Problem Set 3

Due: Wednesday, September 28, 2016.

Collaboration policy: collaboration is *strongly encouraged.* However, remember that

1. You must record the name of every collaborator.
2. You must actually participate in solving all the problems. This is difficult in very large groups, so keep your collaboration groups limited to 3 people at a time.
3. You must write up your own solutions, independently and without relying excessively on notes from your collaboration.
4. **No bibles. This includes solutions posted to problems in previous years.**

Problem 1. Count-Min with Insertions and Deletions. Consider a warehouse that receives items to store and then eventually ships the items elsewhere over the course of a year. For example, the sequence of events might be: store 3 boxes of candy, store 1 box of rice, ship 2 boxes of candy, store 5 bottles of soda, and so on. Let k be some positive number. We call an item type overstocked if items of that type account for at least a $1/k$-fraction of the total items in the warehouse.

At the end of the year, the warehouse wants to know the set of overstocked item types. If the warehouse only received items one at a time and never shipped anything out, this would be an instance of the heavy hitters problem, and we could solve it using the Count-Min algorithm from class.

Let me m be the size of the universe of types of items that the warehouse might receive, and let n be the maximum number of items that the warehouse can store at any one time. Argue how to modify the Count-Min algorithm from class so that it will give the set of overstocked items with probability of success at least 2/3 using $O(k \cdot \log(m) \cdot \log(n))$ bits of space. If you wish, you may assume that each item type has a unique integer ID between 1 and m, inclusive.

Problem 2. A Streaming Lower Bound.

Consider the exact distinct elements problem, which is the problem of deterministically counting the exact number N of distinct elements in a stream in one pass exactly. Assume the elements come from a universe of size m, where m is much larger than N.

(a) Argue that any algorithm for this problem has to remember all of the elements it has seen in the following sense: if the algorithm has seen half the stream, then by looking at the internal state of the algorithm then, we can always determine the set of distinct elements it has seen so far.

(b) How many different possible internal states must an algorithm have if it solves the exact distinct elements problem?

(c) Using both (a) and (b), conclude that for any particular algorithm which solves the distinct elements problem, there exists a sequence of elements which causes the algorithm to use $\Omega(N \cdot \log(m))$ bits of space.

Problem 3. Testing Bipartiteness in the Streaming Model.

Suppose that there is some graph $G = (V, E)$, and we would like to determine whether it is bipartite. We are given a stream consisting of the edges of the graph in some arbitrary (adversarial) order.

(a) Give an $O(|V| \log |V|)$ bit upper bound on the space sufficient to solve this problem deterministically. (Note that storing all the edges we see would take $O(|E| \log |E|)$ space.)

(b) Given an $\Omega(|V| \log |V|)$ bit lower bound on the space needed to solve this problem deterministically.

NONCOLLABORATIVE Problem 4. Which of the following statements about flows are true and which are false? Justify your answer with a (short) proof or counterexample.

(a) If all directed edges in a network have distinct capacities, then there is a unique maximum flow.

(b) If we replace each directed edge in a network with two directed edges in opposite directions with the same capacity and connecting the same vertices, then the value of the maximum flow remains unchanged.

(c) If we add the same positive number λ to the capacity of every directed edge, then the minimum cut (but not its value) remains unchanged.

(d) Flow is transitive: For all graphs G, for all $s, t, u \in G$, if there is a flow of value $v \in \mathbb{R}$ from s to t (i.e. with source s and sink t), and there is a flow of value v from t to u, then there is a flow of value v from s to u.

Problem 5. At a certain point in the season, each team i in the American League has won a certain number w_i of games, and there remain q_{ij} games to be played between teams i and j. Assuming no ties or cancelled games, develop an efficient, flow-based algorithm for deciding if the Red Sox can still win the league pennant, i.e. whether a particular team can still win more games than any other team after all games have been played. Assume you have an algorithm that solves max-flow efficiently.

Problem 6. The JL Lemma in Different Norms

The JL Lemma says that there is a linear operation that takes high-dimensional vectors, outputs low-dimensional vectors, and approximately preserves distances between vectors, or equivalently, the lengths of vectors. The notion of distance it uses for two vectors x and y is the Euclidean distance $\|x - y\|_2$. In this problem, we'll consider preserving other notions of distance.

(a) Define the A-norm of a vector $x \in \mathbb{R}^n$ denoted $\|x\|_A = \sqrt{x^\intercal A x}$. (When $A = I$, this is simply the ℓ_2-norm.) Suppose we have access to a matrix B that behaves like a "square-root" of A in the sense that $A = B^\intercal B$. State and prove a version of the JL Lemma that is the same as what was proved in class, except that it approximately preserves A-norm rather than ℓ_2-norm.

(b) One might hope to prove analogues of the JL Lemma for all "reasonable" norms. Show that for the ℓ_∞ norm, there is no such analogue. IE., give a counterexample showing that if one replaces the 2 with ∞ in the statement of the JL Lemma, it is no longer true. **Hint:** Use the requirement that the operation is linear. (Note however that through more complicated arguments, one can show a counterexample that rules out even nonlinear operations.)

(c) Show that for the ℓ_1 norm, there is also no such analogue. **Hint:** Use the requirement that the operation is linear. (Once again, note that through more complicated arguments, one can show a counterexample that rules out even nonlinear operations.)

Problem 7. 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.