Assignment 2: CS21003 Algorithms 1

Prof. Partha Pratim Chakrabarti and Palash Dey
Indian Institute of Technology, Kharagpur

Submit by 11:59 PM of February 22, 20201

1. There are n jobs $J_1, J_2, \ldots, J_n$ with duration $a_1, a_2, \ldots, a_n \in \mathbb{N}^+$ and deadlines $d_1, d_2, \ldots, d_n \in \mathbb{N}$ respectively. All the jobs are available at the beginning (that is at time 0). If a job $J_\ell, \ell \in [n]$ is completed at time t_ℓ with $t_\ell > d_\ell$, then the utility u_ℓ incurred for job J_ℓ is -1 (which is a penalty); if $t_\ell \leqslant d_\ell$, then $u_\ell = 1$ (which is an award). You have only one machine. So, you cannot process more than one jobs simultaneously. Every job, once started, must run till it finishes. The goal is to find the schedule for these n jobs (that is a sequence accordinig to which these n jobs will be performed) which maximizes $\sum_{\ell=1}^{n} u_\ell$.

   (a) Show, by exhibiting a counter-example, that the greedy algorithm for performing the shortest job first and then recurse does not always output an optimal solution.

   [3 Marks]

   (b) Design a greedy algorithm for the above problem. Clearly explain your algorithm, formally prove its correctness, and analyze its time complexity.

   [2+3+2 Marks]

2. There is a large supply of a set of two types of rods R, where each type 1 rod is of length r_1 with cost c_1 and each type 2 rod is of length r_2 with cost c_2. You are given a set of n required cut-pieces $S = \{x_1, x_2, \ldots, x_n\}$ where each cut-piece x_i is less than both r_1 and r_2. You are required to select the required number of rods of each type to accommodate all the pieces of S so that the total cost of rods used is minimum. Given that, if a rod is partly used, the full rod cost has to be paid, develop an optimal algorithm with the following:

   (a) Present a suitable recursive definition to find the number of rods of each type. Analyse your recursive definition in terms of identical sub-problems produced, if any.

   (b) Present both top-down and iterative Dynamic Programming based algorithms for the recursive definition. Analyse the time and space complexity of your algorithms.

   (c) Show the working of the iterative algorithm on a non-trivial example of at least 15 elements clearly indicating all storage elements memoized and used.

   [(2+1) + (2+2+1) + 2 = 10 Marks]