COS 521: Advanced Algorithm Design
Homework 3
practice problems

Collaboration Policy: You may collaborate with other students on these problems.

1. Recall the weighted paging problem we discussed in class. We designed a randomized algorithm that maintains a distribution of pages in the cache, i.e. maintains probabilities p_i for page i to be in the cache such that $\sum p_i = k$ and these probabilities are updated in an online fashion. In this exercise, you will show how these distributions can be converted into distributions over cache states, i.e. distributions over sets of pages in the cache. For simplicity, we will consider the case where the weights of all pages are 1. We say that distribution D over cache states (i.e. sets of k pages) is consistent with distribution p over pages if $\Pr_{C \in D}[i \in C] = p_i$ for all pages i.

   Let p and q be the probability distributions over pages in two consecutive steps. Given a distribution over cache states D_p consistent with p, you will construct a distribution D_q consistent with q such that the cost of moving from D_p to D_q (i.e. the actual cost of the algorithm) is not much larger than the cost of moving from p to q (i.e. the fractional cost we bound in our analysis). First, we explain what each of these costs are: The fractional cost $C_f(p, q)$ of moving from distribution p to q (over pages) is $\sum_i |p_i - q_i|$. The cost of moving from one cache state C to another cache state C' (both a subset of k pages) is $|C - C'|$. The cost $C(D_p, D_q)$ of moving from distribution D_p to D_q over cache states is the minimum cost matching between distributions D_p and D_q. Instead of defining it precisely, here is how one shows an upper bound on this cost: for each cache state in the support of D_p, you need to specify how the cache state is updated (possibly in a randomized fashion). The resulting distribution over cache states from this change should be D_q. If $Pr[C \to C']$ denotes the probability that the cache contents are updated from C to C', the cost of this move from D_p to D_q is $\sum_{C,C'} |C - C'| \cdot Pr[C \to C']$.

   The goal of this exercise is to show that given p, q and distribution D_p consistent with p, we can construct distribution D_q consistent with q such that $C(D_p, D_q) \leq 2C_f(p, q)$.

   (a) Show without loss of generality that you can assume that q is obtained from p by removing ϵ units of mass from some page a and moving it to some other page b.

   (b) Given p, q as in the part (a), distribution D_p consistent with p, show that you can construct distribution D_q consistent with q such that $C(D_p, D_q) \leq 2\epsilon$.

2. Recall the von Neumann minimax theorem:

$$\min_{x \in \Delta_n} \max_{y \in \Delta_m} y^T A x = \max_{y \in \Delta_m} \min_{x \in \Delta_n} y^T A x$$

where A is an $n \times m$ matrix, $\Delta_n = \{x \in \Re^n | x_i \geq 0, \sum x_i = 1\}$, and $\Delta_m = \{y \in \Re^m | y_i \geq 0, \sum y_i = 1\}$

In this exercise, you will derive Yao's minimax principle, which is a useful technique to show lower bounds on randomized algorithms. Let Π be a problem with a finite set $\mathcal{I}$ of input instances (of a fixed size) and a finite set of deterministic algorithms $\mathcal{A}$. For input $I \in \mathcal{I}$ and algorithm $A \in \mathcal{A}$, let $C(I, A)$ denote the cost of running algorithm A in input I. For a distribution p over inputs, and distribution q over algorithms, let I_p be a random input chosen according to probability distribution p and let A_q be a random algorithm chosen according to q. Then Yao's minimax principle says:

For all distributions p over $\mathcal{I}$ and q over $\mathcal{A}$,

$$\min_{A \in \mathcal{A}} \mathbf{E}[C(I_p, A)] \leq \max_{I \in \mathcal{I}} \mathbf{E}[C(I, A_q)].$$

In other words, the expected cost of the optimal deterministic algorithm for an input distribution p is a lower bound on the expected cost of the optimal randomized algorithm. Lower bounds on the competitive ratio of randomized algorithms are typically obtained by applying Yao's principle with a carefully chosen distribution on inputs.

3. In this exercise, you will use Yao's principle to derive a lower bound on randomized algorithms for bipartite matching. *To be completed.*

4. Let $X = \sum_i X_i$ where the X_i are independent $\{0, 1\}$ random variables and let $\mu = \mathbf{E}[X]$. In class, we proved the following upper bound on the tail of a random variable: for an $0 < \delta \leq 1$,

$$\Pr[X > (1 + \delta)\mu] < exp(-\mu\delta^2/3).$$

Prove the following lower bound (that we stated without proof):

$$\Pr[X < (1 - \delta)\mu] < exp(-\mu\delta^2/2).$$

5. For any $\alpha > 1$, define an α-approximate cut in a graph G as any cut whose size is within a multiplicative factor α of the size of a min-cut in G.

(a) Give a lower bound on the probability that a single iteration of the randomized algorithm for min-cuts will produce some fixed α-approximate min-cut.

(b) Use your result from part (a) to obtain a bound on the number of α-approximate min-cuts.

6. In class, we analyzed an estimator for the second frequency moment $F_2 = \sum_i m_i^2$ (here m_i is the number of copies of element i). Here the basic form of the estimator consisted of a counter $C = \sum_i x_i m_i (x_i \in_R \{+1, -1\})$ and the value of the estimator was C^2. A number of independent copies of this scheme were required to obtain a good estimate with high probability. Consider the following variant to estimate $F_3 = \sum_i m_i^3$: Maintain a counter $C = \sum_i x_i m_i$, where each x_i is picked uniformly and at random from $\{1, \omega, \omega^2\}$ (here ω is a complex cube root of unity). For the purpose of this question, ignore the issues of independence in the choices of x_i and assume that all x_i are picked independently.

   (a) Show that $\mathbf{E}[C^3] = F_3$. (b) Consider the estimator $Re[C^3]$ = the real part of C^3. Give an upper bound on the number of independent copies of this estimator needed to obtain an $1 + \epsilon$ approximation of F_3 with probability $1?\delta$.

7. Given a graph $G(V, E)$, the *sparsest cut* problem is the problem of computing

$$\min_{S \subset V} \frac{|E(S, \bar{S})|}{|S||\bar{S}|}$$

   Consider the following vector program with a vector v_i for every $i \in V$.

$$\begin{aligned} &\min \sum_{(i,j) \in E} (v_i - v_j)^2 \\ \text{such that} \quad &\sum_{i<j} (v_i - v_j)^2 = 1 \end{aligned}$$

   (a) Show that the vector program is a relaxation for sparsest cut.

   (b) Suppose that graph G is a d-regular graph. Show that the optimum value of the vector program is the second eigenvalue of the Laplacian of G (suitably scaled).