ES215: Semester I [2024-2025]
Assignment 1

Due Date: 16-08-2024 (11:59 pm) 150 points

Note: Make appropriate assumptions when required. But, state all the assumptions made and ensure that they do not contradict what has been given in the problem statement. Please type up the solutions and submit them in PDF format. The source files/github link have to be submitted. Please upload all the assignments in Google Classroom. Late assignments will not be graded. Please follow the honor code. Any violations will have severe repercussions.

---

1. Implement a program(s) to list the first 50 fibonacci numbers preferably in C/C++ in the following manner: (***Total: 50 points***)
    a. Using recursion (***10 points***)
    b. Using loop (***10 points***)
    c. Using recursion and memoization (***10 points***)
    d. Using loop and memoization (***10 points***)

    Find the speedup of all the programs on your machine by keeping program (1) as the baseline. (***10 points***).
    Tips: Measure the time taken by the program on the CPU using timespec.

2. Write a simple Matrix Multiplication program for a given NxN matrix in any two of your preferred Languages from the following listed buckets, where N is iterated through the set of values 64, 128, 256, 512 and 1024. N can either be hardcoded or specified as input. Consider two cases (a) Elements of matrix are of data type **Integer** and (b) **Double** In each case, (*i.e.* Bucket 1 for (a) and (b) + Bucket 2 for (a) and(b)) (***Total: 100 points***)
    Bucket1: C, C++, Go
    Bucket2: Python, Java.

a. Report the output of the '**time**' describing the system and CPU times. (***25 points***)
b. Using the '***language hooks***' evaluate the execution time for the meat portions of the program and how much proportion is it w.r.t. total program execution time. (***25 points***)
c. Plot the (a) and (b) execution times for each of the iterations. And compare the performance (System and Program execution times) of the program for given value of N for the languages in both the buckets. –Illustrate your observations. (***50 points***)

Reference:
[1] Timespec
[2] Improving Efficiency of Recursive Functions