

CS 1510 Midterm 1  
Fall 2007

1. (40 points) You wish to drive from point  $A$  to point  $B$  along a highway minimizing the time that you are stopped for gas. You are told beforehand the capacity  $C$  of your gas tank in liters, your rate  $F$  of fuel consumption in liters/kilometer, the rate  $r$  in liters/minute at which you can fill your tank at a gas station, and the locations  $A = x_1, \dots, B = x_n$  of the gas stations along the highway. So if you stop to fill your tank from 2 liters to 8 liters, you would have to stop for  $6/r$  minutes. Consider the following two algorithms:
  - (a) Stop at every gas station, and fill the tank with just enough gas to make it to the next gas station.
  - (b) Stop if and only if you don't have enough gas to make it to the next gas station, and if you stop, fill the tank up all the way.

For each algorithm either prove or disprove that this algorithm correctly solves the problem. Your proof of correctness must use an exchange argument.

2. (20 points) We consider different variations/aspects of the shortest common super-sequence problem. The input is two strings  $A = A_1 \dots A_m$  and  $B = B_1 \dots B_n$ .
  - (a) Give recursive pseudo-code to compute the length of the shortest common super-sequence.
  - (b) Explain as concretely and convincingly as possible why this recursive code may run in exponential time for some inputs.
  - (c) Let the  $m$  by  $n$  table  $T$  be defined as follows:  $T[i, j]$  is the length of the shortest common super-sequence of  $A_1 \dots A_i$  and  $B = B_1 \dots B_j$ . Give pseudo-code, which runs in time  $O(mn)$ , to fill in this table.
  - (d) Show the table  $T$  constructed by your code from the previous part for the two strings  $A = xyxx$  and  $B = xxy$ .

3. (20 points) Give a polynomial time algorithm for the following problem. The input consists of a sequence  $R = R_0, \dots, R_n$  of non-negative integers, and an integer  $k$ . The number  $R_i$  represents the number of users requesting some particular piece of information at time  $i$  (say from a www server). If the server broadcasts this information at some time  $t$ , then the requests of all the users who requested the information strictly before time  $t$  are satisfied. The server can broadcast this information at most  $k$  times. The goal is to pick the  $k$  times to broadcast in order to minimize the total time (over all requests) that requests/users have to wait in order to have their requests satisfied.

As an example, assume that the input was  $R = 3, 4, 0, 5, 2, 7$  (so  $n = 6$ ) and  $k = 3$ . Then one possible solution (there is no claim that this is the optimal solution) would be to broadcast at times 2, 4, and 7 (note that it is obvious that in every optimal schedule that there is a broadcast at time  $n + 1$  if  $R_n \neq 0$ ). The 3 requests at time 1 would then have to wait 1 time unit. The 4 requests at time 2 would then have to wait 2 time units. The 5 requests at time 4 would then have to wait 3 time units. The 2 requests at time 5 would then have to wait 2 time units. The 7 requests at time 6 would then have to wait 1 time units. Thus the total waiting time for this solution would be

$$3 * 1 + 4 * 2 + 5 * 3 + 2 * 2 + 7 * 1$$

4. (20 points) Consider the following problem. The input consists of  $n$  positive integers  $V = \{v_1, \dots, v_n\}$ . Let  $L = \sum_{i=1}^n v_i$ . The problem is to determine if there are two disjoint subsets  $S$  and  $P$  of  $V$  such that  $\sum_{v_i \in S} v_i = \prod_{v_i \in P} v_i$ . That is, no number can be in more than one of  $S$  and  $P$  and you want the sum of the numbers in  $S$  to be equal to the product of the numbers in  $P$ . It is okay for a number to be in neither of  $S$  and  $T$ . Give an algorithm for this problem whose running time is polynomial in  $n$  and  $L$ .