We’ll be doing two major things in recitation today:

- Graphs
- Graph Searching Algorithms

Graphs

- A graph is a pair \((V, E)\), where
  - \(V\) is a set of vertices
  - \(E\) is the set of edges connecting the vertices
- The edges may be either directed (go in one direction) or they may be undirected (go in both directions).
- A path from one vertex to another is a sequence of edges that goes from the first vertex to the other.
- A graph is said to be connected if there is a path from any vertex to any other vertex in the graph.
- A graph that is not connected can be expressed as multiple connected components
- A graph can be expressed in two ways - an adjacency list and an adjacency matrix. Let’s discuss these now
Adjacency List Representation

Adjacency Matrix Representation
Graph Searching Algorithms

Breadth First Search

- Visit all the nodes on a particular level before going to the next level (more in context for trees).

- In graphs, we do this by visiting all the neighbors of a node immediately after visiting the node.

- We’ll do this by maintaining a queue of nodes to visit.

- We also need to keep track of which nodes we have already visited.
Depth First Search

- Keep going down a branch and then backtrack up (more in context for trees).
- In graphs, we do this by visiting a neighbor of a node, then a neighbor of the node we just visited and so on and then go back to the remaining node.
- We’ll do this by maintaining a stack of nodes to visit.
- Again, we also need to keep track of which nodes we have already visited.
Priority First Search

- We visit nodes with higher priority first.
- We'll do this by maintaining a priority queue of nodes to visit.
- Again, we also need to keep track of which nodes we have already visited.
- Interestingly, we can reconstruct BFS and DFS as instances of PFS.
- For DFS, we give strictly increasing priorities, so we stimulate a stack.
- For BFS, we give strictly decreasing priorities, so we stimulate a simple queue.