Algorithms: Assignment 7

Due date: November 2 (Thursday)

Problem 1 (4 points)
Write algorithms for converting (a) an adjacency-list representation of a graph into an adjacency matrix and (b) an adjacency matrix into adjacency lists. Give the time complexity of your algorithms.

Problem 2 (6 points)
Consider a directed graph with n vertices, represented by an adjacency matrix $M[1..n, 1..n]$. A vertex is called a *sink* if it has $(n-1)$ incoming edges and no outgoing edges; note that the graph can have at most one sink.

Give an algorithm that finds the sink vertex and returns its number; if the graph has no sink, it should return 0. The running time of your algorithm should be $O(n)$; if you write a slower algorithm, you will get only 2 points.

Problem 3 (bonus)
This problem is optional, and it allows you to get 2 bonus points toward your final grade.

Consider the following recurrence:

$$
\begin{aligned}
C_0 &= 0 \\
C_n &= 3 \cdot C_{n-1} + \frac{3^n}{n \cdot (n+1)} \qquad \text{(where } n \geq 1\text{)}
\end{aligned}
$$

Give a formula for C_n in terms of n, without the use of recurrence. You need to provide *an exact formula,* rather than a Θ-notation.