# Department of Computing

# School of Electrical Engineering and Computer Science

**CS-250: Data Structure and Algorithms**

**Class: BEEE 13 (Grp1+Grp2)**

# Lab 10: Computational time of algorithms

**Date: 15 April, 2024**

**Time: 10 am - 1 pm**

**Lab Engineer: Anum Asif**

# 

# Lab 10: Computational time of algorithms

**Introduction**

This lab is based on the analysis of the computational time acquired by an algorithm.

**Objectives**

Objective of this lab is to practice of finding the computational time of an algorithm.

**Tools/Software Requirement**

Visual Studio 2012 or gcc or g++

**Helping Material**

Lecture slides, text book

**Lab Tasks**

Consider two different algorithms; listed in lab tasks. You will implement two different approaches for solving the same problem and will figure out the time difference between their successful executions.

**Task 1**

* Calculate the Fibonacci series of a number “n” given by user both *recursively and iteratively*. Figure out the difference in computational time by using both the approaches.

**Recursive Iterative**

E.g. for **N=10000** 20ms 2ms

* Run some timing experiments with your program while trying different values of *n*. Make sure to time only the computation and not the user entering input etc.
* Draw a graph in Excel showing two functions one for recursive function and other for iterative function.

**Task 2**

* Calculate the factorial of a number “n” given by user both *recursively and iteratively*. Figure out the difference in computational time by using both the approaches.
* Draw a graph in Excel showing two functions one for recursive function and other for iterative function.

(Hint: input at least 5 different values of “n” and make their entry in excel sheet. “n” can be of any data type. Try to use larger values of “n” for getting the useful data.)

How to calculate time in seconds in c++:

#include<time.h>

clock\_t startTime = clock();

//YOUR CODE HERE E.G. RECURSIVE CALLS

cout << double(clock()-startTime)/CLOCKS\_PER\_SEC;

**Important Note:** Practice your knowledge of OOP with C++ when creating a solution. Remember to comment your code properly. Inappropriate or no comment may result in deduction of marks.

**Solution:**

|  |
| --- |
| Solution |
| Task 1 Graphs (recursive, iterative):  Task 1 Code (recursive, iterative):  Task 2 Graphs (recursive, iterative):  Task 2 Code (recursive, iterative): |

### Deliverables

Compile a single word document by filling in the solution part and submit this Word file on LMS. Insert the solution/answer in this document. You must show the implementation of the tasks in the designing tool, along with your complete Word document to get your work graded. You must also submit this Word document on the LMS.