



Algorithms Analysis and Design (230213150)

Fall 2023/2024

Practice Assignment #1 (Factorial Calculation)

1. Due date: **Saturday, 11/11/2023** (*No Assignments will be accepted after the mentioned date, with no exceptions.*)
2. The assignment is an **individual** effort. Copying the assignment will be treated as a cheating attempt, which may lead to **FAILING** the course.
3. Try your best to be precise and organized in presenting/analyzing results.
4. The report must be submitted in **pdf** format. You need to include all your **code** as well.

This assignment focuses on factorial calculation, a fundamental mathematical operation that can be approached using various methods. We will explore two such methods: iterative and recursive. By implementing and comparing both approaches, you will gain insights into their efficiency and limitations. This exercise provides an opportunity to measure execution times for different input values, observe how large values affect the stack, and learn how to handle large results by selecting appropriate data types. Through this assignment, you'll strengthen your understanding of programming and algorithmic practices.

Instructions:

- Implement an **iterative** function to calculate the factorial of a given integer n .
- Implement a **recursive** function to calculate the factorial of n .
- Test both functions with various values of n , starting from small values (e.g., 5, 10, 20) and gradually increasing to larger values.
- Measure and record the execution time for each test case. You can use built-in timing libraries or functions available in your chosen programming language.
- Choose a sufficiently large value of n (e.g., $n = 10000$) for both functions.
- Attempt to calculate the factorial for this large value of n to observe the stack overflow issue. Record the outcome.
- **Note:** Use appropriate data types to handle large results like (**long long**)

Report

Write a technical report about the experiments you have made (maximum of 1 page). The report should contain the following:

1. Show execution time results in a presentable way (using **tables** and **charts**).
2. Provide stack overflow observations.
3. You should **discuss** what you have found.
4. Conclude the work as a whole.

All the best