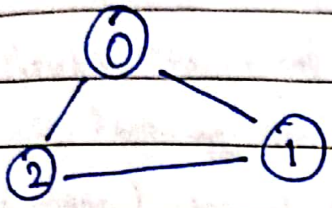
Representation of Graphs:

Scanned with CamScanner

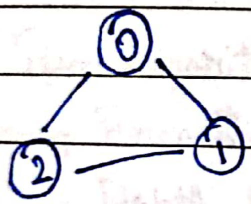
→ Ways to represent a graph:

→ Adjacency list ⇒ Mark the nodes with the list of its neighbours.



ADJACENCY LIST ⇒ 0 → 1 → 2
1 → 0 → 2
2 → 0 → 1

→ Adjacency matrix ⇒ $A_{ij} = 1$ for an edge between i and j, 0 otherwise!



ADJACENCY MATRIX ⇒

	0	1	2
0	0	1	1
1	1	0	1
2	1	1	0

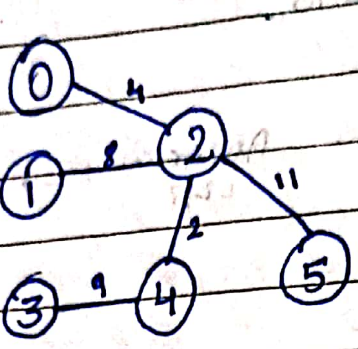
→ ~~E~~ Edge set ⇒ Store the pair of nodes / vertices connected with an edge. Eg - {(0,1); (0,4), (1,4)}

→ Other implementations to represent a graph also exists. For e.g. Compact list representation, cost adjacency list, cost adjacency matrix etc.

→ Cost adjacency matrix :

→ A_{ij} = cost for an edge between i and j, 0 otherwise!

→ If the cost is 0 somewhere:
A_{ij} = cost for an edge between i and j, -1 otherwise.

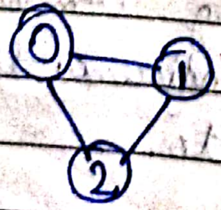


	0	1	2	3	4	5
0	0	0	4	0	0	0
1	0	0	8	0	0	0
2	4	8	0	0	2	11
3	0	0	0	0	9	0
4	0	0	2	9	0	0
5	0	0	11	0	0	0

→ Other Implementations:

→ Edge Set → Store the pair of nodes / vertices connected with an edge.

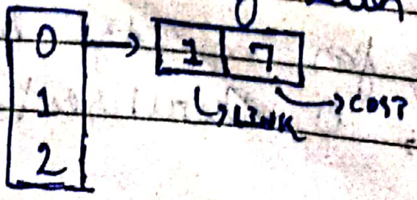
Ex:



⇒ {(0,1), (0,2), (1,2)}

→ Cost adjacency list → Cost is also stored along with the links.

Ex:



→ Compact List representation → The entire graph is stored in a 1d array.

Scanned with CamScanner