CSC 2420 Fall 2012, Assignment 1
Due: October 18 at start of class

1. Consider Graham's LPT algorithm for makespan minimization on m identical machines. The proof that $LPT(\mathcal{J}) \leq \frac{4}{3} - \frac{1}{3m}$ proceeds by contradication. Show that if the approximation does not hold for some input set $\mathcal{J}$, then the smallest processing time $p_n > OPT/3$. (After this the proof argues that ther are at most 2 items per machine and any such configuration of items can be modified to be the solution produced by the LPT algorithm.)

2. Consider the knapsack problem with input items $\{(v_1, s_1), \ldots, (v_n, s_n)\}$ and capacity C. WLG the sizes s_j of all items are at most C. Let $Greedy_{pd}(\mathcal{I})$ be the greedy algorithm that first sorts the items $I_j = (v_j, s_j)$ so that $\frac{v_1}{s_1} \geq \frac{v_2}{s_2} \ldots \geq \frac{v_n}{s_n}$ and then accepts items greedily (i.e. as long as they fit). Consider the following algorithm:

   Let $i^* = argmax_i\{v_i | i = 1 \ldots n\}$ and let $A = \{I_{i^*}\}$ and $B = Greedy_{pd}(\mathcal{I})$ Return the better of the two solutions. Show that this algorithm is a 2-approximation of the optimal.

   - Bonus question: Instead of the partial enumertion PTAS for the knapsack problem, can we obtain a PTAS by enumeratiing over all feasible subsets (of size at most $k = \frac{1}{\epsilon}$) of the largest valued items (rather than all subsets of size at most k) and then accepting greedily?

3. Consider the following scheduling problem. We have n jobs $\mathcal{J} = J_1, \ldots J_n$ with $J_i = (d_i, p_i, v_i)$ where d_i (resp. p_i, v_i) is the deadline (resp. processing time, value) of job J_i. We can assume all paramters are positive integers. A schedule is a mapping $\sigma : \mathcal{J} \rightarrow \mathbb{N} \cup \{\infty\}$ where $\sigma(J_i) = t_i \in \mathbb{N}$ means that job J_i begins processing at time t_i (and then ends at time $t_i + p_i$) and $\sigma(J_i) = \infty$ means that job J_i is not scheduled. A schedule is feasible if all scheduled jobs complete by their deadlines and scheduled jobs do not overlap. (We can say that a job ending at time t and one starting at time t do not overlap.) Show how to use dynamic programming and scaling to provide a FPTAS for this problem.

What is the (asymptotic) time complexity of your algorithm in terms of n and ϵ where you can assume that all arithmetic operations take one step?

4. Suppose we are interested in the makespan problem in the related machines model. Say we want to compute the makespan when there are m_1 machines running at speed $s_1 = 1$ and m_2 machines running at speed $s_2 > 1$. Show how to modify the optimal DP for the identical machines model when there are only a fixed number d different processing times so as to provide an optimal algorithm for the related machines model under the same assumption of having only d different processing times. What is the (asymptotic) time complexity of your algorithm in terms of n, m_1, m_2 and d where n is the number of jobs?

5. Bound the maximum number of iterations for the Jump local search to find a local optimum when the algorithm moves a job J_k it alwsys moves that job to a least loaded machine.

6. Consider the makespan problem for the restricted machines model. Suppose all jobs have processing time/load = 1. Show how to use a max flow algorithm to achieve an optimal makespan solution.