Analysis of Algorithms	**Prelim Practice Questions**
CS6820 Fall 2022	**October 12, 2022**

1. Let $G = (V, E)$ be a directed graph. A *transitive reduction* or *Hasse diagram* of G is a subgraph $G_H = (V, E_H)$ with minimum number of edges such that E and E_H have the same transitive closure.
   (a) Prove that if G is acyclic, then G_H is unique.
   (b) Give an efficient algorithm to construct G_H from G in the case G is acyclic. Your algorithm should have roughly the same complexity as transitive closure. (The problem is NP-complete when G is cyclic.)

2. Show how to construct a topological sort of a directed acyclic graph using depth-first search.

3. An *Euler circuit* in a connected undirected graph $G = (V, E)$ is a cycle that traverses all edges exactly once. It may repeat vertices.
   (a) Prove that G has an Euler circuit iff G is connected and the degree of every vertex is even.
   (b) Give an $O(m)$ algorithm to find an Euler circuit if one exists. Give a proof of correctness and detailed complexity analysis.

4. Modify Dijkstra's algorithm to produce the minimum-weight paths themselves, not just their weights.

5. The following algorithm, known as *Prim's algorithm*, produces a minimum-weight spanning tree T in a connected undirected graph with edge weights. Initially we choose an arbitrary vertex and let T be the tree consisting of that vertex and no edges. We then repeat the following step $n - 1$ times: find an edge of minimum weight with exactly one endpoint in T and include that edge in T.
   (a) Argue that Prim's algorithm is correct.
   (b) Give an implementation that runs in time $O(m + n \log n)$.

6. (a) Given a flow f on a directed graph G, show how to construct the residual graph G_f in $O(m)$ time.
   (b) Using (a), show how to calculate efficiently an augmenting path of maximum bottleneck capacity. (*Hint.* Modify Dijkstra's algorithm.)

7. Give an efficient algorithm for the *s, t-connectivity problem*: Given a directed or undirected graph $G = (V, E)$ and elements $s, t \in V$, $s \neq t$, decide whether there exist at least k edge disjoint paths from s to t and find them if so.

8. Give an efficient algorithm for the *min cut* problem: Given an undirected graph $G = (V, E)$, elements $s, t \in V$, $s \neq t$, and nonnegative edge weights $w : E \to \mathbb{R}_+$, find an s, t-cut of minimum weight; that is, find a partition A, B of V with $s \in A$ and $t \in B$ minimizing

$$\sum_{(u,v) \in E \cap (A \times B)} w(u, v).$$

(Several minor variants of this problem are NP-complete. For example, the *max cut* problem is NP-complete.)

9. Show how matching can be used to give efficient algorithms for the following two problems.

   (a) Given an undirected graph with no isolated vertices, find an edge cover of minimum cardinality. (An *edge cover* is a subset of the edges such that every vertex is an endpoint of some edge in the subset.)

   (b) Find a vertex cover in a given undirected graph that is at most twice the cardinality of the smallest vertex cover. (A *vertex cover* is a set of vertices that includes at least one endpoint of every edge.)

10. Let G be a connected undirected graph. We say that G is *k-connected* if the deletion of any k vertices leaves G connected. Give a polynomial-time algorithm (ideally, $O(k^2mn)$) for testing k connectivity and for finding a set of k disconnecting vertices if G is not k-connected. (*Hint.* Use *Menger's theorem*, which states that G is k-connected if and only if any pair of vertices is connected by at least k vertex-disjoint paths, then use max flow. You need not prove Menger's theorem.