Section 08: Graphs Review

1. Buggy Dijkstra's

Your best friend tried their hand at writing a more complete Dijkstra's pseudocode. However, you noticed something isn't quite right. **Find the bug(s), explain why the behavior isn't correct, and suggest a fix.**

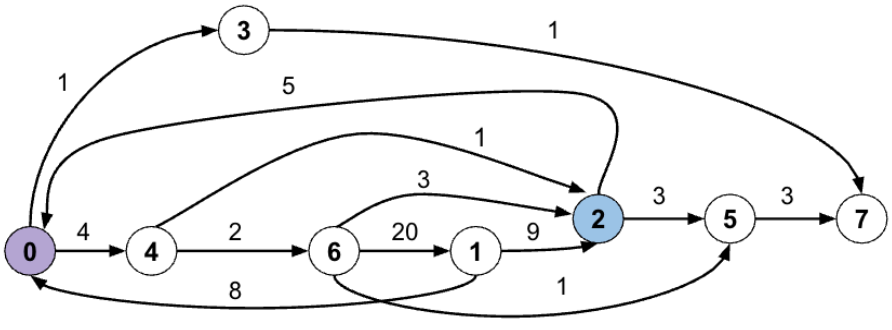
```
dijkstraShortestPath(G graph, V start, V end)
    Set known; Map edgeTo, distTo;
    initialize distTo with all nodes mapped to infinity, except start to 0

    while (there are unknown vertices):
        let u be the closest unknown vertex
        known.add(u)
        for each edge (u,v) to unknown v with weight w:
            if (v == end)
                return path // We found the goal node, we're done!
            oldDist = distTo.get(v)
            newDist = distTo.get(u) + w
            if (newDist < oldDist):
                distTo.put(v, w)
                edgeTo.put(v, u)
```

2. *Extra Practice: A few steps of Dijkstra's*

Run Dijkstra's algorithm on the graph below starting at the **vertex 0**. The table to the left shows the results right **after 2's edges have been relaxed.**

vertex	distTo	edgeTo
0	0	-
1	∞	-
2	5	4
3	1	0
4	4	0
5	8	2
6	6	4
7	2	3



a. In what order were the vertices visited before the vertex 2 was visited?

   0 , ___ , ___ , ___ , 2

   Walk through the steps of the Dijkstra's algorithm starting from 0 and ending at 2.

b. What vertex will be visited next after 2 is visited, and why?

c. Update the table after the vertex you identified from b. is visited (i.e. after all its edges are relaxed).

3. Graph Design: Printing Worksheets

Consider the following scenario:

Anna is commuting from her home to campus to teach lecture. However, before stepping foot in the lecture hall, she needs to print out the lecture worksheets. With this in mind, Anna is trying to calculate the shortest path she can take from her house to the lecture hall that includes a stop at a printer. How might she achieve this?

Assume multiple places on campus have a printer.

Assume edge weights are by distance and nodes are buildings on campus and her home.

Your solution:

4. Graph Design: Dog Walking

Consider the following scenario:

Anna is walking her dog Percy and knows Percy's favorite spots to use the bathroom. She wants to try to make it home whilst simultaneously minimizing the amount of times that she has to stop and let Percy use the bathroom. Return any 5 distinct paths Anna can take to get to her house that minimize the amount of times she has to stop and let Percy use the bathroom?

Her dog can use the bathroom multiple times

No edge weights, and we don't care about distance

Bathroom spots will be marked on the graph as special nodes

Your solution:

5. *Extra Practice - Graph Design: Maze Game*

Suppose we are trying to design a maze within a 2d top-down video-game. The world is represented as a grid, where each tile is either an impassable wall, an open space a player can pass through, or a *wormhole*. On each turn, the player may move one space on the grid to any adjacent open tile. If the player is standing on a wormhole, they can instead use their turn to teleport themselves to the other end of the wormhole, which is located somewhere else on the map.

Now, suppose that there are several coins scattered throughout the map. Your goal is to design an algorithm that finds a path between the player and some coin in the fewest number of turns possible.

1. **Describe how you would represent this scenario as a graph** (what are the vertices and edges? Is this a weighted or unweighted graph? Directed or undirected?).

2. Then, **describe how you would implement an algorithm to complete this task.**