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

2. Let $G = (V, E)$ be a directed, weighted graph. Show that Dijkstra's algorithm does not work if there are negative weight edges but no negative weight cycles. Make your counterexample as simple as possible.

3. (a) Show how to modify Dijkstra's algorithm to solve the single source shortest path problem on a directed, weighted graph if there is exactly one negative weight edge but no negative weight cycles.
   (b) Informally justify the correctness of your algorithm.
   (c) How efficient is your algorithm? Informally justify.