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.

2. Give a linear time, depth-first-search algorithm to find the size of the largest connected component in a graph, where size is measured by the number of vertices. (This should very similar to the algorithm covered in class.)