CS345 : Algorithms II
Semester I, 2016-17, CSE, IIT Kanpur

Assignment 4

Deadline : 6:00 PM, Friday, 7 October 2016

Important Guidelines:

- It is only through the assignments that one learns the most about the algorithms and data structures. You are advised to refrain from searching for a solution on the net or from a notebook or from other fellow students. Before cheating the instructor, you are cheating yourself. The onus of learning from a course lies first on you and then on the quality of teaching of the instructor. So act wisely while working on this assignment
- There are two exercises in this assignments. Each exercise has two problems- one *easy* and one *difficult*. Submit exactly one problem per exercise. It will be better if a student submits a correct solution of an easy problem that he/she arrived on his/her own instead of a solution of the difficult problem obtained by hints and help from a friend. Do not try to be so greedy :-).

1

Attempt exactly one of the two problems.

1.1 Bridges across a river

(marks=30)

This is the 3rd exercise in the Practice sheet. You need to design the algorithm for this problem based on dynamic programming and analyze its time complexity.

1.2 A job scheduling problem

(marks=50)

There are n jobs, each with a deadline d_i and a required processing time t_i. All jobs are available to be scheduled starting at time s. For a job i to be done, it needs to be assigned a period from $s_i \geq s$ to $f_i = s_i + t_i$. Different jobs should be assigned non-overlapping intervals.

We will say that a subset J of the jobs is schedulable if there is a schedule for the jobs in J so that each of them finishes by its deadline. The aim is to compute a schedulable subset of maximum size and also output a schedule for this subset that allows each job to finish by its deadline.

1. Prove that there is an optimal solution J in which the jobs in J are scheduled in the increasing order of their deadlines.

2. Assume that all deadlines d_i and required times t_i are positive integers. Give an algorithm to find an optimal solution. The running time should be polynomial in the number of jobs n, and the maximum deadline $D = \max_i d_i$.

2

Attempt exactly one of the following questions.

2.1 Bag full of boxes

(marks=30)

There are n boxes $b_1, \ldots, b_n$. Box b_i has volume v_i which is a positive integer and it carries a profit p_i. There is a bag with volume m which is also a positive integer. We wish to pick a subset of boxes and put them into the bag to achieve the maximum profit. Notice that the total volume of all the boxes picked must not exceed the volume of the bag. Design a dynamic programming based algorithm to find the subset of boxes that should be picked for the bag to achieve the maximum profit.

2.2 Buying and selling shares multiple times

(marks=50)

There are n consecutive days of a given stock. The days are numbered $i = 1, 2, \ldots, n$; for each day i, there is a price $p(i)$ per share for the stock on that day. For certain (possibly large) values of k we want to study what is called *k-shot strategy*. A k-shot strategy is a collection of m pairs of days $(b_1, s_1), \ldots, (b_m, s_m)$, where $0 \leq m \leq k$ and

$$1 \leq b_1 < s_1 < b_2 < s_2 < \cdots < b_m < s_m \leq n.$$

We view these as a set of up to k non-overlapping intervals, during each of which the investors buy 1000 shares of the stock (on day b_i) and then sell it (on day s_i). The return of a given k-shot strategy is simply the profit obtained from the m buy-sell transactions, namely,

$$1000 \sum_{i=1}^{m} (p(s_i) \; - \; p(b_i))$$

We want to assess the value of k-shot strategies by running simulations on their n-day trace of stock price. Our goal is to design an efficient algorithm that determines, given the sequence of prices, the k-shot strategy with the maximum possible return. Since k may be relatively large in these simulations, the running time of the algorithm should be polynomial in both n and k; it should not contain k in the exponent.