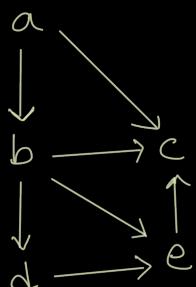
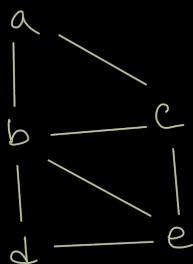


#Graph

Graph \rightarrow Nodes + Edges



Directed Graph



Undirected Graph (2 way travelling)

Travelling permission

Neighbour Node

\hookrightarrow where you can directly travel

\Rightarrow Adjency list

a \rightarrow 2 Node

d \rightarrow No node

```
{
  a : [b, c],
  b : [c, d],
  c : [d, e],
  d : [],
  e : [b]
}
```

Depth First Traversal

a

a, b

a, b, d

a

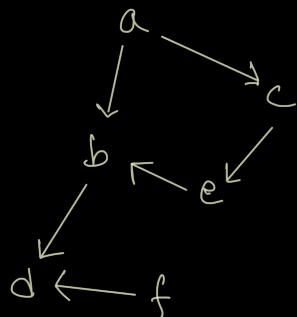
a, c

a, c, e, b, d

NO f traversal

Explore one direction

Stack



Breadth First Traversal

a

a, b

a, b, c

Explore in all direction
(circular exploration)

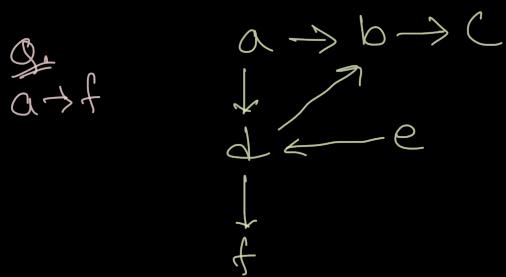
Queue

Solutions

Explore one direction

Stack

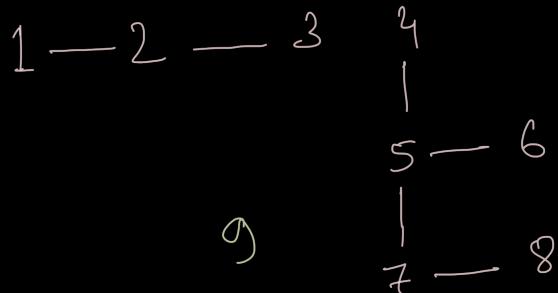
⇒ Directed Path



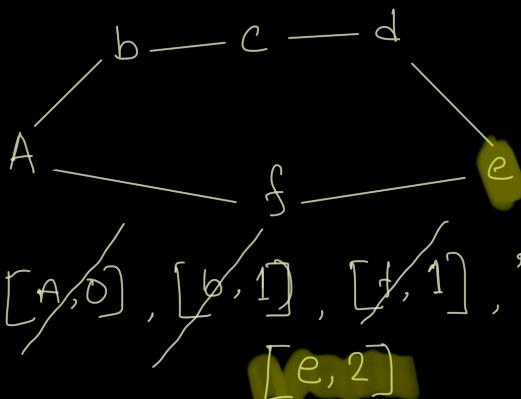
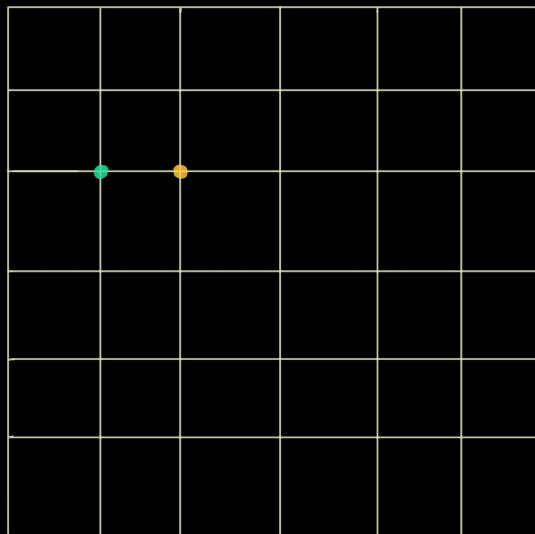
Explore all direction
(Circular exploration)

Queue

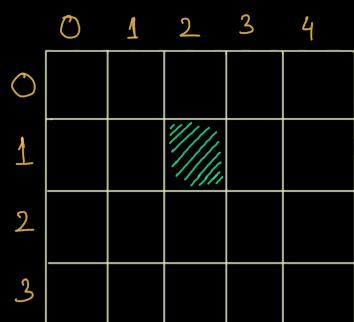
♯ Connected Island / Component



⇒ Shortest Path



$Q = [A, 0], [b, 1], [f, 1], [c, 2], [e, 2]$



General Statement

$(R-1, C)$
 $(R, C-1)$ (R, C) $(R, C+1)$
 $(R+1, C)$

→ Total no. of

Row = 3

Column = 4

→ Point $(R, C) = (1, 2)$

Top = $(0, 2)$

Bottom = $(2, 2)$

Left = $(1, 1)$

Right = $(1, 3)$

Out of Bound Check

	0	1	2	3	4
0					
1					
2					
3					

grid = [[],[],[],[]]

grid.length = 4

Row → $0 \leq \text{Row} < \text{grid.length}$

Column → $0 \leq \text{Column} < \text{grid[0].length}$

```
const grid1 = [
  [1,0,0,1,0],
  [1,0,0,1,1],
  [0,1,0,1,0],
  [0,0,0,0,0],
  [0,0,1,1,1]
]
```

```
const grid = [
  ['L','W','W','L','W'],
  ['L','W','W','L','L'],
  ['W','L','W','L','W'],
  ['W','W','W','W','W'],
  ['W','W','L','L','L']
]
```