

## CSCI 4520 Programming Project

Topic: Sorting, Searching, and Efficiency Analysis

### D2L-→grades-→project for the required search and sorting algorithm

#### Project Introduction

In this project, you will apply theoretical concepts learned in class to a computer simulation that verifies algorithmic results. The project focuses on implementing and analyzing search and sorting algorithms, as well as evaluating their computational efficiency.

#### Project Description

You are required to:

1. Generate 100,000 positive numbers in the range (0, 125).
2. Search for the first occurrence of the number 53 using a sequential search algorithm.
3. Count the total number of occurrences of 53 in the array.
4. After completing steps 1–3, sort the array using one of the required sorting algorithms.
5. Repeat step 3 using a required search algorithm on the sorted array.
6. Compute and compare the time cost between step 3 only and steps 4 and 5 combined.

Run your program three times and record the time for each operation:

- Time to generate 100,000 numbers (Step 1)
- Time to search for the element “53” (Step 2)
- Time to count the occurrences of “53” (Step 3)
- Time to sort the array (Step 4)
- Time to repeat Step 3 on the sorted array (Step 5)

Finally, analyze and compare the performance differences.

#### Project Learning Objectives

Students will:

- Understand the concepts and techniques of computational analysis.
- Develop code while demonstrating an understanding of the software life cycle.
- Become familiar with algorithm efficiency analysis.

#### Computer Programming Learning Objectives

Students will strengthen their knowledge of:

- Arrays (Bounded)
- Algorithm design
- Data structures
- Interface design
- Graphical User Interface (GUI) design

## Overall Learning Outcomes

Upon successful completion of this project, students will:

- Gain a solid understanding of algorithm efficiency analysis.
- Improve their ability to write well-documented, object-oriented programs.

## Program Structure

The final report should include:

1. Algorithm Design – Describe the algorithm in pseudocode.
2. Implementation – Provide source code that implements the algorithms.
3. Efficiency Analysis – Discuss the time and space complexity of your algorithm.

## Deliverables

- Cover Page: Project title, course name, student's full name, and submission date.
- Report Sections:
  - Introduction
  - Theoretical description of the algorithms and efficiency analysis
  - Pseudocode design
  - Discussion of results and applications
  - Conclusion
- Code File: Submit your source code separately (not within the report). Ensure your code includes sufficient comments for readability.

## Grading Rubric

Criterion	Weight
Algorithm design and implementation	15%
Programming code (interfaces and implementation)	20%
Application correctness	10%
Report documentation (clarity, structure, completeness)	65%
Bonus: GUI design	+10%

## Due Date

Last day of class (TBD)

Both the report and code must be submitted on the same day.