College of Computer & Information Science
Northeastern University
CS7880: Network Algorithms and Analysis

Spring 2016
Handout 4
1 April 2016

Problem Set 3 (due Monday, April 18)

Remarks:

- Since the background of the students in the class varies, some may find these problems easy while others may find some problems challenging. The main purpose of these problem sets is for you to learn the course material better by applying ideas learnt in class and/or exploring related problems.
- Please work on these problems on your own, or in collaboration with fellow students in class. Please do not seek out solutions or solution approaches from the web. If you need hints, ask the course instructor.
- Please typeset your solution. Latex, plain text, pdf, and Word are all acceptable formats.
- The total points for the problem set is 75, worth 7.5% of the grade.

Problem 1. (20 points) 2-Universal hash functions

We used a 2-universal hash function in the streaming algorithm for estimating the number of distinct elements in a stream. In this exercise, we construct one class of 2-universal hash functions.

Let $\mathcal{H}$ be a finite collection of hash functions that map a given universe $X = \{0,1\}^n$ into a range $Y = \{0,1\}^m$. The collection $\mathcal{H}$ is 2-universal if for each pair of distinct keys $x, x' \in X$ and pair of elements $y, y' \in Y$, if h is drawn uniformly at random from $\mathcal{H}$, then

$$\Pr[h(x) = y \text{ and } h(x') = y'] = \frac{1}{2^{2m}}.$$

Suppose we represent each element of X as an $n \times 1$ column $0-1$ vector and each element of Y as an $m \times 1$ column vector in the standard manner. For an arbitrary $m \times n$ matrix $A \in \{0,1\}^{m \times n}$ and vector $b \in \{0,1\}^m$, we define the function $h_{A,b} : X \to Y$ as $h_{A,b}(x) = Ax + b$, where all additions and multiplications are modulo two.

Prove that the collection $\mathcal{H} = \{h_{A,b} : A \in \{0,1\}^{n \times m}, b \in \{0,1\}^m\}$ is 2-universal.

Problem 2. (15 points) Estimating the number of distinct colors in an urn of balls

You are given an urn with a large finite number of colored balls. You are asked to estimate the number of distinct colors in the urn. The only operation you are allowed is to take *samples* from the urn, uniformly at random with replacement.

Give a procedure for yielding an unbiased estimator; that is an estimator, whose expectation equals the number of distinct colors. What is the variance of your estimator?

(*Remark:* I hope you find this to be a fun puzzle. Recall that in the streaming algorithm we studied for the second frequency moment, our unbiased estimator consisted of picking a random element in the stream and then returning the number of occurrences in the remainder of the stream. The expectation of this quantity was proportional to $\sum_i f_i \cdot f_i$ and hence gave the second frequency moment. Come up a similar estimator for the zeroth moment in this exercise.)

Problem 3. (10 + 10 = 20 points) Approximating graph bisection by reduction to trees

In the graph bisection problem, you are given an edge-weighted graph $G = (V, E)$ and are asked to determine a subset S of V of size $\lfloor V/2 \rfloor$ that minimizes the weight the cut $(S, \overline{S})$. In this exercise, we show that one can obtain a good approximation algorithm to this problem by reduction to trees.

(a) Give a polynomial-time algorithm to solve graph bisection optimally over trees.

Suppose for any graph edge-weighted $G = (V, E)$, we can compute, in polynomial-time, a probability distribution $\mathcal{D}$ over edge-weighted trees (with new weights) over the same set V of vertices, with the following property for some $\alpha \geq 1$: for any cut $(S, V - S)$ of G, (i) the weight of the cut in any tree in $\mathcal{D}$ is at least the weight of the cut in G; and (ii) the expected weight of the cut in a tree drawn uniformly at random from $\mathcal{D}$ is at most α times the weight of the cut in G.

(b) Show that there exists a randomized polynomial-time algorithm that computes a solution to the graph bisection problem for any graph of expected cost at most α times optimal.

Problem 4. (20 points) Multiset multicover problem

The *multiset multicover problem* is a generalization of set cover in which we have multisets instead of sets and an element may be required to be covered multiple times. Formally, we are given a universe $\mathcal{U}$ of elements, a positive integer r_e for each element $e \in \mathcal{U}$, a collection $\mathcal{C}$ of multisets over $\mathcal{U}$, and a cost $c(S)$ for each multiset $S \in \mathcal{C}$. (A multiset contains a specified number of copies of each element.)

The goal of the multiset multicover problem is to determine a minimum-cost multiset of multisets (thus, you are allowed to pick multiple copies of a multiset) such that each element e is covered at least r_e times. The cost of picking a multiset S, k times, is $k \cdot c(S)$. (You may assume that the number of times that an element e appears in any multiset S is at most r_e.)

Generalize either the greedy algorithm or the randomized rounding algorithm for set cover to achieve an $O(\log(r + n))$ approximation, where n is the number of elements in $\mathcal{U}$ and r is the sum of r_e over all e.