CIS4930 – Problem Solving **Spring 2026**

Homework 3: Deadline Tuesday 4/21/2026

Instructor: Viet Tung Hoang

1. (80 points) Let n be an even number, and let G be the complete undirected graph with n vertices (namely, every two vertices are connected). Prove that the edges of G can be partitioned into exactly $n/2$ spanning trees. (A spanning tree is a connected subgraph that contains all vertices and no cycles.)

[**We expect a rigorous proof.**]

2. (80 points) We are given a directed graph $G = (V, E)$ on which each edge $(u, v) \in E$ has an associated capacity $c(u, v) \geq 0$ that represents the rate of data that we can send in the link (u, v). The graph has $n = |V|$ nodes and $m = |E|$ edges, and assume that $m \geq n$. Let s and t be two nodes in the graph. For a path $s = u_0 \to u_1 \cdots \to u_k = t$, its *bottleneck* is

$$\min\{c(u_0, u_1), \ldots, c(u_{k-1}, u_k)\} \ .$$

Informally, the bottleneck of a path is the limitation of the rate of data that you can send along that path.

a) (40 points) Give an algorithm of $O(m + n)$ time that, given a number $B \geq 0$, finds a path from s to t whose bottleneck is at least B. If many such paths exist, you only need to find one.

[**We expect an informal English description of your algorithm. You need to justify its correctness and running time.**]

b) (40 points) Using part (a), design an $O(m \log(n))$ algorithm that finds the *widest* path from s to t, meaning a path from s to t of the largest bottleneck possible. (Note: Recall that $m = O(n^2)$, so $\log_2(m) = O(\log(n))$.)

[**We expect an informal English description of your algorithm. You need to justify its correctness and running time.**]

3. (80 points) Let $G = (V, E, w)$ be a connected, weighted undirected graph. Assume that the weights are strictly positive. A set $F \subseteq E$ of edges is called a *feedback-edge set* if every cycle of G has at least one edge in F. Design an efficient algorithm to find a minimum-weight feedback-edge set. Analyze the running time of your algorithm.

[**We expect an informal English description of your algorithm. You need to justify its correctness.**]

Hint: For a feedback-edge set F, the *complement feedback graph* G_F is the subgraph of G with the same node set V, but edge set $E \backslash F$. That is, G_F is obtained by deleting edges of F in G. In order to pick a feedback-edge set F of minimum weight, we need to find a complement feedback graph G_F of maximum weight.