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 $Adj[1..n]$, give a $\Theta(n^2)$-time algorithm to compute the adjacency matrix of $G$.

Problem 2. Assume that you have an undirected, weighted graph, where some weights are positive and some are negative. You would like to find a spanning tree who sum of weights on the edges is as close to zero as possible.

(a) Robert Prim suggests that you start at any vertex and grow a tree. Always include a new edge into the tree whose total weight makes the current sum of edge weights as close to zero as possible.
   (i) Does Prim’s algorithm find a desired tree?
   (ii) If so prove it. If not, give a counterexample.

(b) Joseph Kruskal suggests that you start with an empty tree and keep adding edges into the tree (that do not create a cycle) whose total weight of the edges is as close to zero as possible.
   (i) Does Kruskal’s algorithm find a desired tree?
   (ii) If so prove it. If not, give a counterexample.