22**C** : 231 ($CS$ : 5350 : 0001) **Design and Analysis of Algorithms**
**Homework 4**

The homework has two types of problems – reinforcement problems and regular problems. For the reinforcement problems, we are not concerned with originality in coming up with the solution, but rather with how well you write up the solution. You can get help in coming up with the solution – from friends, online, etc. – but understand the solution and explain it in your own words. For the regular problems, the only type of help you can get is collaboration with classmates, and discussion with the instructor or TA. No record or notes, electronic or written, should be taken from such collaborations. For these problems we do care about originality in coming up with the solution.

The homework is worth 12 points. The first six questions are worth one point each, and the last three are worth two points each. The theme is probability and randomized algorithms. The homework is due in class on Thursday, March 29.

## Regular Problems

1. Suppose we have a graph with $n > 5$ vertices that form a ring. That is, the vertex set of the graph is $\{1, 2, \ldots, n\}$ and the edge set is $\{(1, 2), (2, 3), \ldots, (n-1, n), (n, 1)\}$. Consider the following protocol for picking a subset $S$ of the nodes. The number $0 < p < 1$ is a parameter.

   Node $i$ independently picks a random value $x_i$ in the following way: it sets $x_i$ to 1 with probability $p$, and $x_i$ to 0 with probability $1-p$. It then decides to enter the set $S$ if and only if it chose the value 1, its two neighbours chose the value 0, and the two nodes at distance 2 from it chose the value 1.

   What is the expected size of $S$ in terms of $n$ and $p$ ?

2. This question is about the design of the universal family of hash functions. Suppose we want to map a large universe $U$ to a table of size $p$, where $p > 2$ is some prime number. In our situation, $|U|$ is much larger than $p$. For concreteness, suppose that $U = \{0, 1, 2, \ldots, p^3 - 1\}$, so that $|U| = p^3$.

   Consider the following method of choosing a random hash function, which is a simplification of the method we used in class to choose a random hash function from a universal family. We pick a number $a$ uniformly at random from the set $\{0, 1, \ldots, p-1\}$; this defines a hash function $h_a : U \to \{0, 1, \ldots, p-1\}$ given by $h_a(x) = ax \bmod p$.

   Is the following statement true or false : For any two distinct elements $x_1$ and $x_2$ of $U$, the probability, under the random choice of $a$, that $h_a(x_1) = h_a(x_2)$ is at most $1/p$. Substantiate your answer.

3. A monket types on a 26-letter keyboard that has lower case letters only. Each letter is chosen independently and uniformly at random from the alphabet. If the monkey types $1,000,000$ letters, what is the expected number of times the sequence "proof" appears?

4. Suppose we have a sequence of items passing by one at a time. We want to maintain a sample of one item with the property that it is uniformly distributed over all the items that we have seen at each step. Moreover, we want to accomplish this without knowing the total number of items in advance or storing all the items that we see.

   Consider the following algorithm, which stores just one item in memory at all times. When the first item appears, it is stored in memory. When the $k$'th item appears, it replaces the item in memory with probability $1/k$. Explain why this algorithm solves the problem.

5. Suppose that we roll a standard fair die 100 times. Let $X$ be the sum of the numbers that appear over the 100 rolls. Use Chebyshev's inequality to bound $\Pr(|X - 350| \geq 50)$. You'll need to use the fact that for independnent random variables, the variance of the sum is equal to the sum of the individual variances.

6. Alice and Bob play checkers often. Alice is a better player, so the probability that she wins any given game is 0.6, independent of all other games. They decide to play a tournament of $n$ games. Bound the probability that Alice loses the tournament using a Chernoff bound.

7. Consider the following algorithm select$(S, k)$ for selecting the $k$'th smallest in a set $S$ of distinct integers. We pick a uniformly random pivot $x \in S$, and by comparing each element in $S \setminus \{x\}$ with $x$, compute $S_1 = \{y \in S \mid y < x\}$ and $S_2 = \{y \in S \mid y > x\}$. If $|S_1| \geq k$, we recursively call select$(S_1, k)$ and return whatever this call returns; if $|S_1| = k - 1$, we return $x$; if $|S_1| < k - 1$, we recursively call select$(S_2, k - |S_1| - 1)$ and return whatever this call returns.

   We want to bound the expected running time of select$(S, k)$ by $O(n)$ where $n = |S|$. We can do this by bounding the expected number of comparisons. Bound the latter, by following the method used in class for Quicksort with indicator random variables corresponding to each pair of elements.

## Reinforcement Problems

1. Exercise 9 of Chapter 13

2. Exercise 12 of Chapter 13. This refers to the basic edge-contraction algorithm, not the fancy one we discussed on top of this.