Algorithms: Assignment 10

Due date: November 20 (Wednesday)

Problem 1 (3 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 running time of your algorithms.

Problem 2 (3 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 $O(n)$.

Problem 3 (4 points)
Describe a data structure for representing a directed graph that supports all operations listed below. Note that we cannot use the standard adjacency lists or adjacency matrix because they do not satisfy the given time requirements.

- Check the presence of an edge between two given vertices, in $O(\lg V)$ time.
- Add the edge between two given vertices, in $O(\lg V)$ time.
- Perform the breadth-first search, starting from a given vertex, in $O(V + E)$ time.