MS&E 112/212: Combinatorial Optimization
Instructor: Professor Amin Saberi (saberi@stanford.edu)
HW#3 – Due Feb 28, 2017

Homework drop-off: You can turn in your homework in the class or email it to naskoch@stanford.edu.

Collaboration policy: you can only collaborate with ONE registered student. You may not discuss the homework (this includes clarifications, solutions, etc) with anyone but the instructor, the course assistant or the person you picked. Please write the name of your collaborator in your homework.

1. **Kleinberg and Tardos 11.6**

   Recall the basic load balancing problem from section 11.1, we're interested in placing jobs on machines so as to minimize makespan (the maximum load on any one machine). In a number of applications, it is natural to consider cases in which you have access to machines with different processing power, so that a given job may complete more quickly on one of your machines than on another. The question then becomes: How should you allocate jobs to machines in these more heterogeneous systems?

   Here's a basic model that exposes these issues. Suppose you have a system that consists of m slow machines and k fast machines. The fast machines can perform twice as much work per unit time as the slow machines. Now you're given a set of n jobs; job i takes time t_i to process on a slow machine and time $\frac{t_i}{2}$ on a fast machine. You want to assign each job to a machine; as before, the goal is to minimize makespan.

   Give a polynomial-time algorithm that produces an assignment of jobs to machines with a makespan that is at most three times the optimum.

2. **Kleinberg and Tardos 11.9** Consider the following maximization version of the 3-Dimensional Matching Problem. Given disjoint sets X, Y, and Z, and given a set $T \subseteq X \times Y \times Z$ of ordered triples, a subset $M \subseteq T$ is a *3-dimensional matching* if each element of $X \cup Y \cup Z$ is contained in at most one of these triples. The *Maximum 3-Dimensional Matching Problem* is to find a 3-dimensional matching M of maximum size.

   Give a polynomial-time algorithm that finds a 3-dimensional matching of size at least 1/3 times the maximum possible size.

3. Let G be a complete undirected graph in which all edge lengths are either 1 or 2 (clearly, G satisfies the triangle inequality). Give a 4/3 factor algorithm for the traveling salesman problem in this special class of graphs. (Hint: start by finding a minimum cost perfect matching).