CMSC 451: Project, Spring 2002

The project, described in more detail below, is in two parts. The first part is due April 23rd, Tuesday, at the beginning of class. The second part is due May 9th, Thursday, at the beginning of class. These are firm deadlines.

1 The problem

You own a broadcasting company which wants to decide at what times to judiciously broadcast *one specific* news-item, in the following setting. The inputs to the problem are:

(I1) positive integers k and T, with $k \leq T$; and

(I2) a sequence of non-negative integers $a_0, a_1, \ldots, a_{T-1}$.

In our problem, a_0 requests for the news-item arrive at time 0, a_1 requests arrive at time 1, ..., a_{T-1} requests arrive at time $(T-1)$. You are allowed to broadcast the news-item at most k times; furthermore, you can only broadcast at the set of times $S = \{0.5, 1.5, \ldots, T-0.5\}$. For example, if $k = 3$ and $T = 6$, one possible choice for you is to broadcast at times $1.5, 4.5$, and 5.5. If a request arrives at some t and if the earliest time that you choose to broadcast after time t is at some time t', then we say that this request is *satisfied* at time t', and that this request had a *waiting time* of $t' - t$. For instance, if a request arrived at time 3 and if the first broadcast *after time* 3 was at time 5.5, then this request has a waiting time of 2.5.

What is your objective function? Recall that we have some given requests: a_i requests are assumed to arrive at time i, for $i = 0, 1, \ldots, T-1$. You have to choose k times from S at which to broadcast, such that:

(P1) every given request is satisfied (i.e., for every request, there is at least one broadcast after that request arrived); and

(P2) the total waiting time, summed over all the requests, is minimized.

In other words, subject to your constraint that you can broadcast only k times, you want to service all requests, and also minimize their total waiting time.

As an example, suppose $T = 6$, $k = 3$, and $a_0 = 3$, $a_1 = 0$, $a_2 = 5$, $a_3 = 1$, $a_4 = 4$, and $a_5 = 0$. Then, broadcasting at times $0.5, 2.5, 3.5$ is not valid, since there were some requests that arrived at time 4 that did not get satisfied. Instead, suppose we broadcast at times $1.5, 2.5$, and 4.5. Then,

- each of the 3 requests that arrived at time 0 had a waiting time of 1.5;
- each of the 5 requests that arrived at time 2 had a waiting time of 0.5;
- the request that arrived at time 3 had a waiting time of 1.5; and
- each of the 4 requests that arrived at time 4 had a waiting time of 0.5.

Thus, the total waiting time for this valid schedule is $3 \times 1.5 + 5 \times 0.5 + 1 \times 1.5 + 4 \times 0.5 = 10.5$. There are many other valid schedules for this problem, each with its own minimum waiting time.

Given a general instance of such a problem (via the values specified in (I1) and (I2) above), the objective is to find a valid schedule with minimum total waiting time; we wish to develop an efficient algorithm for this problem, given the inputs k, T, and $a_0, a_1, \ldots, a_{T-1}$.

2 Part I of the Project

This part is worth 10 points in total, and is due April 23rd, Tuesday, at the beginning of class.

(i) Given an instance of the above problem, suppose i is the largest integer such that $a_i > 0$. Show that any optimal solution will broadcast at time $i + 0.5$. **(2 points)**

(ii) Develop as efficient an algorithm you can, using dynamic programming, for the given problem. Your algorithm *need not* output an optimal schedule; it just needs to output the total waiting time in an optimal schedule. Analyze the running time of your algorithm. (The problem (i) above may be a useful starting point.) **(8 points)**

The solution to these problems will be given on April 23rd.

Important Note 1: For problem (ii) above, you have the option of "buying a hint" any time before the due date. If you choose to, you can get a hint for this problem from the instructor or the TA (come to their office hours or make an appointment by email). In such a case, you will be given the hint, and your final score for problem (ii) will be half of the score you get. You can buy the hint any time before the due date; so, it is best for you to decide as quickly as possible whether you will need the hint.

Important Note 2: As you are doing the above, please start coding up a simple "exhaustive search" based algorithm for the problem, which basically tries out all possible ways of broadcasting at k different times, and chooses the optimal solution from these. Please do not submit this code with Part I; it is due in Part II of the project. But it will help you greatly to start this implementation right away.

3 Part II of the Project

This part is due May 9th, Thursday, at the beginning of class. Here, you will code up the exhaustive search algorithm mentioned above, as well as the dynamic programming algorithm (which is the heart of Part I, and whose description will be given on April 23rd); you will then compare their running times. The details of how to generate problem instances, report running times, etc., will be announced soon. This part is worth 10 points.