1. Consider the following graph:

   ![Graph Image]

   a. Use Dijkstra’s algorithm to find the costs of the shortest paths from vertex \( a \) to each of the other vertices. Show your work at every step.

   b. Are any of the costs you computed using Dijkstra’s algorithm in part (a) incorrect? Why or why not?

   c. Explain how you can use Dijkstra’s algorithm to recover the actual paths from \( A \) to each of the vertices (rather than just the costs)
2. Suppose you have social network data for some people (including yourself and a famous person). Write pseudocode to find the answers to the following questions:

   a. How would you represent this social network with a graph?

   b. Given two people, how would you determine if they were friends

   c. Find the person with the most friends in the data

   d. Find the length of the shortest path from yourself to the famous person

   e. Find the number of people who do not know anyone you know (or anyone they know, etc).
3. Suppose you are given a graph G. Explain how you would figure out if it has a cycle.

   a. If the graph is undirected?

   b. If the graph is directed?

Disjoint Sets

Reminder: A disjoint sets data structure keeps track of multiple sets that do not share any elements.

UnionFind ADT:
- \textbf{find}(x) \rightarrow \text{Returns a number representing the set that } x \text{ is in}
- \textbf{union}(x, y) \rightarrow \text{Updates the sets so whatever sets } x \text{ and } y \text{ were in are now considered the same sets}

4. How can the Union-Find ADT be used to check whether an undirected graph contains cycle or not? Assume that the graph does not contain any self-loops
5. Given the following trees from our implementation of disjoint sets seen in lecture, answer the following questions (assuming both weighted-unions and path compression):

a. Draw the resulting trees after calling find(5) on the above disjoint sets. Also indicate what value will be returned by this call.

b. Draw the final result of calling union(2,6) on the result of part (a)