# CS 473: Algorithms, Fall 2010
# HBS 10

**Problem 1**. **[k-Regular Bipartite Graphs]**
A $k$-Regular graph is an undirected graph where every vertex has degree $k$. We will prove that if a bipartite graph is k-Regular, then it has a perfect matching. First, recall the following definitions:

*Bipartite Graph*: a graph whose vertices are partitioned into two independent sets, $L$ and $R$.
*Matching*: A matching in a graph $G$ is a set of edges such that no two edges share a common vertex.
*Neighbors*: Let $v$ be a vertex. The neighbors of $v$, denoted by $N(v)$ are the set of vertices connected to $v$.
*Hall's Theorem*: Let $G = (L \cup R, E)$ be a bipartite graph where $|L| = |R|$. Then $G$ has a perfect matching if and only if for every subset $X \subseteq L$, $|N(X)| \geq |X|$.

For the following problems, let $G = (L \cup R, E)$ be a $k$-regular bipartite graph where $|L| = |R|$.

1. Show that the $G$ has a perfect matching via Hall's theorem.
2. Now, construct a flow network $G'$ from $G$ such that the value of the maximum flow in $G'$ is equal to the size of the perfect matching in $G$. *Hint: The flow can be fractional.*

**Problem 2**. **[Dinner Scheduling**
Consider a group of $n$ people who are trying to figure out a dinner schedule over the next $n$ nights where each person needs to cook exactly once. Everyone has scheduling conflicts with some of the nights, so deciding who should cook on which night becomes tricky.

Label the people $\{p_1, \ldots, p_n\}$ and the nights $\{d_1, \ldots, d_n\}$. For each person $p_i$, there's a set of nights $S_i \subset \{d_1, \ldots, d_n\}$ when they are *not* able to cook.

A *feasible dinner schedule* is an assignment of each person to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if $p_i$ cooks on night $d_j$, then $d_j \notin S_i$.

1. Describe a bipartite graph $G$ so that $G$ has a perfect matching if and only if there is a feasible dinner schedule for the group.
2. After generating a schedule, they realize there is a problem. $n - 2$ of the people are assigned to different nights on which they are available: no problem there. However, for the other two people $p_i$ and $p_j$, and the other two days, $d_k$ and $d_l$, both $p_i$ and $p_j$ are assigned to cook on night $d_l$. Show that it's possible to decide in $O(n^2)$ time whether there exists a feasible dinner schedule using the "almost correct" schedule.