Standard Template Library || Built-in Data Structures in C++

library

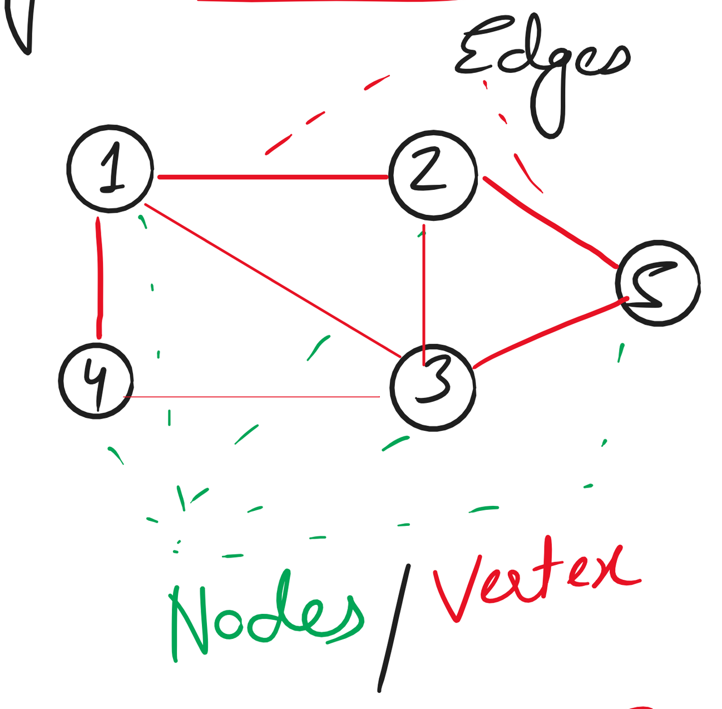
- * Stacks → LIFO
- * queues → FIFO
- * maps → <Key, value> pair
- * sets → Stores Unique Elements
- * lists → (1) forward list → SLL (ii) list → DLL
- * vectors → Dynamic Arrays → we can resize

How to use → # include <library_name>

**Note :→ In the placement drives (Coding rounds) if they don't mention anything, always use STL libraries.

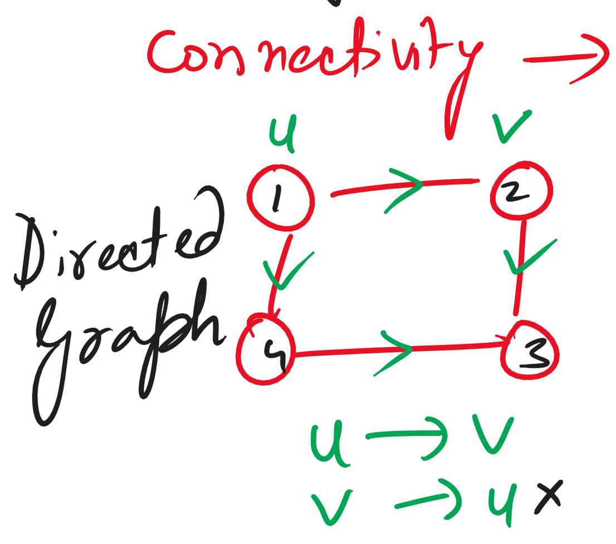
Introduction to Graphs :→ Non-linear Data Structures

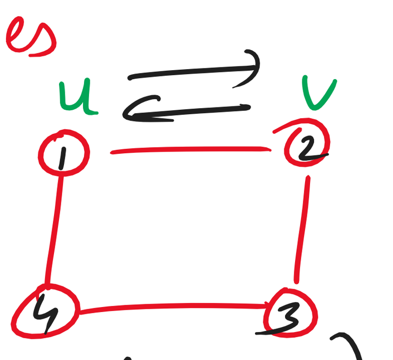
A graph is a data structure containing of an entity called Node. All the nodes of a graph are connected by a connection called Edge. (Google Maps)



Connectivity → 2 types

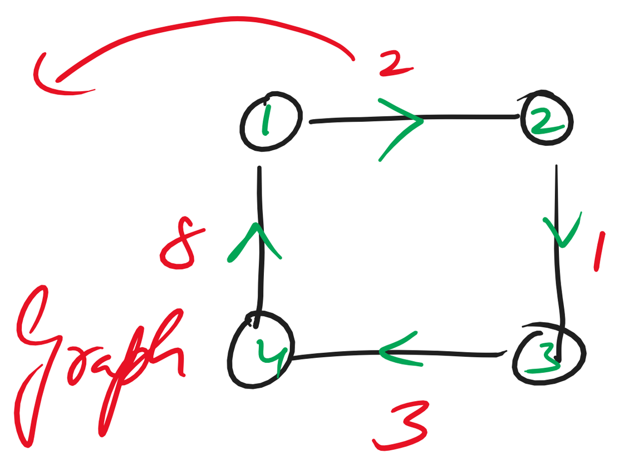
Directed graph



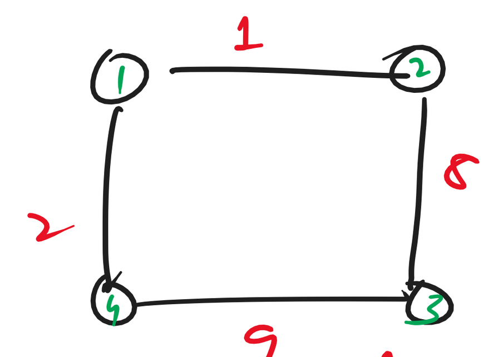


Undirected graph

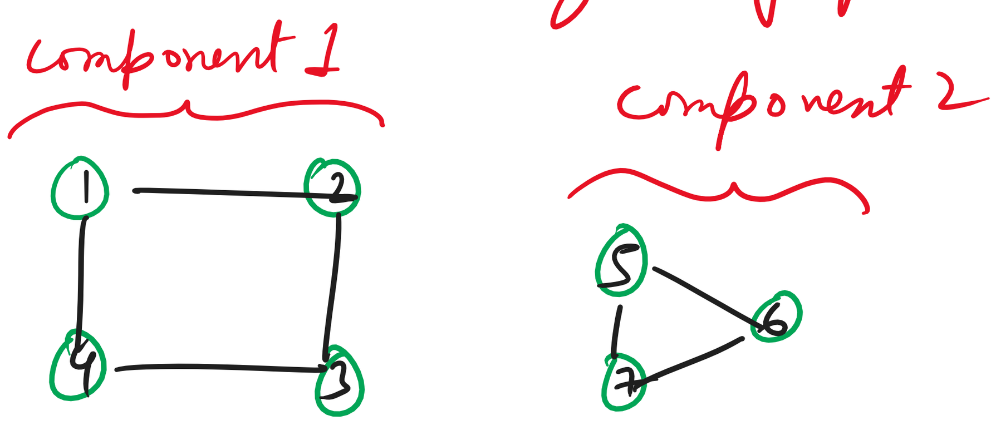
Edge weights



Directed Weighted Graph

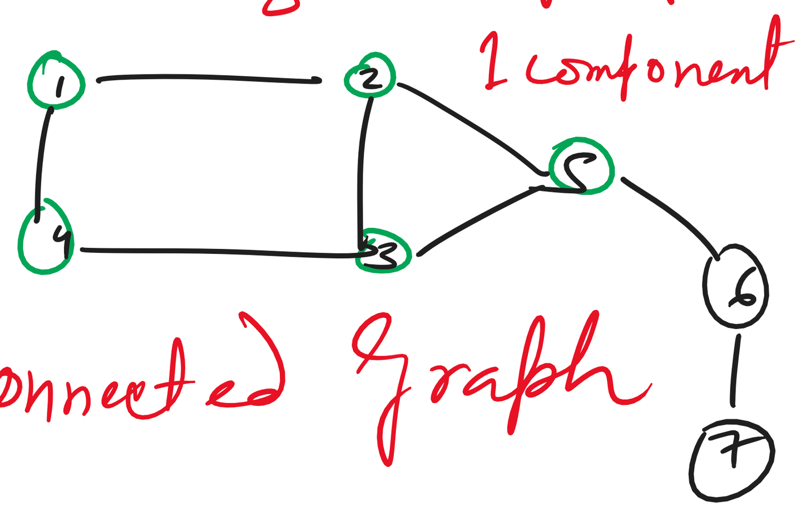


Undirected Weighted graph

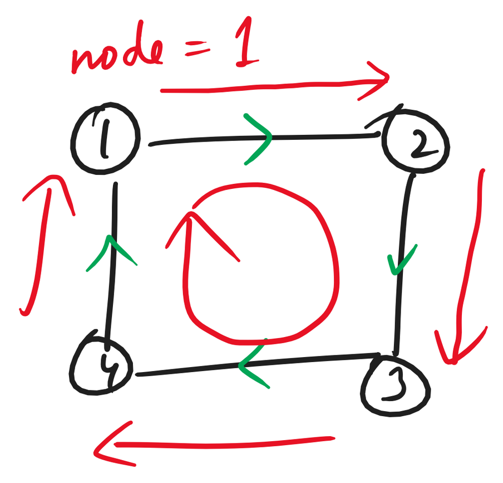


Disconnected graph

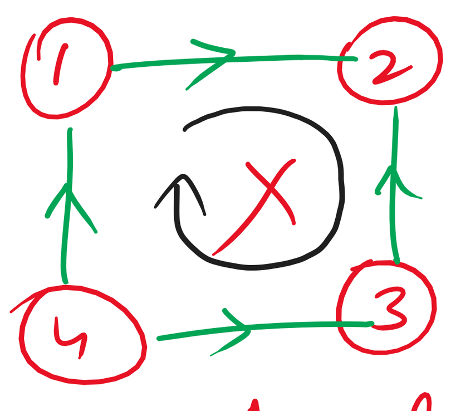
Nodes = 7 nodes



Connected graph

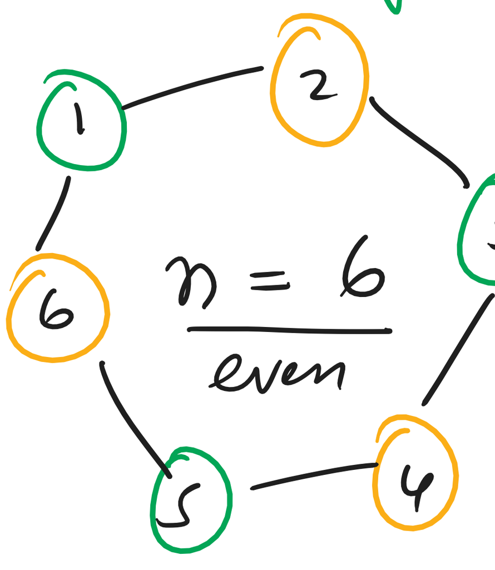


Cyclic graph



Acyclic graph

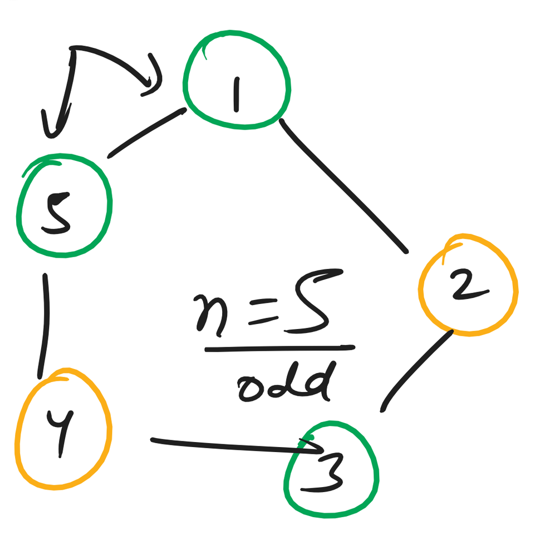
Bipartite graph :→



No adjacent nodes have same colour

Bipartite

same colour



Adjacent nodes have same colour.

Not - Bipartite

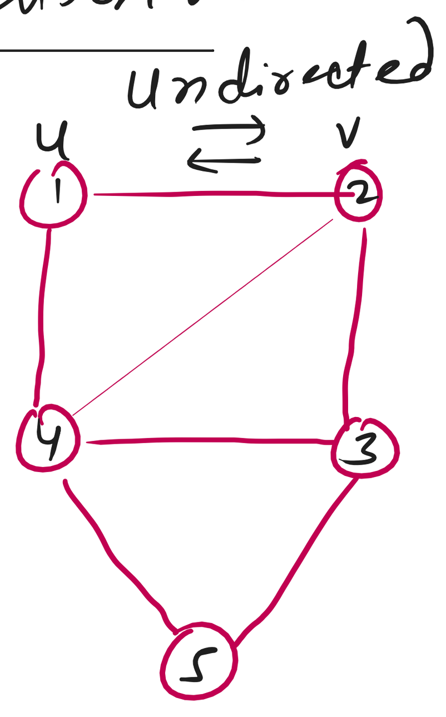
Graph Representation :→

Adjacency Matrix

$n \times n = 5 \times 5$

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

undirected



Adjacency List

Node : List of Neighbours

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