INFORMATION
TECHNOLOGY
UNIVERSITY

Department of Computer and Software Engineering – ITU

SE200T: Data Structures & Algorithms

Course Instructor: Usama Bin Shakeel	**Dated: 11th Sep 2024**
Teaching Assistant: Zainab, Sadia & Ryan	**Semester: Fall 2024**
Session: 2023-2027	**Batch: BSSE2023B**

Assignment 4. Analyzing Time and Space Complexities

Name	**Roll number**	**Obtained Marks/35**

Checked on: ______________________________

Signature: ____________________________

Submission:

• Email instructor or TA if there are any questions. You cannot look at others' solutions or use others' solutions, however, you can discuss it with each other. Plagiarism will be dealt with according to the course policy.
• Submission after due time will not be accepted.

In this assignment you have to do following tasks:

Task 1: Ensure that you have installed all three softwares in your personal computer (Github, Cygwin & CLion). Now, accept the assignment posted in the classroom (e.g Google, LMS etc) and after accepting, clone the repository to your computer. Make sure you have logged into the github app with your account.

Task 2: Review Your Previous Implementations
Task 3: Calculate Time and Space complexities and create a detailed report.
Task 4: Keep your document and code in the respective git cloned folder.
Task 5: Commit and Push the changes through the Github App

Problem Statement:

Calculate the time and space complexities for the sorting algorithms you implemented in previous labs and assignments. This analysis should be based on your specific implementations.

- *Bubble Sort*
- *Quick Sort*
- *Selection Sort*
- *Merge Sort*
- *Radix Sort*
- *Count Sort*
- *Bogo Sort*
- *Bozo Sort*

Instructions:

Create a Word document that includes the following.

Summary Table: *At the beginning of the document, include a summary table with the time and space complexities of each sorting algorithm. This table should be clear and easy to read.*

- ***Detailed Analysis:*** *For each algorithm, provide a detailed analysis of the time and space complexities based on your specific implementations. Include:*
    - ***Time Complexity:*** *Discuss the actual performance of the algorithm in terms of time.*
    - ***Space Complexity:*** *Discuss the actual memory usage of the algorithm.*
- ***Implementation Details:*** *Include any relevant details from your code that help explain the complexities you calculated.*

Important Notice for Your Assignment

Please read the following instructions carefully:

1. *Open your code editor and the theoretical complexity documentation side by side. This will allow you to reference your code while calculating and documenting the complexities.*
2. *Don't try to prove theoretical time and space complexities of sorting algorithms. Your analysis must be based on the actual performance and memory usage of your specific code.*
3. *You must have to submit the code files in which you implemented the sorting algorithms. This is essential for verifying your calculated complexities.*

Any deviation from these rules will result in your assignment receiving a grade of zero.

Assessment Rubric for Assignment

Performance metric	CLO	Able to complete the task over 80% (4-5)	Able to complete the task 50-80% (2-3)	Able to complete the task below 50% (0-1)	Marks
1. Realization of experiment	3	Executes without errors excellent user prompts, good use of symbols, spacing in output. The testing has been completed.	Executes without errors, user prompts are understandable, minimum use of symbols or spacing in output. Some testing has been completed.	Does not execute due to syntax errors, runtime errors, user prompts are misleading or non-existent. No testing has been completed.	
2. Conducting experiment	2	Able to make changes and answer all questions.	Partially able to make changes and few incorrect answers.	Unable to make changes and answer all questions.	
3. Computer use	4	Document submission timely.	Document submission late.	Document submission not done.	
4. Teamwork	4	Actively engages and cooperates with other group member(s) in an effective manner.	Cooperates with other group member(s) in a reasonable manner but conduct can be improved.	Distracts or discourages other group members from conducting the experiment	
5. Laboratory safety and disciplinary rules	2	Code comments are added and do help the reader to understand the code.	Code comments are added and do not help the reader to understand the code.	Code comments are not added.	
6. Data collection	2	Excellent use of white space, creatively organized work, excellent use of variables and constants, correct identifiers for constants, No line-wrap.	Includes name, and assignment, white space makes the program fairly easy to read. Title, organized work, good use of variables.	Poor use of white space (indentation, blank lines) making code hard to read, disorganized and messy.	
7. Data analysis	3	Solution is efficient, easy to understand, and maintain.	A logical solution that is easy to follow but it is not the most efficient.	A difficult and inefficient solution.	
Total (out of 35):					