Wilson Quilli

Professor Elangovan

CMPSC 412: Data Structures Lab

September 23rd, 2025

Lab 4 - Searching and Sorting

In this lab, I worked on two exercises, the first one consisting of using Binary Search Algorithm for a number-guessing game and the second exercise consisting of experimenting with the four sorting algorithms, selection sort, insertion sort, bubble sort and merge sort on data from a text file.

For the first exercise, the program used a range of numbers between 1 and 5000. The algorithm repeatedly guessed the middle number and asked the user for feedback until the number was found. Finally, I calculated the memory size of the variables used in the binary search with sys.getsizeof(). The time complexity of binary search is O(log n), and the space complexity is O(1) since only a few variables are stored.

For the second exercise, I extracted the data from the text file of students using open(file_path, 'r') as file. Then, I implemented each sort to experiment with (selection, insertion, bubble, and merge). While each of these sorting algorithms sorted the data, I also added a timer to calculate the CPU time it took for each sort to perform. Finally, I calculated the memory size used by the data during each algorithm. Selection sort, insertion sort, and bubble sort all have a time complexity of $O(n^2)$ and a space complexity of O(1). Merge sort has a time complexity of O(n log n) and a space complexity of O(n) because it requires additional lists during recursion.

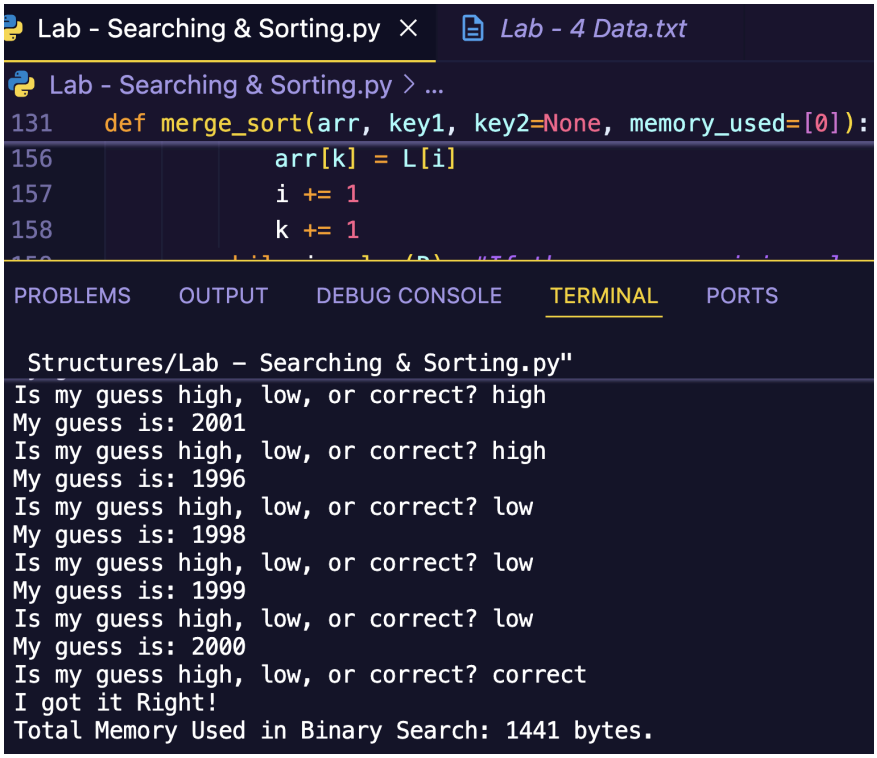
Table 1: CPU Time:

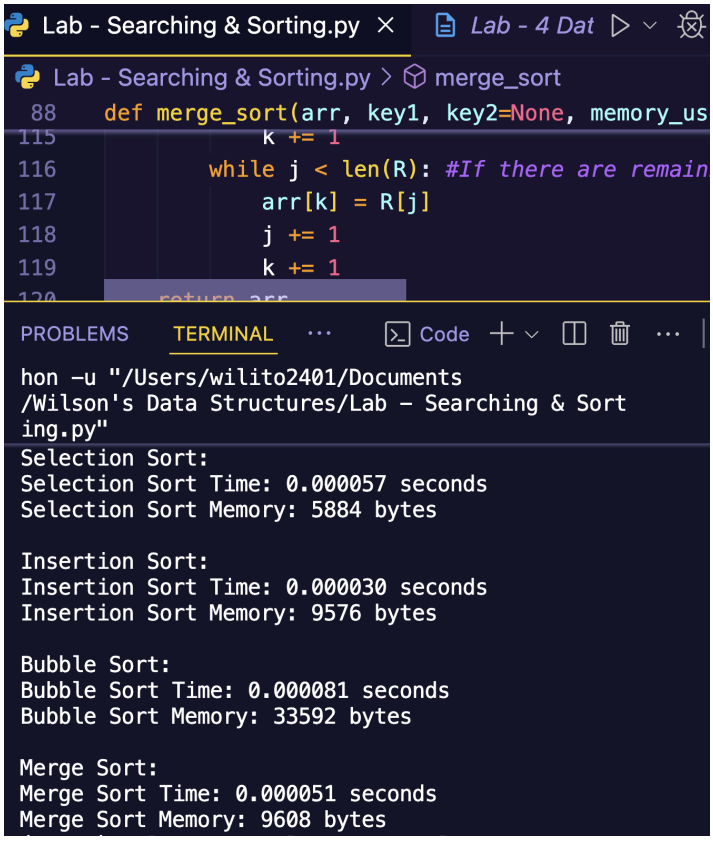
Variables	**Selection Sort**	**Insertion Sort**	**Bubble Sort**	**Merge Sort**
ID	0.186248 Sec.	0.077505 Sec.	0.379274 Sec.	0.003203 Sec.
Name	0.189550 Sec.	0.078564 Sec.	0.377714 Sec.	0.003709 Sec.

Table 2: Memory Size:

Variables	**Selection Sort**	**Insertion Sort**	**Bubble Sort**	**Merge Sort**
ID	373048 Bytes	618912 Bytes	144330808 Bytes	419784 Bytes
Name	373048 Bytes	618912 Bytes	144943672 Bytes	419784 Bytes

Ultimately, after completing both exercises, I learned more about the sorting and searching algorithms and how they function. For example, I learned that insertion sort was the fastest in my experiment, while bubble sort was the slowest in terms of CPU time. I also observed that merge sort uses more memory than the others because it creates sublists during recursion. Overall, both searching and sorting algorithms can perform complex tasks in a short amount of time, but depending on the input size and method, some sorting algorithms are better than others.


Lab - Searching & Sorting.py
Lab - 4 Data.txt
Lab - Searching & Sorting.py > ...
131 def merge_sort(arr, key1, key2=None, memory_used=[0]):
156 arr[k] = L[i]
157 i += 1
158 k += 1
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Structures/Lab - Searching & Sorting.py"
Is my guess high, low, or correct? high
My guess is: 2001
Is my guess high, low, or correct? high
My guess is: 1996
Is my guess high, low, or correct? low
My guess is: 1998
Is my guess high, low, or correct? low
My guess is: 1999
Is my guess high, low, or correct? low
My guess is: 2000
Is my guess high, low, or correct? correct
I got it Right!
Total Memory Used in Binary Search: 1441 bytes.


Lab - Searching & Sorting.py
Lab - 4 Dat
Lab - Searching & Sorting.py > merge_sort
88 def merge_sort(arr, key1, key2=None, memory_us
115 k += 1
116 while j < len(R): #If there are remain
117 arr[k] = R[j]
118 j += 1
119 k += 1
PROBLEMS TERMINAL Code
hon -u "/Users/wilito2401/Documents
/Wilson's Data Structures/Lab - Searching & Sort
ing.py"
Selection Sort:
Selection Sort Time: 0.000057 seconds
Selection Sort Memory: 5884 bytes
Insertion Sort:
Insertion Sort Time: 0.000030 seconds
Insertion Sort Memory: 9576 bytes
Bubble Sort:
Bubble Sort Time: 0.000081 seconds
Bubble Sort Memory: 33592 bytes
Merge Sort:
Merge Sort Time: 0.000051 seconds
Merge Sort Memory: 9608 bytes

Works Cited

- GeeksforGeeks. (n.d.). *Sorting algorithms*. GeeksforGeeks. https://www.geeksforgeeks.org/dsa/sorting-algorithms/
- OpenAI. (2025). ChatGPT (version GPT-4) [Large language model]. https://openai.com/chatgpt