CS 574. Randomized Algorithms
Problem Set 1

Alexandra Kolla

due September 22 , 2015

Collaboration Policy: The homework can be worked in groups of up to 3 students each (2 would be optimal, but 1 and 3 are both accepted).

One submission per team is sufficient. Please write the solution for each of the problems on a separate sheet of paper. Write your team names and netids on each submission and please **staple** all the sheets together.

Extra 5% for typed homeworks (preferably pdf).

Homework is due on or before the end of class, September 22. Only one late homework per person will be allowed. If you submit more than one homework late, you will get no grade for the excess late homeworks.

Problem 1 *Testing Commutativity*

(6 pts.)

Let G be a finite group with group operation $*$. In this question we consider the problem of determining whether G is abelian, i.e., whether all elements commute $x * y = y * x$ for all $x, y \in G$. We assume that G is presented to us as a set of generators, $\{g_1, g_2, \cdots, g_k\}$ (i.e., every element of G can be written as a product of various g_i) together with a black box that returns any desired product $g_i * g_j$ in unit time. Clearly we can test whether G is abelian in $O(k^2)$ time by simply checking if $g_i * g_j = g_j * g_i$ for alli, j. We will now see a randomized algorithm that runs in $O(k)$ time and is correct with high probability. The algorithm makes use of a random product $h = {g_1}^{b_1} * {g_2}^{b_2} * \cdots * {g_k}^{b_k}$, where the $b_i \in \{0, 1\}$ are independent fair coin flips. (Thus a random product is a product of a random subset of the generators, in fixed order.) Here is the algorithm:

- let h, h' be two independent random products
- If $h * h' \neq h' * h$ then output “G is not abelian” else output “G is probably abelian”.

Plainly this algorithm runs in $O(k)$ time, and is always correct when G is abelian. So all that is left is to prove that the error probability is bounded when G is not abelian.

1. Suppose H is any proper subgroup of G. For a random product h, show that $Pr[h \notin H] \geq 1/2$.

2. Deduce from part (1) that, when G is not abelian, the above algorithm reports a correct answer with probability at least 1/4 . [HINT: Let C be the set of elements of G that commute with all elements of G. Then C is always a subgroup of G (called the center of G). Moreover, for any element a of G, the set $Z(a)$ of elements that commute with a is also a subgroup (called the centralizer of a).]

Problem 2 *De-friended*

(3 pts.)

Consider the following simple model for a social network: There are $2n$ users in two groups A and B with $|A| = |B| = n$. For every distinct pair of users i and j, the pair $\{i, j\}$ are friends independently with probability p if they are in the same group, and probability q if they are in different groups. A loner is a user i that has no friends. Let $\mathcal{L}$ denote the event that there exists a loner. Prove that if $p + q >> \frac{\ln n}{n}$,then $Pr[\mathcal{L}] \to 0$ and if $p + q << \frac{\ln n}{n}$, then $Pr[\mathcal{L}] \to 1$. [Here the notation $f(n) >> g(n)$ means that $lim \frac{f(n)}{g(n)} \to \infty$.]

Problem 3 *Surfs' up*

(6 pts)

The Inter-Galactic School of Surfing (IGSS), which you already know from CS473 if you were in my class, strikes back! The IGSS, after a long vacation, wants to send a subset of its surfers to a competition in Kauai. Let V, $|V| = n$, be the set of IGSS surfers, which you can represent as vertices of a graph G. Each surfer v has a group of friends, so that if v and u are friends there is an edge in G between u and v. For a subset of surfers $U \subseteq V$, we call $D \subseteq V$ a *valid team* representing U if for every $u \in U$, either $u \in D$ or some friend of u is in D. Let d be an integer and let U_d be the set of surfers with at least d friends. The IGSS needs to send a *valid team* representing U_d to Kauai but unfortunately there are space limits: a team cannot have more than $N = \lfloor n \frac{\log(d+1)+1}{d+1} \rfloor$ members [1]. Show that there always exist a valid team for U_d with at most N surfers in it.

[1] surfers are encouraged to be social and have many friends, that's why the IGSS initally picks U_d, a set of people with many friends. However, if this set is too big then other surfers might need to be in the team instead

Problem 4 *How to Win in a Casino*

(5 pts.)

Consider, hypothetically, a fair game in a casino: on each play, you may stake any amount S; you win or lose with probability $1/2$ each (all plays being independent); if you win you get your stake back plus S; if you lose you lose your stake.

1. What is the expected number of plays before your first win (including the play on which you win)?

2. The following gambling strategy, known as the "martingale", was popular in European casinos in the 18th century: on the first play, stake 1 dollar; on the second play 2 dollars; on the third play 4 dollars; on the kth play 2^{k-1} dollars. Stop (and leave the casino!) when you first win.

3. Show that, if you follow the martingale strategy, you will leave the casino 1 dollar richer with probability 1. (Maybe this is why the strategy is banned in most modern casinos.)

4. To discover the catch in this seemingly infallible strategy, let X be the r.v. that measures your maximum loss before winning (i.e., the amount of money you have lost before the play on which you win). Show that $E[X] = \infty$. What does this imply about your ability to play the martingale strategy in practice?