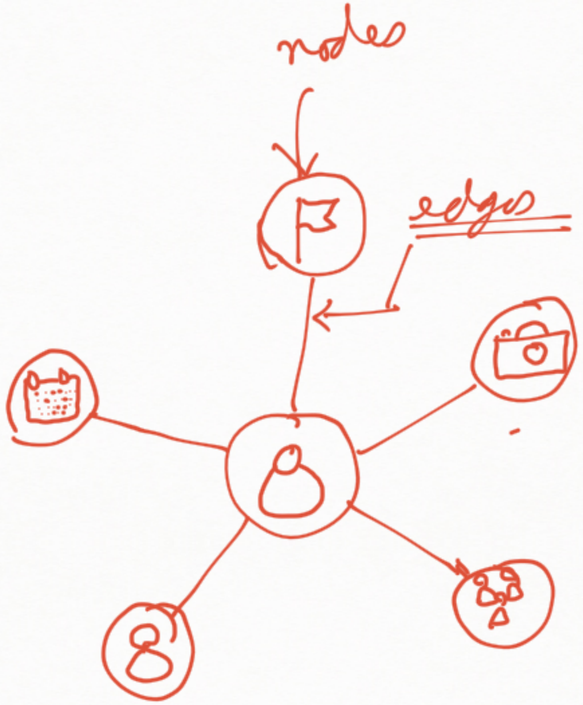
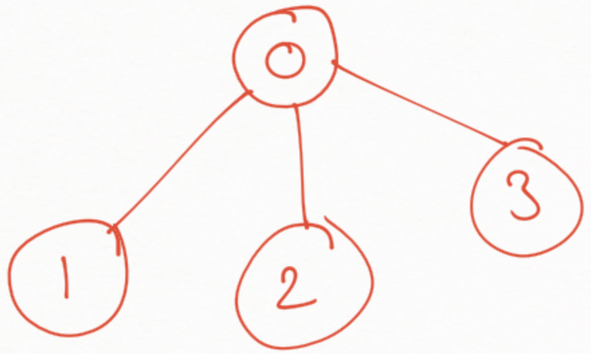
Graph Data Structure:

Facebook



Anything that has data is a node.

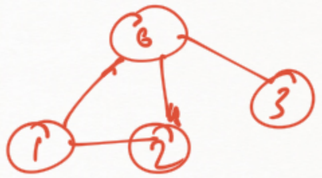


Vertices : {0, 1, 2, 3}
Edges : {(0-1), (0-2), (0-3)}
Graph : {V, E}

a graph is a DS that consists of

- A collection of vertices V
- A collection of edges E

Graph Terminology:



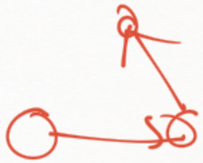
- Adjacency: A vertex is said to be adjacent to another vertex if there is an edge.

  ex: {(0,1), (1,2), (0,2), (0,3)}

  non adjacent vertices: {(2,3), (1,3)}

- Path: A sequence of edges that allows us to go from vertex A to vertex B is called a path

  0-1, 1-2 and 0-2 are paths of vertex 0 - vertex 2



- Directed Graph:

Graph Representation

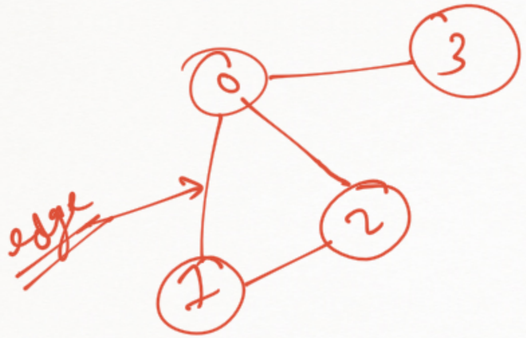
1) Adjacency Matrix

2) Adjacency List

3 - 2 =

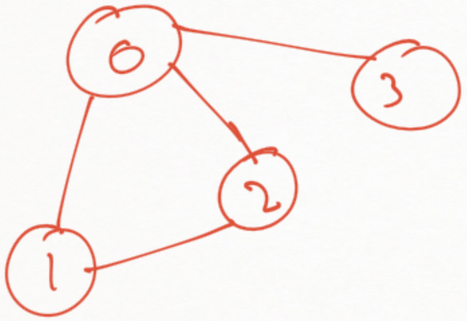
1) AM (adjacency Matrix)

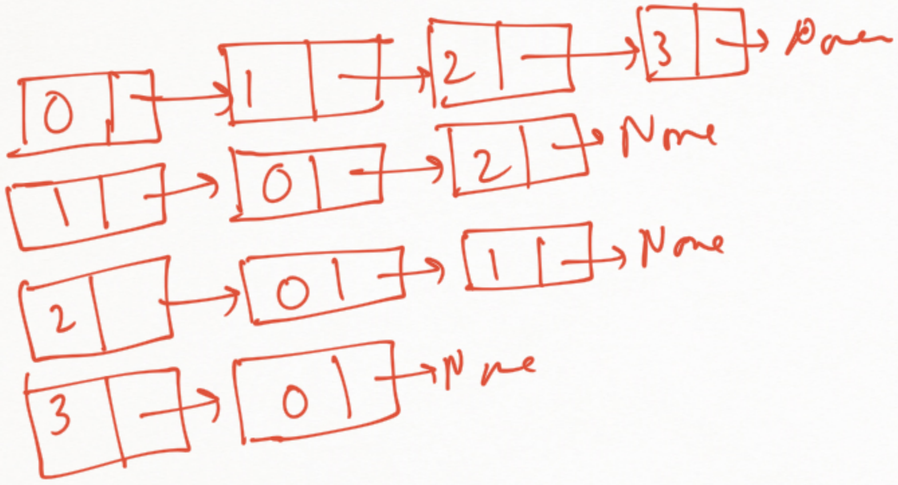
(V x V)



i / j	0	1	2	3
0	0	1	1	1
1	1	0	1	0
2	1	1	0	0
3	1	0	0	0

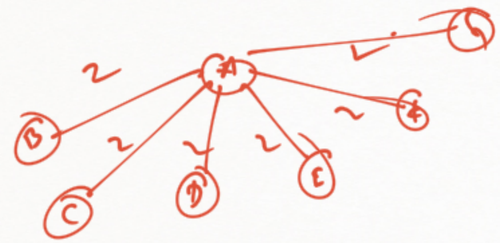
AL (adjacency list)





1) AL represent a graph as an array of linked list

2) The index of the array represents a vertex and each of the element in its linked list represents :
   1) the other vertex that form an edge with the vertex.

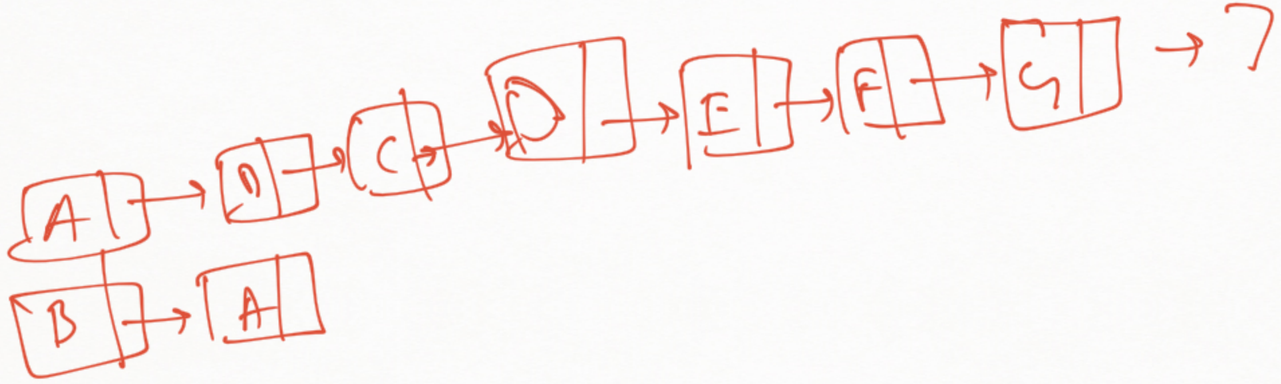


7 → AM 49

∴ AL → 19

0 → 7

1 2



→ 7

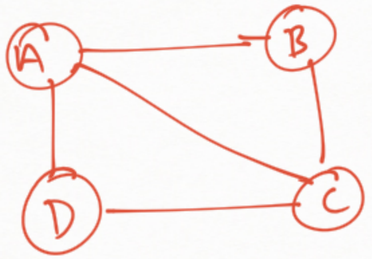
---

1) undirected graph

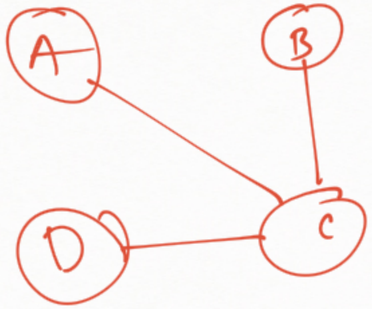
2) connected graph.

1) U G (undirected graph)

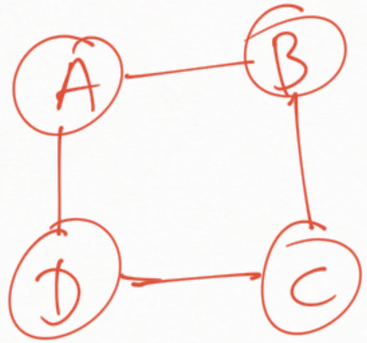
(the edges are bidirectional).



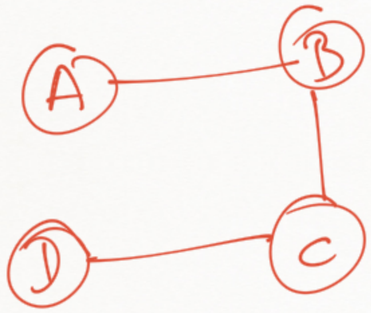
2) connected graph



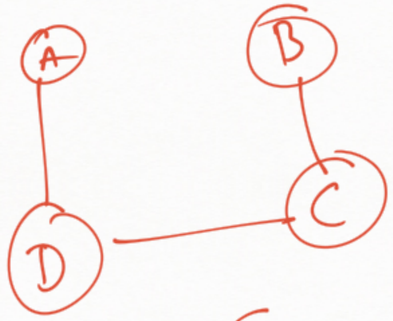
Spanning tree



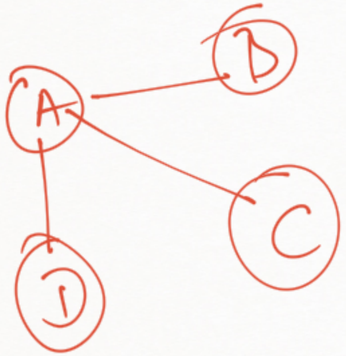
b)

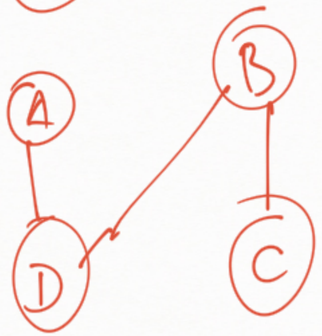


c)

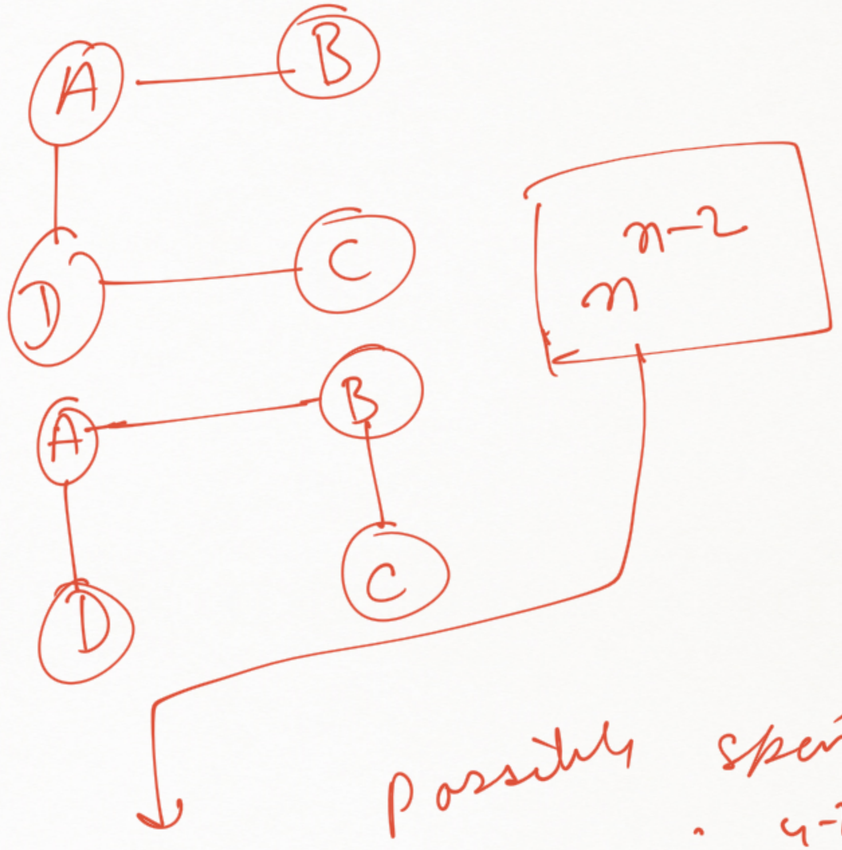


e)





a)

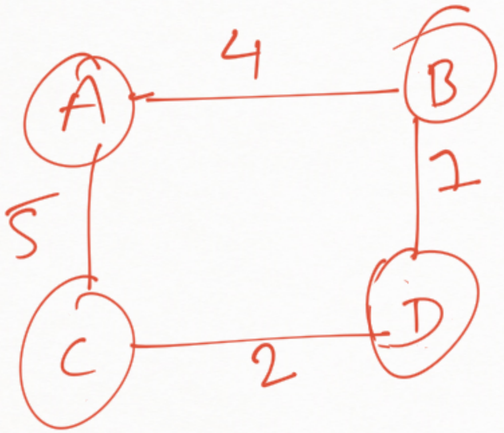


d)

$n = 4$

possible spanning trees $\rightarrow 4^{4-2} = 16$

Minimum Spanning tree

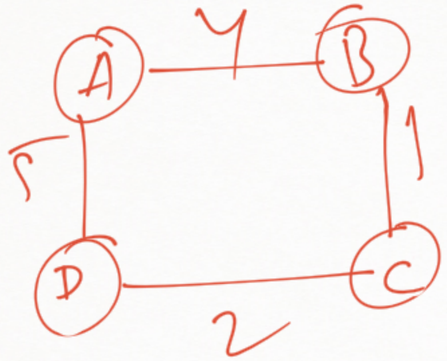


A WG is which each of edges are assigned a weight or cost

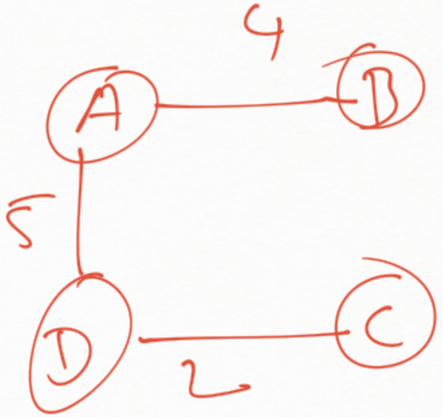
weight → some value, distance time, cost.

weighted graph

MST is a ST in which the sum of the weight of the edges is as minimum as possible.

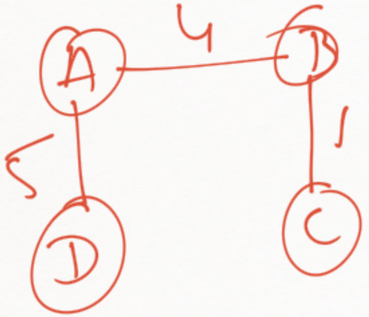


→ ①



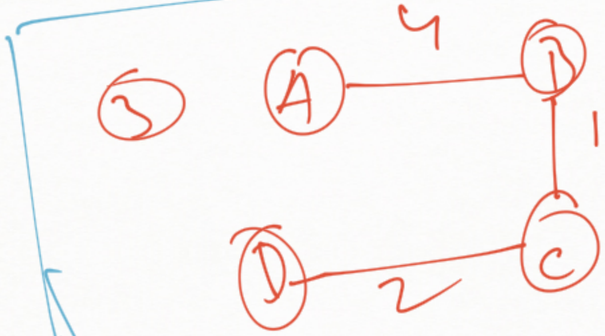
Sum = 11

2)



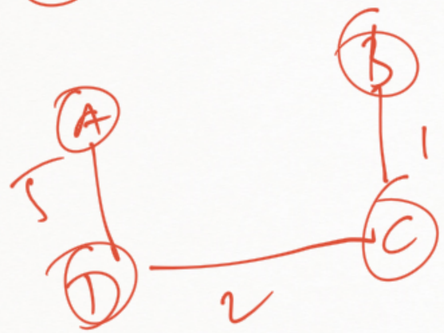
Sum = 10

3)



Sum = 7 ← MST

4)



Sum = 8

a) Prim's algorithm
b) Kruskal's algorithm

ST Applicatin

- Computer network Routing Protocol
- Cluster Analysis
- Civil Network Planning.

etc --

MST Applicatn →

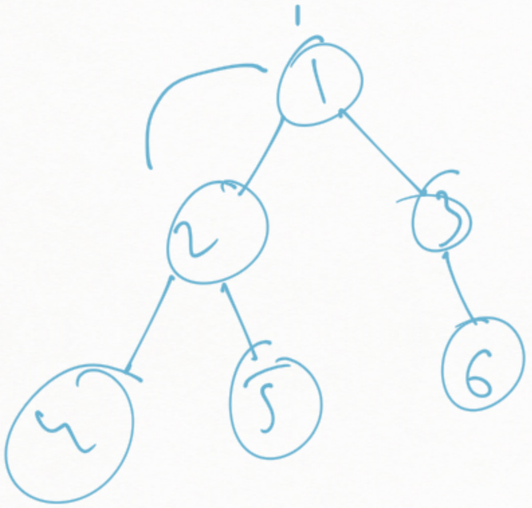
- To find path in the map

BFS

Traversal means visiting all the nodes of a graph

(Breadth First Search)

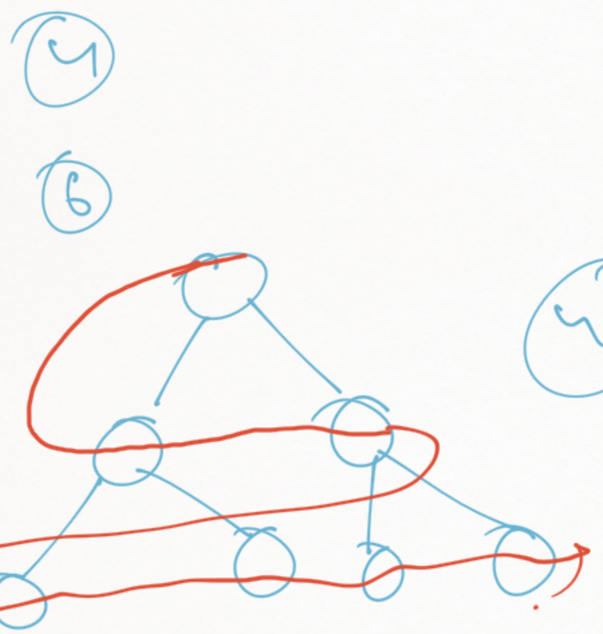
searching graph or a tree



Level 1 ①

Level 2 ② ③

Level 3 ④ ⑤ ⑥



Sense?

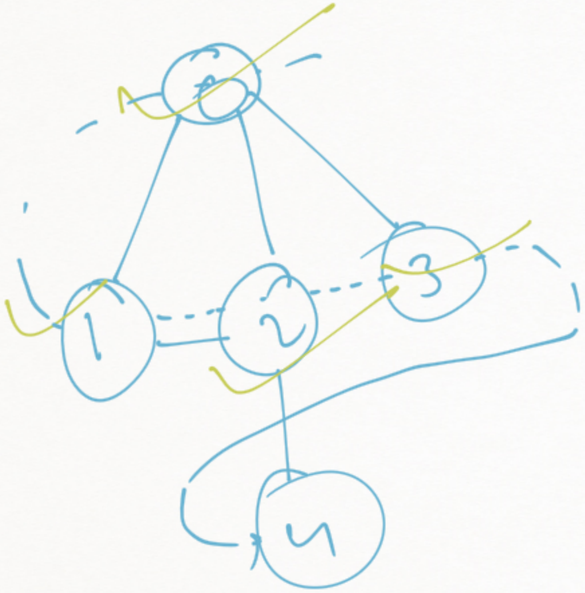
1. Visited
2. Non Visited

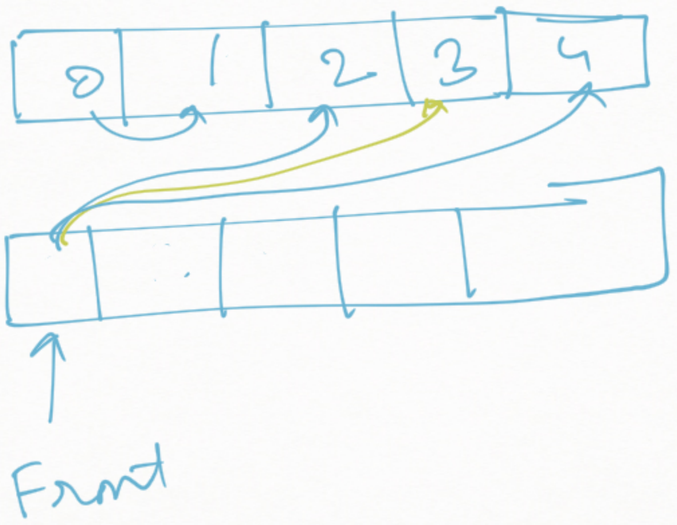
1. Start at the root node and add it to the queue

2. Take the front item of the queue & add it to the visited list

3) Create a list of the vertex's adjacent nodes
Add the ones which aren't visited list.
to the back of the queue.

4) Keep repeating steps 2 and 3 until queue is empty





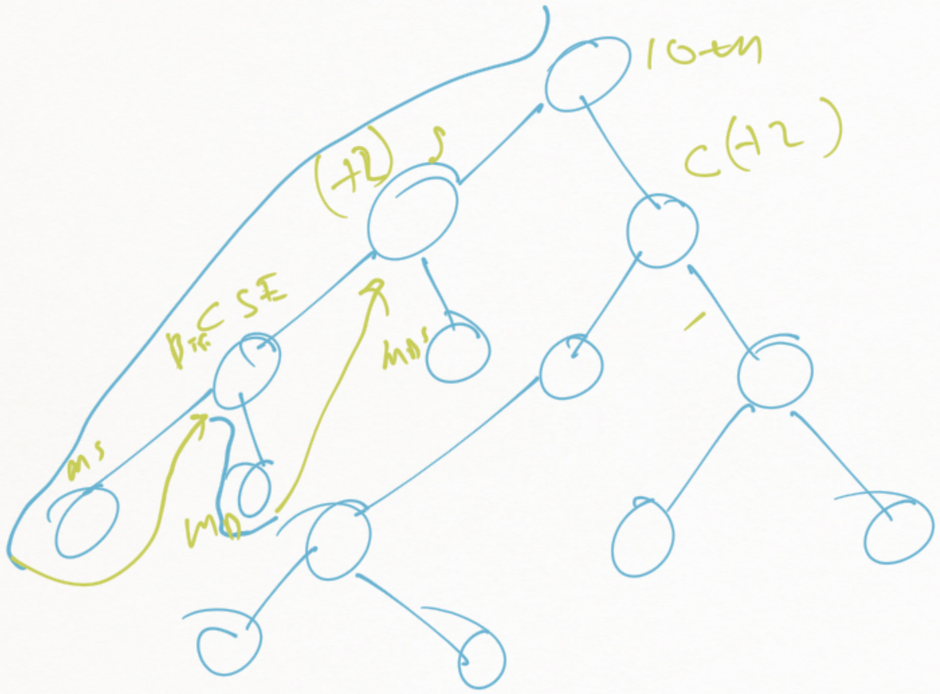
Visited list

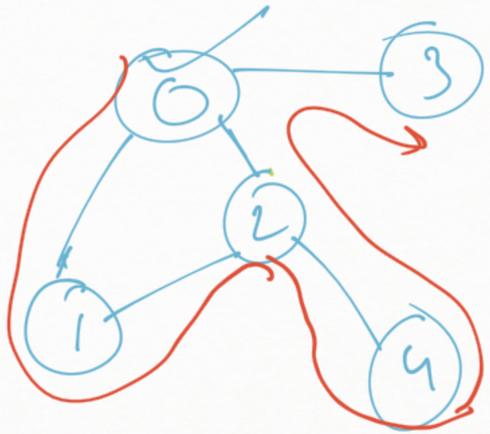
Queue.

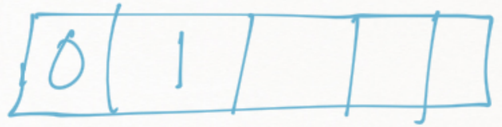
Pseudocode

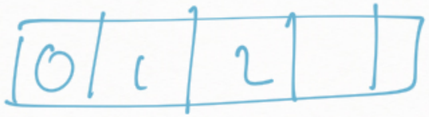
create a queue Q
mark v as visited and put v into Q
while Q is non-empty
remove the head u of Q
mark and enqueue all unvisited
neighbours of Q

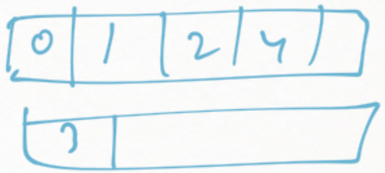
DFS (depth First Search)

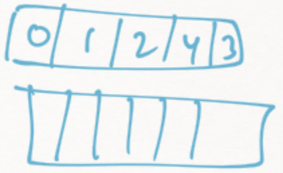












Time Complexity

BFS → $O(V+E)$ where V is no. of nodes
E is number of edges

DFS → $O(V+E)$ "

Space Complexity

BFS → $O(V)$

DFS → $O(V)$

Application of BFS

1) To build a search index
2) GPS Nav.
3) Cycle detection in an undirected graph

Applications of DFS

1. For finding Paths
2. For detecting cycles in a graph.