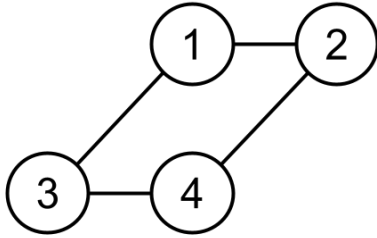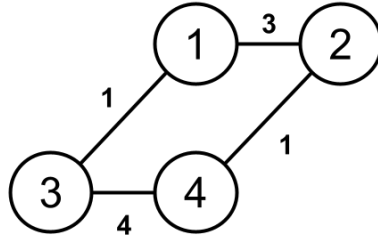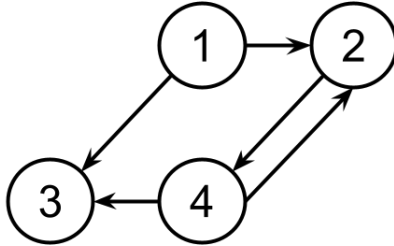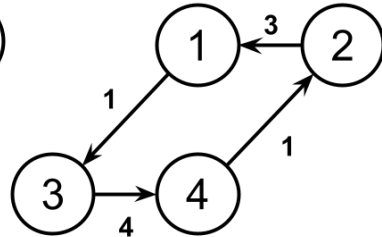# Graphs

What are graphs?

undirected, unweighted

undirected, weighted

directed, unweighted

directed, weighted

- **Graph:** Data structure used to demonstrate relationships between entities

- **Node/vertex:** entity in graph

- **Edge:** connection between two nodes

- **Undirected graph:** edges go both ways

- **Directed graph:** edges go one way

- **Weighted graph:** edges have weights associated with them

- **Acyclic graph:** graph containing no cycles

How are graphs represented?

- **Adjacency list:** each node has its own list of all neighbors

- **Adjacency table:** 2D array stores relationships between each pair of nodes

- **V, E:** Simple list of vertices and list of edges. Common in proofs

- Adjacency lists can take up less space than adjacency tables

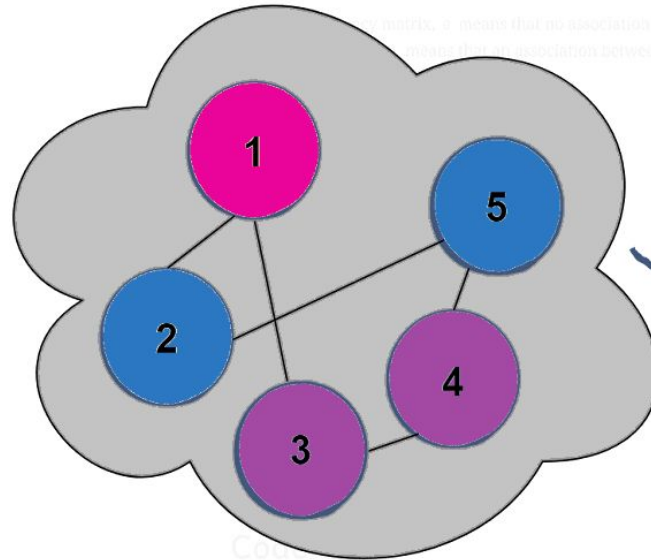- Adjacency lists are what leetcode usually gives you

Adjacency list

- Uses a hashmap of lists, or a list of lists
- Each node's index points to its neighbors
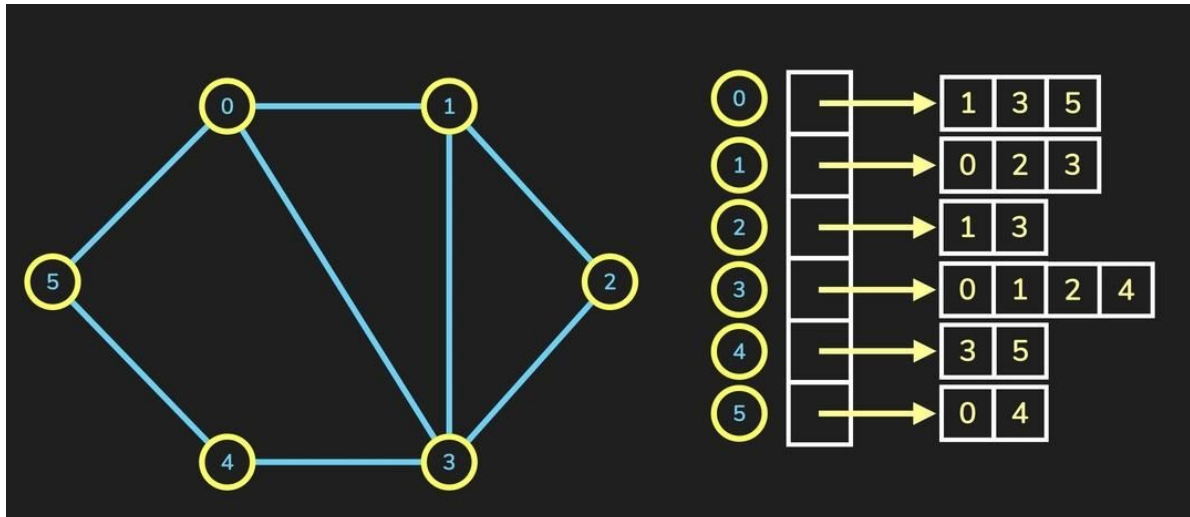
Adjacency matrix

- Represent graph as a matrix M
- M[u][v] is 1 if vertex u is connected to vertex v
- Takes up lots of mem

Algorithms for graphs
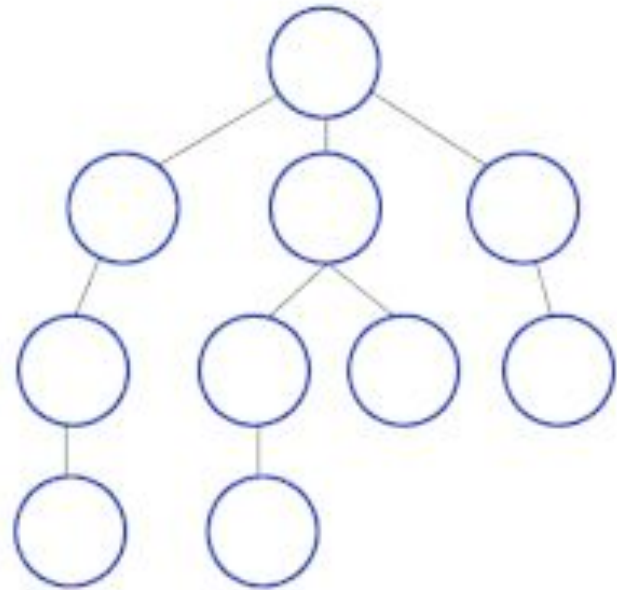
- Traversing
  - Breadth-first, Depth-first

- Topological sort
- Finding shortest path (sort of complicated)
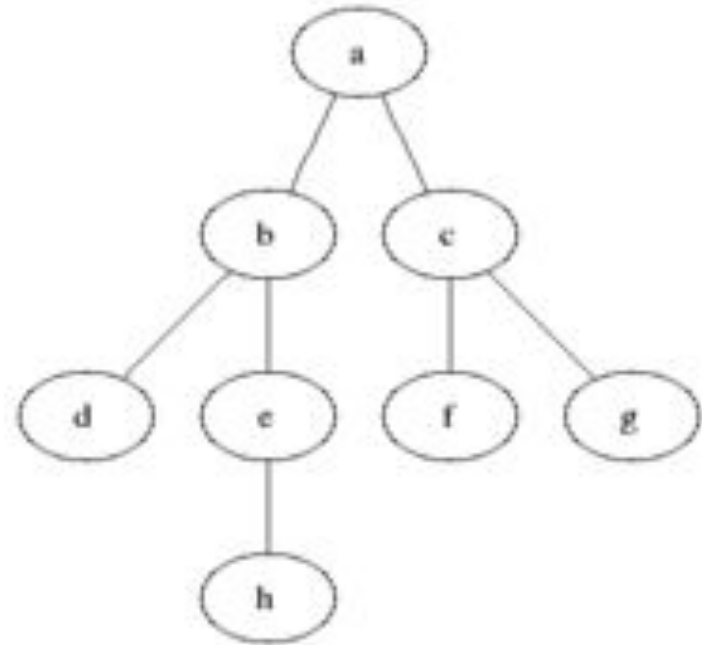 - Dijkstra's algorithm
  - Bellman-Ford

Depth-first traversal

- Pick a node to start out with
- A node recursively traverses all its unvisited neighbors
- Reading pseudocode helps

Breadth-first traversal

- Add first node to queue
- Repeatedly:
  - Remove first node from queue
  - Add all its unvisited children to queue
  - Mark as visited

Terms that you can Google

Useful for today's problems:

- Adjacency list
- Depth-first search
- Breadth-first search

Other:

- Topological sort
- Finding shortest path (sort of complicated)
  - Dijkstra's algorithm
  - Bellman-Ford
- Minimum spanning tree