Markov: For any non-negative random variable X and any $q > 0$, $\Pr\{X > q\} \le \mathrm{E}[X]/q$.
Chernoff: For any binomial(p, n) random variable B and any $\epsilon > 0$,
$\Pr\{B \ge (1+\epsilon)np\} \le e^{-\epsilon^2 np/3}$ and $\Pr\{B \le (1-\epsilon)np\} \le e^{-\epsilon^2 np/2}$.

COMP4804 Midterm Exam — Duration 1h20m **Winter 2010**

Answer all questions in your exam booklet. No notes, calculators, or books allowed.

1. [**2 marks**] Suppose you have a Monte Carlo randomized algorithm $\mathcal{A}$ that, with probability 1/10, gives a correct answer and, with probability 9/10 doesn't give any answer at all. In either case, $\mathcal{A}$ runs in $O(n)$ time. Explain how to use $\mathcal{A}$ to make a Las Vegas algorithm $\mathcal{A}'$ that runs in $O(n)$ expected time and always outputs a correct answer.
2. [**2 marks**] Suppose you have a Las Vegas randomized algorithm $\mathcal{B}$ that always outputs a correct answer and runs in expected time $f(n)$. Describe how to use $\mathcal{B}$ to get a Monte Carlo algorithm $\mathcal{B}'$ that runs in time $20 \cdot f(n)$ in the worst case and that outputs a correct answer with probability at least $1 - 1/2^{10}$.
3. [**3 marks**] Suppose you have a Monte Carlo algorithm $\mathcal{C}$ for solving some problem. When we run $\mathcal{C}$, it runs in $O(n)$ time and produces the correct answer (yes or no) with probability 7/8.

   (a) If we run $\mathcal{C}$ k times, what is the expected number of times $\mathcal{C}$ produces the correct answer?

   (b) Give a tight upper bound on the probability that $\mathcal{C}$ produces the correct answer fewer than $k/2$ times.

   (c) Describe how, using $\mathcal{C}$ as a subroutine, we can get a Monte Carlo algorithm $\mathcal{C}'$ that runs in $O(kn)$ time and produces the correct answer with probability at least $1 - e^{-\Omega(k)}$.

4. [**2 marks**] Suppose we have n elements $x_0, \ldots, x_{n-1}$ arranged in a circular list, so that x_i is adjacent to $x_{(i-1) \bmod n}$ and $x_{(i+1) \bmod n}$. We color this list by assigning, to each x_i, a random color uniformly and independently from the set $\{1, 2\}$

   (a) An element x_i is *bad* if either of its two neighbours receive the same color as x_i. What is the probability that x_i is bad? What is the expected number of bad elements?

   (b) The $t+1$ elements $x_i, \ldots, x_{i+t}$ form a *bad streak* of length t if they all have the same color. For a particular value of i, what is the probability, as a function of t, that $x_i, \ldots, x_{i+t}$ form a bad streak?

   (c) Set $t = 2\log_2 n$ and give a good upper bound on the probability that there exists *any* bad streak (starting at any x_i, for any $i \in \{1, \ldots, n\}$) of length t.

   [FYI: This gives an efficient parallel algorithm for 3-coloring a cycle. Do the random 2-coloring described above (in parallel) and then (sequentially) color each of the bad streaks using 3 colors.]
5. [**2 marks**] Consider an algorithm that works with a list L and runs in m rounds. During round i, the algorithm either appends one element to L (at a cost of $C_i = O(1)$) or deletes some number, $0 \le k_i \le |L|$, of elements from L (at a cost of $C_i = O(1 + k_i)$).

   (a) Define a non-negative potential function $\Phi(L)$ and use it show that the amortized cost of the ith round is $O(1)$.