REPRESENTATION OF GRAPHS

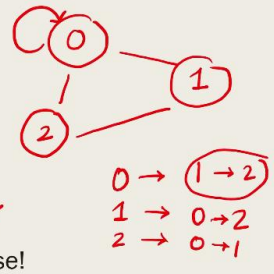
By - CodeWithHarry

Graph Theory so far...

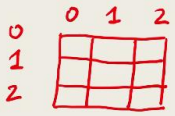
- Graph = Nodes & Edges.
- Used to model real world problems like managing a social network, website links etc
- Helps to solve problems like is there a path between Delhi and California by road. If yes which one is the shortest?
- Helps to solve problems like :

1. Is A friend of B in a Social Network
2. Is Site A linking to Site B ?

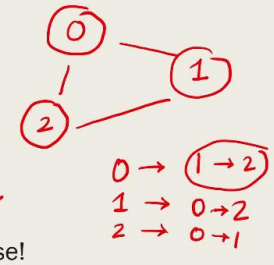
Ways to represent a graph



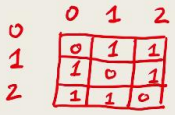
- Adjacency list – Mark the nodes with the list of its neighbors
- Adjacency matrix - A_{ij} = 1 for an edge between i and j, 0 otherwise!
- Edge set – Store the pair of nodes/vertices connected with an edge. Eg – {(0,1), (0,4), (1, 4)}
- Other implementations to represent a graph also exists. For e.g. Compact list representation, cost adjacency list, cost adjacency matrix etc.



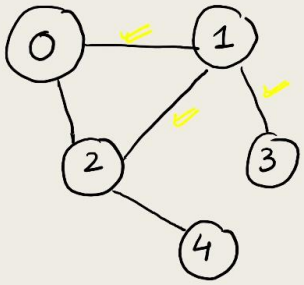
Ways to represent a graph



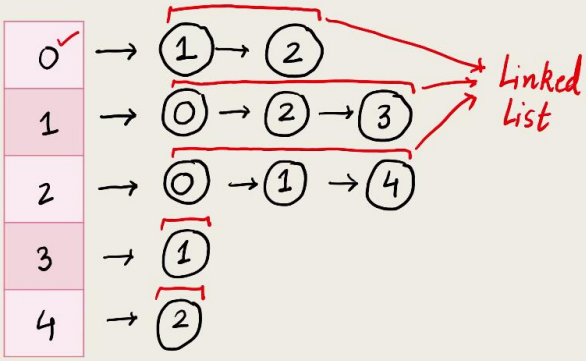
- Adjacency list – Mark the nodes with the list of its neighbors
- Adjacency matrix - A_{ij} = 1 for an edge between i and j, 0 otherwise!
- Edge set – Store the pair of nodes/vertices connected with an edge. Eg – {(0,1), (0,4), (1, 4)}
- Other implementations to represent a graph also exists. For e.g. Compact list representation, cost adjacency list, cost adjacency matrix etc.



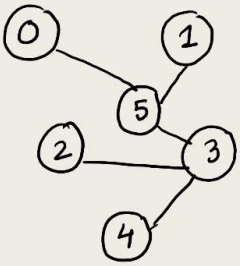
Adjacency list



- Mark the nodes with the list of its neighbors



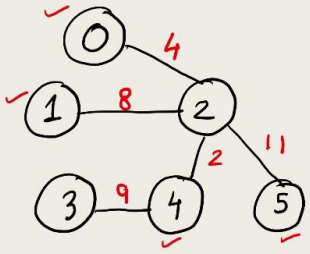
Adjacency matrix



- A_{ij} = 1 for an edge between i and j, 0 otherwise!

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

Cost Adjacency matrix



- A_{ij} = cost for an edge between i and j, 0 otherwise!
- If the cost can be 0:
  A_{ij} = cost for an edge between i and j, ∞ otherwise!

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

Other Implementations

- Edge Set - Store the pair of nodes/vertices connected with an edge. Eg – {(0,1), (0,4), (1, 4)}
- Cost adjacency list – Cost is also stored along with the links
- Compact List representation – The entire graph is stored in a 1d array

Other Implementations

- Edge Set - Store the pair of nodes/vertices connected with an edge. Eg – {(0,1), (0,4), (1, 4)}
- Cost adjacency list – Cost is also stored along with the links
- Compact List representation – The entire graph is stored in a 1d array