CS 344 Homework Zero

Matthew McCaughan

Due January 29th, 2025

Problem 1: Academic Integrity

I affirm that I have not given or received any unauthorized help on this assignment, and that this work is my own.
- Matthew McCaughan

Problem 2: Asymptotics

(a)

The functions below are ordered according to increasing asymptotic growth:
7. $loglog\sqrt{n}$
6. $2^{(\sqrt{log(n)})}$
5. $log(n^5)$
4. $1.1^{1.2n}$
3. The number of proper subsets of [n] of size at least n/3
2. $n!$
1. $n^n \times \sqrt{log(n)}$

I started with log(log(n) as the smallest as the logarithm of a logarithm usually has the smallest growth rate. From there, we compare the limits/logarithms between each function to determine the relationship between their growth

Step 1: Compare $\log(\log(\sqrt{n}))$ **and** $2^{\sqrt{\log(n)}}$

$$\lim_{n\to\infty} \frac{\log(\log(\sqrt{n}))}{2^{\sqrt{\log(n)}}} = 0$$

Since exponential growth dominates logarithmic growth, $\log(\log(\sqrt{n}))$ grows asymptotically slower than $2^{\sqrt{\log(n)}}$.

—

Step 2: Compare $2^{\sqrt{\log(n)}}$ **and** $\log(n^5)$

$$\lim_{n\to\infty} \frac{2^{\sqrt{\log(n)}}}{\log(n^5)} = \infty$$

since exponential growth dominates logarithmic growth, $2^{\sqrt{\log(n)}}$ grows asymptotically faster than $\log\left(n^5\right)$.

—

Step 3: Compare $\log(n^5)$ **and** $1.11^{1.2n}$

$$\lim_{n\to\infty} \frac{\log\left(n^5\right)}{1.11^{1.2n}} = 0$$

since exponential growth dominates logarithmic growth, $\log\left(n^5\right)$ grows asymptotically slower than $1.11^{1.2n}$.
—

Step 4: Compare $1.11^{1.2n}$ and the number of proper subsets of size $\geq n/3$

The number of subsets from [n] with a size of at least n/3 is the combinations that can be made from n $n/3$ is:

$$\sum_{k=\lceil n/3 \rceil}^{n} \binom{n}{k}.$$

Or the total combinations that can be made from n items subtracted by the combinations made from n/3 or less.

$$n! - \frac{n!}{\left(\frac{n}{3}!\left(n - \frac{n}{3}\right)!\right)}$$

Compute the limit:

$$\lim_{n\to\infty} \frac{1.11^{1.2n}}{n! - \frac{n!}{\left(\frac{n}{3}!\left(n-\frac{n}{3}\right)!\right)}} = 0$$

since $n! - \frac{n!}{\left(\frac{n}{3}!\left(n-\frac{n}{3}\right)!\right)}$ grows much faster than $1.11^{1.2n}$,$1.11^{1.2n}$ grows asymptotically slower than the number of subsets.
—

Step 5: Compare $n! - \frac{n!}{\left(\frac{n}{3}!\left(n-\frac{n}{3}\right)!\right)}$ and $n!$

Compute the limit:

$$\lim_{n\to\infty} \frac{n! - \frac{n!}{\left(\frac{n}{3}!\left(n-\frac{n}{3}\right)!\right)}}{n!} \sim 1$$

$n! - \frac{n!}{\left(\frac{n}{3}!\left(n-\frac{n}{3}\right)!\right)}$ grows slower than $n!$, but are in the same class and grow similarly.
—

Step 6: Compare $n!$ and $n^{n\cdot\log(n)^p}$

Compute the limit:

$$\lim_{n\to\infty} \frac{n!}{n^{n\cdot\log(n)^p}} = 0$$

$n!$ grows asymptotically slower than $n^{n\cdot\log(n)^p}$.
(b)

(B)

For the following functions, f(n) and g(n):

n^2 and $(n+2)^2$

Because they grow at the same rate within the same n^2 family, f(n) = $\Theta(g(n))$.

e^{n^2} and $e^{(n+2)^2}$

the exponential $(n+2)^2$ allows g(n) to grow faster than f(n), so therefore, f(n) = $o(g(n))$

$(n^2)^{1/log(n)}$ **and** $(log(n)^{1/loglog(n)}$

both functions are bound by an inverse logarithmic exponent, which will limit and bind the growth of both functions to the same growth family, f(n) = $\Theta(g(n))$.

$n!$ **and** n^n

The function g(n) grows faster than f(n), so f(n) = o(g(n)

$log(5n)$ **and** $log(10n)$

Functions f(n) and g(n) are in the same growth family, but can dominate each other based on the coefficient c provided, so while they are in the same growth family, they can dominate each other. f(n) = $\Theta(g(n)), f(n) = O(g(n)), f(n) = \Omega(g(n))$

Problem 3: Double Recurrence

To Prove by induction that T(n) = O(n) where T is defined as T([7n/10]) + T([n/5]) + 100 * n, and T(0) = T(1) = 1, we must show that T(n) ¡= O(n). If we are trying to assert T(n) $\leq c * n$, then we may also assert that T(n+1) ¡= c(n+1).
T(n+1) $\leq T(7/10 * n + 1) + T(1/5 * n + 1) + 100 * n$
$T(n + 1) \leq c * (7/10 * n) + c * (1/5 * n) + 100 * n$
$T(n + 1) \leq c * (9/10 * n) + 100 * n$
$T(n + 1) \leq n * (9/10 * c) + 100$
We can rearrange this to solve for a value of C, knowing T(n+1) must be at most c*n:
$9/10 * c + 100 \leq c$
$c \geq 1000$

Based on this inductive step, any value c $\geq$ 1000 will suffice for T(n) $\leq$ c*n for when n is some value that is greater or equal to 2 (greater than base cases).

Problem 4: Group Testing

a

A strategy for finding the sick person within log_2n tests is to take your samples and split them in half, with two mixes of samples of people from set S. Whichever mix of samples tests positive between the two, use those new mixes of samples and perform the splitting and testing process, until you have discovered who the sick person is, either by testing them and they are positive, or deducing them from a negative test between one negative and one positive test.

b

If we only have one test kit, then we have to perform the $\Omega(n)$ tests to find the sick person with 100 percent confidence. Assume that with a set size of 1, the first and only individual is the sick person. For any set size greater than 1, you have to test each of the n people individually, which would require a number of tests that scale linearly with the number of people to test in set S. If the sick person is the last one tested, then it is required that all n tests would be necessary to test with 100 percent accuracy.

c

For two test kits, an algorithm for finding the sick person using $O(\sqrt{n})$ tests involves a similar process of splitting two groups into a maximum size of $\sqrt{n}$ individuals. And testing 1 person from each group one at a time with 1 of the two test kits. The sick person will be identified within one of the groups or deduced from all people in the groups testing negative. In either case, continue to test within this positive group to identify the sick person.

d

To find a sick person with k test kits, we are constructing a generalized form for the solutions we found for parts c and d. For k test kits, we should spread our test kits, k, over groups of $\sqrt[k]{n}$ individuals. Much like part c, one person at a time is tested in the group until the sick person is discovered within the group and is subsequently retested for identification. This method performs $\sqrt[k]{n}$ tests on the set S.

Problem 5: Probability

Over all possible arrangements of R red apples and G green apples in the basket, we are looking for the probability the basket is empty when we stop (when we stop when there are no green apples left). We are looking for all the ways that we pick a green apple and it is the last. So, we are selecting G apples from G + R total apples. Based on combinatorics, we are looking for the binomial coefficient, $\binom{G+R}{G}$, is defined by the expression:

$$\binom{G+R}{G} = \frac{(G+R)!}{G!((G+R)-G)!}$$

To express this as a probability over the situation, we are looking for the probability of these events over all possible choices, so the binomial coefficient will be inverted to equivalently show it as a probability.

$$P(BasketEmptyWhenStopped) = \frac{1}{\frac{(G+R)!}{G!((G+R)-G)!}}$$

Or, more simplified:

$$Probability = \frac{1}{\binom{G+R}{G}}$$