

Graphs - I

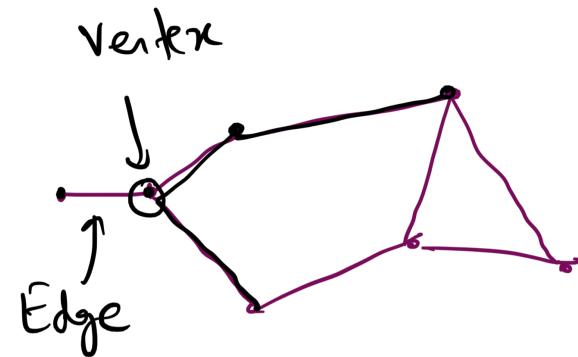
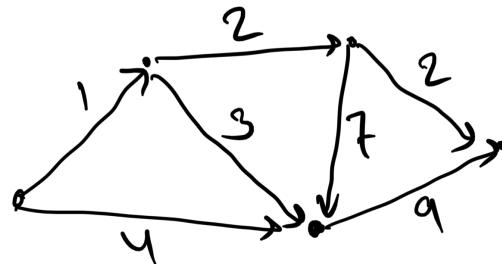


9

Introduction to Graphs Data Structure

Non-linear D.S.

- Directed
- Undirected
weighted



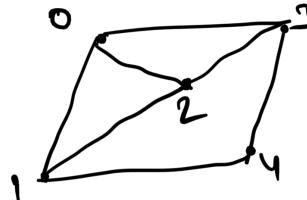
Networking
routes

Implementation of Graphs Data Structure

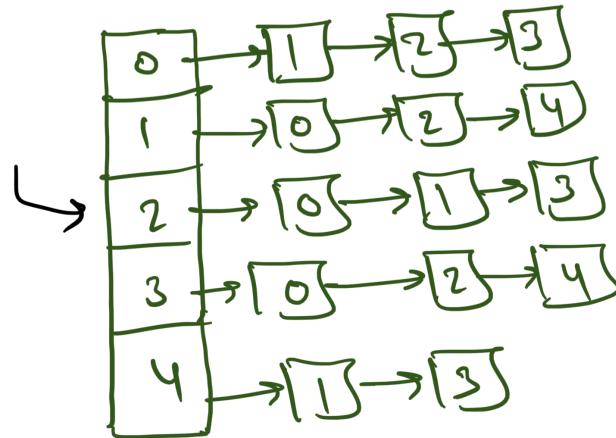
1. Using adjacency matrix

	0	V	X	V	
0	0	1	1	1	0
1	1	0	1	0	1
2	1	1	0	1	0
3	1	0	1	0	1
4	0	1	0	1	0

V
E



2. Using adjacency list Arry. of linked lists



Con: More space

Pro: Constant time edge deletion

$\alpha[2][3]$

Graph: Dense

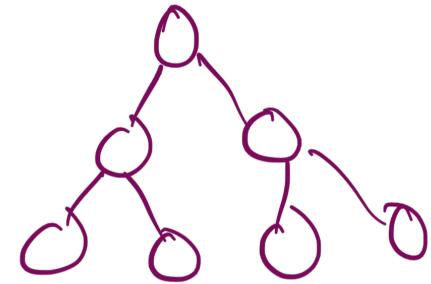
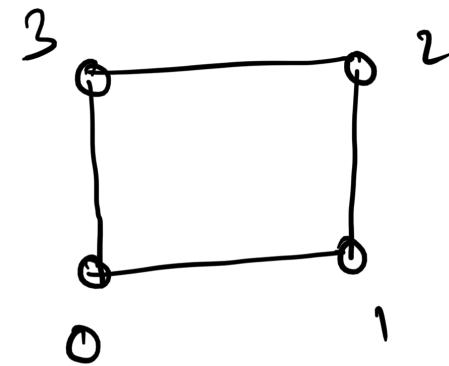
Space: $O(V^2)$

↳ Sparse

$O(V+E)$

	0	1	2	3
0	0	1	0	1
1	1	0	1	0
2	0	1	0	1
3	1	0	1	0

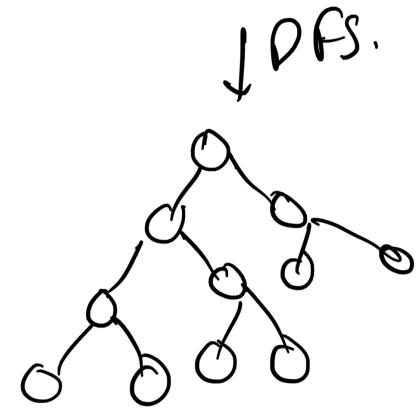
2-D
Matrix.



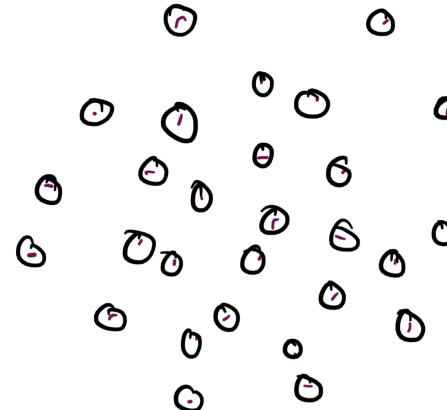
0 →	1	3	
1 →	0	2	
2 →	3	1	
3 →	2	0	

Adjacency
List.

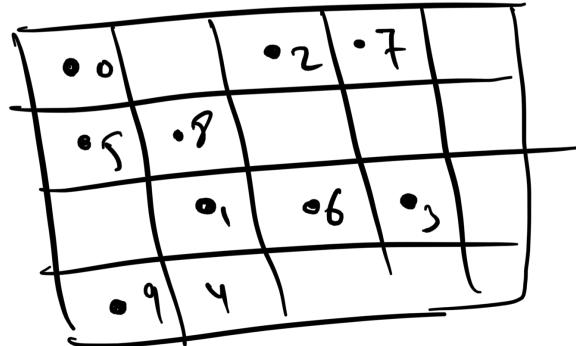
DFS Graph Traversal



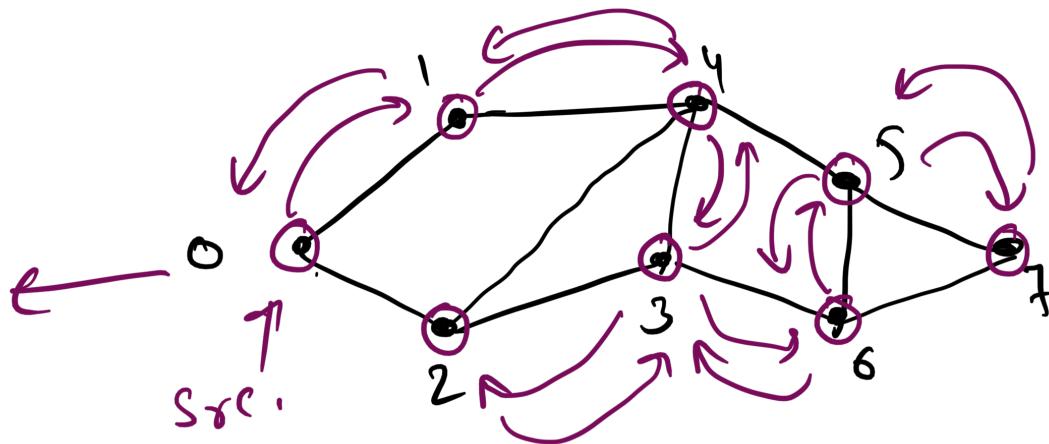
Recursion
+
Stack.



BFS
Greene



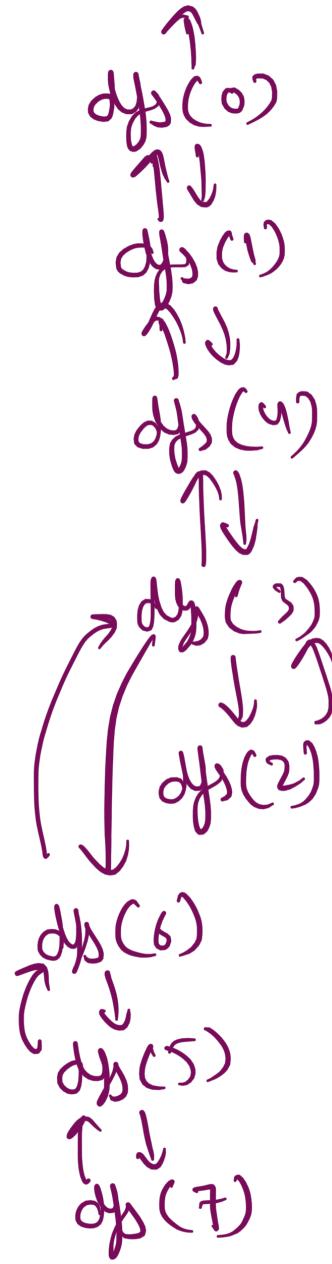
DFS



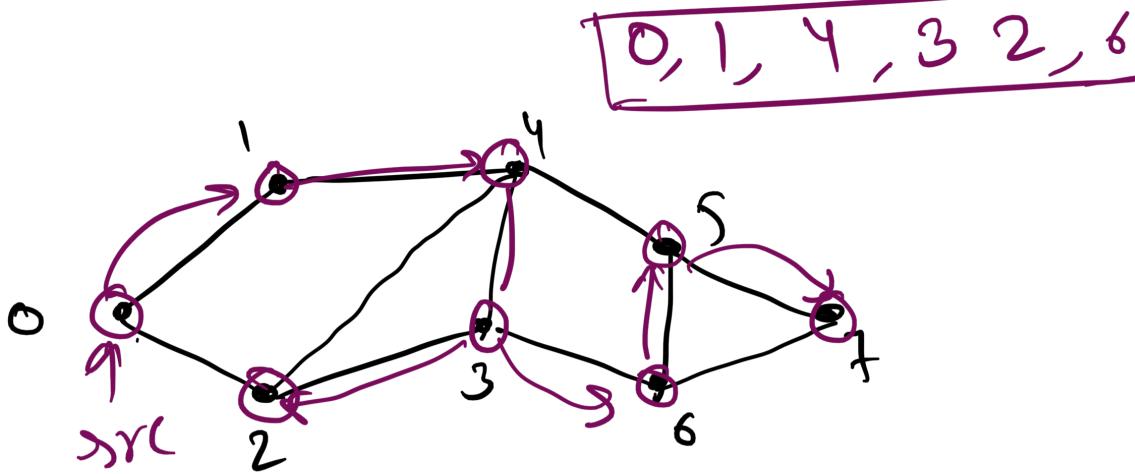
$\text{vis}[j] = \boxed{T | T | T | T | T | T | T | T}$

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Recursively



5
2



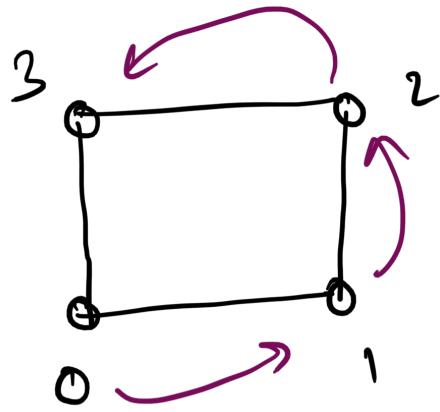
curr = 6

vis[]

T	T	T	T	T	T	T	T	T
0	1	2	3	4	5	6	7	

Par[]

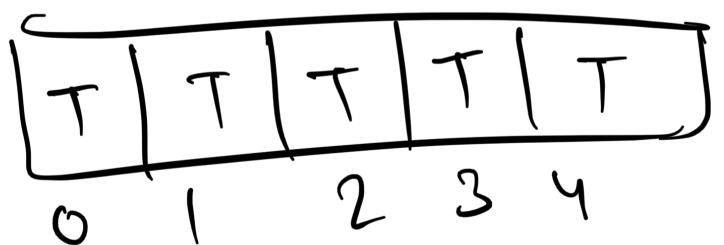
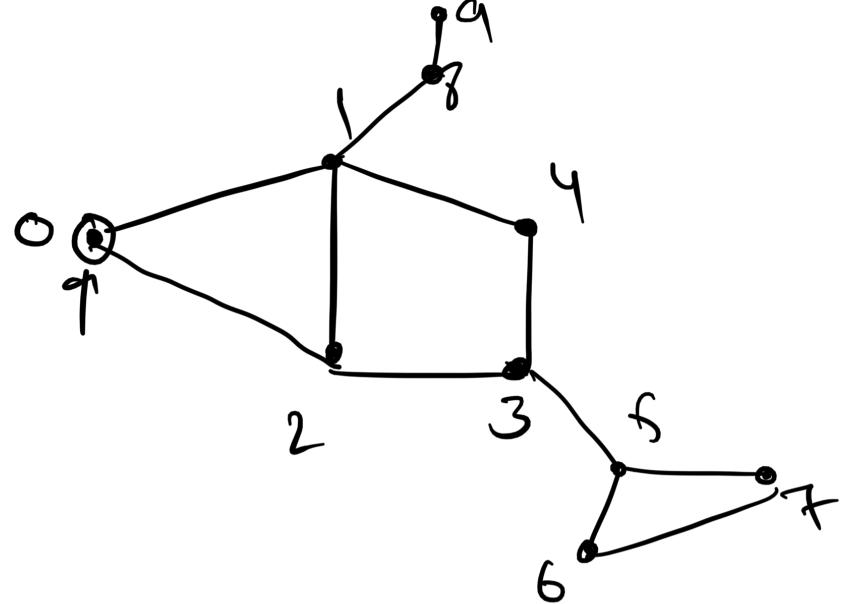
-1	0	0	2	2	6	3	6	
0	1	2	3	4	5	6	7	



0, 1, 2, 3

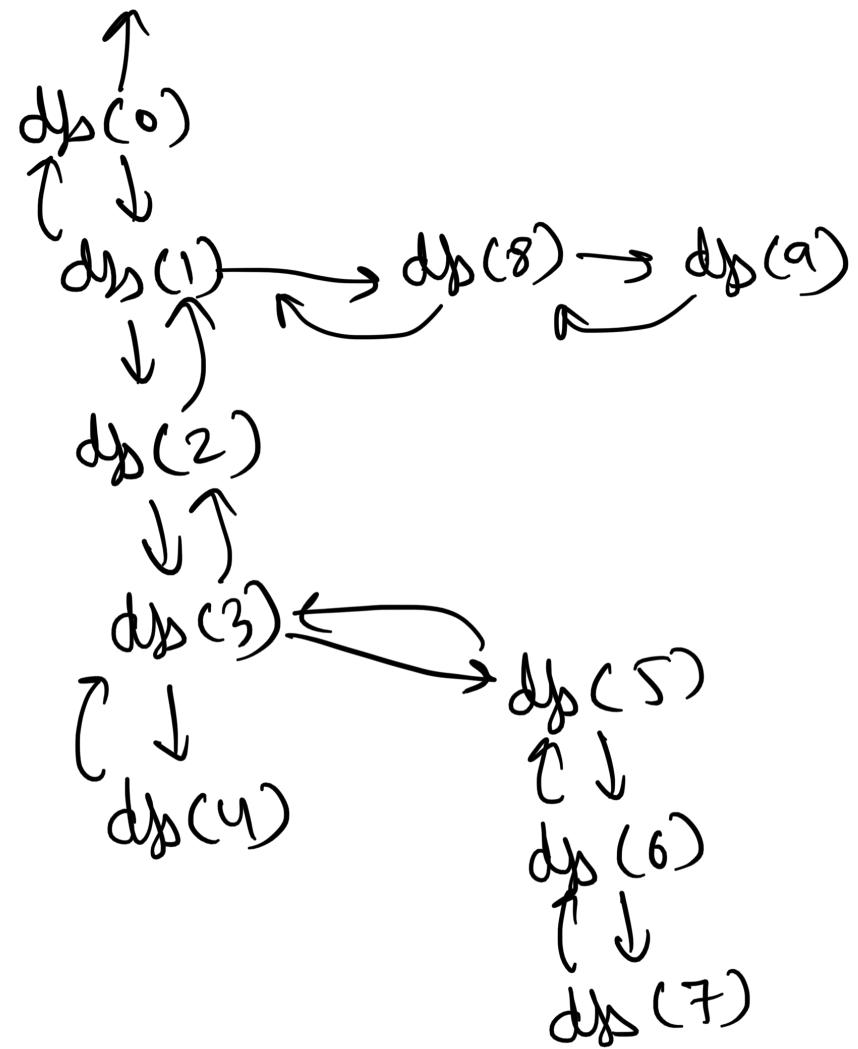
0 →	1	3	
1 →	0	2	
2 →	3	1	
3 →	2	0	

\uparrow
 $dys(0)$
 \downarrow
 $dys(1)$
 \downarrow
 $dys(2)$
 \downarrow
 $[dys(3)]$



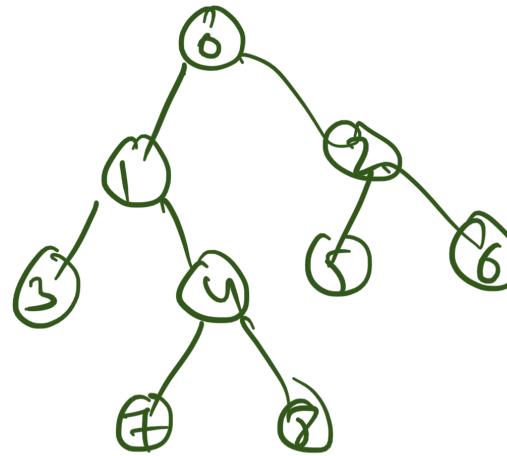
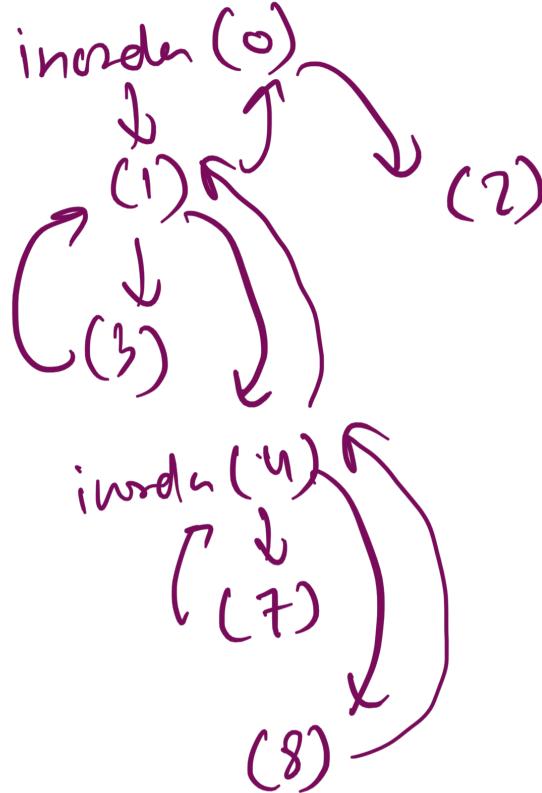
dp

0, 1, 2, 8, 4, 3, 9, 5, 6, 7



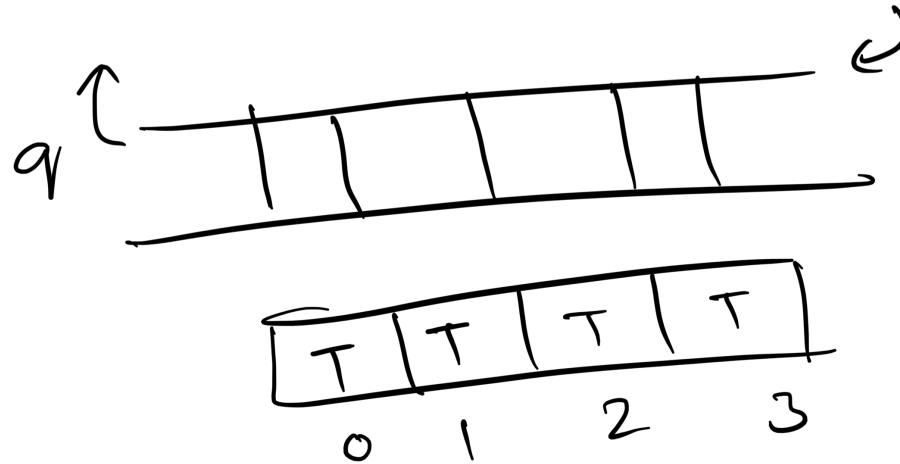
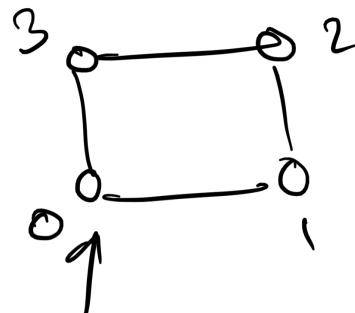
3

LNR



BFS Graph Traversal

0, 1, 3, 2



Practice Problems

- 1. Learn More about Graphs from: <https://www.programiz.com/dsa/graph>