Practice problems: Graph Problems

1. [CLRS 22.1-4] Given an adjacency-list representation of a multigraph \( G = (V, E) \), describe an \( O(|V| + |E|) \) time algorithm to compute the adjacency-list representation of the “equivalent” undirected graph \( G' = (V, E') \), where \( E' \) consists of the edges in \( E \) with all multiple edges between two vertices replaced by a single edge and with all self-loops removed.

2. CLRS 22.2-6

3. Consider a directed weighted graph with non-negative weights which is formed by adding an edge from every leaf in a binary tree to the root of the tree. Let the graph/tree have \( n \) vertices. An example of such a graph with \( n = 7 \) could be the following:

![Graph Example](image)

We want to design an algorithm for finding the shortest path between two vertices in such a graph.

(a) How long time would it take Dijkstra’s algorithm to solve the problem?

(b) Describe and analyze a more efficient algorithm for the problem.

4. A bipartite graph is an undirected graph \( G = (V, E) \) in which \( V \) can be partitioned into two sets \( V_1 \) and \( V_2 \) such that \( (u, v) \in E \) implies either \( u \in V_1 \) and \( v \in V_2 \) or \( u \in V_2 \) and \( v \in V_1 \); that is, all edges go between the two sets \( V_1 \) and \( V_2 \). Give an efficient algorithm to determine if an undirected graph is bipartite.
5. A *bike-chain graph* is an undirected weighted graph consisting of simple cycles called *links*. Two links have a single vertex in common. Such a vertex is called a *pin*. Pins have degree four while all other vertices have degree two. The following is an example of a bike-chain with three links and two pins.

(a) Assume that a bike-chain graph $G$ has $k$ links, $n$ vertices, and $m$ edges. Give $m$ as a function of $k$ and $n$.

(b) Describe and analyze an efficient algorithm for computing the minimal spanning tree of $G$.

6. *(CPS130 final spring 2000)* Consider a *pole-graph* which is an undirected graph with positive edge weights, consisting of two poles connected through a layer of nodes as follows:

Let $n$ be the number of vertices in a pole-graph and assume that the graph is given in normal edge-list representation.

(a) How long would it take Dijkstra’s algorithm to find the single-source-shortest-paths from one of the poles in a pole-graph to all other nodes?

(b) Describe and analyze a more efficient algorithm for solving the single-source-shortest-paths problem on a pole-graph. Remember to prove that the algorithm is correct.