Quiz 2 Prep

Heap as an array:
i's left child = 2i+1
i's right child = 2i+2
i's parent = (i-1)/2

Each level is inserted level at a time

Kahn's (Topological Sort)
- Add indegree of 0 to queue
- Remove from queue and decrement neighbors indegrees
  - add to sorted result

Prim's (MST)
- Add random node to visited
- From visited pick minimum edge weight
- Add minimum edge neighbor to visited

Kruskal's (MST)
- Pick minimum weight edge and add to MST solution
- If minimum weight edge connecting two nodes in same tree then ignore

Dijkstra's (shortest path)
- Start at start node and add to visited
- Pick shortest path that leads to node not in visited, update value for node
- Check adjacent edge of unvisited node to check for shorter path, update value for the unvisited node.
- Calculate distance from source, not local distance

Problems with Dijkstra's:

1. Doing updates is unnecessary. Compare edges in terms of weight + shortest distance to that edge
2. Fails if there are negative weights

Bellman Ford: is Dijkstras modified for negative weight edges,
Longest path in graph is V-1 length, otherwise there is cycle

A* is general case of Dijkstras.
1. add heuristic function value to each edge weight
2. Do Dijkstra's with heuristics applied to weights

Floyd-Warshall (Shortest Distance)
dist is a VxV matrix initialized to ∞
for each edge (u,v)
    dist[u][v] = weight of (u,v)
for each vertex v
    dist[v][v] = 0
for k, 1 to V
    for i, 1 to V
        for j, 1 to V
            if dist[i][j] > dist[i][k] + dist[k][j]
                dist[i][j] = dist[i][k] + dist[k][j]