Problem Set 6 Solutions

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.

Solution: Ad 1 and 3 are discarded. Ad 2, 5 and 6 are rejected. Ad 4, 7 and 8 are completed with total profit 13.7.

(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?

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.

Solution: Ad 1, 4, 7, and 8 are completed. The others are rejected. The profit of OPT is 14.7.

(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.)

Solution: Let $0 < \epsilon < 1/2$. For $i \geq 0$, let $a_i = i(1-\epsilon)$ be the arrival time and $p_i = (\alpha+\epsilon)^i$ be the profit of job i. Since $a_i \leq a_{i+1} < a_i + 1$ and $p_{i+1} = (\alpha+\epsilon)p_i > \alpha p_i$, AD-SCHEDULE discards job i and starts job $i+1$. By using induction on i, if there are $2n$ jobs, then AD-SCHEDULE only completes the last job giving the profit $p_{2n} = (\alpha+\epsilon)^{2n}$.

Since $\epsilon < 1/2$, the intervals $[a_0, a_0+1)$, $[a_2, a_2+1)$, etc. are disjoint. Therefore it is possible to complete jobs $0, 2, \ldots, 2n$. This means that the profit of OPT is at least

$$\sum_{i=0}^{n} p_{2i} = \sum_{i=0}^{n} (\alpha+\epsilon)^{2i} = \frac{(\alpha+\epsilon)^{2(n+1)} - 1}{(\alpha+\epsilon)^2 - 1}.$$

If $\alpha = 1$, then $(\alpha+\epsilon)^2 - 1 = \epsilon^2 + 2\epsilon$ can be arbitrarily small, hence, OPT can produce arbitrarily good profits compared to AD-SCHEDULE.

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

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

Solution: The proof is by induction on the ads. After the first ad arrives, we have $j = 1$. AD-SCHEDULE starts displaying j and it has not completed or discarded any ads. Therefore, we have $S_t = D_t = \emptyset$. Say an ad k arrives at time t, while AD-SCHEDULE is displaying the ad l. (If no ad is being displayed when k arrives, then the inductive step is trivial.) For the inductive step, we assume that before the arrival of ad k, we have $(\alpha - 1) \sum_{i \in D_t} p_i \leq p_l + \sum_{i \in S_t} p_i$. On arrival of the ad k at time t', if AD-SCHEDULE decides not to display it, then both D_t and S_t remain unchanged and we trivially prove the inductive step. If AD-SCHEDULE discards l and starts displaying k, then we have $p_k > \alpha p_l$ and $D_{t'} = D_t \cup \{l\}$. Therefore, we have $(\alpha - 1) \sum_{i \in D_{t'}} p_i = (\alpha - 1)(\sum_{i \in D_t} p_i + p_l) = (\alpha - 1)(\sum_{i \in D_t} p_i) + \alpha p_l - p_l \leq (\alpha - 1)(\sum_{i \in D_t} p_i) + p_k - p_l = p_k + \sum_{i \in S_t} p_i$. Done.

(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^*.

Solution: OPT* starts only those ads which OPT starts and completes.

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

Solution: Define $f(j)$ as follows: For an ad $j \in S^*$, pick $i = f(j) \in S \cup D_\infty$, such that i is the latest job such that $a_i \leq a_j$. (Note that i could be the same as j.) It is easy to see that $a_j - 1 < a_i$, otherwise i would be done by the time j arrived, and AD-SCHEDULE would do j. Now for this i, we have $\alpha p_i \geq p_j$, otherwise AD-SCHEDULE would discard i and start j. Since j is in S^*, the intervals $[a_j - 1, a_j]$ are non-overlapping. Therefore, the function 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

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

Solution: Every ad whose profit is counted on the left hand side is in S^*. Therefore, we have

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

Notice that f maps to $D_\infty \cup S$ and that $a_{f(i)} \le a_i \le t$ for all $i \in S_t^*$, and $a_{f(j^*)} \le a_{j^*} \le t$. Therefore, $f(j^*)$ and each $f(i)$ are in the set $D_t \cup S_t \cup \{j\}$. Also notice that f is injective. Therefore, we have

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

(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.

Solution: From (g) we infer that

$$\begin{aligned} \Phi(t) &\ge \alpha^2 \left(\sum_{i \in S_t} p_i + p_j \right) - (\alpha - 1)\alpha \left(p_j + \sum_{i \in D_t} p_i + \sum_{i \in S_t} p_i \right) \\ &= \alpha \left(\sum_{i \in S_t} p_i + p_j \right) - (\alpha - 1)\alpha \left(\sum_{i \in D_t} p_i \right) , \end{aligned}$$

which is by (d)

$$\ge \alpha(\alpha - 1) \left(\sum_{i \in D_t} p_i \right) - (\alpha - 1)\alpha \left(\sum_{i \in D_t} p_i \right) = 0 .$$

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

Solution: Since Φ is positive, α^2 times the profit of AD-SCHEDULE is at least $(\alpha - 1)$ times the profit of OPT* and we conclude that AD-SCHEDULE is $\alpha^2/(\alpha - 1)$-competitive.

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

Solution: $\alpha = 2$.

(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 $((\alpha^2/(\alpha-1)) - 1)P$.

Solution: Let $0 < \epsilon < 1/n$. For $0 \le i \le n$, let $a_{2i} = i(1-\epsilon)$ be the arrival time and $p_{2i} = (\alpha+\epsilon)^i$ be the profit of job $2i$. For $0 \le i \le n$, let $a_{2i+1} = i$ be the arrival time and $p_{2i+1} = \alpha(\alpha+\epsilon)^i$ be the profit of job $2i+1$.

Since $0 \le \epsilon < 1/n$, $a_{2i} \le a_{2i+1} < a_{2(i+1)} < a_{2i} + 1$. If AD-SCHEDULE runs job $2i$, then

- it rejects job $2i+1$ because $p_{2i+1} = \alpha p_{2i}$ after which
- it discards job $2i$ and starts job $2(i+1)$ because $p_{2(i+1)} = (\alpha+\epsilon)p_{2i} > \alpha p_{2i}$.

By using induction on i, AD-SCHEDULE only completes job $2n$ giving the profit $p_{2n} = (\alpha+\epsilon)^n$.

Notice that the intervals $[a_1, a_1+1)$, $[a_3, a_3+1)$, etc. are disjoint. Therefore it is possible to complete jobs $1, 3, \ldots, 2n+1$. This means that the profit of OPT is at least

$$\sum_{i=0}^{n} p_{2i+1} = \sum_{i=0}^{n} \alpha(\alpha+\epsilon)^i = \frac{\alpha(\alpha+\epsilon)^{n+1} - 1}{\alpha+\epsilon-1}.$$

By taking ϵ arbitrarily small, the profit of OPT is $(\alpha^2 p_{2n} - 1)/(\alpha - 1) \ge ((\alpha^2/(\alpha - 1)) - 1)p_{2n}$ for n large enough.

(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.

Solution: We can prove that

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

in a manner similar to part (d). Observe also that whenever OPT* is working, AD-SCHEDULE is working as well. Since the profit is the same as the length of the job, we have

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

Therefore, both the inequalities we need to prove that the potential function remains positive are true and we can use the same potential function to prove the same competitive ratio.

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.

Solution: For RB-INSERT, consider a complete red-black tree with an even number of levels in which nodes at odd levels are black and nodes at even levels are red. When a node is inserted as a child of one of the leaves, then $\Omega(\lg n)$ color changes will be needed to fix the colors of nodes on the path from the inserted node to the root.

For RB-DELETE, consider a complete red-black tree in which all nodes are black. If a leaf is deleted, then the "double blackness" will be pushed all the way up to the root, with a color change at each level (case 2 of RB-DELETE-FIXUP), for a total of $\Omega(\lg n)$ color changes.

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.

Solution: All cases except for case 1 of RB-INSERT-FIXUP and case 2 of RB-DELETE-FIXUP are terminating.

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 $\Phi(T') = \Phi(T) - 1$.

Solution: Case 1 of RB-INSERT-FIXUP reduces the number of red nodes by one, a fact that can be seen in Figure 13.4 in CLRS. Hence, $\Phi(T') = \Phi(T) - 1$.

(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.

Solution: TREE-INSERT causes one node insertion and a unit increase in potential. The nonterminating case of RB-INSERT-FIXUP (Case 1) makes three color changes and decreases the potential by one. The terminating cases of RB-INSERT-FIXUP (Cases 2 and 3) cause one rotation each and do not affect the potential.

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

Solution: The number of structural modifications and amount of potential change resulting from TREE-INSERT and the terminating cases of RB-INSERT-FIXUP are constant, so the amortized cost of these parts are constant. The nonterminating case of RB-INSERT-FIXUP may repeat up to $O(\lg n)$ times, but its amortized cost is 0, since by our assumption the unit decrease in the potential pays for the structural modifications needed. Therefore, the worst-case amortized cost of RB-INSERT is constant.

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 $O(1)$.

Solution: From Figure 13.5 of CLRS, we see that Case 1 of RB-INSERT-FIXUP makes the following changes to the tree:

- Changes a black node with two red children to a red node (node C), resulting in a potential change of -2.
- Changes a red node to a black node with one red child (node A in the top diagram; node B in the bottom diagram), resulting in no potential change.
- Changes a red node to a black node with no red children (node D), resulting in a potential change of 1.

The total change in potential is -1, which pays for the structural modifications performed, and thus the amortized cost of Case 1 (nonterminating case) is 0. Because the terminating cases of RB-INSERT-FIXUP cause constant structural changes and constant change in potential, since $w(v)$ is based solely on node color and the number of color changes caused by termintaing cases is constant. The amortized cost of the terminating cases is at most constant. Hence, the overall amortized cost of RB-INSERT-FIXUP is constant.

(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 $O(1)$.

Solution: Figure 13.6 of CLRS shows that Case 2 of RB-DELETE-FIXUP makes the following changes to the tree:

- Changes a black node with no red children to a red node (node D), resulting in a potential change of -1.
- If B is red, then it loses a black child, with no effect on potential.
- If B is black, then it goes from having no red children to having one red child, resulting in a potential change of -1.

The total change in potential is either -1 or -2, depending on the color of B. In either case, one unit of potential pays for the structural modifications performed, and thus the amortized cost of Case 2 (nonterminating case) is at most 0. Because the terminating cases of RB-DELETE cause constant structural changes and constant change in potential, since $w(v)$ is based solely on node color and the number of color changes caused by termintaing cases is constant. The amortized cost of the terminating cases is at most constant. Hence, the overall amortized cost of RB-DELETE-FIXUP is constant.

(h) Complete the proof of Theorem 1.

Solution: Since the amortized cost of each operation is bounded above by a constant, the actual number of structural modifications for any sequence of m RB-INSERT and RB-DELETE operations on an initially empty red-black tree cause $O(m)$ structural modifications in the worst case.