CS302 - Design and Analysis of Algorithm
Fall 2024
Project Description
Total Marks: 100. You need to work as a Group of 3

In this project, you are required to

1) **[40 Points]** Implement the following divide and conquer algorithms using any language of your choice with user defined input size of at least 10. You do not need to show any GUI.
   - **a)** [Source: Introduction to the Design and Analysis of Algorithms, Anany Levitin, 3rd Edition, Exercise 5.1 (2), Page 174]
     1. Write pseudocode for a divide-and-conquer algorithm for finding values of both the largest and smallest elements in an array of n numbers.
     2. Set up and solve (for $n = 2^k$) a recurrence relation for the number of key comparisons made by your algorithm.
     3. How does this algorithm compare with the brute-force algorithm for this problem?
   - **b)** [Source: Introduction to the Design and Analysis of Algorithms, Anany Levitin, 3rd Edition, Exercise 5.1 (3), Page 174]
     1. Write pseudocode for a divide-and-conquer algorithm for the exponentiation problem of computing a^n where n is a positive integer.
     2. Set up and solve a recurrence relation for the number of multiplications made by this algorithm.
     3. How does this algorithm compare with the brute-force algorithm for this problem?
   - **c)** [Source: Introduction to the Design and Analysis of Algorithms, Anany Levitin, 3rd Edition, Exercise 5.1 (9), Page 175]
     1. Let $A[0..n-1]$ be an array of *n* real numbers. A pair $(A[i], A[j])$ is said to be an ***inversion*** if these numbers are out of order, i.e., $i < j$ but $A[i] > A[j]$. Design an $O(n \log n)$ algorithm for counting the number of inversions.
   - **d)** [Source: Introduction to the Design and Analysis of Algorithms, Anany Levitin, 3rd Edition, Section 5.2, Page 176]
     1. What is the worst-case and best-case complexity of the quicksort algorithm?
   - **e)** [Source: Introduction to the Design and Analysis of Algorithms, Anany Levitin, 3rd Edition, Exercise 5.5 (1), Page 197]
     1. For the one-dimensional version of the closest-pair problem, i.e., for the problem of finding two closest numbers among a given set of *n* real numbers, design an algorithm that is directly based on the divide-and-conquer technique and determine its efficiency class.

2. Is it a good algorithm for this problem?

f) [Source: Algorithm Design, Jon Kleinberg, 1st Edition, Page 242]

1. Solved Exercise 1

g) [Source: Algorithm Design, Jon Kleinberg, 1st Edition, Page 244]

1. Solved Exercise 2

h) [Source: Algorithm Design, Jon Kleinberg, 1st Edition, Exercises (1-7), Page 246-249]

1. Any three exercises

2) [10 Points] Implement the following divide and conquer algorithms using any language of your choice. Generate 10 random sample inputs for at least 2 of the above problems with least input size > 100 and save it on text files. So in summary you have 10 x 2 different files of varying complexities

(i) Finding the closed pair of points
(ii) Integer Multiplication

3) [5 Points] Apply these algorithms on the input datasets

4) [30 Points] You need to show a very nice user interface where user can select any file input file (assumed valid), and then show the working of both algorithms using nice user interface

5) [15 Points] You are required to submit strictly a 3-page report (maximum 18% similarity) with the following sections

a. Abstract
b. Introduction
c. Your proposed system (make a clear diagram of your system here along with discussion)
d. Experimental Setup (The details of each input dataset including values of random numbers generated)
e. Results and Discussion (Here, show results from benchmarks i.e. correct solution obtained using various files)
f. Conclusion
g. References

Submission Instruction:

The project should be submitted in three phases. You have to submit each deliverable within the due date. In case of non-compliance the maximum obtainable marks would be reduced by 25% each day after the due date.

Submission 1:

- **Deliverables:** Project part 1.
- **Due Date:** October 22, 2024

Submission 2:

- **Deliverables:** Project part 2, 3, and 4.
- **Due Date:** November 10, 2024

Submission 3:

- **Deliverables:** Project part 5.
- **Due Date:** November 20, 2024