CMPT-405/705

January 13, 2016

ASSIGNMENT 1

Problem 1

Assume we have a set of n jobs to execute, each of which takes unit time. At any time we can serve exactly one job. Job i, $1 \leq i \leq n$ earns us a profit $g_i > 0$ if and only if it is executed no later than time d_i.

We call a set of jobs *feasible* it there exists at least one sequence that allows each job in the set to be performed no later than their respective deadline. It is easy to show that a set of jobs is feasible if and only if the sequence "earliest deadline first" is feasible.

Show that the following greedy algorithm is optimal: Add in every step the job with the highest value of g_i among those not yet considered, provided that the chosen set of jobs remains feasible.

Hint: show first that it is always possible to re-schedule two feasible sequences (one computed by Greedy) in a way that every job common to both sequences is scheduled at the same time. Of course, this new sequence might contain gaps.

Problem 2

The *fractional Knapsack problem* is defined as follows. A knapsack with total capacity C and n items with value v_i and weight w_i, $1 \leq i \leq n$, are given. The goal is to choose $x_1 \dots x_n$ between 0 and 1, such that $\sum_{i=1}^{n} v_i x_i$ is maximized and $\sum_{i=1}^{n} w_i x_i \leq C$.

Develop a greedy algorithm for that problem and show that it solves the problem optimally.

Problem 3

- Prove that a binary tree that is not full cannot correspond to an optimal prefix code. (A binary tree is said to be full if each internal vertex has exactly two children.)
- Show by induction that the number of degree-2 nodes in any non-empty binary tree is 1 less than the number of leaves.

Problem 4

A *clique* in a graph $G = (V, E)$ is a set of vertices $U \subseteq V$ which form a complete subgraph of G. (So *all* edges between vertices in U are present in G.)

Consider the following GREEDY algorithm for finding the maximum size clique in a graph. (1.) Delete from the graph a vertex that is not connected to every other vertex. (2.) Repeat (1) until the remaining graph is a clique.

Give an example to show this does not always give the the optimal solution.