Homework 1 **Graphs and Algorithms**

Matchings, Hall's theorem, Flows

Date: 5/2/2016 Due: 24/2/2016

1. **Picking disjoint subsets:** Let A be a finite set with m elements, and let $A_1, \ldots, A_n$ be subsets of A. Let $b_1, \ldots, b_n$ be positive integers. We would like to pick subsets $B_i \subseteq A_i$ such that $|B_i| = b_i$ and the B_i's are disjoint (i.e. $B_i \cap B_j = \emptyset$ for all $1 \leq i < j \leq n$.)

   (a) (7 pts) Show that this possible if and only if it holds that

   $$| \cup_{i \in I} A_i| \geq \sum_{i \in I} b_i$$

   for all subsets of indices $I \subseteq \{1, \ldots, n\}$.

   (b) (3 pts) How do we find such B_i's in polynomial time (or detect that they do not exist).

2. (10 pts) **Orienting Edges:** Given an undirected graph G, show that its edges can be oriented such that the indegree and outdegree at any vertex differ by at most 1.

   [Hint: Euler tours]

3. (10 pts) **Matchings:** Let G be a bipartite graph with vertex parts X and Y, such that $d(x) \geq 1$ for all $x \in X$ and $d(x) \geq d(y)$ for all edges $(x, y) \in E$, where $x \in X$ and $y \in Y$. Show that G has a matching covering every vertex of X.

4. **Graph Orientations:** Given an undirected graph and an integer k, we wish to orient the edges in such a way that each vertex has at most k incoming edges.

   (a) (5 pts) Show that this can be done using max-flow.

   (b) (5 pts) Use max-flow min-cut to show that such an orientation exists if and only if for every subset of vertices S, the number of edges with both endpoints in S is at most $k|S|$.

5. (10 pts) **Distributed Computation on two processors:** There are n jobs (processes) and two processors. Job i runs in a_i time on processor 1 and in b_i time on processor 2. The jobs also communicate among each other. So, if two jobs i and j are assigned to different processors, this is not desirable and an overhead of c_{ij} is incurred.

   Find an assignment of jobs to machines that minimizes the total execution time plus the total overhead. So if S is the set of jobs assigned to processor 1 and T is the set of jobs assigned to processor 2, then the objective is

   $$\sum_{i \in S} a_i + \sum_{j \in T} b_j + \sum_{(i,j): i \in S, j \in T} c_{ij}.$$

   [Hint: Formulate this as a min-cut problem on an undirected graph]