Problem 1. Let $G = (V, E)$ be a directed graph.

(a) Assuming that G is represented by an adjacency matrix $A[1..n, 1..n]$, give a $\Theta(n^2)$-time algorithm to compute the adjacency list representation of G. (Represent the addition of an element v to a list l using pseudocode by $l \leftarrow l \cup \{v\}$.)

(b) Assuming that G is represented by an adjacency list $\text{Adj}[1..n]$, give a $\Theta(n^2)$-time algorithm to compute the adjacency matrix of G.

Problem 2. An undirected graph is 2-colorable if each vertex can be assigned either Red or Blue so that no two vertices that share an edge have the same color.

(a) Use breadth-first-search to determine if an undirected graph $G = (V, E)$ is 2-colorable, and if so 2-color it.

(b) Use depth-first-search to determine if an undirected graph $G = (V, E)$ is 2-colorable, and if so 2-color it.

Problem 3. Let $G = (V, E, p)$ be a directed graph representing a network of roads between cities. The weight $p(e)$ is the probability that road e will be open, so that $0 \leq p(e) \leq 1$. The probabilities are assumed to be independent. You want to take a trip from city a to city b.

(a) Give an algorithm to find the route that has the most chance of being open.

(b) How fast is your algorithm?