COMS 5110: Assignment 1
Due: Sept. 18^{th}, 11:59pm
Total Points: 50

Late submission policy. Any assignment submission that is late by not more than two business days from the deadline will be accepted with a 20% penalty for each business day. That is, if a homework is due on Friday at 11:59 PM, then a Monday submission gets a 20% penalty and a Tuesday submission gets another 20% penalty. After Tuesday no late submissions are accepted.

Submission format. Homework solutions will have to be typed in. You can use word, LaTeX, or any other typesetting tool to type your solution. Your submission file should be in pdf format. Do **NOT** submit a photocopy of handwritten homework, except for diagrams that can be drawn and scanned. We reserve the right **NOT** to grade homework that does not follow the formatting requirements. Name your submission file: `<Your-net-id>-5110-hw1.pdf`. For instance, if your netid is `asterix`, then your submission file will be named `asterix-5110-hw1.pdf`. Each student must hand in their own assignment. If you discussed homework or solutions with others, a list of collaborators must be included with each submission. Each of the collaborators has to write the solutions in their own words (copies are not allowed). If you received help with the assignment from AI tools, the names and websites of the AI tools must be included with your submission.

General Requirements

- When proofs are required, do your best to make them both clear and rigorous.
- Even when proofs are not required, you should justify your answers and explain your work.

Some Useful (in)equalities

- $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$
- $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$
- $2^{\log_2 n} = n$, $a^{\log_b n} = n^{\log_b a}$, $n^{n/2} \leq n! \leq n^n$, $\log x^a = a \log x$
- $\log(a \times b) = \log a + \log b$, $\log(a/b) = \log a - \log b$
- $a + ar + ar^2 + ... + ar^{n-1} = \frac{a(r^n - 1)}{r-1}$
- $1 + \frac{1}{2} + \frac{1}{2^2} + ... + \frac{1}{2^n} = 2(1 - \frac{1}{2^{n+1}})$
- $1 + 2 + 4 + ... + 2^n = 2^{n+1} - 1$

Problem 1

(10 pints) **Model of Computation** Consider the following algorithm (written in pseudocode):

```
1 Alg(X, Y)
    Input: Decimal numbers X with m digits (x_1 x_2 ... x_m) and Y with n digits
           (y_1 y_2 ... y_n)
2   k = min(m, n)
3   for i = 1 to k do
4       z_i = max(x_{m-k+i}, y_{n-k+i})
5   report Z with k digits (z_1 z_2 ... z_k)
```

In the *Random Access Machine (RAM)* model, each word has $c \log_2 \max(m, n)$ bits for some constant $c \geq 1$ so that each word can hold each of the values of m and n. Let $v(m, n)$ be the result saved in any variable at the end of the above algorithm. The function $v(m, n)$ fits a word in the RAM model if the number of bits used to save $v(m, n)$ is in $O(\log_2 \max(m, n))$; $v(m, n)$ does not fit a word in the RAM model if the number of bits used to save $v(m, n)$ is not in $O(\log_2 \max(m, n))$.

1. (4 pts) Determine whether the value stored in each variable in the above algorithm fits a word in the RAM model. The algorithm uses the following types of variables: m, n, k, i, $m - k + i$, $n - k + i$; x_i, y_i, z_i. You need to justify your answer by following the above definition of whether the result in any variable fits a word in the RAM model. Note that the numbers X, Y, and Z are represented by storing their digits in variables.

2. (2 pts) What is the input size for the above algorithm?

3. (4 pts) Obtain an expression that is a tight upper bound for the running time of the algorithm and derive a tight big O notation for the expression. Assume that a positive constant d is an upper bound for the time to run each of lines 2-4 in the above algorithm once. Then line 5 takes at most $d \times k + d$ time, where reporting each digit of Z takes time bounded by d and confirming that the last digit of Z has been reported also takes time bounded by d.

Problem 2

(10 pints) **Mathematical Induction** Show by induction that $32 \log_2 n \leq n$ for all $n \geq n_0$ for some $n_0 > 0$.

Problem 3

(10 pints) **Sizes of left and right subarrays in Merge Sort** Let $A[p..r]$ be a subarray of $A[1..n]$ with $1 \leq p < r \leq n$. In the recursive algorithm MERGE-SORT(A, p, r), the subarray $A[p..r]$ is partitioned into left and right subarrays $A[p..q]$ and $A[q + 1..r]$ with

$q = \lfloor (p+r)/2 \rfloor$. Let $m = r - p + 1$ be the size of the subarray $A[p..r]$. Show that the size of $A[p..q]$ is $q - p + 1 = \lceil \frac{m}{2} \rceil$, and that the size of $A[q+1..r]$ is $r - (q+1) + 1 = \lfloor \frac{m}{2} \rfloor$. You just need to prove it for the case where $p + r$ is odd, that is, $p + r = 2y + 1$ for some integer y.

Problem 4

(20 points) **Solving the recurrence by induction** Use mathematical induction to show that $T(n) \leq cn - b(\log_2 n)^2$ for all $n \geq n_0$ for some constants $n_0 > 0$, $c > 0$, and $b > 0$, where $T(n)$ is defined in the following recurrence:

$T(1) = d$,

$T(n) = T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + d(\log_2 n)^2$ if $n > 1$, where d is a positive constant.