Algorithm Design and Analysis (Fall 2021)
Assignment 5
Due: Sunday, May 29, 2022

1. (35 points) In this question, we will prove König-Egerváry Theorem, which states that, in any bipartite graph, the size of the maximum matching equals to the size of the minimum vertex cover. Let $G = (V, E)$ be a bipartite graph.

   (a) (5 points) Explain that the following is an LP-relaxation for the maximum matching problem.

   $$\begin{aligned} \text{maximize} \quad & \sum_{e \in E} x_e & \\ \text{subject to} \quad & \sum_{e:e=(u,v)} x_e \le 1 & (\forall v \in V) \\ & x_e \ge 0 & (\forall e \in E) \end{aligned}$$

   (b) (5 points) Write down the dual of the above linear program, and justify that the dual program describes the fractional version of the minimum vertex cover problem.

   (c) (10 points) Show by induction that the *incident matrix* of a bipartite graph is totally unimodular. (Given an undirected graph $G = (V, E)$, the incident matrix A is a $|V| \times |E|$ zero-one matrix where $a_{ij} = 1$ if and only if the i-th vertex and the j-th edge are incident.)

   (d) (10 points) Use results in (a), (b) and (c) to prove König-Egerváry Theorem.

   (e) (5 points) Give a counterexample to show that the claim in König-Egerváry Theorem fails if the graph is not bipartite.

2. (30 points) You and your $n-1$ roommates are always sharing expenses (bills, groceries, pizza, etc.) but it's terribly inconvenient to split each bill equally. You agree that each bill should be paid by one person (a possibly different one for each bill) who writes down what a subset of the roommates owe him. At the end of the year you aggregate everything in a *debt network* $G = (V, E, w)$ where $V = \{1, \ldots, n\}$ and $w(u, v)$ is the net (positive) amount that u owes v. If u owes v nothing then $(u, v) \notin E$. Prove that all debts can be settled with at most $n - 1$ person-to-person payments *such that if u pays v then $(u, v) \in E$.*

   (Hint: Prove that the debt network can be equivalently transform to a network with at most $n - 1$ edges. Some intuitions from Ford-Fulkerson algorithm may be helpful.)

3. (35 points) The network flow problem only restricts the capacity of each edge. Consider the following variant, each edge e does not only have a capacity c_e, but also a demand d_e. Finally, the edge should have flow $d_e \leq f_e \leq c_e$. Please find out how to solve the following problems by reducing them to the original max flow problem.

   (a) (20 points) In the original network flow problem, finding a feasible flow is easy. (a zero flow is surely feasible.) However, whether there exists a feasible flow is not straightforward in this new variant. How to determine the existence of a feasible flow by using the original max flow algorithm? (i.e., each f_e should satisfy $d_e \leq f_e \leq c_e$ and the flow conservation constraint should hold at all vertices other than s and t.)

   (b) (15 points) Based on the previous part, can we further find a maximum feasible flow? Please also use the original max flow algorithm.

4. How long does it take you to finish the assignment (including thinking and discussion)? Give a score (1,2,3,4,5) to the difficulty. Do you have any collaborators? Please write down their names here.