# Homework 4

Due: June 30th, 2009

1. **(6)**
   Give an optimal algorithm to solve $(P|pmtn|C_{max})$, that is, minimizing makespan on identical parallel machines when preemption is allowed. Justify optimality. What is the running time of your algorithm? *(Hint: Think of the lower bounds done in class)*

2. **(6+6)**

   (a) We saw that List-Scheduling returned a schedule with $C^S_{max} \leq (2 - 1/m)OPT$. Find an example of an ordering of jobs such that the above inequality holds with equality. Recall we did for $m = 2$ in class. For partial credit do $m = 3, 4$.

   (b) We saw that LPT returned a schedule with $C^S_{max} \leq (4/3 - 1/3m)OPT$. Find an example of an ordering of jobs such that the above inequality holds with equality. Recall we did $m = 2$ in class. For partial credit do $m = 3, 4$.

   *(Hint: Note that if the inequality is tight, all the inequalities used in the analysis must be tight. Use this to construct examples)*

3. **(6)**
   Consider the LPT algorithm for $(P||C_{max})$. If the machine $i$ which processes the job which finishes last, processes $k$ other jobs, then prove that LPT returns a factor $\frac{k+1}{k}$ approximation algorithm.

4. **(1+5)**
   In class, we saw that list scheduling (using any order of the jobs) is a 2-approximation for $P||C_{\max}$. Consider the problem with l release dates, $P|r_j|C_{\max}$.

   (a) Generalize the lower bound $OPT \geq p_j \ \forall j$ to take into account the release times $r_j$ for the jobs.

   (b) Consider the following list scheduling rule:

   Whenever a machine becomes available, schedule an *available*, unprocessed job.

   Show that this is a factor 2-approximation for $P|r_j|C_{\max}$.
   *(Hint: Let $\ell$ be the last job to finish. Try to modify the proof done in class for normal list scheduling using the bound in a) and considering the times between $r_\ell$ and $t_\ell$, the release time and the time when $\ell$ actually starts being processed.)*