Problem Set 6

MIT students: This problem set is due in lecture on **Monday, November 7, 2005.** The homework lab for this problem set will be held 2–4 P.M. on Sunday, November 6, 2005.

Reading: Chapter 17 and the handout on competitive analysis.

Both exercises and problems should be solved, but *only the problems* should be turned in. Exercises are intended to help you master the course material. Even though you should not turn in the exercise solutions, you are responsible for material covered in the exercises.

Mark the top of each sheet with your name, the course number, the problem number, your recitation section, the date and the names of any students with whom you collaborated. **Please staple and turn in your solutions on 3-hole punched paper**.

You will often be called upon to "give an algorithm" to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of the essay should provide the following:

1. A description of the algorithm in English and, if helpful, pseudo-code.
2. At least one worked example or diagram to show more precisely how your algorithm works.
3. A proof (or indication) of the correctness of the algorithm.
4. An analysis of the running time of the algorithm.

Remember, your goal is to communicate. Full credit will be given only to correct solutions *which are described clearly*. Convoluted and obtuse descriptions will receive low marks.

Exercise 6-1. Do Exercise 17.1-1 on page 410 in CLRS.

Exercise 6-2. Do Exercise 17.3-3 on page 416 in CLRS.

Exercise 6-3. Do Exercise 17.3-6 on page 416 in CLRS.

Exercise 6-4. Do Exercise 17.3-7 on page 416 in CLRS.

Exercise 6-5. Do Exercise 17.4-2 on page 425 in CLRS.

Problem 6-1. Electronic Billboard

You are starting a new Electronic Billboard company called E-Bill. For now, you have just one billboard and you can display one advertisement on the billboard at a time. Your advertising contract with the customers says that if you display their advertisement for one week, then they will pay you, otherwise they won't. Customers come to you with their advertisements at arbitrary times and offer you some money to display their ad. When they come to you, you must decide immediately whether you will display their advertisement starting immediately. If you are already displaying some other advertisement, you may drop it and lose all the profit due to the dropped ad. The goal is to device an algorithm to maximize your profit.

Formally, an advertisement i arrives at time a_i and has a profit of p_i. Upon arrival of an ad, the algorithm must decide immediately whether to display it or not. The algorithm receives the profit p_i of ad i arriving at time a_i only if it displays the ad from time a_i up to $a_i + 1$. Assume that only one job arrives at a time. Assume also that if ad i completes at time t, the billboard can start displaying another ad immediately.

Consider the following algorithm AD-SCHEDULE, where $\alpha > 1$ is some constant. When an ad i with profit p_i arrives,

1. if no ad is being displayed, then start displaying i,
2. if an ad k of profit p_k is being displayed, then discard k if and only if $p_i > \alpha p_k$, otherwise don't accept i.

The ads that an algorithm displays for a full week are ***completed*** ads, and the ads that the algorithm starts displaying, but does not display for a full week are ***discarded*** ads. The ads that the algorithm does not accept at all are ***rejected*** ads. The algorithm earns profit from only the completed ads.

(a) Consider the arrival sequence shown in Figure 1. Give the execution trace of algorithm AD-SCHEDULE with $\alpha = 1.5$, that is, show what actions occur when each ad arrives and which ads are completed, discarded, and rejected. Compute the total profit earned by AD-SCHEDULE.

Ad	*Arrival Time*	*Profit*
1	1.0	1.0
2	1.7	1.3
3	1.8	1.8
4	2.0	3.0
5	2.1	2.3
6	2.5	4.0
7	3.0	4.5
8	4.1	6.2

Figure 1: A sample input to AD-SCHEDULE.

(b) Let OPT be the algorithm that knows the sequence of ads in advance and schedules the ads so as to maximize the profit. Which of the ads in Figure 1 are completed, rejected, and discarded by OPT? What is the profit of OPT?

(c) Show that if $\alpha = 1$, then AD-SCHEDULE can produce arbitrarily poor profits compared with OPT. (*Hint:* Give a sequence of ads on which OPT does well and AD-SCHEDULE does poorly.)

Consider a set of ads $A = \{1, 2, \ldots, n\}$. A subset $S \subseteq A$ is a solution to the billboard-scheduling problem if and only if some algorithm can complete all ads in S. That is, no two ads in the solution X overlap. The profit P_S of a solution S is $P_S = \sum_{i \in X} p_i$.

For a sequence A of ads, suppose that AD-SCHEDULE generates a solution $S \subseteq A$ and OPT generates the solution $S^* \subseteq A$. For any time t, let $S_t \subseteq S$ be the set of ads that AD-SCHEDULE completes by (up to and including) time t and $D_t \subseteq A - S$ be the set of ads that AD-SCHEDULE discards by time t. Similarly, let $S_t^* \subseteq S^*$ be the set of ads that OPT completes by time t.

(d) Suppose that j is the ad that AD-SCHEDULE is displaying at time t. Prove by induction that

$$(\alpha - 1) \sum_{i \in D_t} p_i \le p_j + \sum_{i \in S_t} p_i \,.$$

(e) Prove that OPT never needs to discard an ad after it starts displaying it. In other words, if an optimal algorithm OPT generates a solution S^* where the algorithm sometimes discards ads, then there is another algorithm OPT* that never discards an ad and generates the same solution S^*.

(f) Show that there exists an injective mapping $f : S^* \to D_\infty \cup S$ such that for all $j \in S^*$, we have $\alpha p_{f(j)} \ge p_j$ and $a_j - 1 < a_{f(j)} \le a_j$. (Remember to show that the mapping is injective.)

(g) Let j^* and j be the ads that OPT* and AD-SCHEDULE are displaying at time t. Prove that, for all t, we have

$$p_{j^*} + \sum_{i \in S_t^*} p_i \le \alpha \left(p_j + \sum_{i \in D_t} p_i + \sum_{i \in S_t} p_i \right) .$$

(h) Consider the potential function

$$\Phi(t) = \alpha^2 \left(\sum_{i \in S_t} p_i + p_j \right) - (\alpha - 1) \left(\sum_{i \in S_t^*} p_i + p_{j^*} \right) ,$$

where AD-SCHEDULE and OPT* are displaying ads j and j^* respectively at time t. Using parts (d) and (g), prove that this potential function always stays positive. Alternatively, you may prove this part using induction directly, if you wish.

(i) Conclude that AD-SCHEDULE is $\alpha^2/(\alpha - 1)$-competitive.

(j) What is the optimal value of α to minimize the competitive ratio?

(k) *Optional:* Give an example of a sequence of ads where, if AD-SCHEDULE's profit is P, then OPT's profit is at least $((\alpha^2/(\alpha-1))-1)P$.

(l) *Optional:* Consider a variation of the billboard-scheduling problem in which ad lengths may vary and profit is proportional to ad length. That is, each ad i has a profit p_i and must be displayed for time p_i. Give a competitive analysis of AD-SCHEDULE for this variation of the problem.

Problem 6-2. The cost of restructuring red-black trees.

There are four basic operations on red-black trees that modify their structure: node insertions, node deletions, rotations, and color modifications. We have seen that RB-INSERT and RB-DELETE use only $O(1)$ rotations, node insertions, and node deletions to maintain the red-black properties, but they may make many more color modifications.

(a) Describe how to construct a red-black tree on n nodes such that RB-INSERT causes $\Omega(\lg n)$ color modifications. Do the same for RB-DELETE.

Although the worst-case number of color modifications per operation can be logarithmic, we shall prove the following theorem.

Theorem 1 *Any sequence of m RB-INSERT and RB-DELETE operations on an initially empty red-black tree causes $O(m)$ structural modifications in the worst case.*

(b) Examine Figures 13.4, 13.5, and 13.6 in CLRS closely. Some of the cases handled by the main loop of the code of both RB-INSERT-FIXUP and RB-DELETE-FIXUP are *terminating*: once encountered, they cause the loop to terminate after a fixed, constant number of operations. For each of the cases of RB-INSERT-FIXUP and RB-DELETE-FIXUP, specify which are terminating and which are not.

We shall first analyze the structural modifications when only insertions are performed. Let T be a red-black tree, and let $\Phi(T)$ be the number of red nodes in T. Assume that 1 unit of potential can pay for the structural modifications performed by any of the three cases of RB-INSERT-FIXUP.

(c) Let T' be the result of applying Case 1 of RB-INSERT-FIXUP to T. Argue that $\Phi(T') = \Phi(T) - 1$.

(d) Node insertion into a red-black tree using RB-INSERT can be broken down into three parts. List the structural modifications and potential changes resulting from TREE-INSERT, from nonterminating cases of RB-INSERT-FIXUP, and from terminating cases of RB-INSERT-FIXUP.

(e) Using part (d), argue that the amortized number of structural modifications (with respect to Φ) of RB-INSERT is $O(1)$.

We now wish to prove the theorem for both insertions and deletions. Define

$$w(v) = \begin{cases} 0 & \text{if } v \text{ is red,} \\ 1 & \text{if } v \text{ is black and has no red children,} \\ 0 & \text{if } v \text{ is black and has one red child,} \\ 2 & \text{if } v \text{ is black and has two red children.} \end{cases}$$

Let the potential of a red-black tree T be defined as

$$\Phi(T) = \sum_{v \in T} w(v) ,$$

and let T' be the tree that results from applying any nonterminating case of RB-INSERT-FIXUP or RB-DELETE-FIXUP to T.

(f) Show that $\Phi(T') \leq \Phi(T) - 1$ for all nonterminating cases of RB-INSERT-FIXUP. Argue that the amortized number of structural modifications (with respect to Φ) performed by RB-INSERT-FIXUP is $O(1)$.

(g) Show that $\Phi(T') \leq \Phi(T) - 1$ for all nonterminating cases of RB-DELETE-FIXUP. Argue that the amortized number of structural modifications (with respect to Φ) performed by RB-DELETE-FIXUP is $O(1)$.

(h) Complete the proof of Theorem 1.