Sample Midterm 1

6:00-8:00pm, 28 February

Read these instructions carefully

1. This is a **closed book** exam. Calculators **are** permitted.
2. This midterm consists of 10 questions. The first seven questions are multiple choice; the remaining three require written answers.
3. Answer the multiple choice questions by **circling** the correct answer. You should be able to answer all of these from memory, by inspection, or with a very small calculation. Incorrect answers will receive a negative score, so if you do not know the answer you should **not** guess.
4. Write your answers to the other questions in the spaces provided below them. None of these questions requires a long answer, so you should have enough space; if not, continue on the back of the page and state clearly that you have done so. **Show all your working**.
5. The questions vary in difficulty: if you get stuck on some part of a question, leave it and go on to the next one.

Your Name:

---

1. Alice and Bob are two people in a group of size n. The group is ordered randomly in a line. The probability that there are *exactly* k people between Alice and Bob is

$$\frac{k}{n} \qquad \frac{n-k}{n(n-1)} \qquad \frac{(n-k-1)}{n!} \qquad \frac{(n-k-1)}{n(n-1)} \qquad \frac{2(n-k-1)}{n(n-1)}$$

---

2. Each cereal box contains one coupon, chosen independently and uniformly at random from a set of n different coupons.

(a) The expected number of boxes that need to be bought before a copy of *some particular* coupon is obtained is

$$n^2 \qquad n\ln\ln n \qquad \sqrt{n} \qquad n \qquad n\ln n \qquad \mathrm{e}^n$$

(b) The expected number of boxes that need to be bought before at least one copy of *all* n coupons is obtained is on the order of

$$n^2 \qquad n\ln\ln n \qquad \sqrt{n} \qquad n \qquad n\ln n \qquad \mathrm{e}^n$$

[continued overleaf]

3. An unbiased coin is flipped repeatedly until *both* heads and tails are obtained. The expected number of times the coin is flipped is

$$2 \qquad \frac{5}{2} \qquad 3 \qquad \frac{10}{3} \qquad \frac{7}{2} \qquad \frac{9}{2}$$

4. Three fair six-sided dice are rolled. *Given that* at least one of the dice comes up 6, the probability that *exactly* one of them comes up 6 is

$$\frac{25}{216} \qquad \frac{75}{216} \qquad \frac{125}{216} \qquad \frac{25}{91} \qquad \frac{75}{91} \qquad \frac{1}{3}$$

5. $10n$ balls are thrown at random into n bins.

(a) The probability that the first bin contains exactly k balls is

$$\tbinom{10n}{k}(\tfrac{1}{n})^k(\tfrac{n-1}{n})^{10n-k} \qquad \tbinom{n}{k}(\tfrac{1}{n})^k(\tfrac{n-1}{n})^{10n-k} \qquad 10\tbinom{n}{k}(\tfrac{1}{n})^k(\tfrac{n-1}{n})^{10n-k} \qquad (\tfrac{1}{n})^k(\tfrac{n-1}{n})^{10n-k} \qquad \tbinom{10n}{k}(\tfrac{1}{n})^{10n}$$

(b) As $n \to \infty$, for fixed k, this probability becomes very close to

$$\tfrac{1}{10} \qquad \mathrm{e}^{-1}\tfrac{1}{k!} \qquad \mathrm{e}^{-10}\tfrac{1}{k!} \qquad \mathrm{e}^{-1}\tfrac{10^k}{k!} \qquad \mathrm{e}^{-10}\tfrac{10^k}{k!}$$

6. X and Y are arbitrary *independent* random variables satisfying $E[X] = E[Y] = 1$ and $\mathrm{Var}[X] = \mathrm{Var}[Y] = 2$. Circle those <u>three</u> of the following statements that <u>must</u> be true:

$$\Pr[X = 1] < 1 \qquad E[1/X] = 1 \qquad E[X^2] = 1$$

$$\Pr[X \geq 2] \leq \tfrac{1}{2} \qquad E[XY] = 1 \qquad \mathrm{Var}[2X + Y] = 10$$

7. **(a)** A coin with heads probability p is tossed n times. The expected number of heads is

$$\tfrac{1}{p} \qquad \tfrac{n}{2} \qquad np \qquad np(1-p) \qquad \tfrac{n}{p}$$

(b) A *run* in a sequence of coin tosses is a maximal subsequence of either heads or tails. (Thus, for example, the sequence `HHHTTHTHH` contains five runs.) The expected number of runs in n tosses of a coin with heads probability p is

$$\tfrac{n}{2} \qquad np \qquad np(1-p) \qquad 2(n-1)p(1-p)+1 \qquad \tfrac{np}{1-p}$$

[continued overleaf]

8. Finding a long path in a graph

Let $G = (V, E)$ be a directed graph with n vertices $1, 2, \ldots, n$. Suppose we want to determine whether G contains a simple path of length at least 5 starting from a given vertex s. (A *simple* path is one which does not visit any vertex more than once.) A brute-force search from s would take time $O(n^5)$, which is prohibitive for large n. In this problem, we will come up with a randomized algorithm that works with high probability and runs in linear time! The algorithm is as follows:

(1) Generate a random permutation π of the vertices $\{1, 2, \ldots, n\}$
(2) Delete from G all edges (i, j) for which $\pi(i) > \pi(j)$
(3) In the resulting graph (which must be acyclic), find a longest path from s in linear time
(4) Output this path if it is of length at least 5

(You should have seen in CS170 how to solve the longest path problem for acyclic graphs by dynamic programming, as required in step (3). However, you won't need to remember the details here.) The algorithm clearly runs in time linear in the size of the graph: step (1) takes $O(n)$ time, steps (2) and (3) take linear time, and step (4) is trivial.

(a) Verify the claim in line (3) that the graph remaining after the edge deletions is acyclic.

(b) Let π be a random permutation of n items. Show that the probability $\Pr[\pi(1) < \pi(2) < \ldots < \pi(k)]$ is exactly $\frac{1}{k!}$. [Hint: In a random permutation, what is the distribution of the positions of items $1, 2, \ldots, k$?]

(c) Now suppose that G contains a simple path from s of length 5. What is the probability that $\mathcal{P}$ survives the edge deletion process in the algorithm?

(d) Deduce that, if G contains such a path then the algorithm will find one with probability at least $\frac{1}{720}$.

(e) Suppose you wanted to boost the probability in part (c) to $\frac{99}{100}$. Explain how you would do this; you need not perform a complete calculation, but you should indicate what calculation you would need to perform.

[continued overleaf]

9. The ABC of 123

Alice, Bob and Charlie want to choose one of the numbers 1,2 and 3 with equal probabilities. All they have is a biased coin that comes up heads with probability p, where $0 < p < 1$.

Alice suggests the following method. Toss the coin twice. If the two flips are HH, output 1, if they are HT, output 2, if they are TH, output 3. If the flips are TT, repeat the experiment.

Bob suggests the following method. Toss the coin twice. Output the number of heads obtained, plus 1.

Charlie suggets the following method. Toss the coin three times. If heads is obtained only in the i-th toss, where $1 \leq i \leq 3$, output i. Otherwise, repeat the experiment.

(a) For which values of p, if any, does Alice's method produce the numbers 1,2 and 3 with equal probabilities?

(b) For which values of p, if any, does Bob's method produce the numbers 1,2 and 3 with equal probabilities?

(c) For which values of p, if any, does Charlie's method produce the numbers 1,2 and 3 with equal probabilities?

[continued overleaf]

(d) What is the expected number of tosses used by Alice's method.

(e) What is the variance of the number of tosses used by Charlie's method.

(f) Suppose that you have a supply of biased coins, each with an unknown, and possibly different, bias. You are allowed to use each coin only twice. Can you use the coins to select one of the numbers 1,2 and 3 with equal probabilities? Show how or explain why this is not possible.

[continued overleaf]

10. Max3Sat

In the problem MAX3SAT, we are given a 3CNF formula ϕ (i.e., an AND of OR's wherein each clause contains three literals) and we want to find an assignment that satisfies the maximum possible number of clauses. We will restrict ourselves to formulas ϕ such that any clause contains exactly three distinct variables, and any two distinct clauses share at most two variables. MAX3CNF is an NP-hard problem, so we do not expect to find an efficient algorithm that solves it exactly. Here is a very simple linear-time randomized algorithm that gives a pretty good approximation:

(1) Randomly and independently assign each variable x_i a value 0 or 1 with probability $\frac{1}{2}$ each

(2) Output the resulting assignment

Let the r.v. X denote the number of satisfied clauses for the assignment output by the algorithm. In addition, for every clause c in ϕ, let X_c be the indicator r.v. that assumes the value 1 if c is satisfied and 0 otherwise.

(a) Write down the equation relating the random variable X and the random variables X_c.

(b) Show that $E[X] = \frac{7}{8}m$ where m is the number of clauses in ϕ. Deduce that $E[X] \geq \frac{7}{8}\text{OPT}$, where OPT is the maximum number of satisfied clauses over all possible assignments to ϕ.

(c) Let $Y = m - X$. Explain why Y is a non-negative r.v., and use Markov's inequality to obtain an upper bound for $\Pr[Y \geq \frac{1}{4}m]$.

(d) Let p denote the probability that the assignment output by the algorithm satisfies at least $\frac{3}{4}\text{OPT}$ clauses. Using part (c), show that $p \geq \frac{1}{2}$.

[continued overleaf]

11. Randomized algorithms with two-sided errors

Let Π be some problem with yes/no answers on every input. We will denote by $\Pi(x)$ the answer to Π on input x. This question is concerned with reducing the error probability of randomized algorithms with *two-sided* errors.

(a) We begin with the familiar case of one-sided errors. Suppose Algorithm $\mathcal{A}$ behaves as follows: on every input x,

(i) if $\Pi(x) =$ yes then $\mathcal{A}$ outputs "yes" with probability $\geq \frac{1}{2}$, and "no" otherwise;
(ii) if $\Pi(x) =$ no then $\mathcal{A}$ outputs "no" with probability 1.

Explain how to modify $\mathcal{A}$ so that the probability of error in case (i) is at most ϵ, for any desired $\epsilon > 0$, and the increase in running time is a factor of $\mathrm{O}(\log(\frac{1}{\epsilon}))$.

(b) In preparation for dealing with two-sided errors, we first derive a useful fact about coin-tossing. Show using a Chernoff bound that, when a biased coin with Heads probability $\frac{3}{4}$ is tossed n times, the probability that at most $\frac{n}{2}$ Heads are observed is at most $\exp(-cn)$, for some constant c (which you should specify).

(c) Now consider Algorithm $\mathcal{B}$, which has two-sided error and behaves as follows: on every input x,

(i) if $\Pi(x) =$ yes then $\mathcal{B}$ outputs "yes" with probability $\geq \frac{3}{4}$, and "no" otherwise;
(ii) if $\Pi(x) =$ no then $\mathcal{B}$ outputs "no" with probability $\geq \frac{3}{4}$, and "yes" otherwise.

Show how to modify $\mathcal{B}$ so that the probability of error in both cases is at most ϵ, for any desired $\epsilon > 0$, and the increase in running time is a factor of $\mathrm{O}(\log(\frac{1}{\epsilon}))$. [Hint: Use part (b).]

[The End]