1. Why is an adjacency list representation used in BFS? (Consider the running time of the algorithm if an adjacency matrix is used instead.)

2. An Euler tour of a connected, directed graph $G = (V, E)$ is a cycle that traverses every edge in $E$ exactly once (and each vertex at least once).
   
   (a) Prove that $G$ has an Euler tour if and only if the in-degree of $v$ is equal to the out-degree of $v$, for all $v \in V$.
   
   (b) Write a $O(m)$ time algorithm that finds an Euler tour of $G$, if one exists. (Hints: use DFS and merge a sequence of edge-disjoint cycles.)

3. Write an algorithm that detects whether an undirected graph is bipartite. This problem may also be described as the 2-color problem in which each vertex must be colored using two colors and the colors of adjacent vertices must be different. (Hint: Use BFS.)

4. Rewrite the DFS algorithm in your book so that it is iterative instead (but does exactly the same thing). (Combine the two separate functions into a single iterative function.)