Problem Set 11

Due: TBD.

Problem 1. Consider a collection of faculty available to give you advice about your future path in life. Since they are busy, each only answers yes or no questions. Some are wiser than others, and are more often correct in their advice. You would like, looking back, to have taken the advice of the wisest faculty member, but ahead of time you don't know who that is. Consider the following online algorithm: each time you have a question, you ask all the faculty, and go with a "weighted majority" opinion as follows:

1. Initially, each faculty member has a weight of 1.
2. After asking a question, take the (yes/no) answer of larger total weight.
3. Upon discovering which answer is correct, halve the weight of each faculty member who answered incorrectly.

You will show that, using this algorithm, you make at most $2.41(m + \lg n)$ mistakes, where m is the number of mistakes made by the wisest faculty, and n is the number of faculty. (Thus, in the language of asymptotic competitive ratios, you are 2.41-competitive against asking the best faculty member.)

(a) Prove that, when the online algorithm makes a mistake, the total weight of the faculty decreases by a factor of 4/3. Use this to upper bound the total weight assigned to faculty.

(b) Lower-bound the weight assigned to faculty by considering the weight of the wisest faculty member in terms of m.

(c) Combine the above two parts to prove the claim.

With randomization, you can do better. We modify the online algorithm as follows:

1. Instead of going with the majority, choose one faculty member with probability proportional to his weight, and follow his opinion.
2. Instead of multiplying incorrect faculty weights by 1/2, multiply them by some $\beta < 1$ to be determined later.

(d) For a given question, let F denote the fraction of the weight of faculty with the wrong answer. Give an expression for the multiplicative change in the total weight as a result of reweighting for this question.

(e) Arguing much as in the deterministic case, prove that the (expected) number of wrong answers will be at most

$$\frac{m \ln(1/\beta) + \ln n}{1 - \beta}$$

so for example, setting $\beta = 1/2$ gives $1.39m + 2 \ln n$, for a competitive ratio of 1.39.

(f) Show that if one uses the "right" β, one can limit the number of errors to $m + \ln n + O(\sqrt{m \ln n})$, achieving a 1-competitive algorithm (in the asymptotic sense). (This β can actually be found online by "repeated doubling" as the mistakes build up.)

Problem 2. Consider the problem of finding a lifetime companion. Among the pool of k potential partners at the university (MIT students being snobby about this) you can choose and date any one for a while and by doing so measure their suitability compared to all your previous dates'. You then decide whether to stay together forever or break up forever (no way your exes will take you back). Your goal is to find the most suitable companion, as measured by a rank ordering of all the possibilities.

(a) Show that any deterministic strategy for choosing dates and deciding whether to break up is terrible from a competitive perspective: you can be forced by fate to end up with the absolutely worst possible choice.

(b) Devise a randomized strategy that does better, giving you a constant probability of ending up with the absolute best companion.

(c) Suppose that you want to play for slightly less high stakes. Give an algorithm minimizing the *expected rank* of your final choice. You will receive full credit for achieving rank $O(\log k)$ and partial credit for worse bounds.

OPTIONAL (d) Show how to achieve expected rank $O(1)$.

Problem 3. In many applications, one wants to do range searching among objects other than points. In this problem, we will see that we can reduce several problems of this flavor to normal orthogonal range searching.

(a) Let S be a set of n axis-parallel rectangles in the plane (i.e., the sides of the rectangles are vertical and horizontal). We want to be able to report all rectangles in S that are completely contained in a query rectangle $[x : x'] \times [y : y']$. Describe a data structure for this problem that uses $O(n \log^3 n)$ storage and has $O(\log^4 n + k)$ query time, where k is the number of reported answers. (**Hint:** Transform the problem to an orthogonal range searching problem in some higher-dimensional space.)

(b) Let P consist of a set of n polygons in the plane. Again, describe a data structure that uses $O(n \log^3 n)$ storage and has $O(\log^4 n + k)$ query time to report all polygons completely contained in the query rectangle, where k is the number of reported answers (note that as a special case, you can report containment of line segments).

NONCOLLABORATIVE Problem 4. Suppose you're implementing a video game in which the player can walk around a planar environment made up of walls, and at any time the screen displays only the walls that are (partially) visible by the player. More precisely, the player is modeled as a single point; the walls are modeled as noncrossing line segments; two points are *visible* if the line segment connecting them does not intersect any walls except at its endpoints; and a wall is *visible* from a point if at least one point on the wall is visible from the point. You can assume that no wall endpoints are collinear with the player's location. Give an $O(n \lg n)$-time algorithm to compute the set of walls visible from the player. **Hint**: Use a line-sweep algorithm, but instead of sweeping a horizontal line, sweep (rotate) a half-line around a point.

Problem 5. Read through the handout on the final project, which is linked from the course website. Submit a paragraph-long (or so) proposal for the final project detailing the topic and scope of the proposed project. Include citations to relevant papers. If you are collaborating with other students, submit just one copy with all of your names on it. Please limit your groups to 3 students. Working alone is permitted only with a good justification. Email your proposal to Professor Karger at `karger@mit.edu` and CC the TAs. Make the subject line of your email "6.854 project" (without quotes) so we can find it (you won't get credit for a different subject line).

If you are seeking a collaborator, feel free to post on the 6.854-students mailing list—you may want to outline the kind of project you'd like to do.

Problem 6. How long did you spend on this problem set? Please answer this question using the Google form that is located on the course website. This problem is mandatory, and thus counts towards your final grade. It is due by the Monday 2:30pm after the pset due date.