1. Prove that a tree has at most one perfect matching.

2. Consider a flow network $G = (V, E)$ with source $s$, sink $t$ and non-negative edge capacities, $w : E \to \mathbb{R}_{\geq 0}$. Suppose $(S, S^c)$ and $(T, T^c)$ are both minimum $s-t$ cuts in $G$. Show that both $(S \cup T, (S \cup T)^c)$ and $(S \cap T, (S \cap T)^c)$ are also minimum $s-t$ cuts in the network.

3. **Kleinberg and Tardos 7.9** Network flow issues come up in dealing with natural disasters and other crises, since major unexpected events often require the movement and evacuation of large numbers of people in a short amount of time.

   Consider the following scenario. Due to large-scale flooding in a region, paramedics have identified a set of $n$ injured people distributed across the region who need to be rushed to hospitals. There are $k$ hospitals in the region, and each of the $n$ people needs to be brought to a hospital that is within a half-hour’s driving time of their current location (so different people will have different options for hospitals, depending on where they are right now).

   At the same time, one doesn’t want to overload any one of the hospitals by sending too many patients its way. The paramedics are in touch by cell phone, and they want to collectively work out whether they can choose a hospital for each of the injured people in such a way that the load on the hospitals is balanced: Each hospital receives at most $\lceil n/k \rceil$ people.

   Reduce this problem to a maximum-flow problem, for which you know how to solve.

4. **Kleinberg and Tardos 7.27** Some of your friends with jobs out West decide they really need some extra time each day to sit in front of their laptops, and the morning commute from Woodside to Palo Alto seems like the only option. So they decide to carpool to work.

   Unfortunately, they all hate to drive, so they want to make sure that any carpool arrangement they agree upon is fair and doesn’t overload any individual with too much driving. Some sort of simple round-robin scheme is out, because none of them goes to work every day, and so the subset of them in the car varies from day to day.

   Here’s one way to define fairness. Let the people be labeled $\{p_1, \ldots, p_k\}$. We say that the total driving obligation of $p_j$ over a set of days is the expected number of times that $p_j$ would have driven, had a driver been chosen uniformly at random from among
the people going to work each day. More concretely, suppose the carpool plan lasts for $d$ days, and on the $i$-th day a subset $S_i \subset S$ of the people go to work. Then the above definition of the total driving obligation $\Delta_j$ for $p_j$ can be written as $\Delta_j = \sum_{i: p_j \in S_i} \frac{1}{|S_i|}$. Ideally, we would like to require that $p_j$ drives at most $\Delta_j$ times; unfortunately, $\Delta_j$ may not be an integer.

So let’s say that a driving schedule is a choice of a driver for each day—that is, a sequence $p_{i_1}, p_{i_2}, \ldots, p_{i_d}$ with $p_{i_t} \in S_{i_t}$—and that a fair driving schedule is one in which each $p_j$ is chosen as the driver on at most $\lceil \Delta_j \rceil$ days.

- Reduce this problem to a max-flow problem.
- Prove that for any sequence of sets $S_1 \ldots S_d$ driving schedule, there exists a fair driving schedule.

5. **Kleinberg and Tardos 7.12** You are given a flow network with unit capacity edges: it consists of a directed graph $G = (V, E)$, a source $s \in V$ and a sink $t \in V$; and $c_e = 1$ for every $e \in E$. You are also given a parameter $k$.

The goal is to delete $k$ edges so as to reduce the maximum $s - t$ flow in $G$ by as much as possible. In other words, you should find a set of edges $F \subseteq E$ so that $|F| = k$ and the maximum $s - t$ flow in $G' = (V, E \setminus F)$ is as small as possible. Give a polynomial time algorithm to solve this problem.