15-451, Spring 2017, Recitation 12

Competitive Analysis and Random Incremental Analysis

1 Where to put the Safe?

An office building has one safe where valuables are kept. There are two rooms in the building numbered $1, 2$. The distance between the rooms is 1. There is a sequence of requests where an employee in some room needs to access the safe. If the employee is in the room with the safe, the cost is 0. If the employee is in the other room, the cost is 1. Management is monitoring these activities, and has the option to move the safe from time to time to a different room. The cost of moving the safe is p.

The requests are adversarily generated, and future requests are unknown by management. After each request management has the option of moving the safe from one place to another. Management's goal is to obtain a deterministic algorithm with low total cost. (The total cost includes the employee movement costs plus the costs incurred by moving the safe.) The criterion of any management algorithm is the competitiveness of the algorithm, as defined in class.

We will analyze a counter-based algorithm, where the threshold is $2p$. This means that when a request is from the room without the safe, the algorithm processes it in the current room (at a cost of 1), and increments the counter. If the counter has reached $2p$ it moves the safe to the just requested room (at a cost of p), and resets the counter to 0. Call this on-line algorithm A, and call the adversary's algorithm B.

Prove that this algorithm is 3-competitive. Hint: use the following potential function:

$$\Phi(S_A, c, S_B) = \begin{cases} 2c & \text{if } S_A = S_B \\ 3p - c & \text{if } S_A \neq S_B \end{cases}$$

Solution: It's just a matter of going through all the cases and proving that in each case the amortized cost to A of the operation is at most three times the cost to B of the operation. In the analysis below we let C_A be the cost of the operation to A. C_B is analogous. We let AC_A be the amortized cost to A.

- Case: The request is free to A:

  The cost to A is 0, the change in potential is 0, and the cost to B is ≥ 0.

- Case: The request costs A 1:

  If A and B are in the same state then $C_A = C_B = 1$. Also, $\Delta\Phi = 2$. So $AC_A = 3$. Thus $AC_A \leq 3C_B$.

  If A and B are in different states, then $C_A = 1, C_B = 0$. Also, $\Delta\Phi = -1$. So $AC_A = 0$. and we have $AC_A \leq 3C_B$.

- Case: A moves the safe:

  In this case c changes from $2p$ down to 0.

If $S_A = S_B$ before and $S_A \neq S_B$ after, then the potential changes from $4p$ down to $3p$, for a net change of $-p$. $C_A = p$, so $AC_A = 0 \leq 3C_B$.

If $S_A \neq S_B$ before and $S_A = S_B$ after, then the potential changes from p before to 0 after, for a net change of $-p$. $C_A = p$, so $AC_A = 0 \leq 3C_B$.

- Case: B moves the safe:

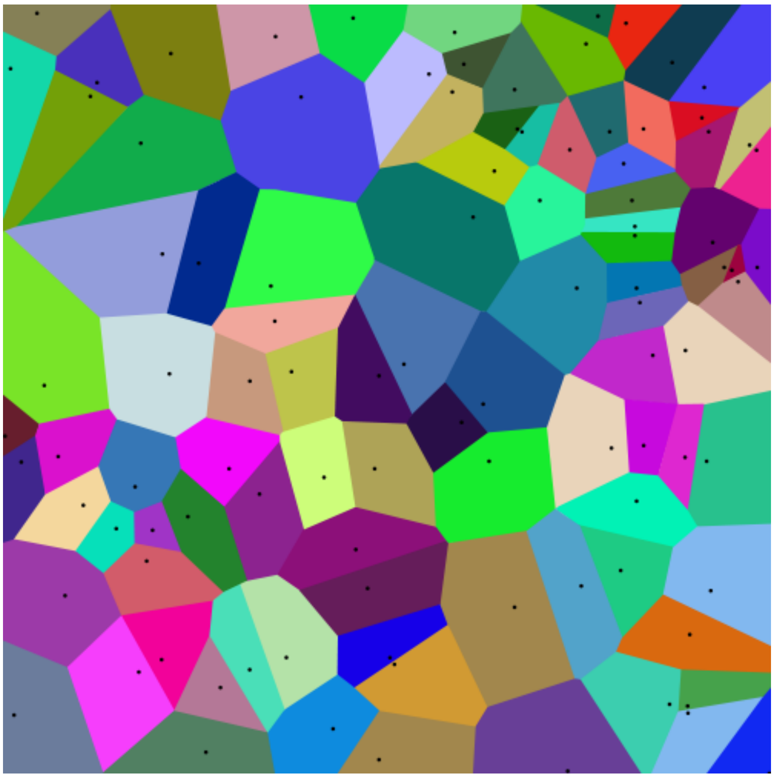
$$\Delta\Phi \leq |(3p - c) - 2c| = |3p - 3c|$$

Since $0 \leq c \leq 2p$, we know that $|3p - 3c| \leq 3p$. So $AC_A \leq 3p \leq 3C_B$.

This completes the proof.

2 Painting

A sequence of n distinct points $p_1, p_2, \ldots p_n$ in the unit square is specified. In a *painted voronoi diagram* the part of the unit square closest to point p_i is painted with color c_i. Here is an example:



This problem concerns the amount of paint used when by a certain painting algorithm:

> **Painting Algorithm:** Insert the points one at a time in the given order. Each time a point is inserted, paint the region of the unit square that is closer to point p_i than any other point that has been inserted so far.

Note that one unit of paint is required to paint the entire square.

1. Devise a sequence of n points and prove that for this sequence the painting algorithm uses $\Omega(n)$ paint.

   Solution: Just take n points, all on the left half of the square, evenly spaced along the perpendicular bisector of the left side. If you add the points from left to right, each point will repaint the entire right hand side, giving at least $n/2$ cost.

2. Returning to the original formulation, where the sequence of points is specified, suppose that before applying the painting algorithm, we randomly permute the n points. Prove that the expected total amount of paint used by this algorithm is at most $1 + \ln n$.

   Solution:

   We use backward analysis. We remove the points one at a time at random. When there are i points left and we remove a random one, the expected area of the repainted region is $1/i$. This is because each of the i points has a disjoint region and the total area of all of them is 1. So the average of the region sizes is $1/i$.

   Thus, the expression above is just

   $$\sum_{i=1}^{n} \frac{1}{i} = H_n \leq 1 + \ln n$$