Princeton Univ Fall '20	COS 521:Advanced Algorithms
Homework 4	
Out: *Oct 24*	Due: *Nov 13*

Instructions:

- Upload your solutions (to the non-extra-credit) to each problem as a **separate PDF** file (one PDF per problem) to codePost. Please make sure you are uploading the correct PDF! Please anonymize your submission (i.e., do not list your name in the PDF), but if you forget, it's OK.
- If you choose to do extra credit, upload your solution to the extra credits as a single separate PDF file to codePost. Please again anonymize your submission.
- You may collaborate with any classmates, textbooks, the Internet, etc. Please upload a brief "collaboration statement" listing any collaborators as a separate PDF on codePost (if you forget, it's OK). But always **write up your solutions individually**.
- For each problem, you should aim to keep your writeup below one page. For some problems, this may be infeasible, and for some problems you may write significantly less than a page. This is not a hard constraint, but part of the assignment is figuring out how to easily convince the grader of correctness, and to do so concisely. "One page" is just a guideline: if your solution is longer because you chose to use figures (or large margins, display math, etc.) that's fine.
- Each problem is worth twenty points (even those with multiple subparts), unless explicitly stated otherwise.

Problems:

§1 This exercise is to remind you of some basic linear algebra.

(a) Show that every eigenvalue of a real symmetric matrix $M \in \mathbb{R}^{n \times n}$ is real.
Hint: Use $x^T M \overline{x} = \overline{x}^T M x$ for any n dimensional complex vector x.

(b) Show that any two eigenvectors corresponding to distinct eigenvalues of a real symmetric matrix $M \in \mathbb{R}^{n \times n}$ are orthogonal.

(c) Show that there exist n orthogonal eigenvectors of any real symmetric matrix $M \in \mathbb{R}^{n \times n}$ that span the entire n dimensional space.

(d) Using the previous part, show that the spectral norm of any matrix $A \in \mathbb{R}^{m \times n}$, defined to be $\|A\|_2 := \max_x \frac{\|Ax\|_2}{\|x\|_2}$, is achieved when x is the eigenvector corresponding to the largest eigenvalue of $A^T A$.

(e) Show that if the columns of a real square matrix $M \in \mathbb{R}^{n \times n}$ are orthonormal (i.e., have unit ℓ_2 norm and are pairwise orthogonal), then the rows of M are also orthonormal. (Hence, M is a unitary matrix.)

§2 This problem asks you to prove a bound on the spectral norm of random matrices which is valuable in analyzing many randomized algorithms.

Construct a random *symmetric* matrix $R \in \mathbb{R}^{n \times n}$ by setting $R_{ij} = R_{ji}$ to $+1$ or -1, uniformly at random. Prove that, with high probability,

$$\|R\|_2 \leq c\sqrt{n \log n},$$

for some constant c. This is much better than the naive bound of $\|R\|_2 \leq \|R\|_F = n$.

Hint: For a symmetric matrix R, show that another way to write the spectral norm besides $\|R\|_2 := \max_x \frac{\|Rx\|_2}{\|x\|_2}$ is that $\|R\|_2 = \max_x \frac{|x^T Rx|}{x^T x}$.

Hint: Try to bound $\frac{x^T Rx}{x^T x}$ for one particular x, and then extend the result to hold for all x, simultaneously, by taking a ϵ-net for $\epsilon = \frac{1}{\mathsf{poly}(n)}$ from Lecture 12. It is possible to solve this problem using the standard Hoeffding bound for bounded random variables – so don't use exotic concentration bounds!

§3 A *matroid* on $[n]$ elements is a collection of sets that generalized the concept of linear independence for vectors. Specifically, a matroid $\mathcal{I}$ satisfies:

- **Non-trivial:** $\emptyset \in \mathcal{I}$.
- **Downwards-closed:** If $S \in \mathcal{I}$, then $T \in \mathcal{I}$ for all $T \subseteq S$.
- **Augmentation:** If $S, T \in \mathcal{I}$, and $|S| > |T|$, then there exists an $i \in S \setminus T$ such that $T \cup \{i\} \in \mathcal{I}$.[1]

(a) Prove that the following collections are matroids:

i. Sets of size at most k (that is, the elements are $[n]$, and $\mathcal{I} = \{X \subseteq [n] | \ |X| \leq k\}$).

ii. Acyclic subgraphs of any undirected graph $G = (V, E)$ (that is, the elements are E and $\mathcal{I} = \{X \subseteq E | X \text{ contains no cycles}\}$).

iii. Let $G = (L, R, E)$ be a bipartite graph. The elements are L, and $\mathcal{I} = \{X \subseteq L | \ |N(S)| \geq |S| \ \forall S \subseteq X\}$ ($N(S)$ are the neighbors of S: $\{x \in R | \ \exists y \in S, (x, y) \in E\}$). That is, $X \in \mathcal{I}$ if and only if all nodes in X can be simultaneously matched to R.

(b) Given weights $w_i \geq 0, i \in [n]$, and some collection of feasible sets $\mathcal{I}$ where $\emptyset \in \mathcal{I}$, your goal is to find the max-weight feasible set: $\arg\max_{S \in \mathcal{I}} \{\sum_{i \in S} w_i\}$. Consider a greedy algorithm that first sorts the elements in decreasing order of w_i (i.e. picks a permutation σ such that $w_{\sigma(i)} \geq w_{\sigma(i+1)}$ for all i), then iteratively does the following (initializing $A = \emptyset$, $i = 1$, go until $i > n$): Check if $A \cup \{\sigma(i)\} \in \mathcal{I}$. If so, add $\sigma(i)$ to A. Update $i := i + 1$. Prove that this greedy algorithm finds the max-weight feasible set no matter what non-negative weights are input *if and only if* $\mathcal{I}$ is a matroid (that is, prove that the algorithm succeeds whenever $\mathcal{I}$ is a matroid. Also, if $\mathcal{I}$ is not a matroid, provide an instance of weights for which the algorithm fails).

[1] Think of this as a generalization of linear independence: if I give you a set S of k linearly independent vectors, and T of $< k$ linearly independent vectors, then there is some vector in S not spanned by T.

§4 In the submodular welfare problem, there are n bidders and m items. The value of bidder $i \in \{1, \ldots n\}$ for a subset of items $S \subseteq \{1, \ldots, m\}$ is given by a monotone submodular function $f_i(S)$ where $f_i(\emptyset) = 0$. We want to allocate the m items to the n bidders, i.e., find an item partition where bidder i gets susbset $S_i \subseteq \{1, \ldots, m\}$ and $S_i \cap S_j = \emptyset$ for $i \neq j$, and the goal is to maximize the welfare $\sum_{i \in \{1,\ldots,n\}} f_i(S_i)$. Show that the greedy algorithm which considers the items one-by-one in an arbitrary order and allocates the next item j to the bidder which has the highest marginal value (i.e., which maximizes $f_i(S \cup j) - f_i(S)$ where S is the currently allocated set of items to bidder i), gives a 2-approximation algorithm for the submodular welfare problem.

Hint: Note that a submodular function remains submodular even if you "contract" a set, i.e., $f_S(A) := f(S \cup A) - f(S)$ is also a submodular function on elements $\{1, \ldots, m\} \setminus S$.

§5 (Discrepancy Theory) Suppose you are given a matrix $A \in \{0,1\}^{n \times n}$ such that each column of A has at most s ones, i.e., column sparsity is at most s. We will analyze the following polynomial time algorithm to find a coloring $\vec{\varepsilon} \in \{-1, +1\}^n$ such that $\|A\vec{\varepsilon}\|_\infty = \|\sum_t \vec{a}_t \varepsilon_t\|_\infty = O(s)$, where $\vec{a}_t$ is the t-th column of A.

The algorithm will iteratively color more and more columns of A. Initially, all columns are "uncolored" and a column $\vec{a}_t$ gets "colored" when ε_t is defined.

- In any iteration, if the number of remaining uncolored columns is at most $2s$, color them arbitrarily and halt.
- Otherwise, we solve an LP with fractional variables denoting the fractional colors of the uncolored columns. For every row i that contains more than $2s$ uncolored ones, put an LP constraint that its fractional discrepancy is zero, i.e., $\sum_t a_t(i) Y_t = 0$ where Y_t denotes LP variable $x_t \in [-1, 1]$ for uncolored columns t, and Y_t denotes constant ε_t for colored columns t.
- Find a basic solution x^* of the above LP, and for any uncolored column t that gets $x^*_t \in \{-1, +1\}$ in this basic solution, we permanently set its ε_t to that color.
- Repeat.

(a) Show that in any iteration, the number of row constraints that we put is at most half the number of currently uncolored columns. Thus a basic solution will integrally color many columns in each iteration.

(b) Show that this algorithm obtains a solution with discrepancy $O(s)$.

Remark: There is nothing special about $A \in \{0,1\}^{n \times n}$ vs. $A \in \{0,1\}^{n \times T}$, i.e., having $T \gg n$ columns. We can easily extend the above result to T columns using the idea of finding a basic solution in the beginning with at most n uncolored columns.

§6 (Online set cover): In the online set cover problem, we are given a universe $U := \{1, \ldots, n\}$ of n elements and a family $\mathcal{S} := \{S_1, \ldots, S_m\}$ of m sets where each $\bigcup_i S_i = U$. The algorithm starts with $A = \emptyset$ which denotes the collection of selected sets. In each time step $t \in \{1, \ldots, T\}$, an adversary reveals an element $e_t \in U$, and the online algorithm has to immediately ensure that $e_t \in \bigcup_{S \in A} S$, i.e., if e_t is already covered

then the algorithm doesn't need to select a new set, and otherwise the algorithm has to select a set into A that contains e_t. The goal of the algorithm is to minimize the size of A. Show that there are problem instances such that the offline optimal solution has $|A| = O(1)$ but any deterministic online algorithm has size $\Omega(\log m)$.

Hint: You may prove the lower bound even against an online algorithm that is allowed to maintain a fractional set cover, i.e., it maintains $\vec{x} \in [0,1]^m$ such that $\sum_{i \,:\, e_t \in S_i} x_i \geq 1$ and the algorithm is only allowed to increase the coordinates of $\vec{x}$.

Extra Credit:

§1 (Extra credit) Calculate the eigenvectors and eigenvalues of the (adjacency matrix of the) n-dimensional boolean hypercube, which is the graph with vertex set $\{-1,1\}^n$ and x, y are connected by an edge iff they differ in exactly one of the n locations. (Hint: Use symmetry extensively.)

§2 (Open Problem) Can you extend the discrepancy result in Problem 5 to every prefix, i.e., find a coloring $\vec{\varepsilon}$ such that $\max_t \| \sum_{i \leq t} \vec{a}_i \varepsilon_i \|_\infty = O(s)$? Can you prove this result for some other function $f(s)$ that does not depend on n and T (like $f(s) = 2^s$)?