 Sid Banerjee (sbanerjee@cornell.edu)

Problem 1: (Practice with Asymptotic Notation)

An essential requirement for understanding scaling behavior is comfort with asymptotic (or 'big-O') notation. In this problem, you will prove some basic facts about such asymptotics.

Part (a)

Given any two functions $f(\cdot)$ and $g(\cdot)$, show that $f(n) + g(n) = \Theta(\max\{f(n), g(n)\})$.

Part (b)

An algorithm ALG consists of two *tunable* sub-algorithms ALG_A and ALG_B, which have to be executed serially (i.e., one run of ALG involves first executing ALG_A followed by ALG_B). Moreover, given any function $f(n)$, we can tune the two algorithms such that one run of ALG_A takes time $O(f(n))$ and ALG_B takes time $O(n/f(n))$. How should we choose f to minimize the overall runtime of ALG (i.e., to ensure the runtime of ALG is $O(h(n))$ for the smallest-growing function h)?

How would your answer change if ALG_A and ALG_B could be executed in parallel, and we have to wait for both to finish?

Part (c)

We are given a recursive algorithm which, given an input of size n, splits it into 2 problems of size $n/2$, solves each recursively, and then combines the two parts in time $O(n)$. Thus, if $T(n)$ denotes the runtime for the algorithm on an input of size n, then we have:

$$T(n) = 2T(n/2) + O(n)$$

Prove that $T(n) = O(n \log n)$.
Hint: Note that for a constant size input, the algorithm takes $O(1)$ time. How many recursions does it require to reduce a problem of size n to constant size subproblems? What is the total runtime overhead at each recursive level?

Problem 2: (Some important asymptotes)

Part (a)

In class, we defined the harmonic number $H_n = \sum_{i=1}^{n} 1/i$. Argue that:

$$\int_1^{n+1} \frac{1}{x} dx \leq H_n \leq 1 + \int_1^n \frac{1}{x} dx$$

Thus, prove that $H_n = \Theta(\ln n)$.
Hint: Bound the $1/x$ function from above and below by a step function.

Part (b)

Next, we try to find the asymptotic growth $n!$. As in the previous part, argue that:

$$\int_1^n \ln x dx \leq \ln n! \leq \int_1^{n+1} \ln x dx$$

Thus, prove that $n! = \Theta(n \ln n)$.

Part (c)

(Stirling's approximation) We now improve the estimate in the previous part to get the familiar form of Stirling's approximation. First, argue that for any integer $i \geq 1$, we have:

$$\int_i^{i+1} \log x dx \geq \frac{\log i + \log(i+1)}{2}$$

Using this, show that:

$$n! \leq e\sqrt{n}\left(\frac{n}{e}\right)^n$$

Hint: Given any $i > 1$, where does the line joining the two points $(i, \ln i)$ and $(i+1, \ln(i+1))$ lie with respect to the function $\log x$?

Problem 3: (The Geometric Distribution)

A random variable X is said to have a *Geometric*(p) distribution if for any integer $k \geq 1$, we have $\mathbb{P}[X = k] = p(1-p)^{k-1}$.

Part (a)

Suppose we repeatedly toss a coin which gives HEADS with probability p. Argue that the number of tosses until we see the first HEADS is distributed as *Geometric*(p).

Part (b)

(Memoryless property) Using the definition of conditional probability, prove that for any integers $i, k \geq 1$, the random variable X obeys:

$$\mathbf{P}[X = k + i | X > k] = \mathbf{P}[X = i]$$

Also convince yourself that this follows immediately from the characterization of the Geometric r.v. in Part (a).

Part (c)

Show that: $(i)\mathbf{E}[X] = \frac{1}{p}$, and $(ii)Var[X] = \frac{1-p}{p^2}$
Hint: Note that by the memoryless property, a Geometric(p) random variable X is 1 with probability p, and $1+Y$ with probability $(1-p)$, where Y also has a Geometric(p) distribution. Now try writing the expectation and variance recursively.

Problem 4: (Upper Bounds on Collision Probabilities)

Let $X_{m,n}$ denote the number of collisions when m balls are dropped u.a.r. into n bins. In class, we showed that then the expected number of collisions is $\binom{m}{2}/n$. We now upper bound the probability that no collision occurs.

Assume that $n > m$ (clearly this is required for no collisions!). First, using the law of total probability, argue that:

$$\mathbf{P}[\text{No collisions when } m \text{ balls dropped u.a.r. in } n \text{ bins}] = \prod_{i=1}^{m-1}\left(1 - \frac{i}{n}\right)$$

Next, using the inequality $e^{-x} \geq (1-x)$, simplify the above to show:

$$\mathbf{P}[\text{No collisions when } m \text{ balls dropped u.a.r. in } n \text{ bins}] \leq e^{-\mathbf{E}[X_{m,n}]}$$

Problem 5: (Posterior Confidence in Verifying Matrix Multiplication)

In class, we saw Freivald's algorithm for checking matrix multiplication, which, given matrices A, B and C, returned the following:

- If $AB = C$, then the algorithm always returned TRUE
- If $AB \neq C$, then the algorithm returned TRUE with probability at most $1/2$

Part (a)

Given any $\epsilon > 0$, how many times do we need to run Freivald's algorithm to be sure that $\{AB = C\}$ with probability greater than $1 - \epsilon$?

Part (b)

Suppose we started with the belief that the events $\{AB = C\}$ and $\{AB \neq C\}$ were equally likely (i.e., $\mathbf{P}[AB = C] = \mathbf{P}[AB \neq C] = 1/2$). Moreover, suppose k independent runs of Freivald's algorithm all returned TRUE. Then what is our new (or *posterior*) belief that $\{AB = C\}$?