Important points about Hashmap :→

* Acts as a data - structure for $<K, V>$ pair in C++ & Java.

* In python it is represented using "Dictionaries"

data = { 102 : Sadana } ;

Note : In C++ we don't have hashmap, rather we use unordered_set, unordered_map

* How do we maintain insertion - order in an Hashmap ? *** Infosys

⟹ We use LinkedHashMap

↳ Built - in Linked List

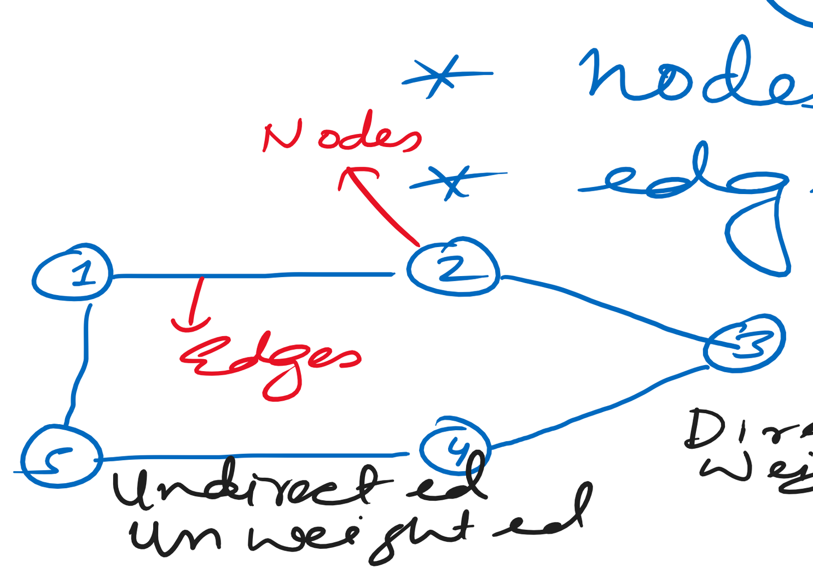
* Graphs ? By default → weight = 1

Non - linear data structures :

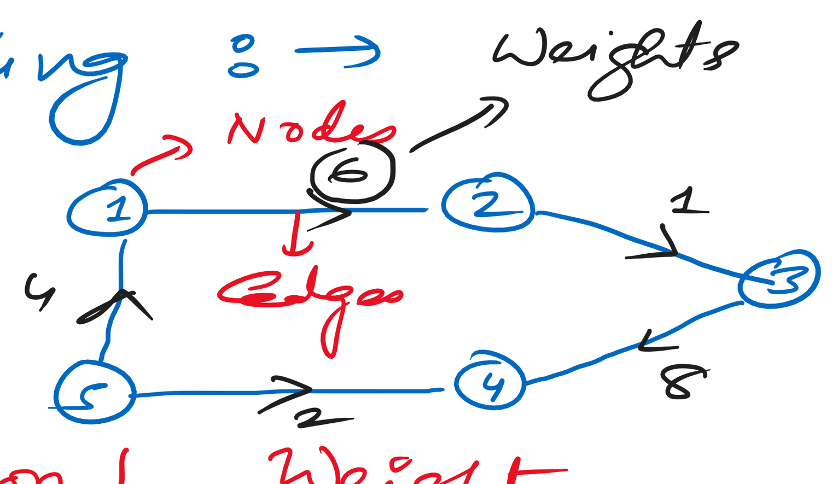
It is an entity having :→ Weights

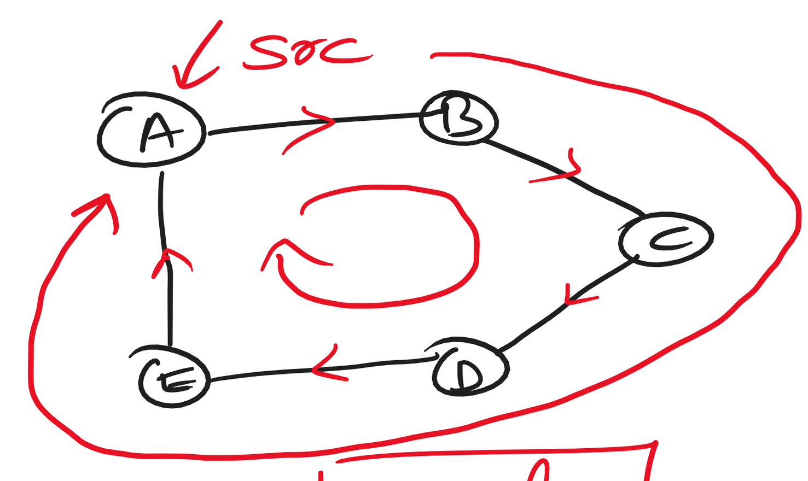
* nodes

* edges



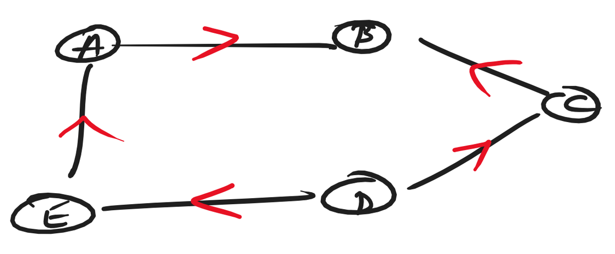
Direction | Directed Weighted graph |



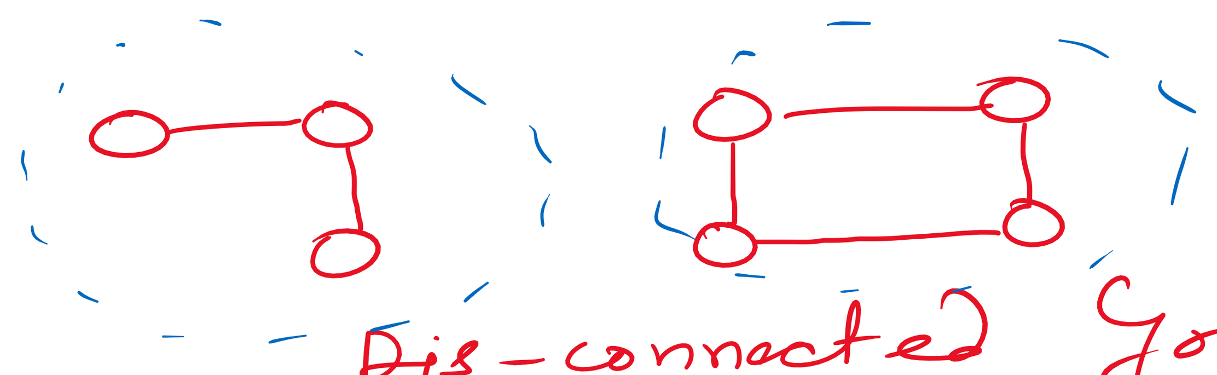


|cycle|

Directed Cyclic graph



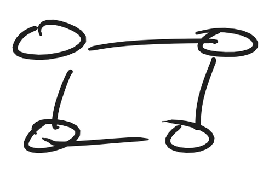
DAG
Directed Acyclic graph



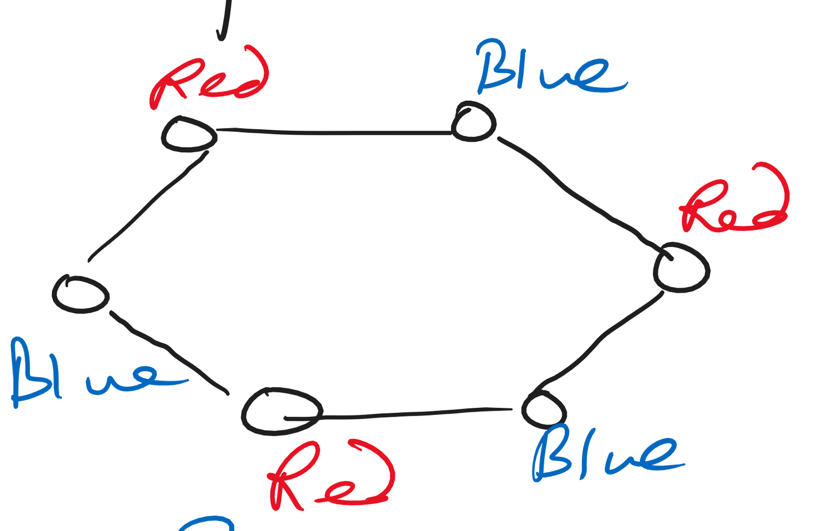
Dis - connected Graph

Component 1 Component 2

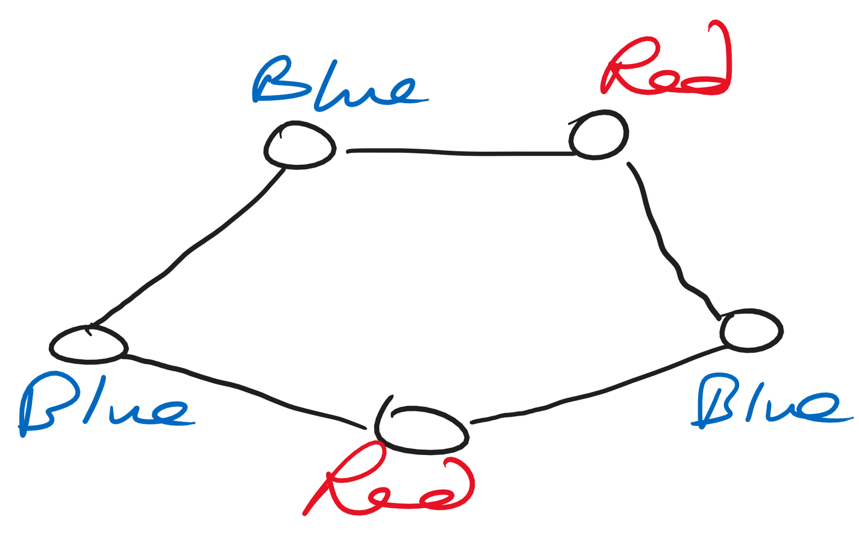
* When all the nodes are not connected to each other.


→ connected graph

Bipartite Graph :→



Bipartite

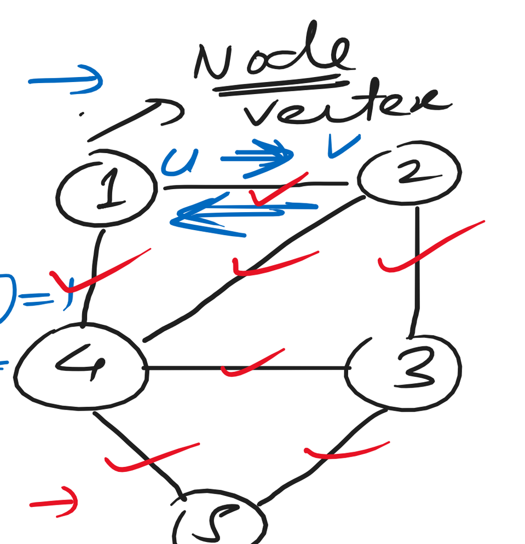


Not Bipartite

How to represent graphs :→

① Adjacency Matrix

Ⓘ Adjacency List



Adjacency Matrix :

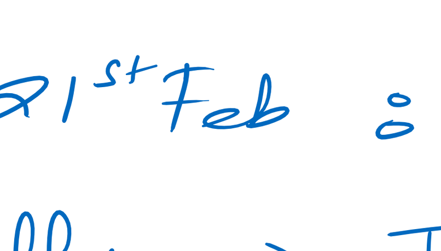
	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	1	0	1	0
2	0	1	0	1	1	0
3	0	0	1	0	1	1
4	0	1	1	1	0	1
5	0	0	0	1	1	0

Adj List :→

Nodes : List of neighbors

0 →
1 → 2, 4
2 → 1, 3, 4
3 → 2, 4, 5
4 → 1, 2, 3, 5
5 → 3, 4

Directed Graph :



Directed

Adj list :

1 → 2
2 →
3 → 2
4 → 1, 3

Plan → 21st Feb : [DSA 450.com

9185

Graphs → Traversals

Greedy Algorithms → Idea / Approach Problems

Dynamic Programming → Recursion → Memoisation → Tabulation → Space optimisation

(Problems) !!!