Homework 5

Problem 1. (30 points) [Multi-Source, Multi-Sink Bounded Flow]

Let $G = (V, E)$ be a directed graph where each edge $(u, v) \in E$ has a capacity $c(u, v) \geq 0$. Instead of a single source and sink, you are given a set of *source nodes* $S \subset V$ and a set of *sink nodes* $T \subset V$, with $S \cap T = \emptyset$. Each source $s_i \in S$ has a maximum flow supply $\text{Supply}(s_i)$ that can originate from it. Similarly, each sink $t_j \in T$ has a maximum flow demand $\text{Demand}(t_j)$ that can terminate at it.

The objective is to find the *maximum total flow* that can be sent from the sources in S to the sinks in T, subject to:

1. The flow on every edge respects its capacity $c(u, v)$.
2. The total flow leaving any source $s_i \in S$ must be at most $\text{Supply}(s_i)$.
3. The total flow entering any sink $t_j \in T$ must be at most $\text{Demand}(t_j)$.

Design an efficient algorithm that computes this maximum total flow. Describe your solution, write pseudocode, analyze the time complexity, and prove the correctness of your algorithm.

Problem 2. (30 points)[Flow Reduction with Unit Capacities]

You are given a directed graph $G = (V, E)$ with a source $s \in V$, a sink $t \in V$, and unit capacities $c(e) = 1$ for all $e \in E$. You are also given an integer parameter $k \geq 0$. The goal is to delete exactly k edges so as to make the maximum s–t flow in the remaining network as small as possible. Formally, find $F \subseteq E$ with $|F| = k$ that minimizes the maximum s–t flow value in $G' = (V, E \setminus F)$.

Design an efficient algorithm that find this set F that minimize the max flow. Describe your solution, write pseudocode, analyze the time complexity, and prove the correctness of your algorithm.

Problem 3. (40 points) Minimum Stations to Close

You are given an undirected, unweighted graph $G = (V, E)$ representing a rail network, where each vertex corresponds to a train station and each edge represents a direct rail connection between two stations. Two special vertices $A, B \in V$ represent the infected town and your hometown, respectively.

To stop the spread of infection, you can *close* certain stations (that is, remove vertices from the graph) so that no train can travel from A to B. However, you are *not allowed* to close A or B themselves. You may assume that there is no edge connecting A and B. The goal is to determine the smallest number of stations that must be closed so that there is no remaining path from A to B.

In this problem, you will design an algorithm to compute this minimum number by reducing it to a *maximum flow / minimum cut* problem. You need to describe a reduction that transforms each instance of the station closing problem into an instance of the maximum flow / minimum cut problem. Describe your solution, provide pseudocode for the reduction, and analyze its time complexity. Moreover, you must prove the correctness of your reduction by explaining how each solution in one problem corresponds to a solution in the other (the bidirectional relationship we discussed in class).