Randomized Algorithms (Blum/Gupta) **Homework 5**
Date: Monday Mar 21, 2011 **Due:** Monday Apr 4, 2011

Groundrules

- Homeworks will generally consist of *exercises*, easier problems designed to give you practice, and *problems*, that may be harder, trickier, and/or somewhat open-ended. You should do the exercises by yourself, but you may work with a friend on the harder problems if you want. One exception: no fair working with someone who has already figured out (or already knows) the answer. If you work with a friend, then write down who you are working with.
- If you've seen a problem before (sometimes we'll give problems that are "famous"), then say that in your solution (it won't affect your score, we just want to know). Also, if you use any sources other than the textbook, write that down too (it's fine to look up a complicated sum or inequality or whatever, but don't look up an entire solution).

Exercises

1. (Problem 4.9 in M&R) Show that there exists a permutation that would cause $2^{\Omega(n)}$ congestion under random-order shortest-path routing on the hypercube. That is, rather than fixing bits in a left-to-right order, you fix the bits in a random order (which corresponds to taking a random shortest path from each source to its destination).
2. **(Centers and FRT don't mix.)** In the lecture notes, we saw how to reduce k-median on general metrics to k-median on trees. This used random tree embeddings that satisfy two properties: (a) distances in the trees are greater, and (b) the expected distances are not much greater. Show such a reduction fails for k-center. Specifically, as a counter-example,

   - construct a metric (V, d), and a distribution $\mathcal{D}$ over trees on the vertex set V which have (a) $d_T \geq d$, and (b) for every x, y, $E_{\mathcal{D}}[d_T(x, y)] \leq O(1)d(x, y)$, such that
   - if we draw a random tree T from this distribution, solve k-center optimally on this tree T to get centers F_T, the expected cost $E_{T \leftarrow \mathcal{D}}[\max_{v \in V} d(v, F_T)]$ is much larger than the cost of the optimal k-center solution on the original metric. (Ideally, the expected cost should exceed the optimal cost by n^ϵ for some constant $\epsilon > 0$.)

   Note: it is not enough to point out where the reductions break down—we want a counter-example. Note #2: you can do this with $k = 2$.

   Extra Credit: Show such an example for the k-means problem as well.
3. **(How long, how long?)** For n points S picked independently and uniformly from the unit square $[0, 1]^2$, show that $E[TSP(S)] = \Theta(\sqrt{n})$. Note you need to show both the upper and lower bounds. (Hint: have a read over the geometric lemma from class.)

Problems

1. **(External regret vs Swap regret.)** In Rock-Paper-Scissors, Rock beats Scissors (winner has loss 0, loser has loss 1), Scissors beats Paper, and Paper beats Rock; if both players play the same action, they tie (each gets loss of 1/2).

   Consider playing T games of Rock-Paper-Scissors against an opponent who first plays Rock $T/3$ times, then plays Scissors $T/3$ times, then plays Paper $T/3$ times.

(a) Thinking of Rock, Paper, and Scissors as three "experts", describe in words what Randomized Weighted Majority would do against such an opponent. To be concrete, consider the version of RWM that, when expert i incurs loss ℓ, updates using $w_i \leftarrow w_i(1-\epsilon)^\ell$. Assume a learning rate $\epsilon \gg 1/T$, or if you like, you can think of $\lim_{\epsilon \to 1}$. Approximately (ignoring terms that are $o(T)$) what is the total loss of RWM and how does that compare to the loss of the best expert?

(b) What approximately is the *swap regret* of RWM (ignoring terms that are $o(T)$)?

(c) Since external regret is defined as the difference between the loss of the algorithm and the loss of the best expert, any two sequences of actions with the same total loss will result in the same external regret. Is this true for swap regret? In the context of this Rock-Paper-Scissors example, is there a behavior with approximately the same total loss as RWM but with much less swap regret?

(d) Extra credit: what would the swap-regret algorithm from class do in the above situation? Feel free to code it up.

2. **(I Stream, You Stream.)** Recall the data streaming model: elements from $[D]$ stream by, and frequency vector is $x \in \mathbb{Z}_{\geq 0}^D$ where x_i counts the number of occurences of element $i \in [D]$ seen so far. We want a streaming algorithm that stores information about the stream so that when it is eventually queried with some index $q \in [D]$, returns a value $\hat{x}_q \approx x_q$ with probability at least $1-\delta$. One way to do this is to store x explicitly, but we want to use less space.

Consider the following algorithm:

> *Keep a global hash function $H : [D] \to [d]$, and also d counters $C_1, C_2, \ldots, C_d$ (initially zero), each with its own hash function $h_i : [D] \to \{-1, +1\}$. If you see element $e \in [D]$, first hash it using the global hash function H to get the bucket number $H(e)$, and then update*
>
> $$C_{H(e)} \leftarrow C_{H(e)} + h_{H(e)}(e)$$
>
> *When faced with the query q, output $A(q) := h_{H(q)}(q) \cdot C_{H(q)}$.*

Assume that H and the h_i's are indepenedently picked, and each hash function is itself pairwise independent.

(a) Show that $E[A(q)] = x_q$.

(b) Show that the variance of $A(q) = \frac{1}{d}(F_2 - x_p^2) \leq F_2/d$.

(c) Show that if we set $d = 1/\epsilon^2\delta$, we get an estimate $A(q) \in x_q \pm \epsilon\sqrt{F_2} = x_q \pm \epsilon\|x\|_2$ w.p. $1-\delta$.

(d) Finally, consider an extension of this idea: maintain t independent copies of the above data structure. On a query for q, if the answers are $A_1(q), A_2(q), \ldots, A_t(q)$, return the median $M(q)$ of these t answers.

Show that with $t = 10 \log 1/\delta$, and $d = 3/\epsilon^2$, you get $M(q) \in x_q \pm \epsilon\|x\|_2$ w.p. $1-\delta$.

Note: This shows that the element q is such that x_q is large compared to $\|x_2\|$, then we get a good estimate. In many data streams, there are a few very frequent elements (the "elephants") and others are fairly rare (the "mice"). This method allows us to estimate the size of any elephant well.

3. **(Large girth graphs.)** Recall the lower bound on low-diameter decompositions: we used the existence of graphs with at sufficiently many edges and yet large girth (no short cycles). In this problem, we prove that such graphs exist.

Consider the random graph $G_{n,p}$ with $p = c/n$. Delete all edges that lie on cycles of length $\frac{1}{2}\log_c n$. Now use the probabilistic method to infer that for sufficiently large n, there exist graphs with at least $(c-1)(n-1)/2$ edges and girth at least $\frac{1}{2}\log_c n$.