Lab Sheet 1 **21AIE212 Design and Analysis of Algorithms**

Academic Year: March-July 2023 S4 B.Tech CSE-AI, 2021 Batch

1 Prelab

We would be using two different platforms for the lab.
1. Google colab
2. Leet code.
The first prelab assignment is to get familiarize with google colab. We use Python language in colab.

Colab Assignments.

1. Create a notebook in colab.
2. Add a code cell and type in a Python statement to print Hello world Message. Play the cell and observe the output.
3. Add another code cell and write Python code for reading a number and print its square.
4. Add a text cell and write the algorithm for parenthesis matching using stack.
5. Add a code cell and implement the parenthesis matching algorithm. Test your code.
6. Learn how to plot charts in colab.
   Refer https://colab.research.google.com/notebooks/charts.ipynb

For the questions in the remaining part of this lab sheet, copy paste the question to a text cell in the notebook and write your code in a code cell immediately below it. If the question is to explain your observation, write the answer in a text cell immediately below the question.

How to submit the solution?

Simple!, Print the notebook (in PDF format) and upload..

2 Basic Algorithm Analysis

For each of the following questions, implement each algorithm as a separate Python function. To analyse and plot the running time do the following.

1. Include a counter variable in each function to count the number of iterations/recursions done for a given input.
2. Find the bit length (input size) of the input number using the Python method $(n).bit\_length()$ that returns the bit length of the number n.

Plot the bit length of the input number(s) in x-axis against corresponding counter value(s) and execution time in the y-axis and draw the chart.

Questions.

1. Write a program to compute the n^{th} Fibonacci number using
   a. Recursive Fibonacci Algorithm
   b. Iterative algorithm.
   Count the number of iterations/recursions for each n. Plot the operation count against the bit length of n.

2. Write a program to find the Greatest Common Divisor (GCD) of two numbers using
   a. Euclid's algorithm
   b. Binary GCD algorithm
   Count the number of iterations/recursions for each input. Plot the operation count against the bit length of largest among the two input numbers.

3. For each of the three gcd algorithms, identify the best case and worst case inputs and find the running times in each case. Write your answers in a text cell.

4. Write a program to determine if a positive integer, N, is prime. Write the answers for the following questions in a text cell.
   a. In terms of N, what is the worst-case running time of your program?
   b. Let B equal the number of bits in the binary representation of N. What is the value of B? (as a function of N)
   c. In terms of B, what is the worst-case running time of your program?

5. Write a program to compute all primes up to and including n. (The Sieve of Eratosthenes method).