

Agenda → Introduction to graphs

what are graphs?

applications

terminologies

Implementation

General fun

Pre-requisite

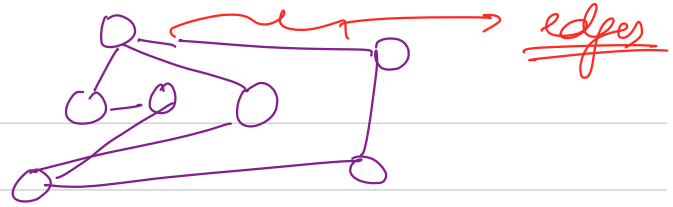
familiar with arrays /
programming lang

C/C++/Java/
Python/ruby/
JS

familiarity with arrays

& if possible linked list

What are graphs??



A graph is a "non-linear" data structure, which comprises of ^{vertices} nodes and edges where nodes denote entities and the edges denote the relationship between the entities. So graph is used to represent data which is shown or depicted via the medium of relationship.

applications of graphs →

→ flight path management

→ google maps

→ facebook's graph API

→ E-commerce

Retailing

graphs

→ shortest path



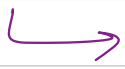
Represent graphs



Adjacency Matrix



Adjacency List

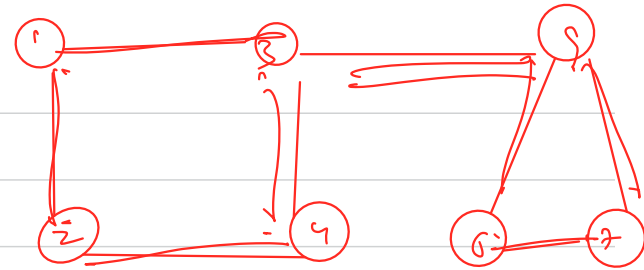


Adjacency Map



Edge list

[1] [3]
[3] [1]



cols → no of nodes

	1	2	3	4	5	6	7
1	.	T	T
2	T
3	T	.	.	.	F	.	.
4	.	.	T
5
6
7

row = no of nodes



adjacency matrix \rightarrow both row & col repr node

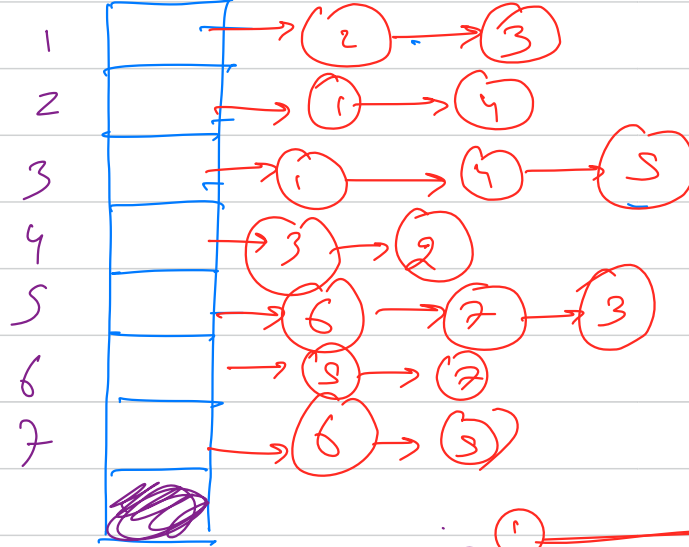
\rightarrow disadvantage \rightarrow space

$n \rightarrow$ nodes \rightarrow $O(n^2)$ space

Adjacency list

array of linked list

~~array of array?~~
~~array of vector~~
~~arraylist?~~



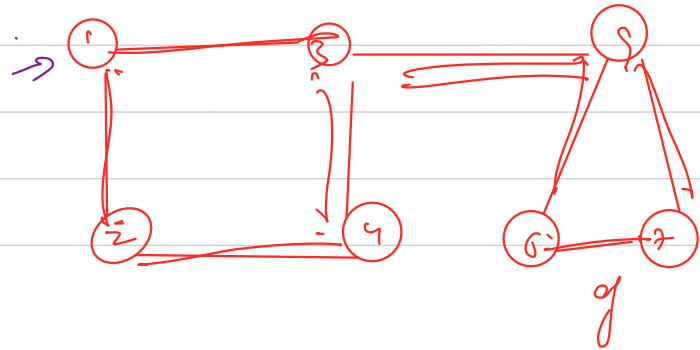
array

array \rightarrow all the nodes

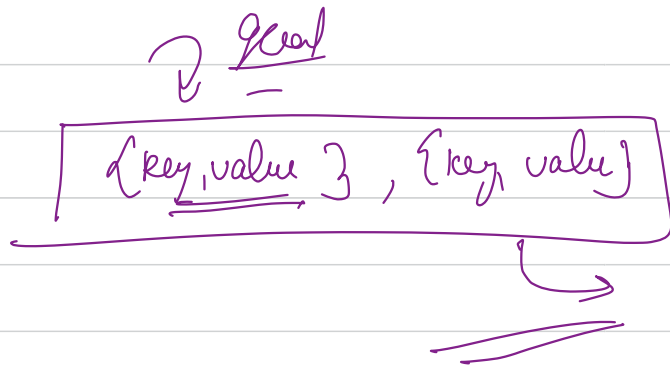
for any index i

$a[i]$ \rightarrow store all the neighbours of

the i^{th} node



why we don't have array of array \rightarrow we don't know n
why we don't use array of array list / vector ^{side} ??



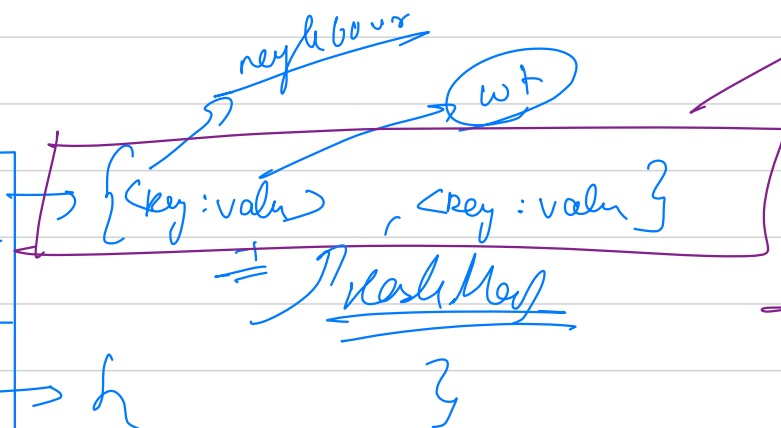
unordered-map



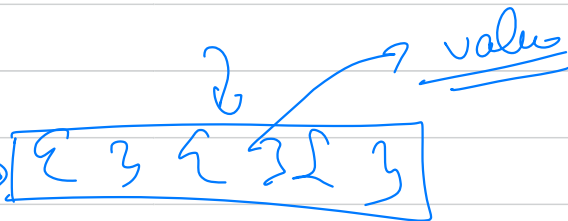
Adjacency

map

map of maps



dict / HM
 $O(n)$
 $O(1)$



Hash Map

$\text{adj}[u][v] \leq \text{wt}$
↓
wt

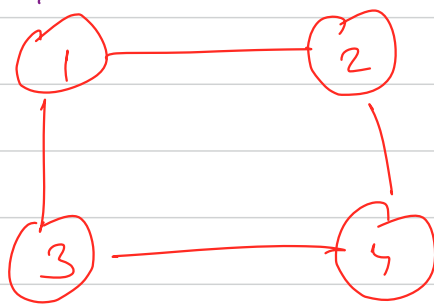
Edge list?



List of edges



Kruskals ^{nst}



~~[{1-2}, {1-3}, {3-4}, {2-4}]~~

{
src
dst
wt
}

< , , >

How many different types of graphs are there ?

↳ Based on edge property

↳ Weighted

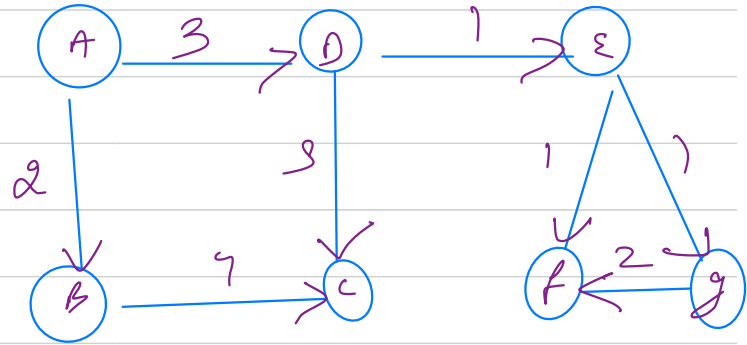
↳ Unweighted

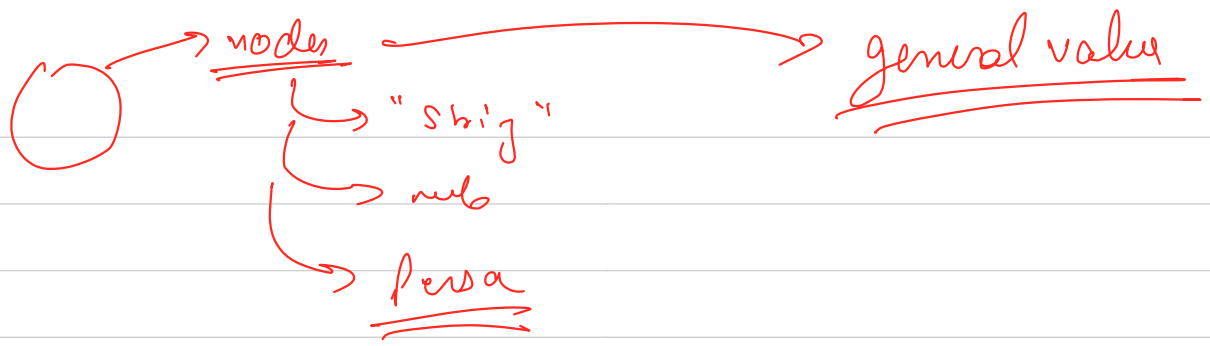
weight \rightarrow magnitude in any property of relationship

↳ Based on direction

↳ Directed

↳ Undirected

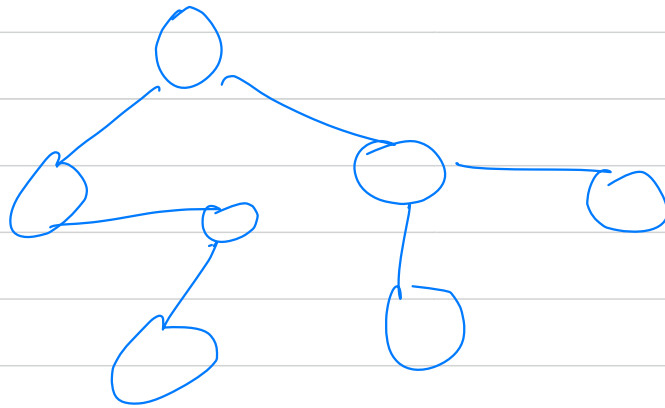




↳ In a graph we can have cycles

↳ if a graph doesn't contain cycle (acyclic graph)

it is also called a tree



Every tree is a graph, but not every graph is a tree.

if from any node of a graph you can reach
to any other node (via any path), then it's a
connected graph.

