Due: Wednesday, Jan 29, beginning of lecture

NOTE: Each problem set only counts 5% of your mark, but it is important to do your own work. These assignments will be followed by term tests, each worth 15% of your final grade. You may consult with others concerning the general approach for solving problems on assignments, but you must write up all solutions entirely on your own. Anything else is plagiarism, and is subject to the University’s Code of Behavior.

1. Consider the unit profit job scheduling problem where the $i^{th}$ job has release time $r_i$, processing time $p_i$ and deadline $d_i$. All parameters are positive integers. A schedule is a mapping $\sigma : \{1, \ldots, n\} \to \{1, \ldots, \max_j d_j\} \cup \{\infty\}$. That is, $\sigma$ schedules job $j$ to start at time $\sigma(j)$ if $\sigma(j) \neq \infty$ and does not schedule jobs with $\sigma(j) = \infty$. A feasible schedule $\sigma$ satisfies the properties:
   1) $\sigma(j) \neq \infty$ and $\sigma(k) \neq \infty$ implies $[\sigma(j), \sigma(j) + p_j] \cap [\sigma(k), \sigma(k) + p_k] = \emptyset$
   2) $\sigma(j) \neq \infty$ implies $r_j \leq \sigma(j)$
   3)$\sigma(j) \neq \infty$ implies $\sigma(j) + p_j \leq d_j$.

Define the ECT greedy algorithm for job scheduling on one processor as follows: In each iteration the algorithm gives highest priority to that job (if any) which has the earliest possible completion time consistent with jobs already scheduled. When a job can be scheduled, ECT schedules it as early as possible. (ECT for job scheduling is a generalization of EFT for activity scheduling.)

The objective is to maximize the number of jobs scheduled in a feasible schedule $\sigma$.

(a) Provide pseudo code (similar to that presented in the notes) for the ECT job scheduling algorithm.

(b) Show that there is some input $I$ for which $|OPT(I)| \geq 2 \cdot |ECT(I)|$ where $|A(I)|$ denotes the number of jobs scheduled by $A$.

(c) Show that ECT always achieves a 2-approximation of the optimal; that is, $\forall I |OPT(I)| \leq 2 \cdot |ECT(I)|$.

2. We consider the following “unique edge cost” modification of the MST (minimum spanning tree problem). We are given a graph $G = (V, E)$ and edge costs $c : E \to \mathbb{R}$. Suppose the graph satisfies the property that there exists a MST $T$ in which no two edges have the same cost. Prove that Kruskal’s algorithm constructs such a MST.

3. Consider the following multiple copy knapsack problem. Given a set of $n$ items with positive integer weights $w_1, \ldots, w_n$ and positive values or gains $g_1, \ldots, g_n$ and a knapsack of capacity $C$, construct a sequence $S$ of non-negative integer multiplicities $m_1, \ldots, m_n$ such that
   - $\sum_{1 \leq i \leq n} m_i \cdot w_i \leq C$ i.e. $S$ is a feasible sequence
   - The sequence $S$ produces the maximum value $g(S)$ for all feasible sequences where $g(S) = \sum_{1 \leq i \leq n} m_i \cdot g_i$. 

Design a dynamic programming solution with complexity $O(nC)$ for this problem by

(a) Defining an appropriate semantic array $A$ and show how to compute the optimal value from $A$.

(b) Giving an equivalent recursively defined computational array $\tilde{A}$.

(c) Prove that $A = \tilde{A}$

(d) Show how to obtain an optimal solution.

Hint: A one dimensional array $A$ will suffice.