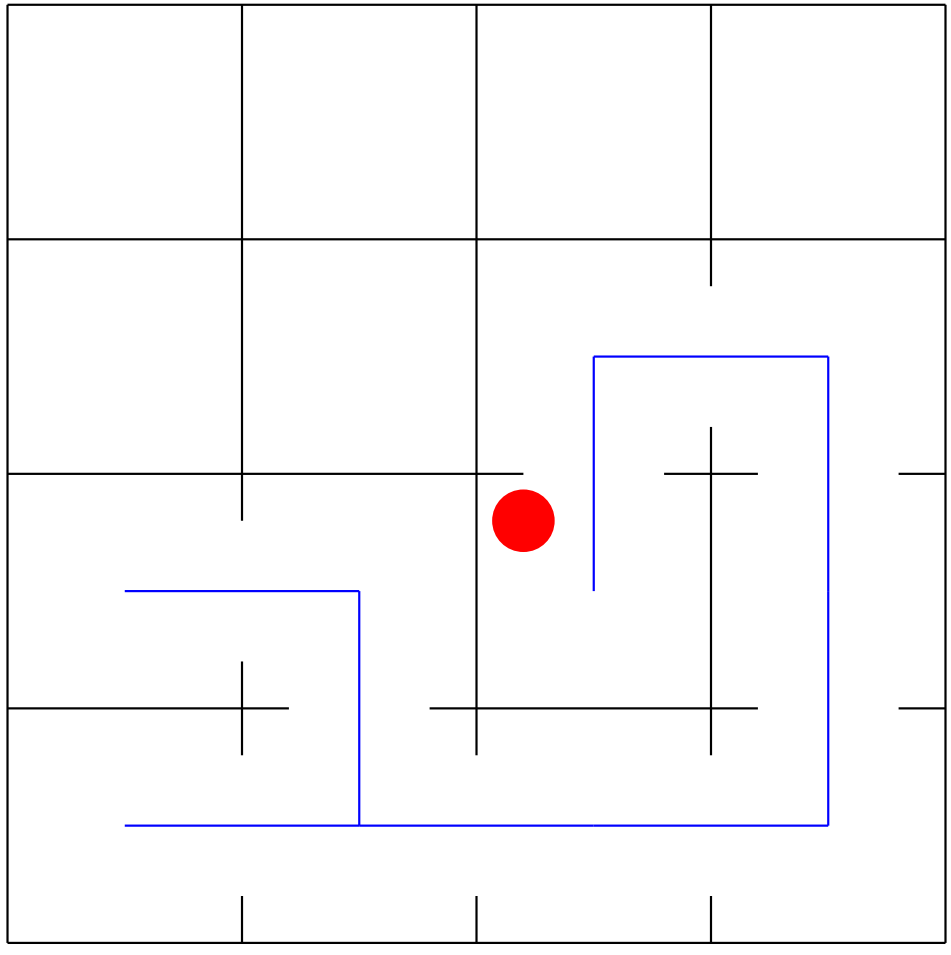


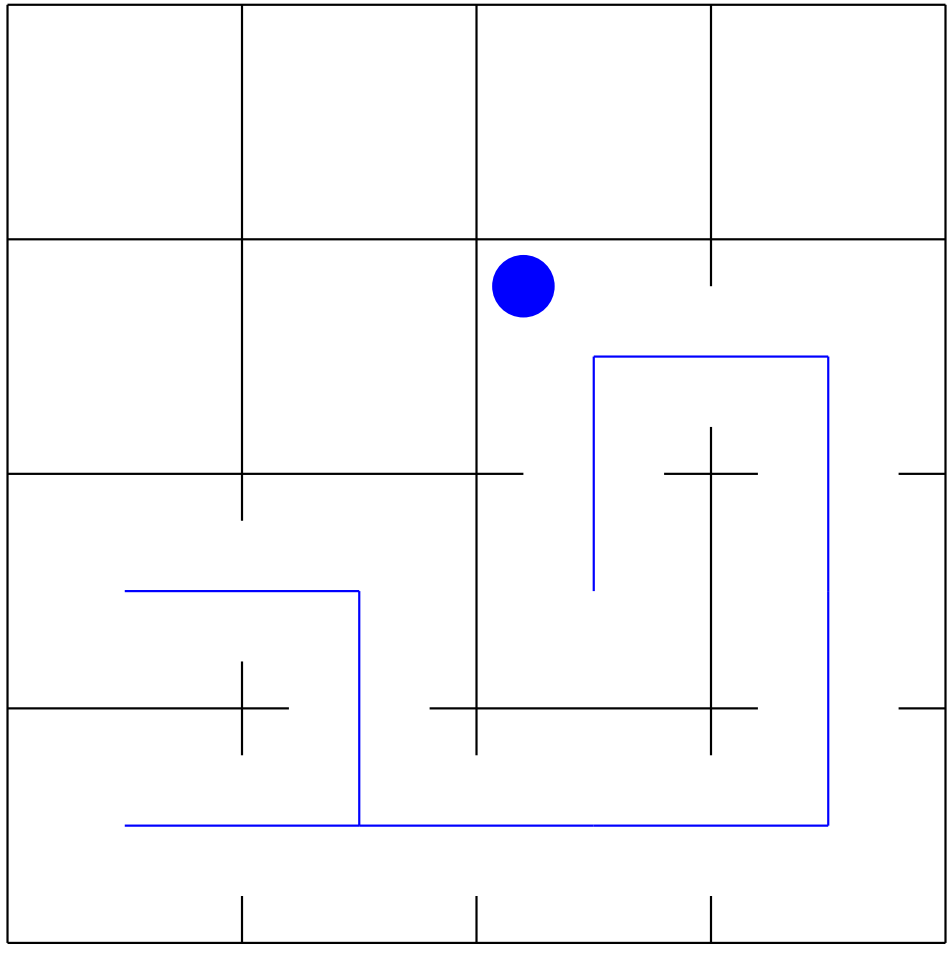


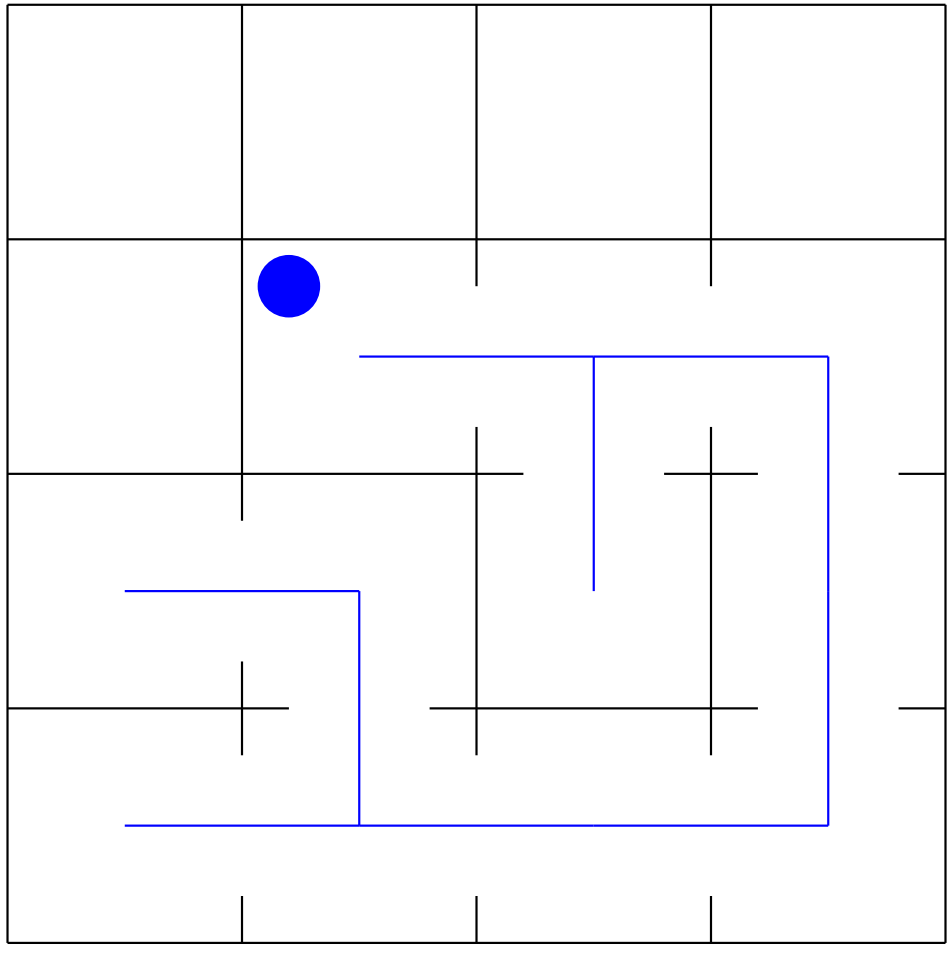


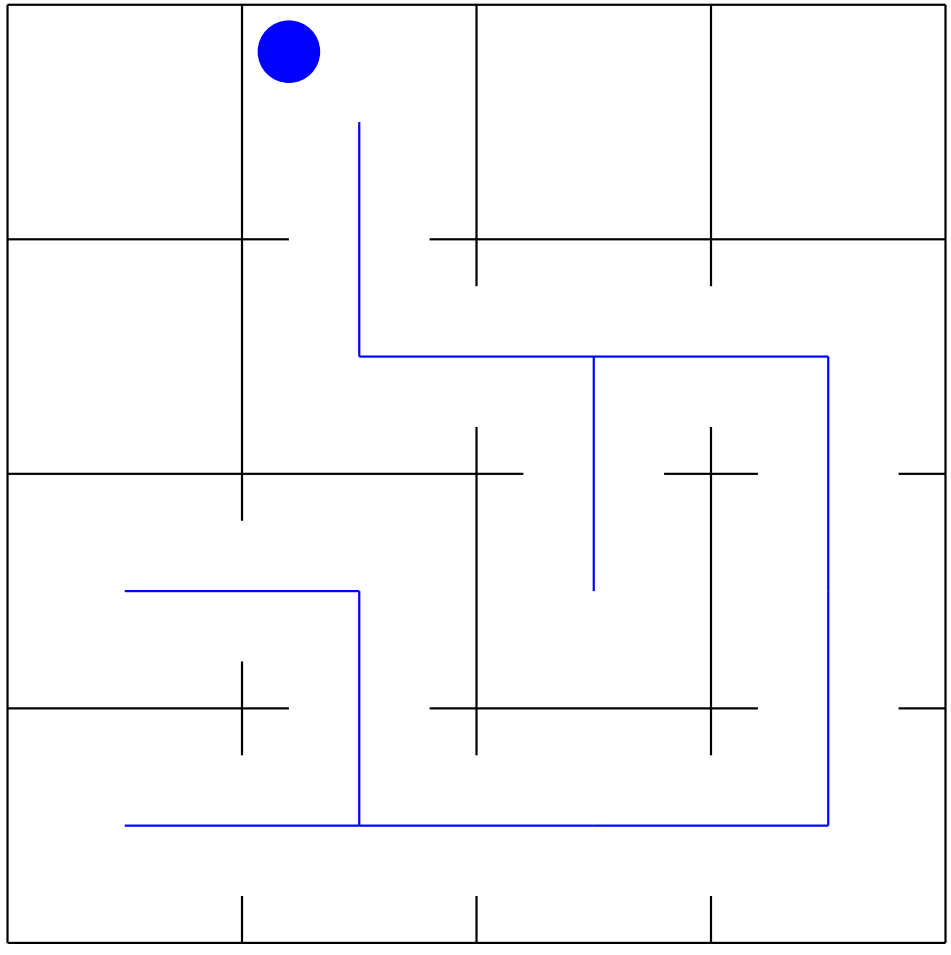


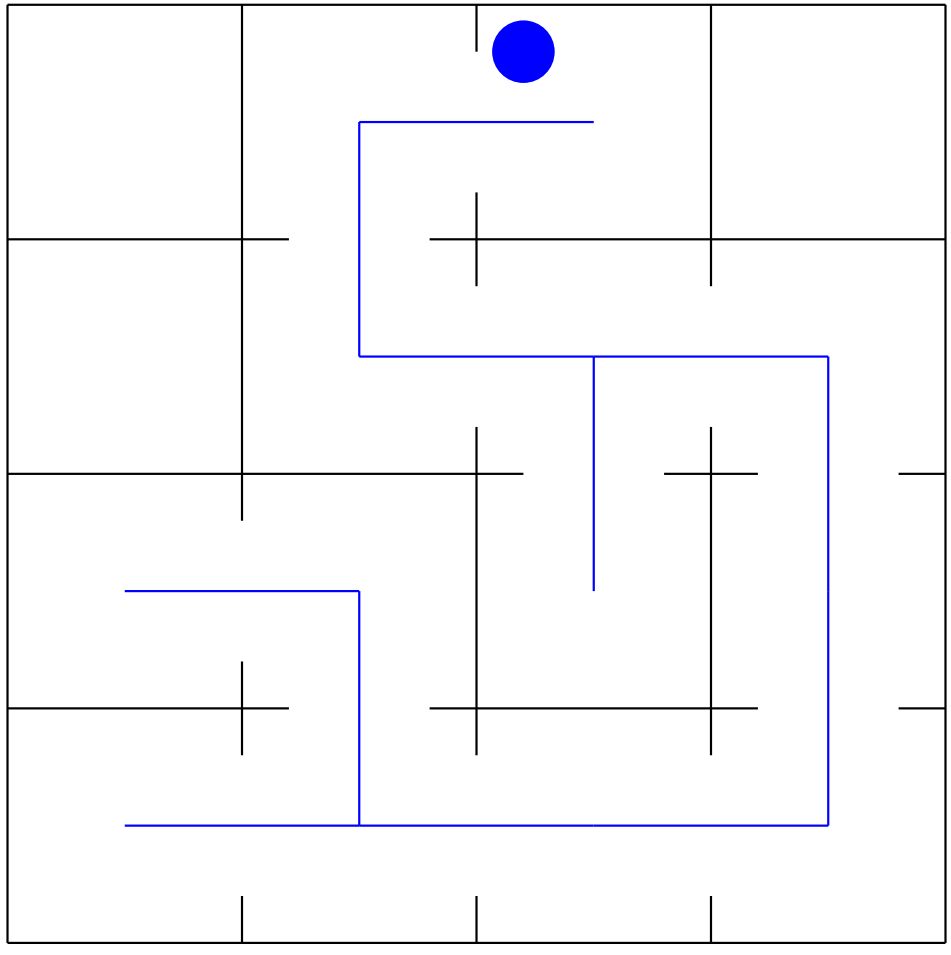


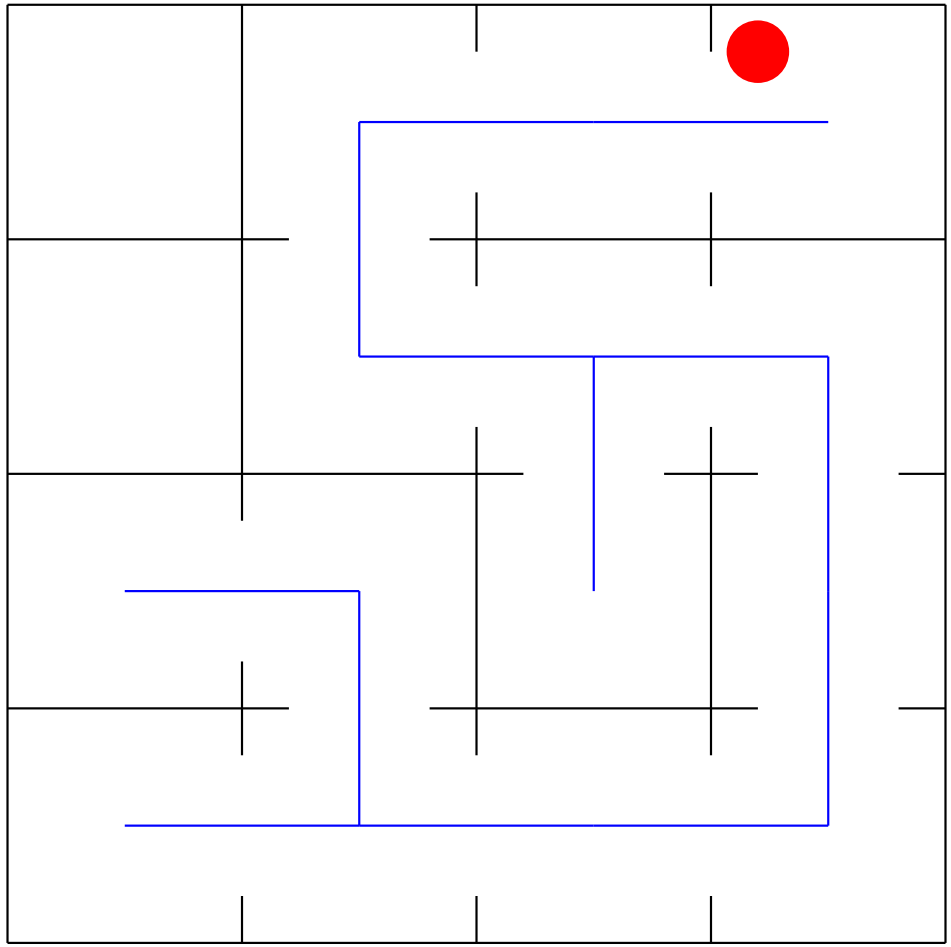


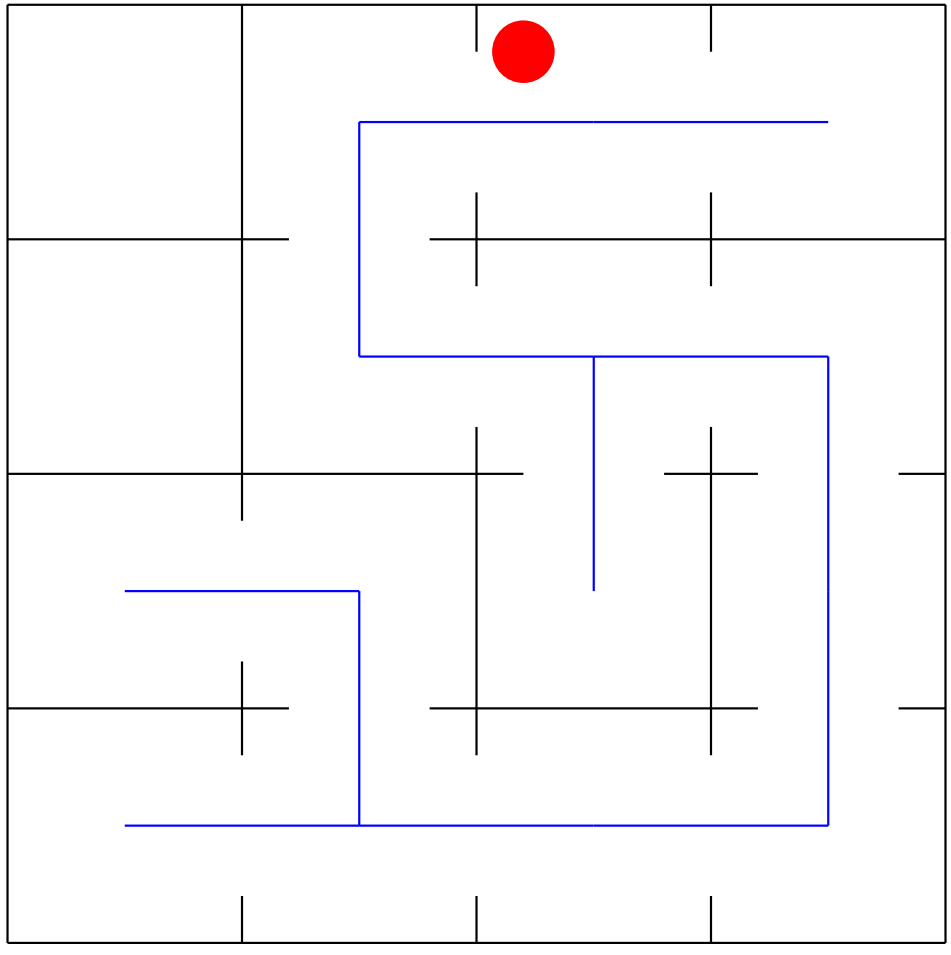


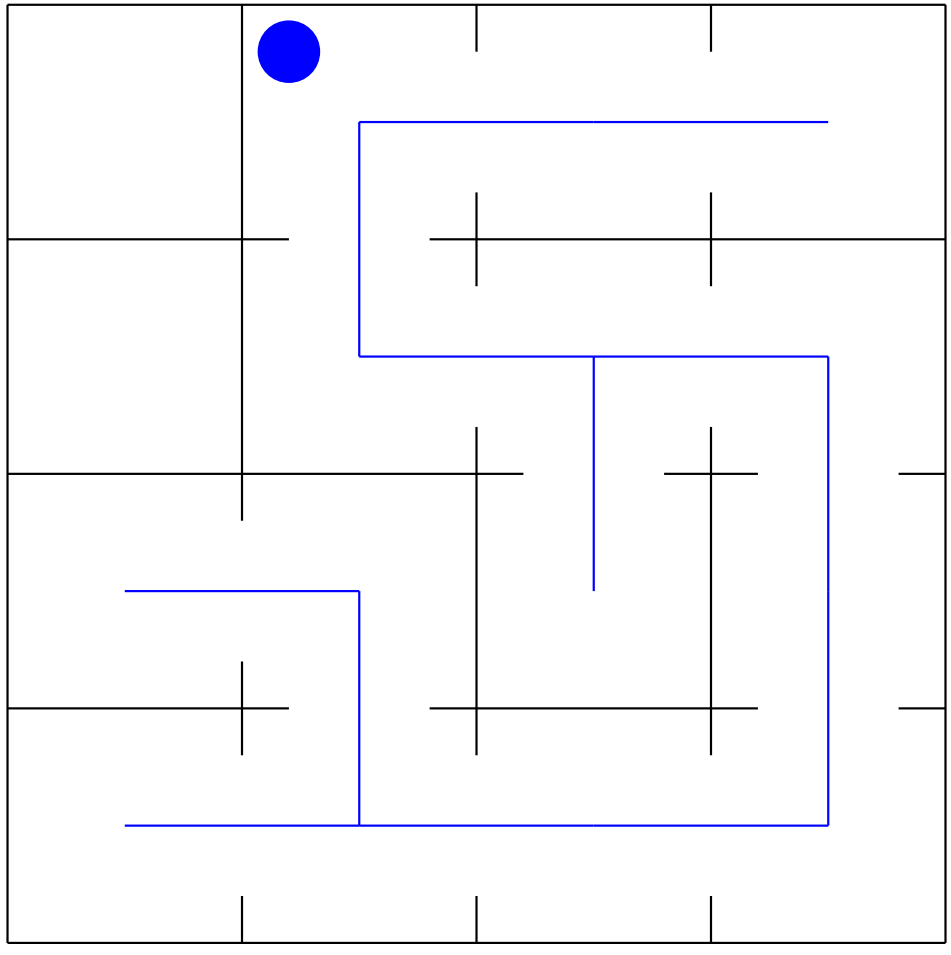


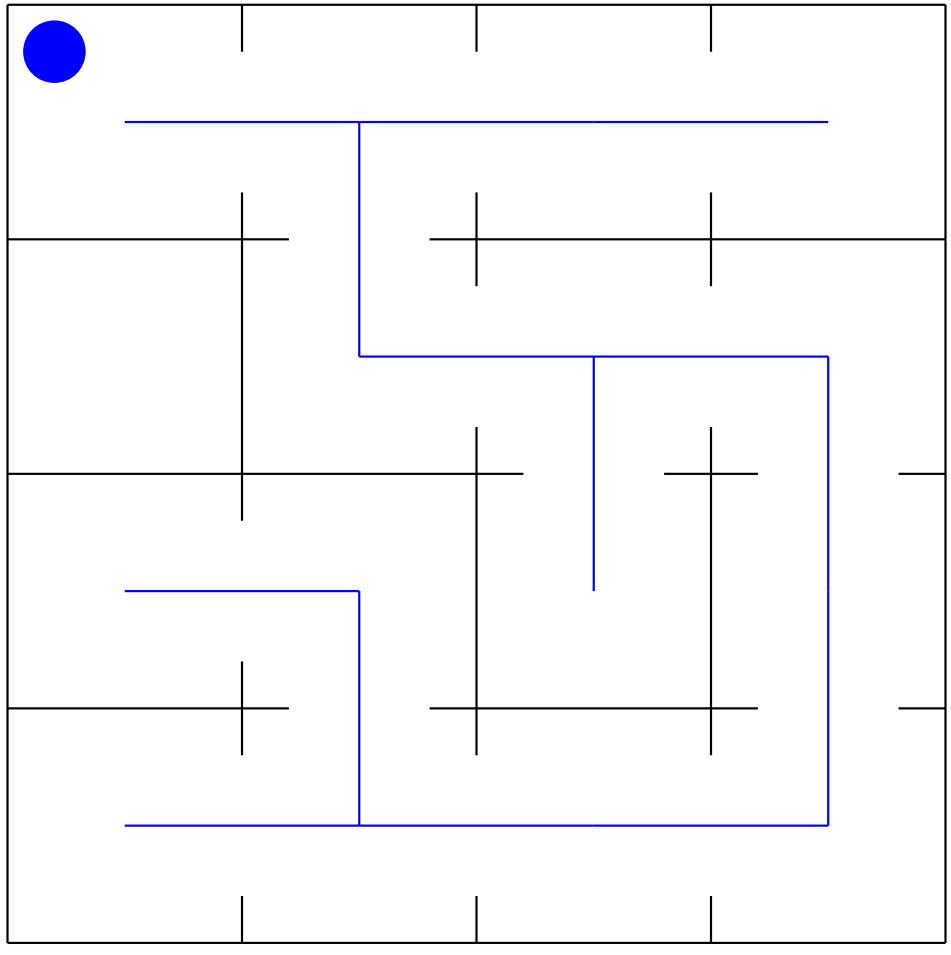


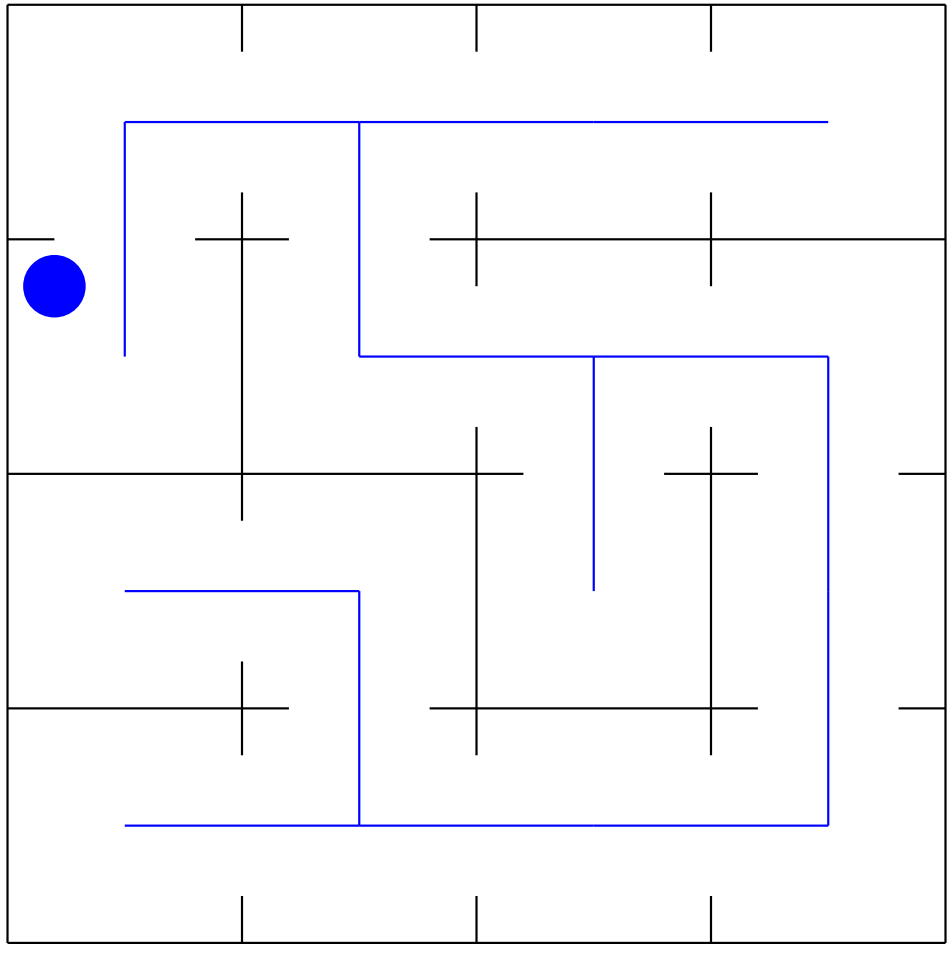


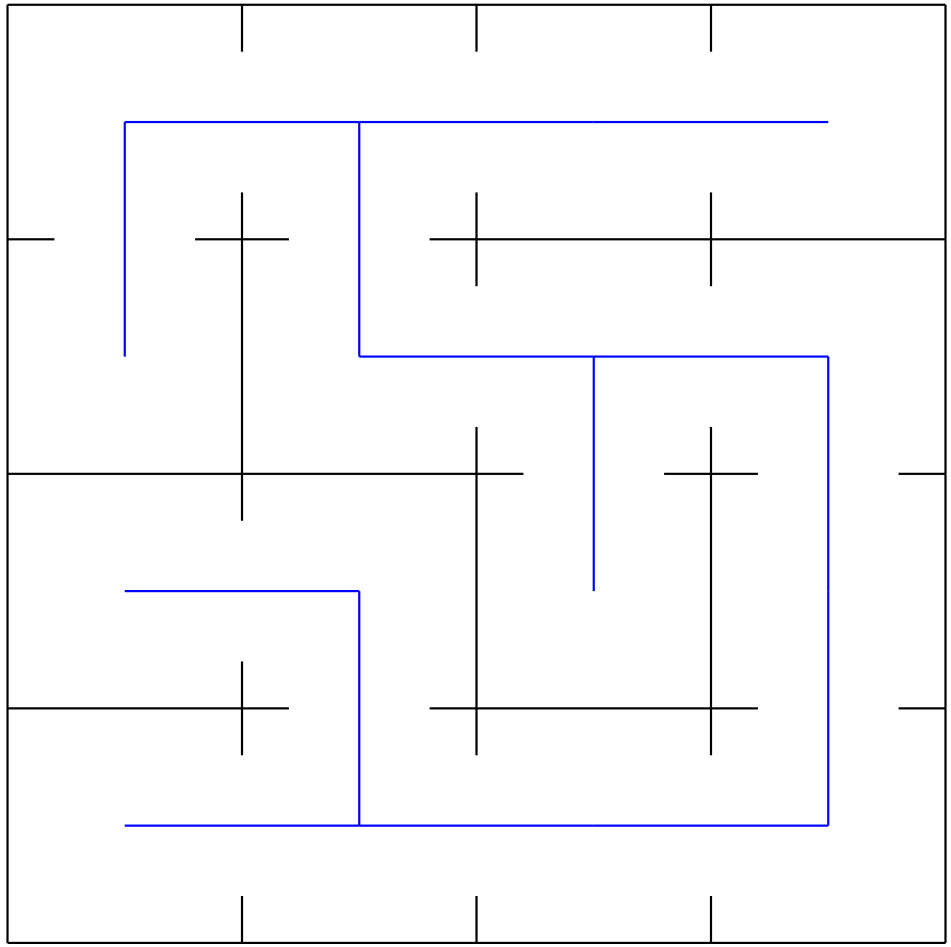






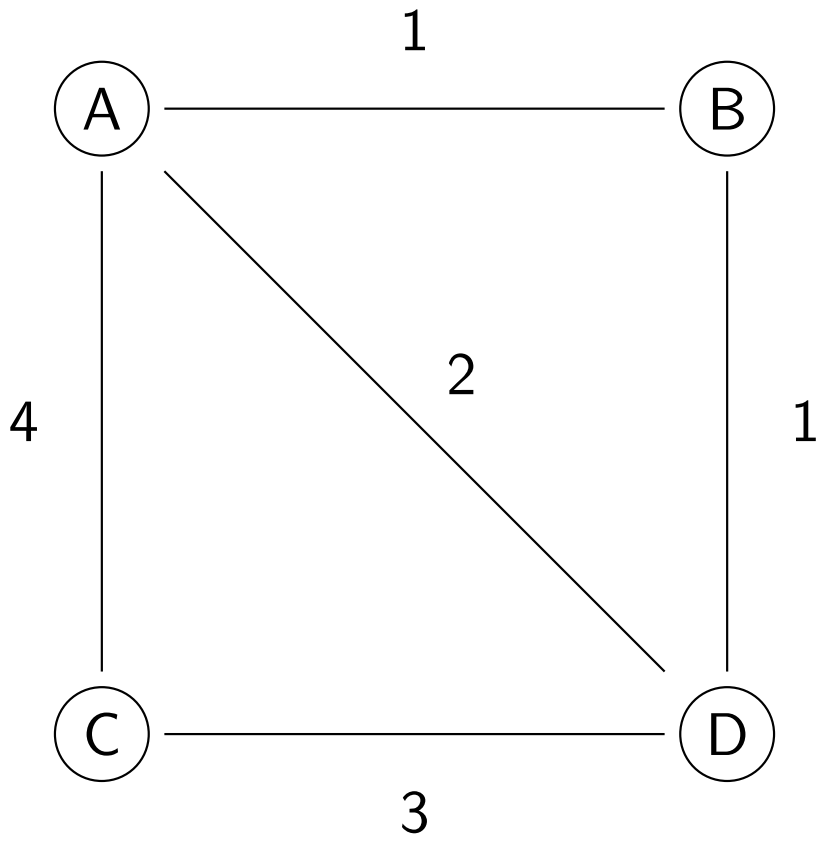




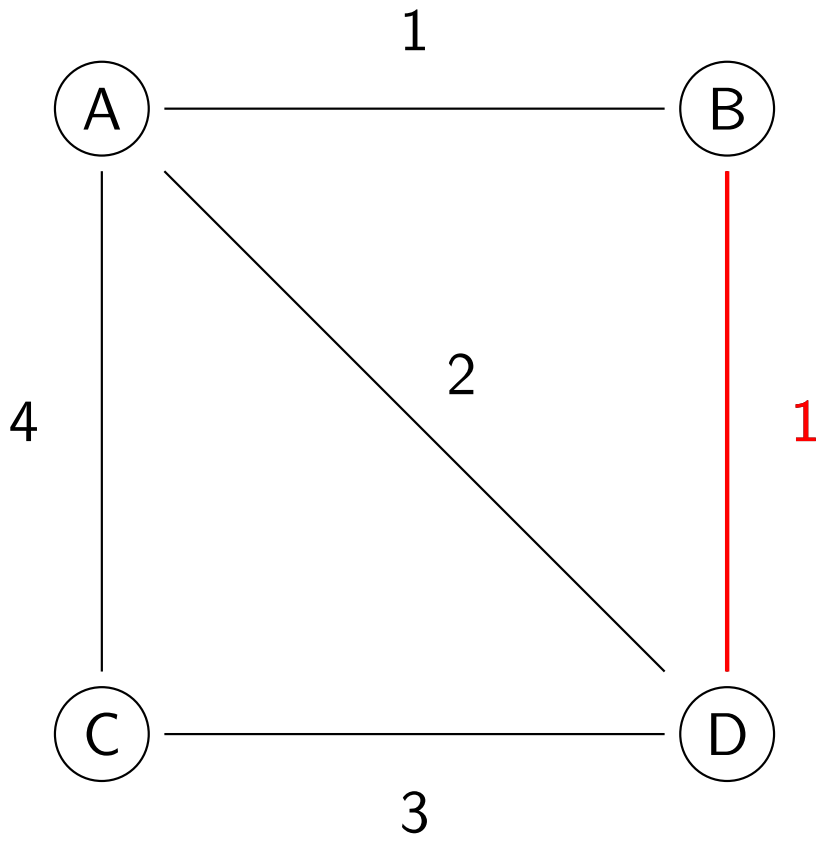


Perfect Maze!

- Minimum Spanning Tree

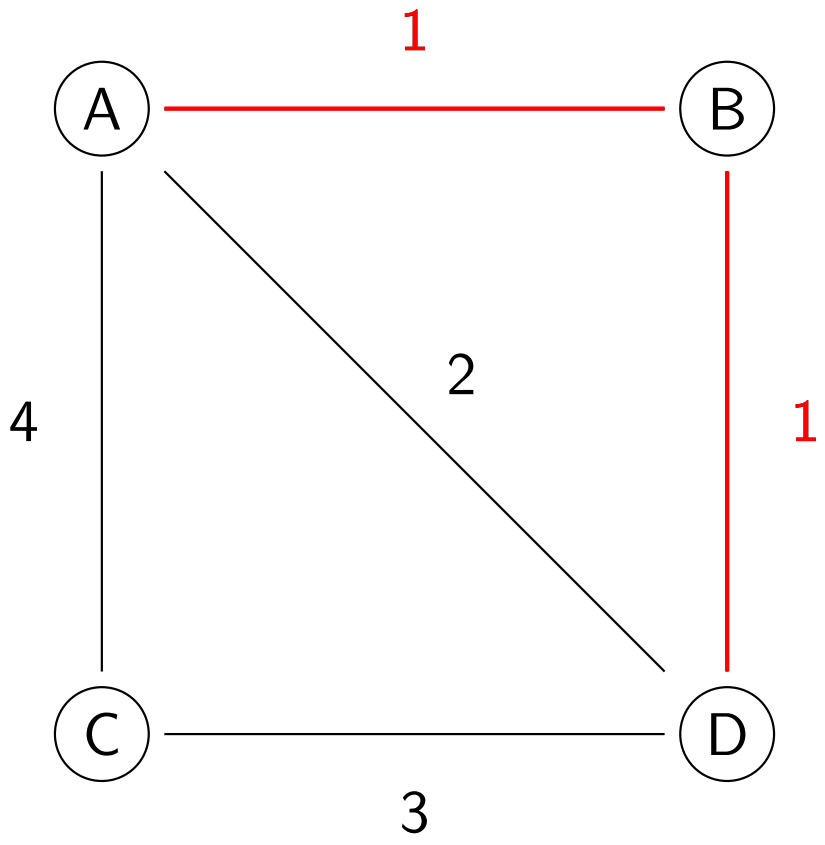


- Minimum Spanning Tree



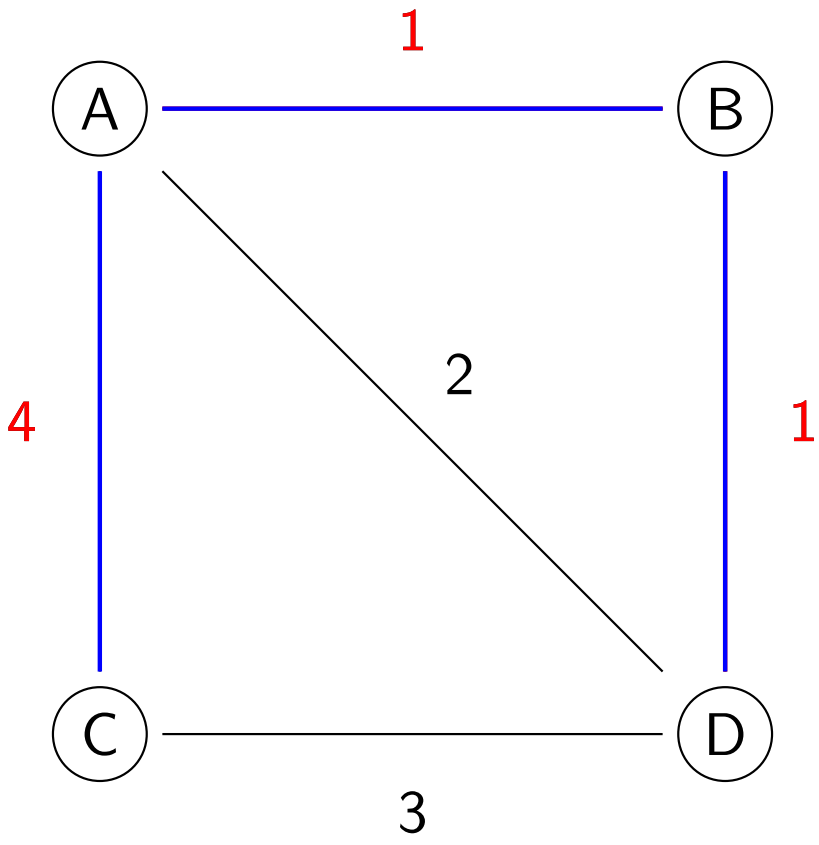
Not Spanning Tree

- Minimum Spanning Tree



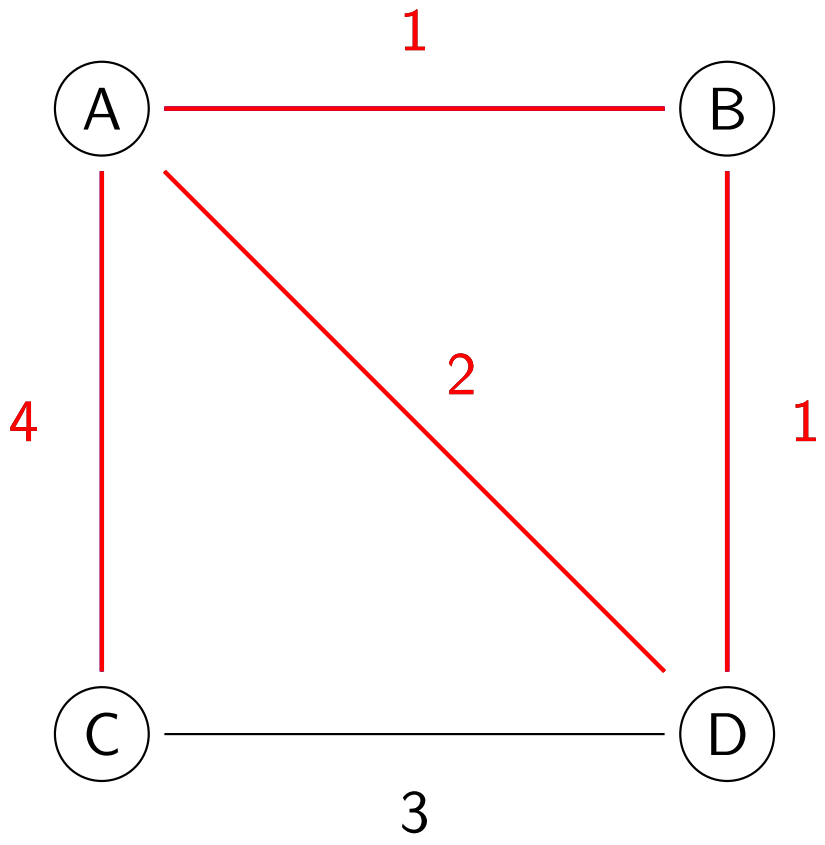
Not Spanning Tree

- Minimum Spanning Tree



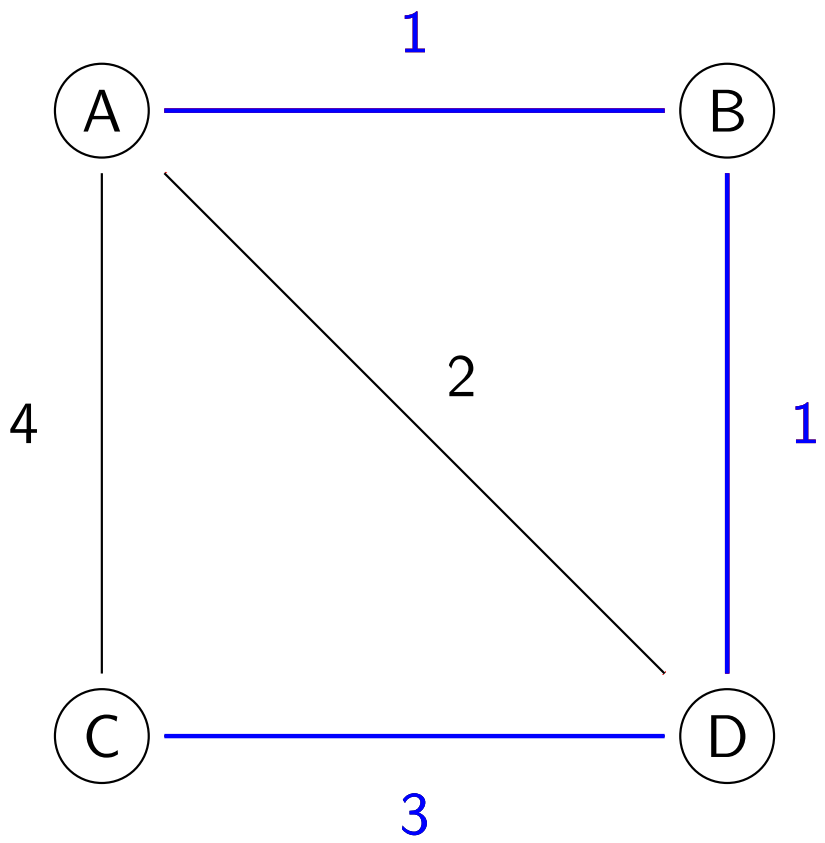
Spanning Tree
With Not Minimum Weights

- Minimum Spanning Tree



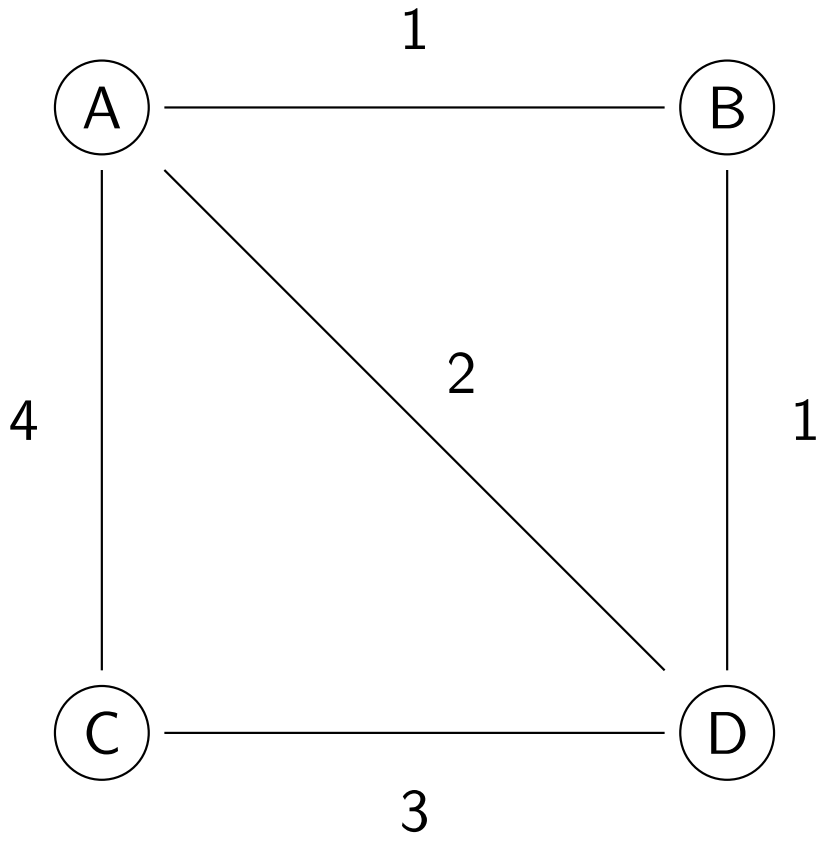
Not Spanning Tree(Cycle Exists)

- Minimum Spanning Tree

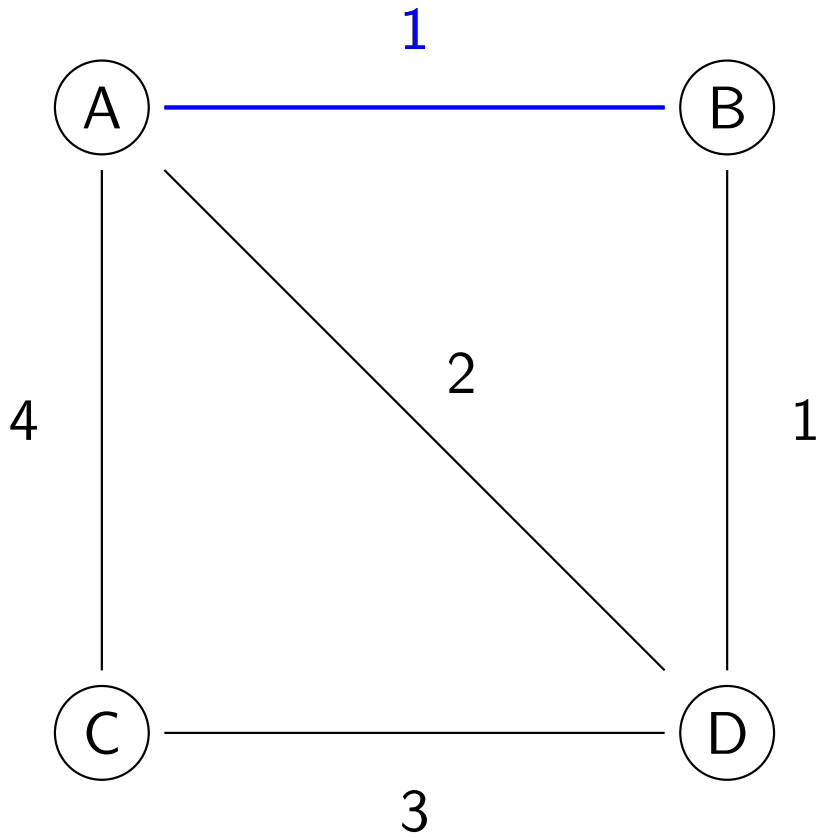


Spanning Tree
With Minimum Weights

- Kruskal's Algorithm : Finding Minimum Spanning Tree



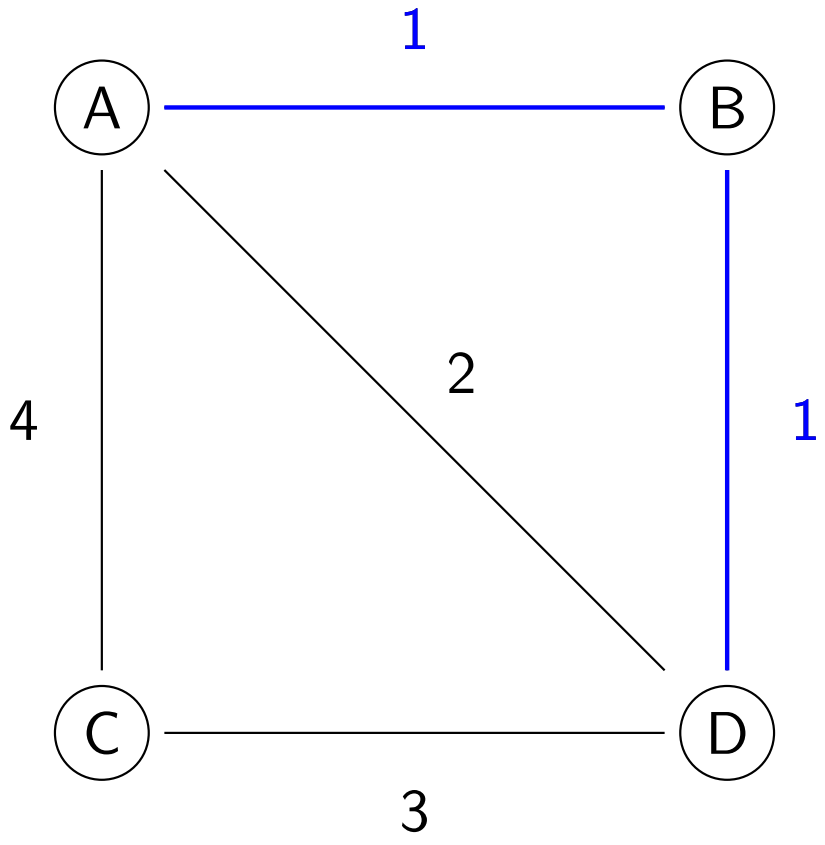
- Kruskal's Algorithm : Finding Minimum Spanning Tree



Subgraph S

1

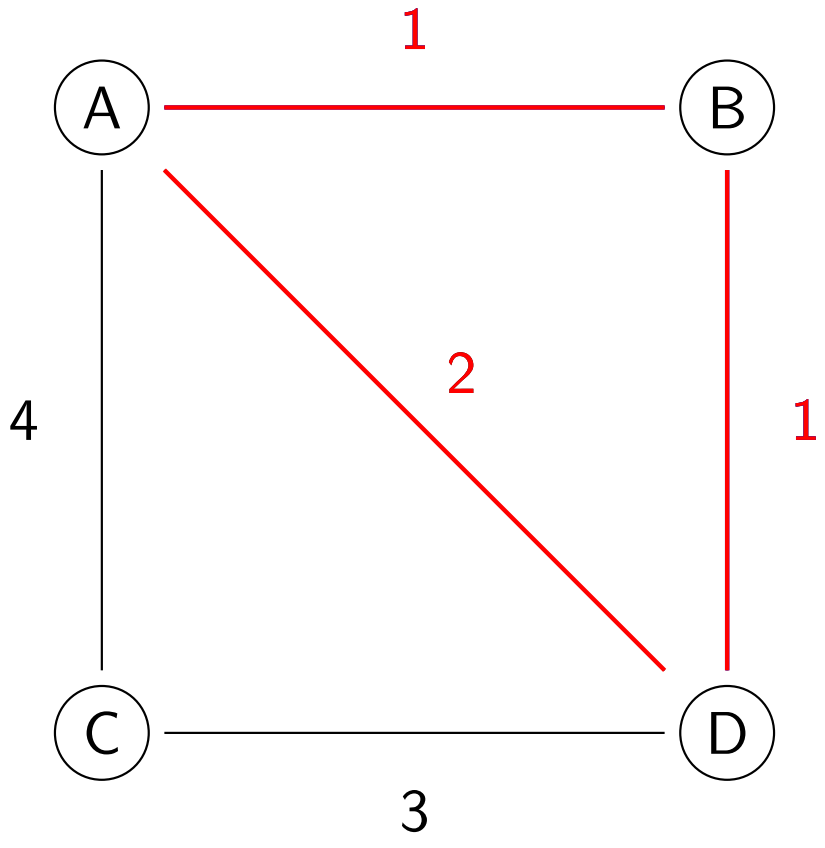
- Kruskal's Algorithm : Finding Minimum Spanning Tree



Subgraph S

1 — 1

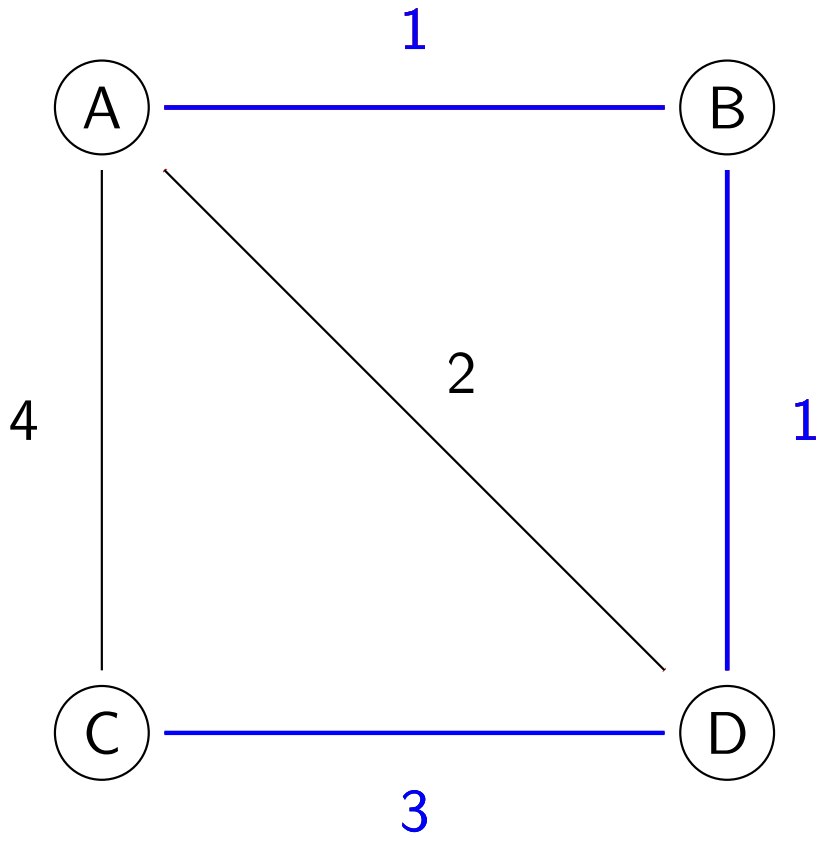
- Kruskal's Algorithm : Finding Minimum Spanning Tree



Subgraph S (Cycle Exists)

$1 \text{ —— } 1 \text{ —— } 2$

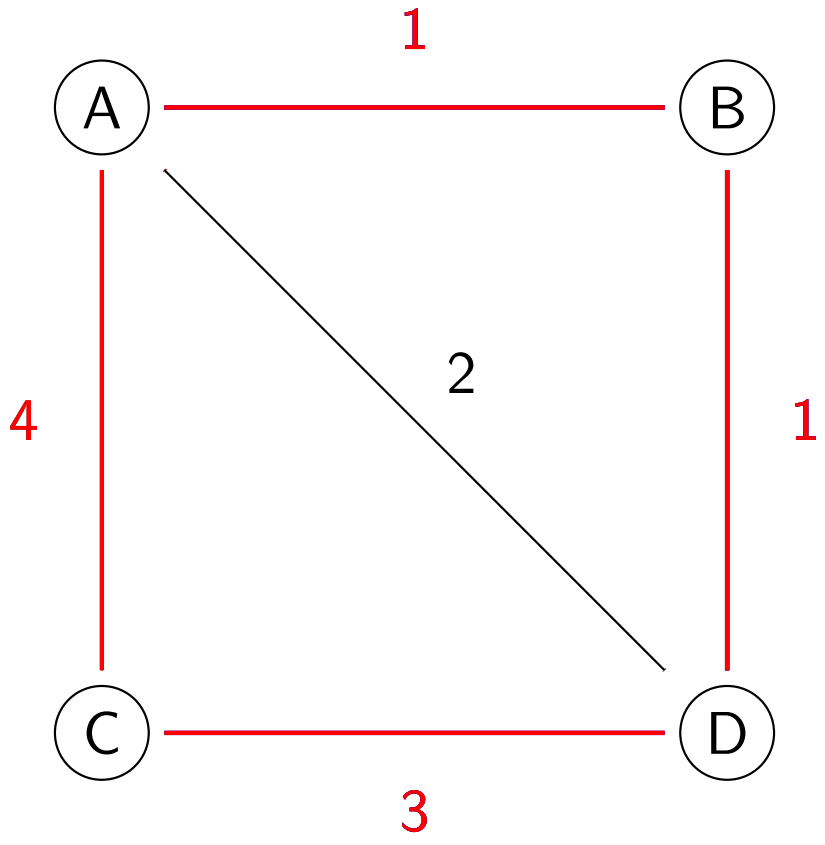
- Kruskal's Algorithm : Finding Minimum Spanning Tree



Subgraph S

1 — 1 — ~~2~~ — 3

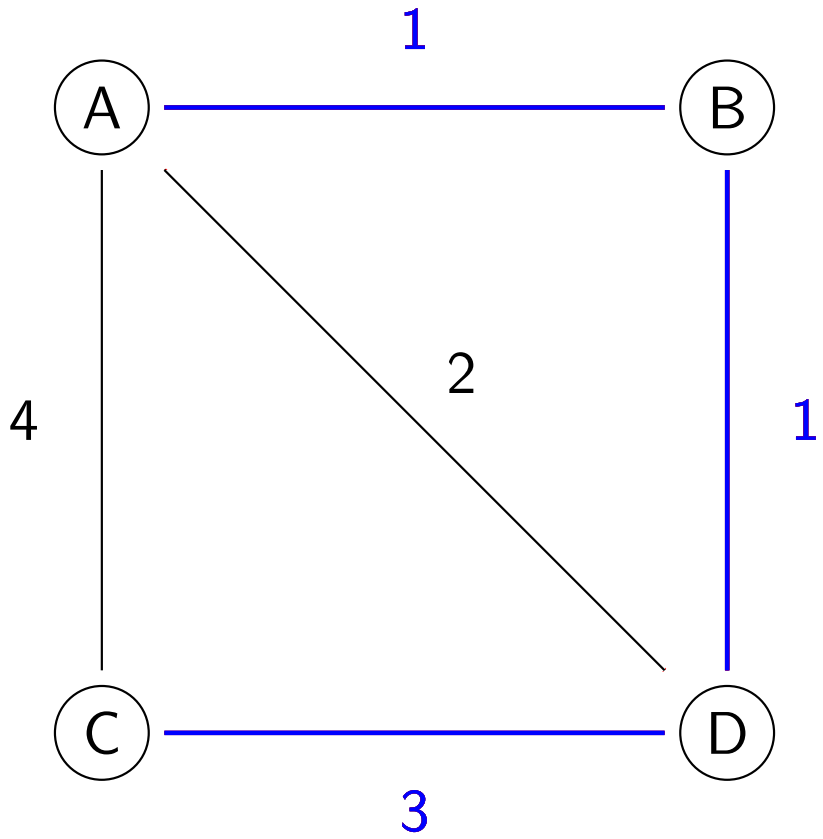
- Kruskal's Algorithm : Finding Minimum Spanning Tree



Subgraph S (Cycle Exists)

1 — 1 — ~~2~~ — 3 — 4

- Kruskal's Algorithm : Finding Minimum Spanning Tree



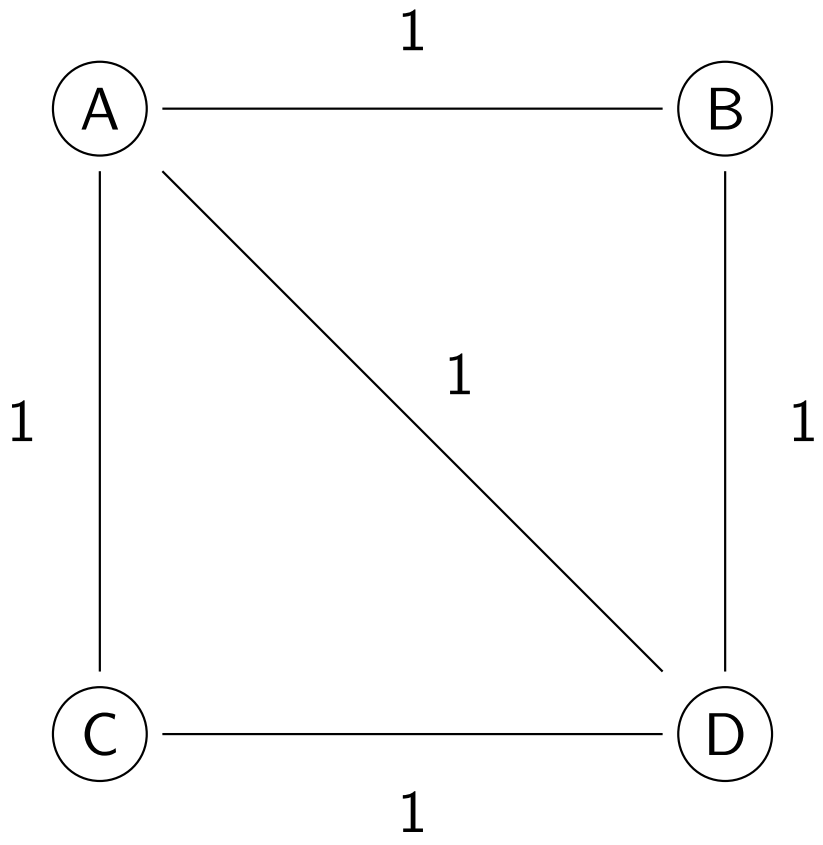
Subgraph S is minimum spanning tree!

$1 \text{ —— } 1 \text{ —— } \cancel{2} \text{ —— } 3 \text{ —— } \cancel{4}$

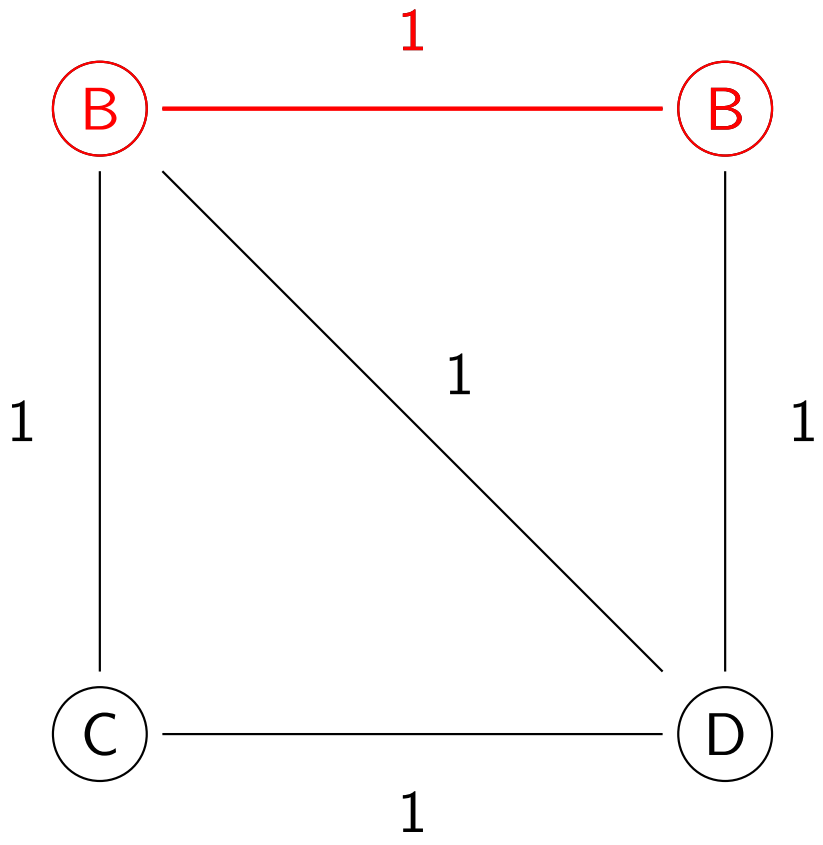
- If all weights are same, we can ignore order of the weights.

- If all weights are same, we can ignore order of the weights.
- Just check a node makes cycles or not, for random order.

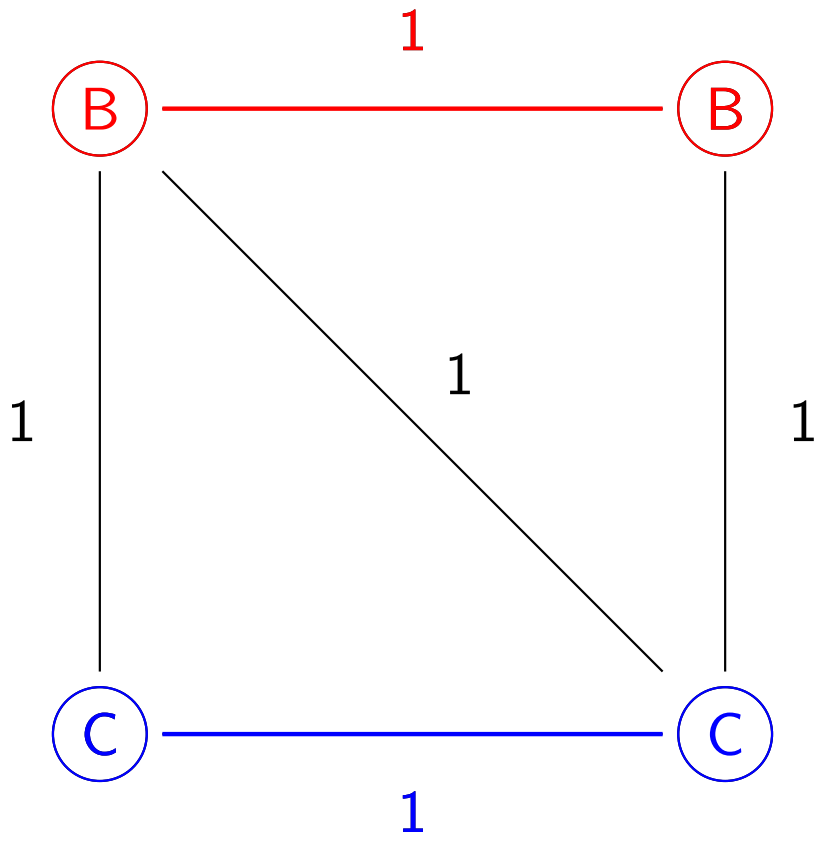
- 'Randomized' Kruskal's Algorithm



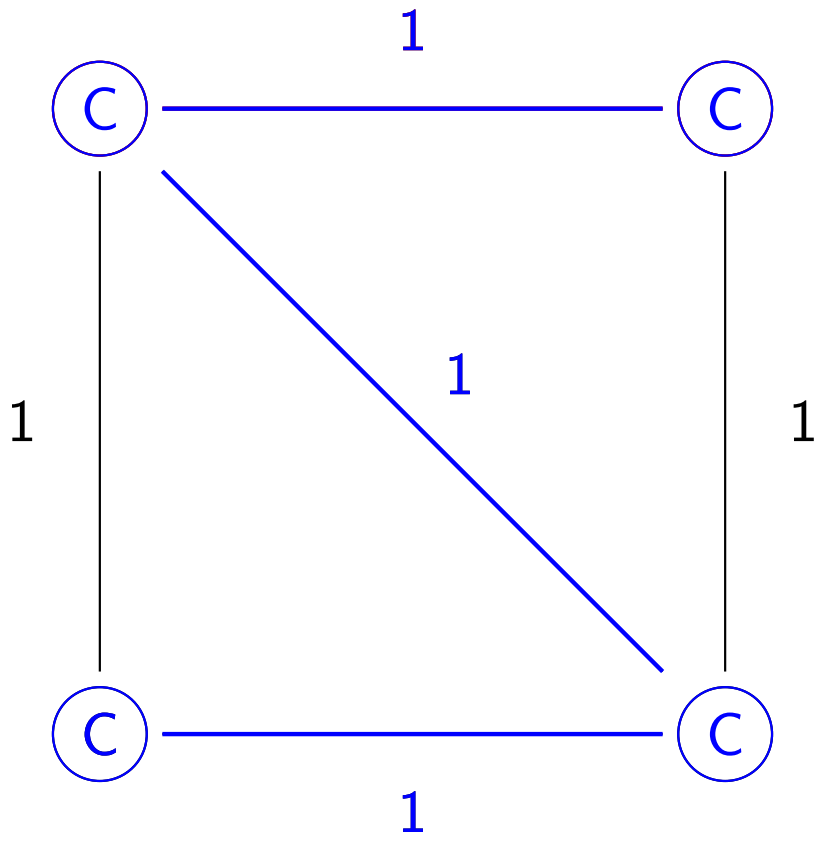
- 'Randomized' Kruskal's Algorithm



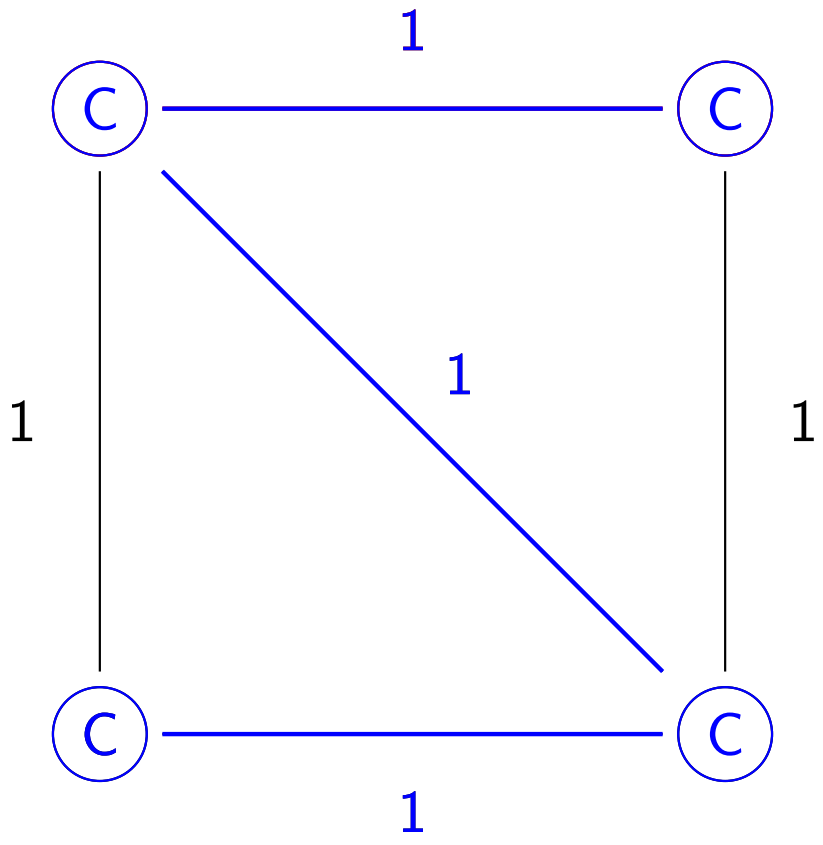
- 'Randomized' Kruskal's Algorithm



- 'Randomized' Kruskal's Algorithm

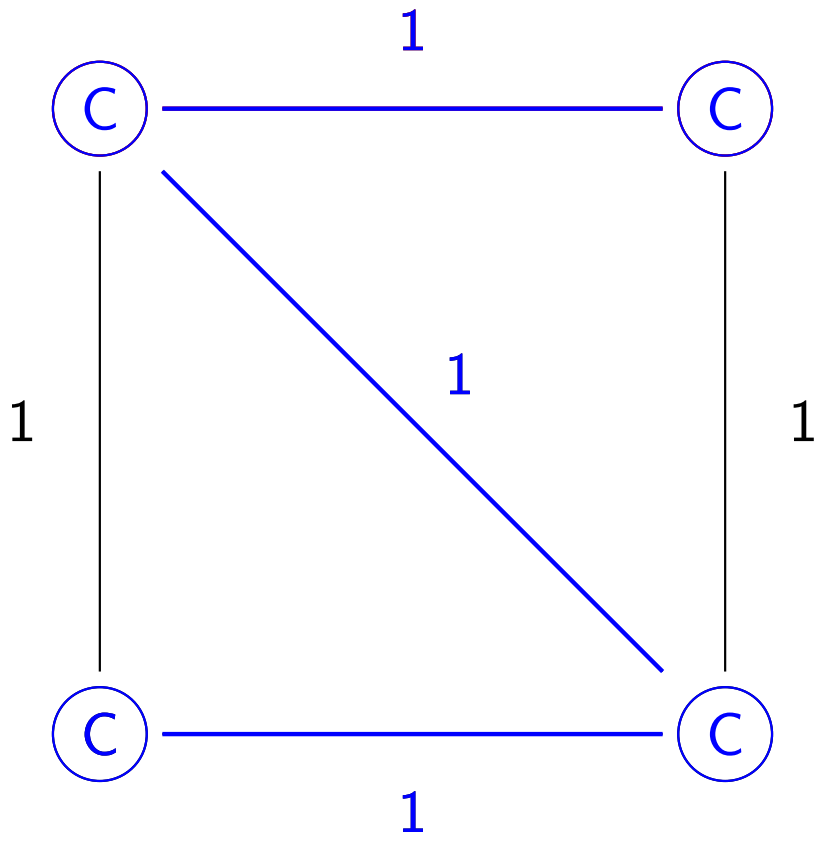


- 'Randomized' Kruskal's Algorithm

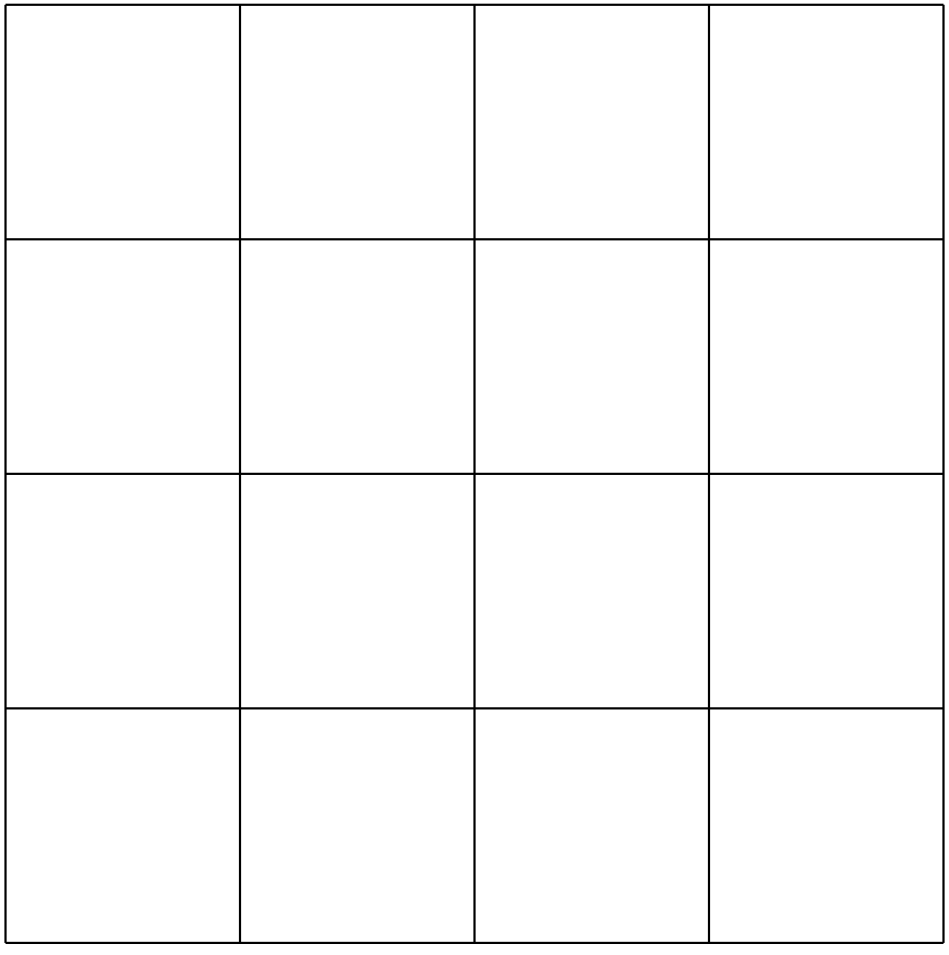


Minimum Spanning Tree

- 'Randomized' Kruskal's Algorithm

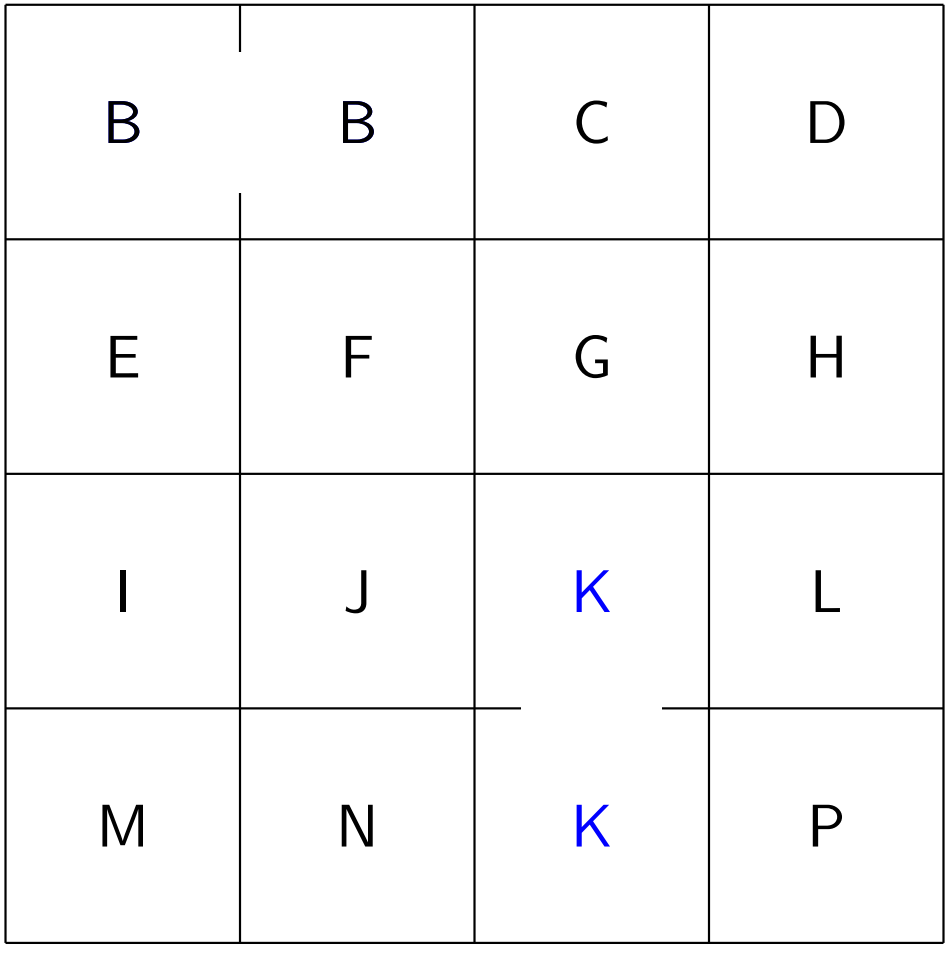


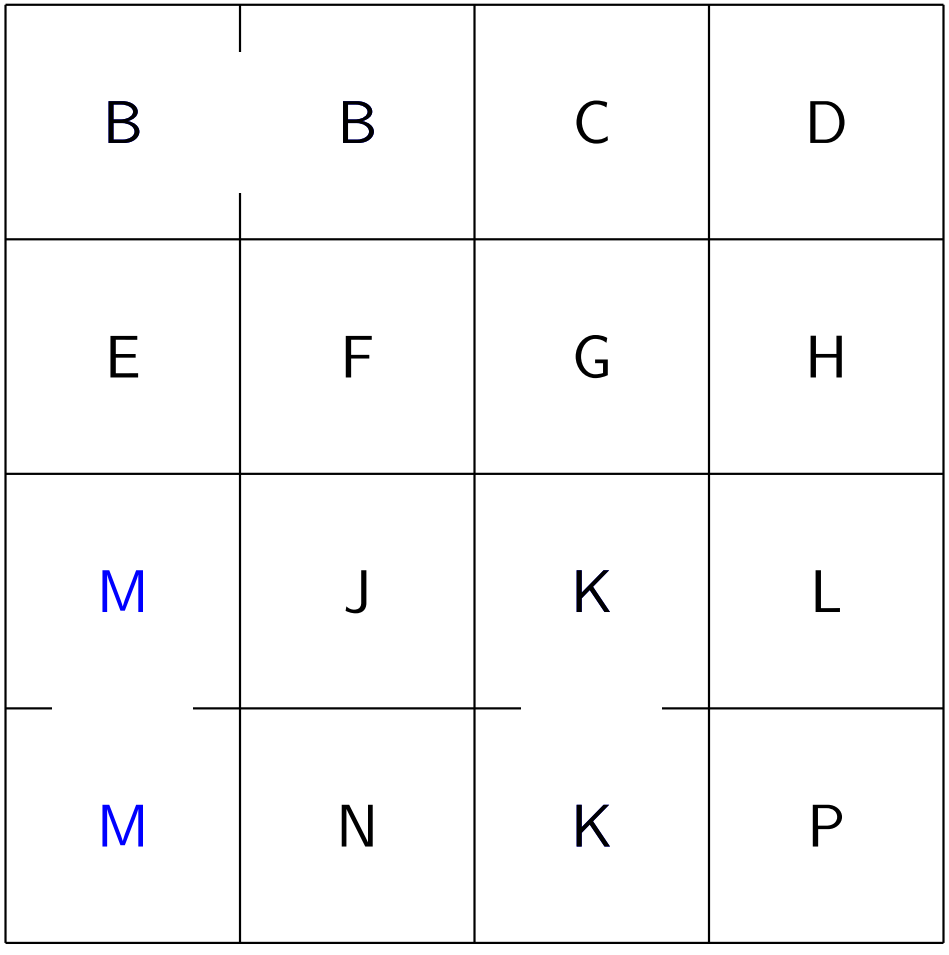
Minimum Spanning Tree
Also 'Perfect' Graph!

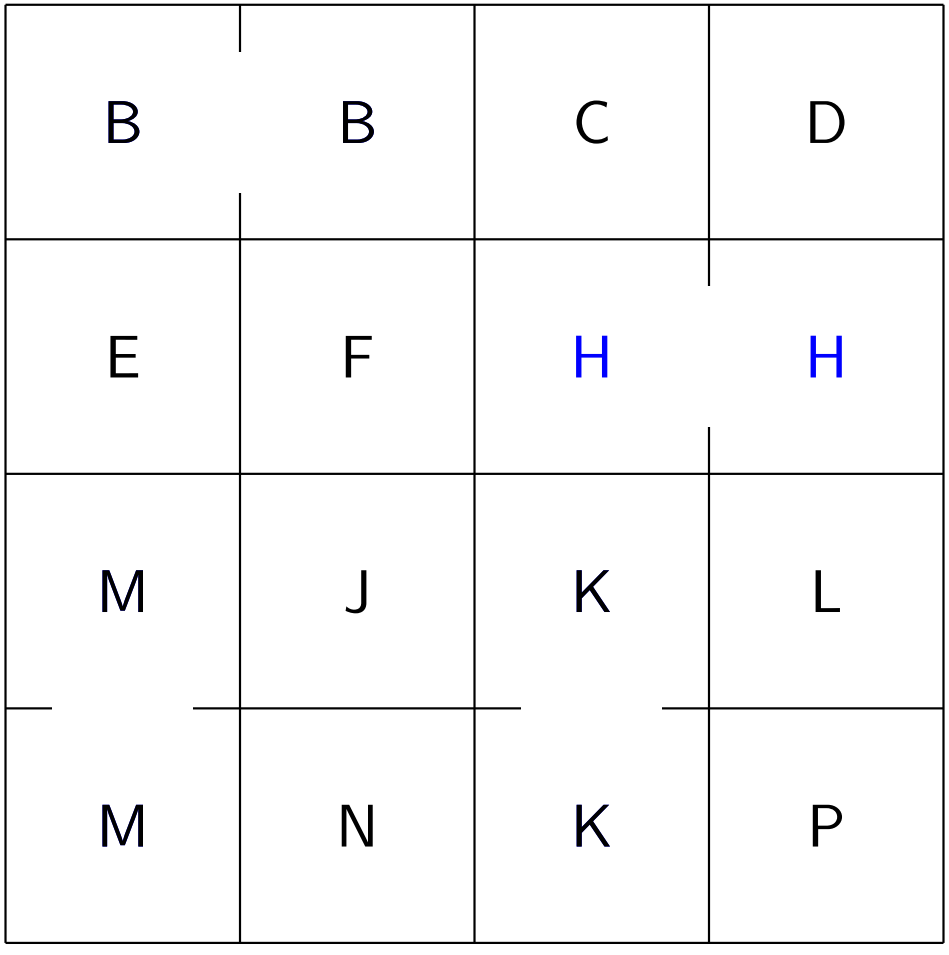


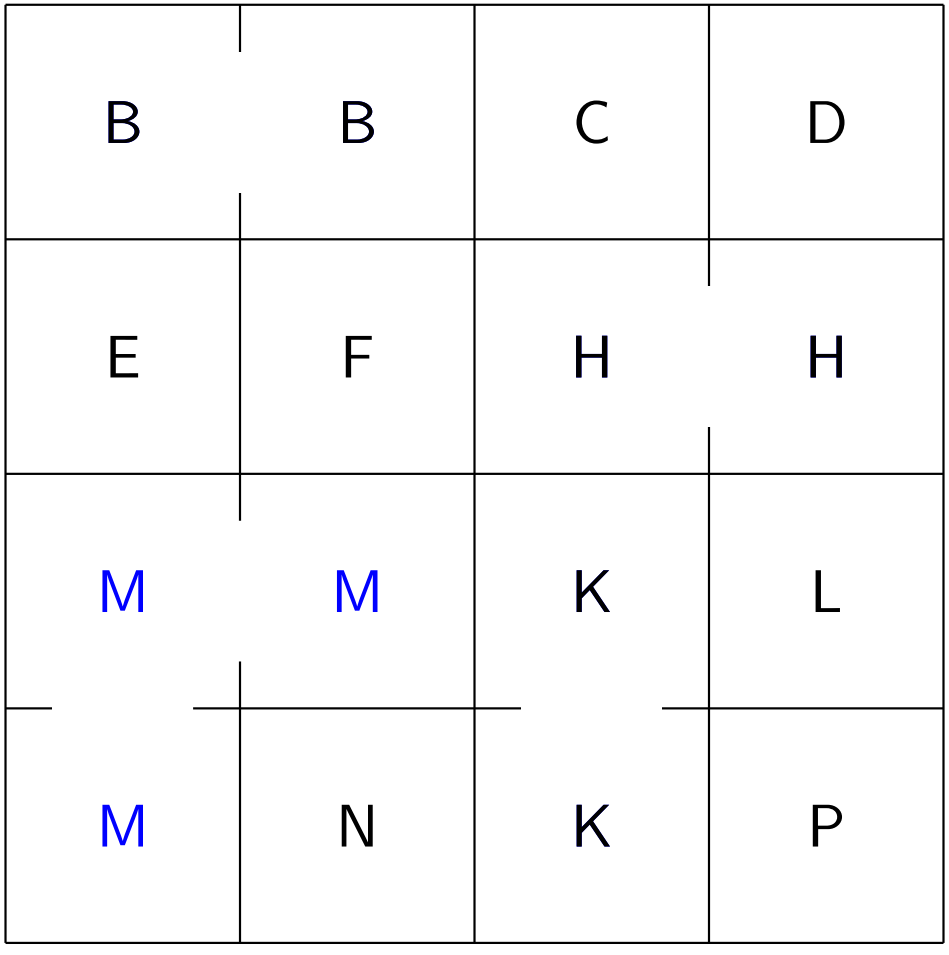
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

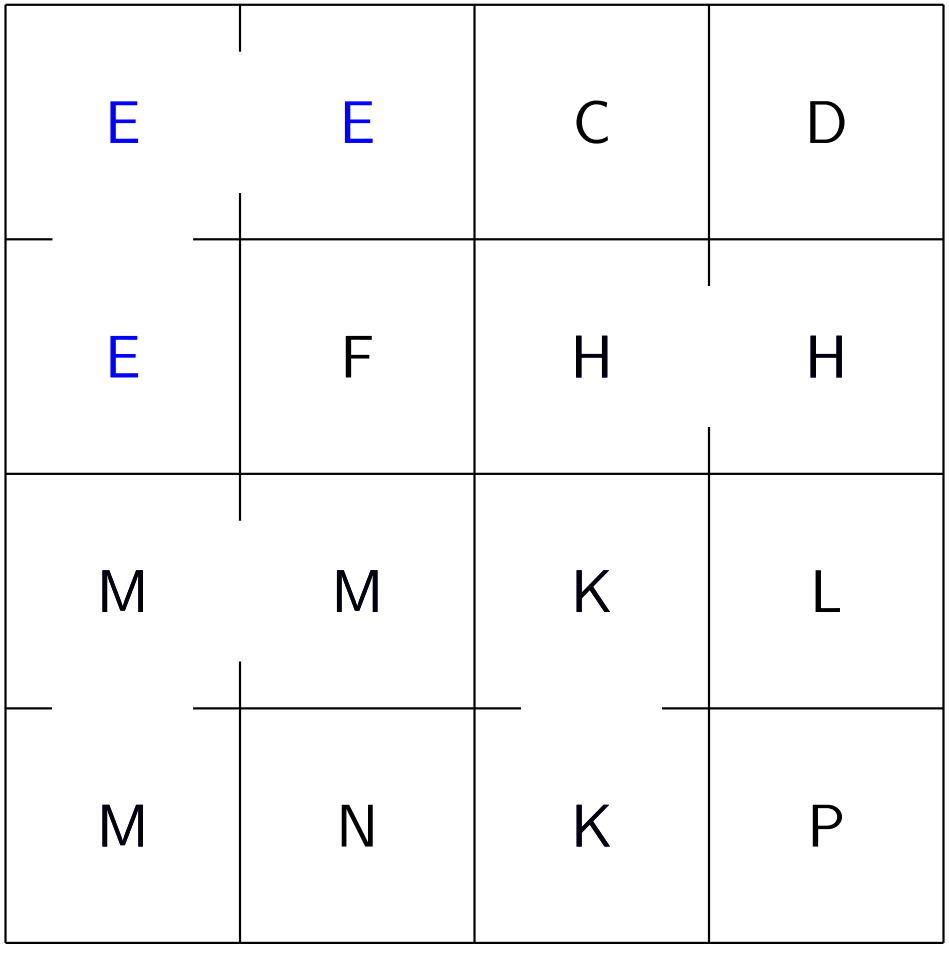
B	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

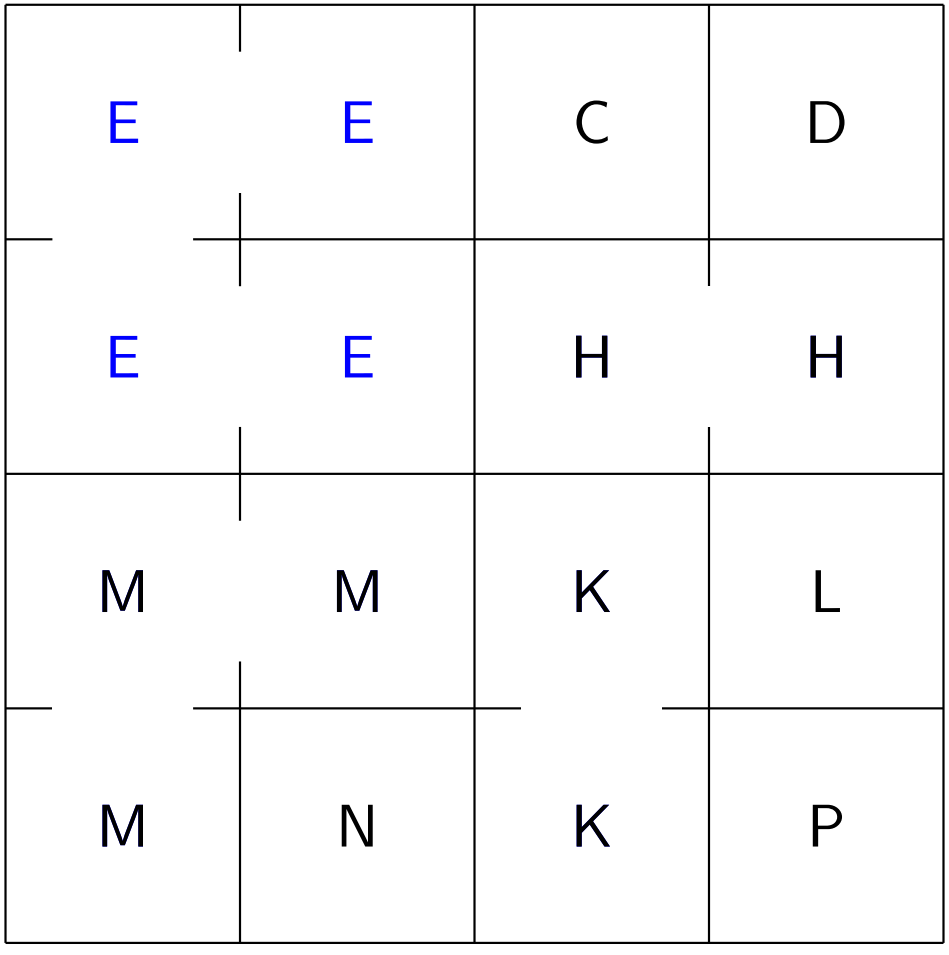

B
B
C
D
E
F
G
H
I
J
K
L
M
N
K
P

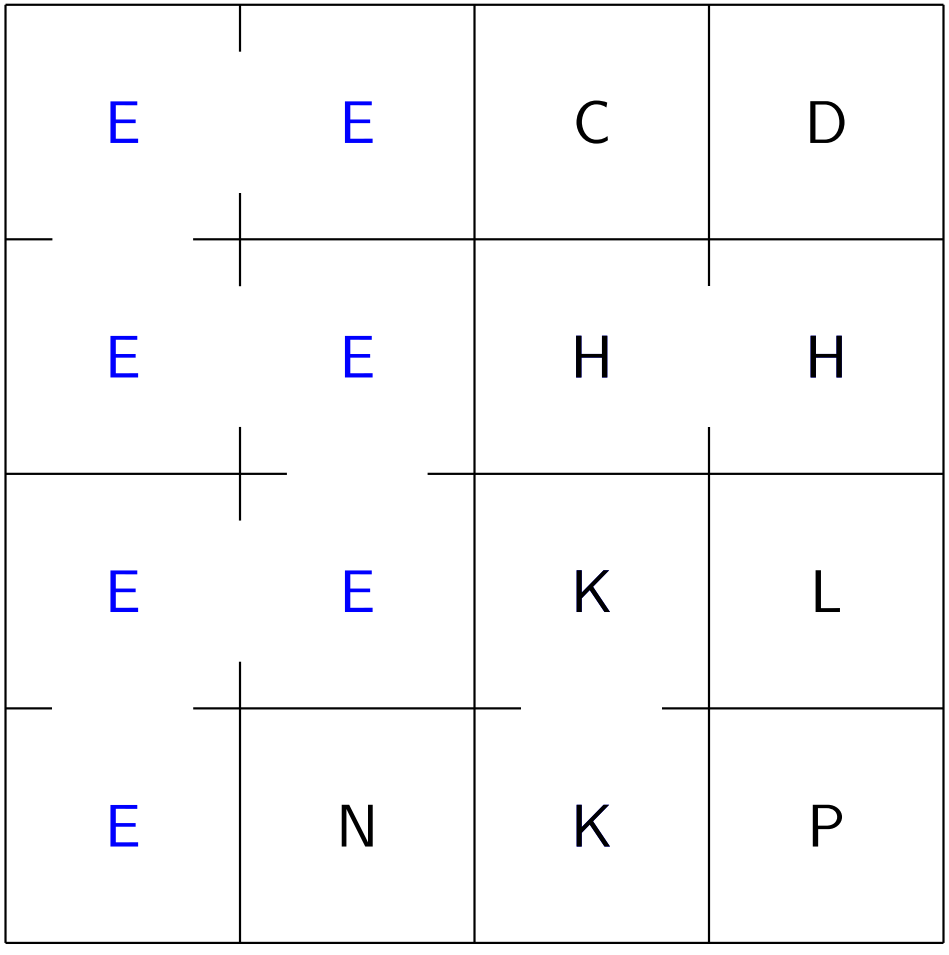

B
B
C
D
E
F
G
H
M
J
K
L
M
N
K
P

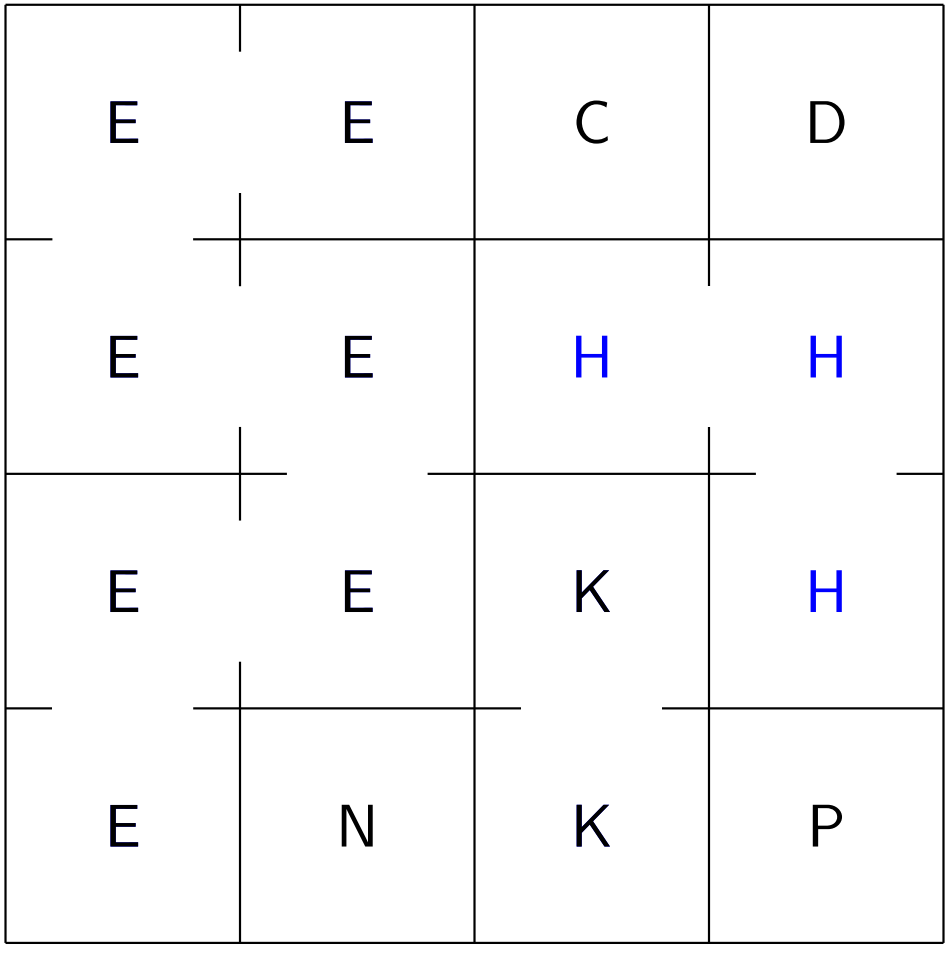

B
B
C
D
E
F
H
H
M
J
K
L
M
N
K
P

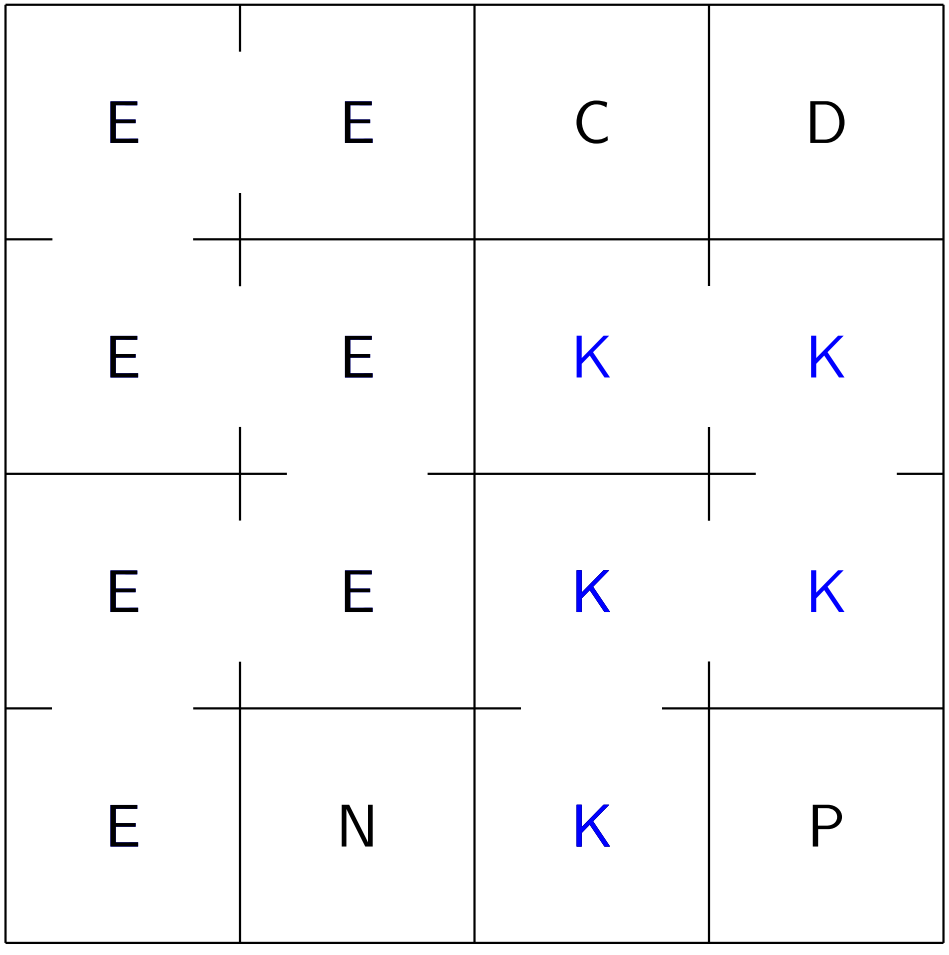

B
B
C
D
E
F
H
H
M
M
K
L
M
N
K
P

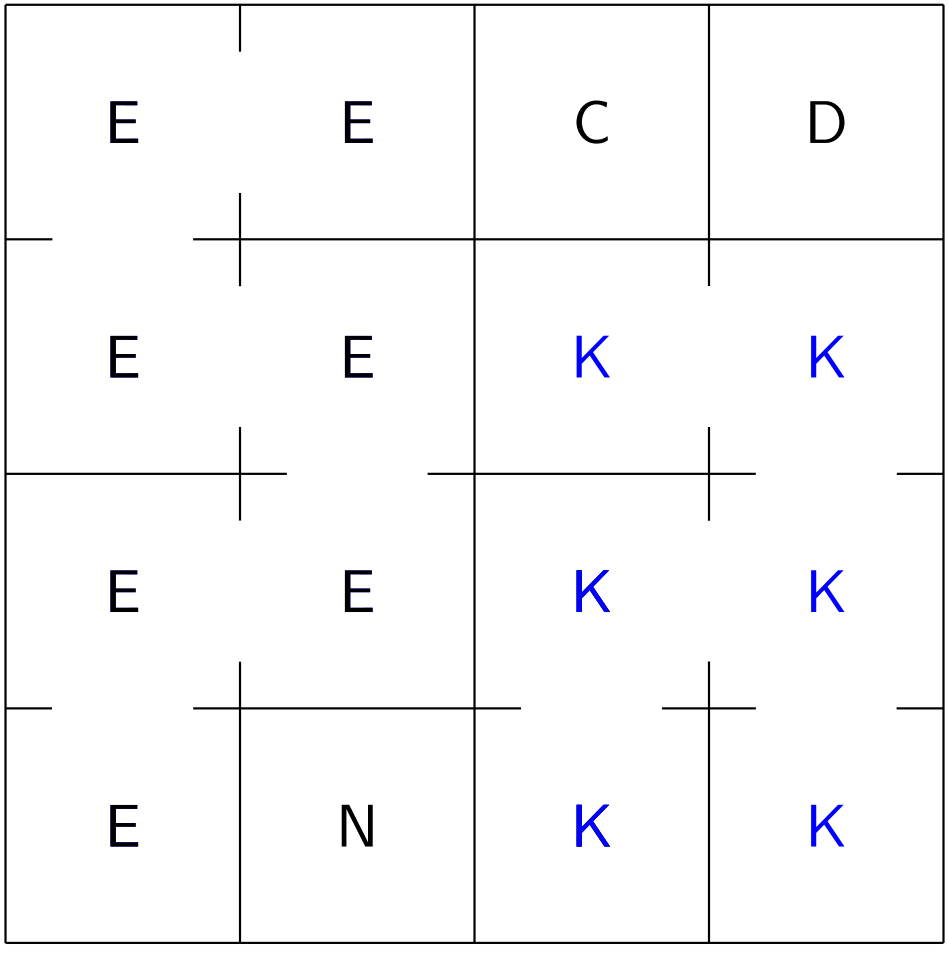

E
E
C
D
E
F
H
H
M
M
K
L
M
N
K
P

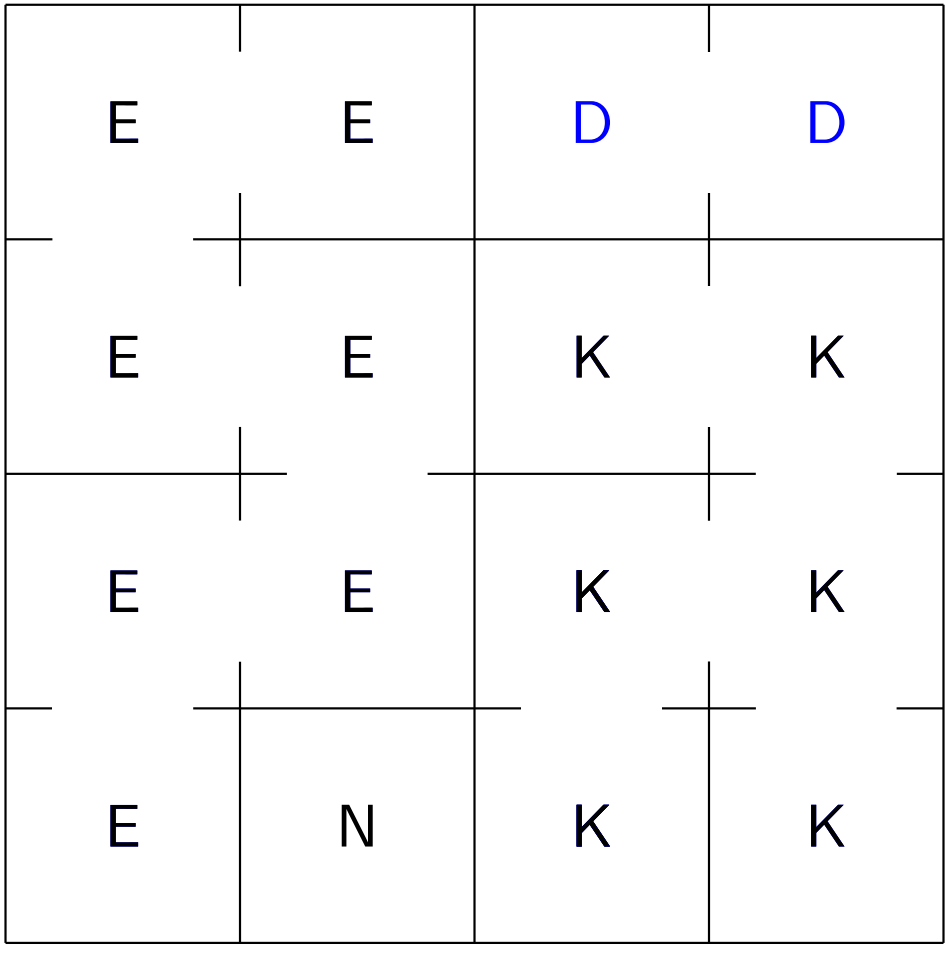

E
E
C
D
E
E
H
H
M
M
K
L
M
N
K
P

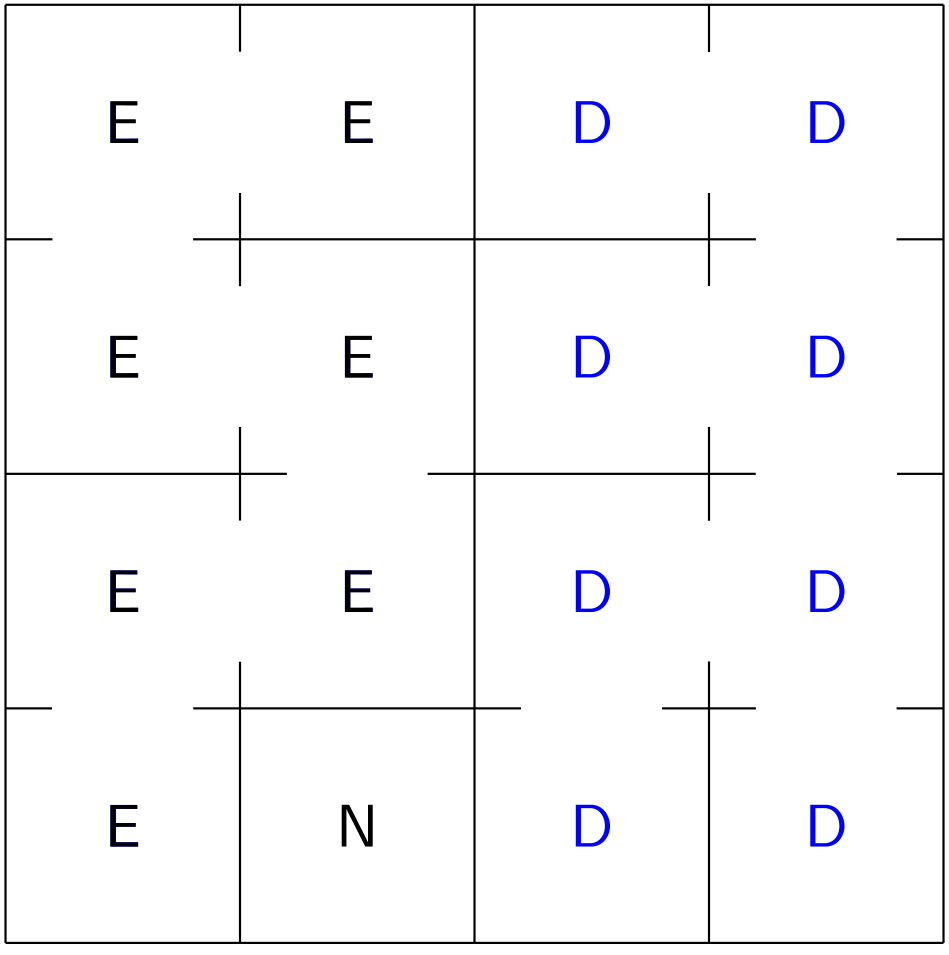

E
E
C
D
E
E
H
H
E
E
K
L
E
N
K
P

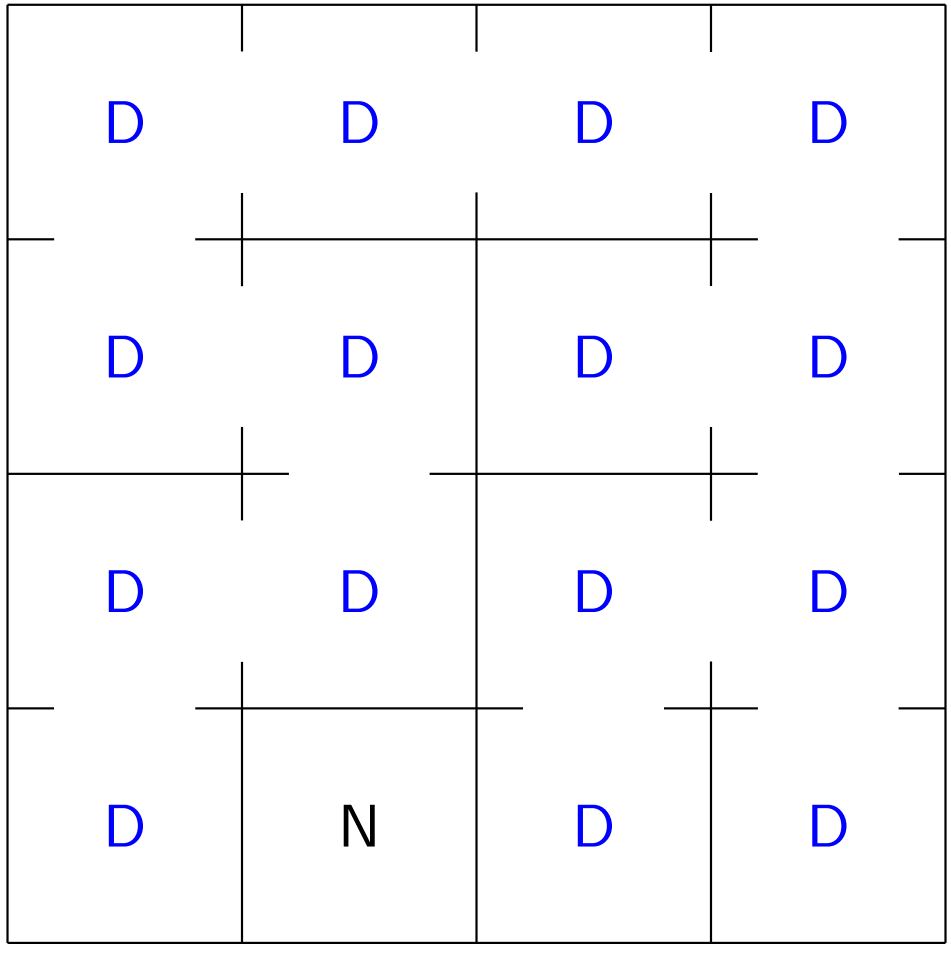

E E C D
E E H H
E E K H
E N K P

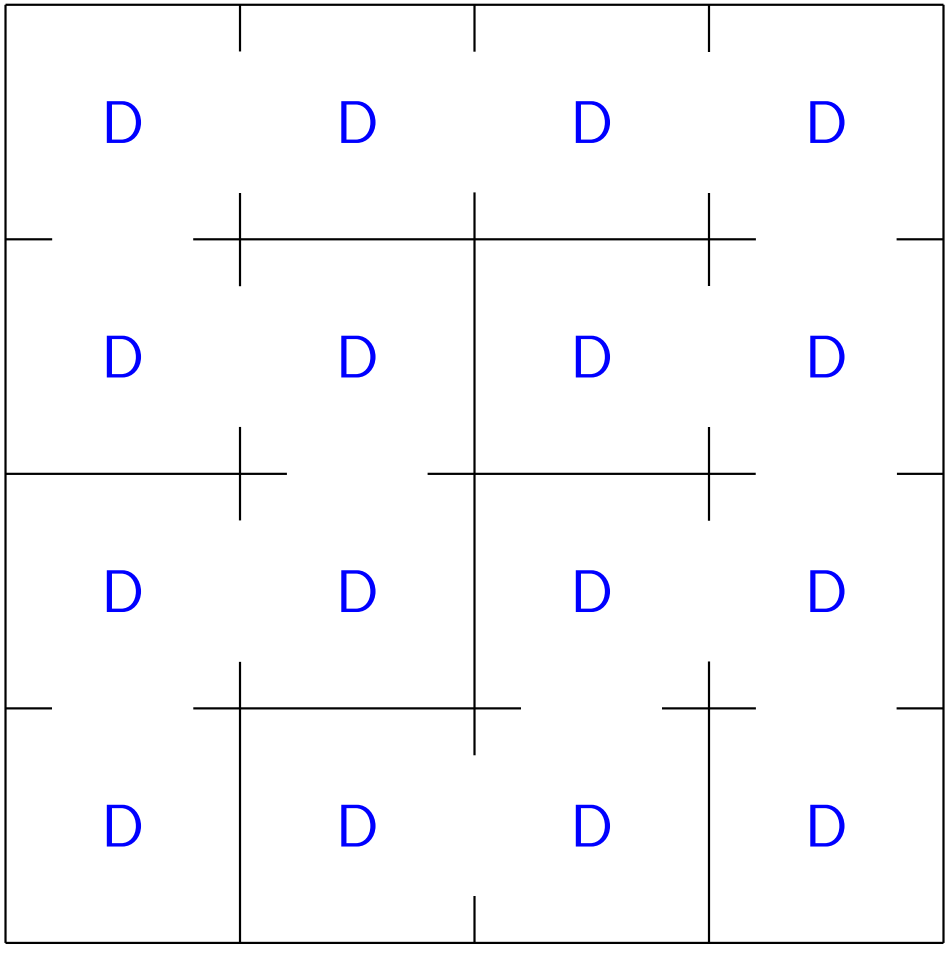

E E C D
E E K K
E E K K
E N K P

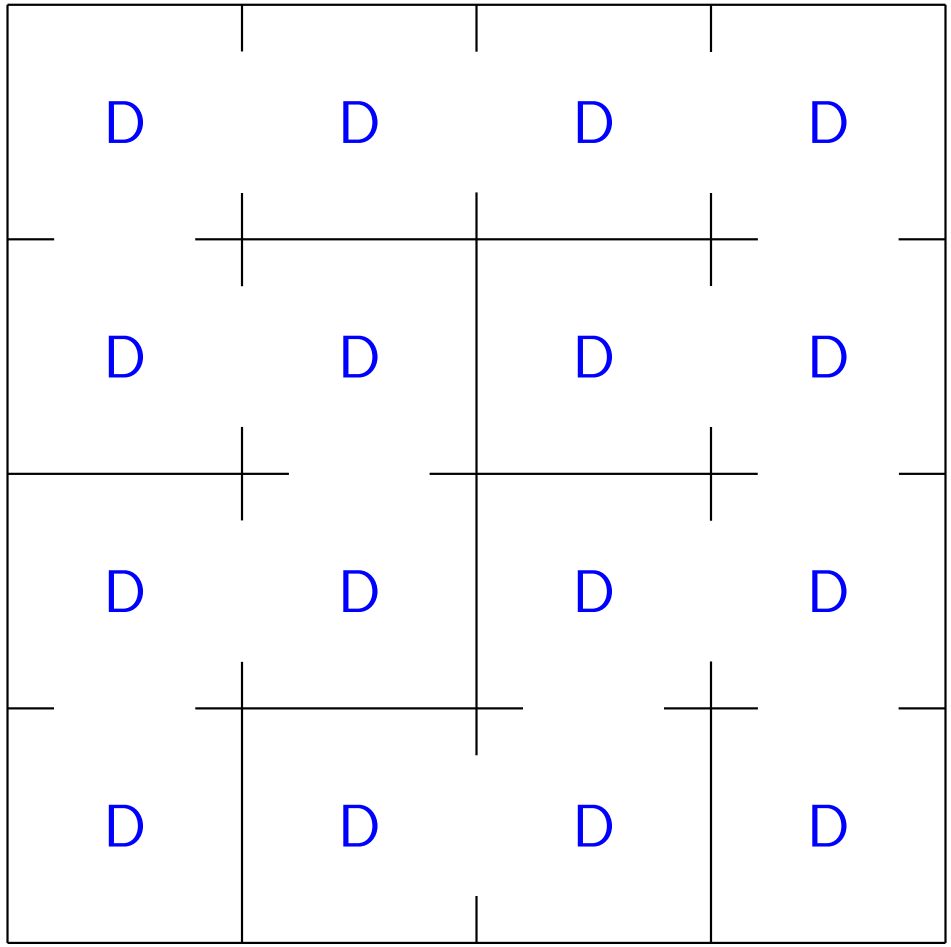

E
E
C
D
E
E
K
K
E
E
K
K
E
N
K
K


E E D D
E E K K
E E K K
E N K K


E E D D
E E D D
E E D D
E N D D


D
D
D
D
D
D
D
D
D
D
D
D
D
N
D
D


D
D
D
D
D
D
D
D
D
D
D
D
D
D
D
D



Perfect Maze!