(Sol) HW01 Graph algorithms (1)

Due: Tuesday, Sep. 5th, 2023 on Canvas

KT 4.4

Duplicate, edit and export as a pdf to submit on Canvas. No late submission is accepted.

1. [4 points] Reliability Graph
   We are given a directed graph $G = (V, E)$ on which each edge (u, v) has an associated value $r(u, v)$, which is a real number in the range $0 \leq r(u, v) \leq 1$ that represents the reliability of a communication channel from vertex u to vertex v. We interpret $r(u, v)$ as the probability that the channel from u to v will not fail, and we assume that these probabilities are independent.

   a. [2 points] Give an efficient algorithm to find the most reliable path between the two given vertices.
      Answer: The reliability of a path is just the product of the r() values along that path. So, all we need to do is find a path from u to v that maximizes this product. We can do this by running Dijkstra's algorithm with three changes:

      i. The "distance" the product of r() values, instead of the sum of the weights

      ii. The "distance" for an unknown vertex starts at 0 instead of ∞

      iii. We're trying to maximize instead of minimize the distance

   b. [2 points] Explain why your answer finds the most reliable path.
      Answer: The most reliable path is the path with the highest probability of success. Since the probabilities are independent, the probability of success of a channel from u to v is equal to the product of of $r()$ values on the path from u to v. The modified Dijkstra's algorithm finds the maximum cost path from u to any other node, including v, so the modified Dijkstra's algorithm will find the path from u to v with the best probability of success.

2. [7 points] Connection Graph

   In LinkedIn, you can search for any public profile and see how many "connections" are between you and the public figure. For example, Vitalik Buterin is a 3rd

connection from me. One of the features of LInkedIn premium is for you to be able to see who are the common connections between you and the person you searched. For anyone who is a 2nd connection from you, you can see who are the mutual connections. For example, Andrew Ng is a 2nd connection from me and Dr. Cindi Thompson, the founder of Climate Companion is a mutual connection. For anyone who is a 3rd connection, you can see the 2 people between you and the person. For example, I am connected to Tao Sun, a USF CS alum, Tao is connected to Jason Lohrentz (I don't know him), and Jason is connected to Vitalik (I actually know him).

a. [3 points] Between BFS and Dijkstra, which one is more efficient to solve this problem? Explain why.
   Answer: BFS. There is no weight on the edges, and we only need to find one (not all) shortest path from the member to the target person. BFS only takes $O(|V| + |E|)$ while Dijkstra takes $O(|V|^2)$ or $O(|E| \log |V|)$, so BFS is more efficient.

b. [4 points] Give an efficient algorithm to compute the connection distance, i.e. compute n for the n^{th} connection.

Answer: BFS with the modification labeling each node in the graph with the distance.

```
public void bfsN (int source) {
    boolean [] visited = new boolean[vertices];
    int[] label = new int[vertices];
    bfs(source, visited, label);
}
void bfs(int v, boolean [] visited, int[] label) {
    Queue q = new Queue();
    visited[v] = true;
    q.enqueue(v);
    label[v]=0;
    while(!q.empty()) {
        v = q.dequeue();
        for (int adj : adjacent(v)) {
            if(!visited[adj]) {
                q.enqueue(adj);
                visited[adj] = true;
                label[adj] = label[v]+1;
            }
        }
    }
}
```