

# CS300 – Spring 2020-2021 - Sabancı University

## Homework #4 – Sorting and Searching

Due: May 22<sup>nd</sup> Saturday at 23:55

### Brief Description

In this homework, you are given an **unsorted** PhoneBook and asked to sort it in **alphabetical** order. Since you are a CS student, you care about the time taken to sort the PhoneBook so you want the **fastest** comparison-based sorting algorithm to perform this operation. Therefore, you must compare the given **4** Sorting algorithms and conclude which one is faster. Also, since fundamental usage of a PhoneBook is search for some names, you are expected to implement **2** search algorithms to compare.

The sorting algorithms you will compare are:

- **Insertion Sort**
- **Quick Sort** (pivot=median) (use **Insertion Sort** when size < 10)
- **Merge Sort** (in place; without using auxiliary memory storage)
- **Heap Sort**

The searching algorithms you will compare are:

- **Sequential Search**
- **Binary Search**

### Input

First, your program should ask for an input file (PhoneBook file) then it asks for a query entry where you should enter a full/partial contact name in one line. Here is a detailed simple input txt example.

PhoneBook.txt

Harold Stenseth +905547710771 Izmir Elbert Womack +905551674361 Istanbul Ashley Jepsen +905514534601 Izmir
--

As is shown in the sample PhoneBook.txt file, every line is for one person and Name, Surname, Telephone, and City are the provided information. You do not have any assumption on the

length of the name and surname which are separated by a space. The telephone has a standard length of 13 characters. You also do not have any assumption on the length of the city name, but it is one word for sure.

Another input you will ask is a search query. There are two types of searches you need to observe over the dataset, which is partial and full name search:

- Full Contact Name Search  
In this case, the user provides the whole name and surname and you will be searching for an exact match in both.
- Partial Contact Name Search  
In this case, your input is part of the name starting from the beginning and you do not have pre-knowledge of where it ends. **In the case of repeated partial names and key matches, you need to detect every match and print them out.**

## Output

After you receive the inputs (dataset and search query), you should first display the sorting times in Nanoseconds and then the speedups of the sorting algorithms. Afterward, the search times need to be printed, and last, the speedup list for search algorithms (check the sample runs to have a clearer image). After getting the execution times, you will be able to compute the speedups and display them. **Additionally, you are expected to print the search results (see sample runs).**

## Program Flow

After taking the inputs, Firstly, you have to read the input file and load the PhoneBook into different vectors where you are going to apply a sorting algorithm on each one. There are 2 types of search that should be valid in your implementation, Partial and Full name search. They are described in the previous part of this document.

After loading the PhoneBook into each vector copy, your program should ask for a search query (Partial or Full), then all 4 vectors should be sorted according to the sorting algorithms listed above. Concerning the search operation, it should be performed on the same sorted PhoneBook copy so you can make comparisons between the search running times on both algorithms.

For the comparisons, you should measure the running time without printing anything to the screen then display it for each structure. You are also required to calculate the speedups between algorithms, the equation for calculating the speedup is given below:

$$\text{Speedup} = \frac{\text{Slower Sorting Algorithm exec time}}{\text{Faster Sorting Algorithm exec time}}$$

The same formula applies when comparing between the vector search times:

$$\text{Speedup} = \frac{\text{Slower Search exec time}}{\text{Faster Search exec time}}$$

### Important notes

- You are asked to print out the search results to the screen, your search functions should return correct results, this will be tested during the grading process.
- In Quick Sort Algorithm, you are expected to choose **the pivot as the median**.
- In Merge Sort, you are expected to have an **in-place version**. Means no extra memory usage.
- Your classes should be **template-based**.
- **Note:** The provided sample runs were taken from a machine with the following specifications:
  - CPU: Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz
  - RAM: 16GB

Therefore, we understand that you have different machines, it can be powerful or weak in terms of specifications, so, we don't expect the speedup ratios to be exactly the same as in the sample runs, however, the results should be reasonable and conform with the correctness of the results, for example:

- In **sample run 4**, the speedup ratio (Insertion Sort / Quick Sort) is so high (31.779) so we will accept lower ratios like 30 or 20 or even 10. However, ratios

beneath 2 would be suspicious and ratios like 0.9 or less (less than 1) are not acceptable and it shows that the code is problematic somehow.

→ In order to find the exact time, you can run the search function N times then you take the measured time and divide it by N, you will have much sharper timings.

**Example:**

```
int N = 1000; // if it takes too much time then N = 100
int timeStart // start measuring
for (int i = 0; i < N; i++)
    search_function()
int timeEnd // end measuring
executionTime = (endTime - startTime) / N
```

## Sample runs

### Sample Run 1

Please enter the contact file name:

PhoneBook-sample1.txt

Please enter the word to be queried :

Mario

Sorting the vector copies

=====

Quick Sort Time: 309111 Nanoseconds

Insertion Sort Time: 219424 Nanoseconds

Merge Sort Time: 259642 Nanoseconds

Heap Sort Time: 409247 Nanoseconds

Searching for Mario

=====

Search results for Binary Search:

MARIO does NOT exist in the dataset

Binary Search Time: 51550 Nanoseconds

Search results for Sequential Search:

MARIO does NOT exist in the dataset

Sequential Search Time: 165588 Nanoseconds

SpeedUp between Search Algorithms

=====

(Sequential Search/ Binary Search) SpeedUp = 3.21218

SpeedUps between Sorting Algorithms

=====

(Insertion Sort/ Quick Sort) SpeedUp = 0.709855

(Merge Sort / Quick Sort) SpeedUp = 0.839964

(Heap Sort / Quick Sort) SpeedUp = 1.32395

## Sample run 2

Please enter the filename:

PhoneBook-sample2.txt

Please enter the word to be queried :

Robert

Sorting the vector copies

=====

Quick Sort Time: 4686352 Nanoseconds

Insertion Sort Time: 15465172 Nanoseconds

Merge Sort Time: 15308845 Nanoseconds

Heap Sort Time: 8123151 Nanoseconds

Searching for Robert

=====

Search results for Binary Search:

ROBERT CAMACHO +905556997035 Istanbul

ROBERT COY +905559274775 Bursa

ROBERT DELGENIO +905535624463 Bursa

ROBERT FISCHER +905536567181 Istanbul

ROBERT KINCHEN +905585715179 Ankara

ROBERT MCMILLIN +905566459683 Mugla

ROBERT MOSS +905596290308 Mugla

ROBERTA CARLYON +905592886254 Mugla

Binary Search Time: 109381 Nanoseconds

Search results for Sequential Search:

ROBERT CAMACHO +905556997035 Istanbul

ROBERT COY +905559274775 Bursa

ROBERT DELGENIO +905535624463 Bursa

ROBERT FISCHER +905536567181 Istanbul

ROBERT KINCHEN +905585715179 Ankara

ROBERT MCMILLIN +905566459683 Mugla

ROBERT MOSS +905596290308 Mugla

ROBERTA CARLYON +905592886254 Mugla

Sequential Search Time: 1765180 Nanoseconds

SpeedUp between Search Algorithms

=====

(Sequential Search/ Binary Search) SpeedUp = 16.1379

SpeedUps between Sorting Algorithms

=====

(Insertion Sort/ Quick Sort) SpeedUp = 3.30005

(Merge Sort / Quick Sort) SpeedUp = 3.26669

(Heap Sort / Quick Sort) SpeedUp = 1.73336

### Sample Run 3

Please enter the filename:

PhoneBook-sample3.txt

Please enter the word to be queried :

Tina Gulko

Sorting the vector copies

=====

Quick Sort Time: 2499417 Nanoseconds

Insertion Sort Time: 4530183 Nanoseconds

Merge Sort Time: 4998825 Nanoseconds

Heap Sort Time: 3905338 Nanoseconds

Searching for Tina Gulko

=====

Search results for Binary Search:

TINA GULKO +905555214353 Izmir

Binary Search Time: 156526 Nanoseconds

Search results for Sequential Search:

TINA GULKO +905555214353 Izmir

Sequential Search Time: 937277 Nanoseconds

SpeedUp between Search Algorithms

=====

(Sequential Search/ Binary Search) SpeedUp = 5.988

SpeedUps between Sorting Algorithms

=====

(Insertion Sort/ Quick Sort) SpeedUp = 1.8125

(Merge Sort / Quick Sort) SpeedUp = 2

(Heap Sort / Quick Sort) SpeedUp = 1.5625



#### Sample Run 4

Please enter the filename:

PhoneBook-shuffled.txt

Please enter the word to be queried :

Gulsen Demiroz

Sorting the vector copies

=====

Quick Sort Time: 101234080 Nanoseconds

Insertion Sort Time: 3217155950 Nanoseconds

Merge Sort Time: 2795505390 Nanoseconds

Heap Sort Time: 214487870 Nanoseconds

Searching for Gulsen Demiroz

=====

Search results for Binary Search:

GULSEN DEMIROZ does NOT exist in the dataset

Binary Search Time: 312429 Nanoseconds

Search results for Sequential Search:

GULSEN DEMIROZ does NOT exist in the dataset

Sequential Search Time: 26556300 Nanoseconds

SpeedUp between Search Algorithms

=====

(Sequential Search/ Binary Search) SpeedUp = 84.9994

SpeedUps between Sorting Algorithms

=====

(Insertion Sort/ Quick Sort) SpeedUp = 31.7794

(Merge Sort / Quick Sort) SpeedUp = 27.6143

(Heap Sort / Quick Sort) SpeedUp = 2.11873

## Sample Run 5

Please enter the filename:

PhoneBook-shuffled.txt

Please enter the word to be queried :

William

Sorting the vector copies

=====

Quick Sort Time: 103670750 Nanoseconds

Insertion Sort Time: 3013754750 Nanoseconds

Merge Sort Time: 2668554670 Nanoseconds

Heap Sort Time: 195263440 Nanoseconds

Searching for William

=====

Search results for Binary Search:

WILLIAM AMMANN +905511932805 Bursa

WILLIAM ARTHUR +905562451286 Mugla

WILLIAM BAIG +905550568328 Izmir

WILLIAM BASH +905586488892 Izmir

WILLIAM BOOTHBY +905589026291 Mugla

WILLIAM BYRD +905524927636 Ankara

WILLIAM CHILDS +905533803958 Ankara

WILLIAM CHOQUETTE +905594109713 Izmir

WILLIAM CLINE +905585739452 Ankara

WILLIAM CONSTANT +905526153465 Ankara

WILLIAM CORONADO +905527551169 Istanbul

WILLIAM COULSON +905547697706 Ankara

WILLIAM CRAWFORD +905588504000 Istanbul

WILLIAM DILLON +905560055969 Izmir

WILLIAM EVANS +905516331172 Ankara

WILLIAM FELLHOELTER +905582753443 Izmir

WILLIAM FRANKLIN +905556802033 Mugla

WILLIAM FULTS +905541092648 Istanbul

WILLIAM GARCIA +905518961107 Istanbul

WILLIAM GRAY +905538931487 Bursa

WILLIAM GREEN +905525236487 Mugla

WILLIAM GRIMES +905536548349 Izmir

WILLIAM HARRISON +905586333675 Istanbul

WILLIAM HAYES +905551962201 Istanbul  
WILLIAM HERNANDEZ +905553079127 Istanbul  
WILLIAM HINTZE +905549889332 Izmir  
WILLIAM HOLDER +905512104631 Bursa  
WILLIAM HOLLEMAN +905514849636 Mugla  
WILLIAM HOLLOBAUGH +905524353742 Ankara  
WILLIAM HULL +905556815354 Ankara  
WILLIAM HUMPHREYS +905579083699 Ankara  
WILLIAM ISLAS +905550159250 Bursa  
WILLIAM JIMENEZ +905576076829 Izmir  
WILLIAM JOHNSON +905546114538 Istanbul  
WILLIAM KELLY +905554525839 Izmir  
WILLIAM KOCI +905577674896 Izmir  
WILLIAM LEWIS +9055419074294 Ankara

Binary Search Time: 156207 Nanoseconds

Search results for Sequential Search:

WILLIAM AMMANN +905511932805 Bursa  
WILLIAM ARTHUR +905562451286 Mugla  
WILLIAM BAIG +905550568328 Izmir  
WILLIAM BASH +905586488892 Izmir  
WILLIAM BOOTHBY +905589026291 Mugla  
WILLIAM BYRD +905524927636 Ankara  
WILLIAM CHILDS +905533803958 Ankara  
WILLIAM CHOQUETTE +905594109713 Izmir  
WILLIAM CLINE +905585739452 Ankara  
WILLIAM CONSTANT +905526153465 Ankara  
WILLIAM CORONADO +905527551169 Istanbul  
WILLIAM COULSON +905547697706 Ankara  
WILLIAM CRAWFORD +905588504000 Istanbul  
WILLIAM DILLON +905560055969 Izmir  
WILLIAM EVANS +905516331172 Ankara  
WILLIAM FELLHOELTER +905582753443 Izmir  
WILLIAM FRANKLIN +905556802033 Mugla  
WILLIAM FULTS +905541092648 Istanbul  
WILLIAM GARCIA +905518961107 Istanbul  
WILLIAM GRAY +905538931487 Bursa  
WILLIAM GREEN +905525236487 Mugla  
WILLIAM GRIMES +905536548349 Izmir

WILLIAM HARRISON +905586333675 Istanbul  
WILLIAM HAYES +905551962201 Istanbul  
WILLIAM HERNANDEZ +905553079127 Istanbul  
WILLIAM HINTZE +905549889332 Izmir  
WILLIAM HOLDER +905512104631 Bursa  
WILLIAM HOLLEMAN +905514849636 Mugla  
WILLIAM HOLLOBAUGH +905524353742 Ankara  
WILLIAM HULL +905556815354 Ankara  
WILLIAM HUMPHREYS +905579083699 Ankara  
WILLIAM ISLAS +905550159250 Bursa  
WILLIAM JIMENEZ +905576076829 Izmir  
WILLIAM JOHNSON +905546114538 Istanbul  
WILLIAM KELLY +905554525839 Izmir  
WILLIAM KOCI +905577674896 Izmir  
WILLIAM LEWIS +9055419074294 Ankara

Sequential Search Time: 26470800 Nanoseconds

SpeedUp between Search Algorithms

=====

(Sequential Search/ Binary Search) SpeedUp = 169.46

SpeedUps between Sorting Algorithms

=====

(Insertion Sort/ Quick Sort) SpeedUp = 29.0704

(Merge Sort / Quick Sort) SpeedUp = 25.7407

(Heap Sort / Quick Sort) SpeedUp = 1.8835

## Sample Run 6

Please enter the filename:

PhoneBook-shuffled.txt

Please enter the word to be queried :

Virg

Sorting the vector copies

=====

Quick Sort Time: 103523720 Nanoseconds

Insertion Sort Time: 3200932540 Nanoseconds

Merge Sort Time: 2754943720 Nanoseconds

Heap Sort Time: 214178500 Nanoseconds

Searching for Virg

=====

Search results for Binary Search:

VIRGIE DILORENZO +905522148183 Mugla

VIRGIL SCHIRO +905520736869 Ankara

VIRGINIA ATKINS +905599534420 Bursa

VIRGINIA BASQUEZ +905577945155 Istanbul

VIRGINIA BOWMAN +905554551074 Bursa

VIRGINIA BROWN +905575396124 Izmir

VIRGINIA BURNS +905575713316 Bursa

VIRGINIA HARTLE +905559017152 Istanbul

VIRGINIA HOBERG +905574933865 Izmir

VIRGINIA KAYSER +905588959312 Mugla

VIRGINIA MAJOR +905524248481 Bursa

VIRGINIA MEEKS +905559851680 Ankara

VIRGINIA OVERSTREET +905521897621 Bursa

VIRGINIA ROBINSON +905535242466 Ankara

VIRGINIA RUDDY +905559278674 Bursa

VIRGINIA SCHMITZ +905575344091 Mugla

VIRGINIA TESTER +905566236472 Bursa

VIRGINIA WOODHAMS +905543498083 Bursa

VIRGINIA ZAPATA +905591602872 Mugla

Binary Search Time: 156538 Nanoseconds

Search results for Sequential Search:

VIRGIE DILORENZO +905522148183 Mugla  
VIRGIL SCHIRO +905520736869 Ankara  
VIRGINIA ATKINS +905599534420 Bursa  
VIRGINIA BASQUEZ +905577945155 Istanbul  
VIRGINIA BOWMAN +905554551074 Bursa  
VIRGINIA BROWN +905575396124 Izmir  
VIRGINIA BURNS +905575713316 Bursa  
VIRGINIA HARTLE +905559017152 Istanbul  
VIRGINIA HOBERG +905574933865 Izmir  
VIRGINIA KAYSER +905588959312 Mugla  
VIRGINIA MAJOR +905524248481 Bursa  
VIRGINIA MEEKS +905559851680 Ankara  
VIRGINIA OVERSTREET +905521897621 Bursa  
VIRGINIA ROBINSON +905535242466 Ankara  
VIRGINIA RUDDY +905559278674 Bursa  
VIRGINIA SCHMITZ +905575344091 Mugla  
VIRGINIA TESTER +905566236472 Bursa  
VIRGINIA WOODHAMS +905543498083 Bursa  
VIRGINIA ZAPATA +905591602872 Mugla

Sequential Search Time: 26269600 Nanoseconds

SpeedUp between Search Algorithms

=====

(Sequential Search/ Binary Search) SpeedUp = 167.816

SpeedUps between Sorting Algorithms

=====

(Insertion Sort/ Quick Sort) SpeedUp = 30.9198

(Merge Sort / Quick Sort) SpeedUp = 26.6117

(Heap Sort / Quick Sort) SpeedUp = 2.06888

## General Rules and Guidelines about Homeworks

The following rules and guidelines will be applicable to all homework unless otherwise noted.

### How to get help?

You may ask questions to TAs (Teaching Assistants) of CS300. Office hours of TAs can be found [here](#). Recitations will partially be dedicated to clarifying the issues related to homework, so it is to your benefit to attend recitations.

### What and Where to Submit

Please see the detailed instructions below/in the next page. The submission steps will get natural/easy for later homework.

### Grading

Careful about the semi-automatic grading: Your programs will be graded using a semi-automated system. Therefore, you should follow the guidelines about input and output order; moreover, you should also use the exact same prompts as given in the Sample Runs. Otherwise, the semi-automated grading process will fail for your homework, and you may get a zero, or in the best scenario, you will lose points.

#### Grading:

- ☐ **We will hold a demo session for your homework grading. Each one of you must show that your code is running as expected and explain several parts of your code if necessary. Wait for an announcement for the demo schedule.**
- ☐ A late penalty is 10% off the full grade and only one late day is allowed.
- ☐ **Having a correct program is necessary, but not sufficient to get the full grade. Comments, indentation, meaningful and understandable identifier names, informative introduction and prompts, and especially proper use of required functions, unnecessarily long program (which is bad) and unnecessary code duplications will also affect your grade.**
- ☐ Please submit your own work only (even if it is not working). It is really easy to find “similar” programs!
- ☐ For detailed rules and course policy on plagiarism, please check out <http://myweb.sabanciuniv.edu/gulsend/courses/cs201/plagiarism/>

## Plagiarism will not be tolerated!

Grade announcements: Grades will be posted in SUCourse, and you will get an Announcement at the same time. You will find the grading policy and test cases in that announcement.

Grade objections: Since we will grade your homework with a demo session, there will be very likely no further objection to your grade once determined during the demo.

## What and where to submit (IMPORTANT)

Submission guidelines are below. Most parts of the grading process are automatic. Students are expected to strictly follow these guidelines to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade.

Add your name to the program: It is a good practice to write your name and last name somewhere in the beginning program (as a comment line of course).

Name your submission file:

- ☐ Use only English alphabet letters, digits or underscore in the file names. Do not use blank, Turkish characters or any other special symbols or characters.
- ☐ Name your cpp file that contains your program as follows.  
    **“SUCourseUserName\_yourLastname\_yourName\_HWnumber.cpp”**
- ☐ Your SUCourse username is actually your SUNet username which is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII, and Turkish characters in the file name. For example, if your SUCourse username is cago, the name is Çağlayan, and the last name is Özbugsızkodyazaroglu, then the file name must be:  
    **cago\_ozbugsizkodyazaroglu\_caglayan\_hw4.cpp**
- ☐ Do not add any other character or phrase to the file name.
- ☐ Make sure that this file is the latest version of your homework program.
- ☐ You need to submit ALL .cpp and .h files including the data structure files in addition to your main.cpp in your VS solution.
- ☐ The name of the main cpp file should be as follows.  
    **“SUCourseUserName\_yourLastname\_yourName\_HWnumber.cpp”**  
For example zubosman\_Osmanoglu\_Zubeyir\_hw4.cpp is a valid name, but  
hw5\_hoz\_HasanOz.cpp, HasanOzHoz.cpp are NOT valid names.

Submission:

Submit via SUCourse ONLY! You will receive no credits if you submit by other means (e-mail, paper, etc.).

**Successful submission is one of the requirements of the homework. If for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.**

***Good Luck!***

***Emre Vardar and Gülşen Demiröz***