

Fall 2023 – CS 201 Data Structures and Algorithms  
Homework-3

Maximum points: 200. Individual Work Only.

Due Date: November 2, 2023 before 11:59pm (late submissions will get a score of zero)

### Objective

- Select the best data structure and design an efficient algorithm for search and retrieval

### Problem Description

1. Write a C++ program that reads a file which contains movie names and actors separated by “/” and store the movie name and actors using a data structure such that we can search for either the movie or the actor. If the movie name is specified as the search query, then the search result must be all the actors in that movie printed one per line. If the actor’s name is specified as the search query, then the search result must be all the movies that the actor has acted in printed one per line. If either the movie name or actor name is not found, then an appropriate “Not Found” message must be printed. The file that contains the movie names and actors and the search query file are provided as command-line arguments to the program. The key objective of this homework is to use the best data structure and design an efficient algorithm that provides the best search and retrieval times. You are free to use any of the data structures or algorithms provided by the C++ standard library (i.e., std namespace). A sample program to read the two files and parse them is provided in Blackboard (HW3Sample.cpp), you can use this program to get started.
2. After testing your program for correctness and efficiency, use the different input files provided in Blackboard along with the given query file (query.txt) and complete the table below:

| DB File     | # of records | Time taken to create<br>data structure | Time taken to<br>search | Total time taken |
|-------------|--------------|--|-------------------------|------------------|
| dbfile1.txt | 442 lines    | 0.050224 seconds                       | 0.008575 seconds        | 0.058799 seconds |
| dbfile2.txt | 7065 lines   | 0.942367 seconds                       | 0.01031 seconds         | 0.952677 seconds |
| dbfile3.txt | 14129 lines  | 1.75858 seconds                        | 0.008529 seconds        | 1.76711 seconds  |

3. Include the table in the report and write a short description that describes the rationale for your choice of the data structure and corresponding algorithm. Determine the complexity of the search and retrieval operation, compare the search time for different db file sizes, and explain the results obtained in the table.

### Program Documentation and Testing

1. Use appropriate variables names and indentation in your source code.
2. Include meaningful comments to indicate various operations performed by the program.
3. Programs must include the following header information within comments:

```
/*  
    Name:  
    Email:  
    Course Section: Fall 2023 CS 201  
    Homework #:  
    Instructions to compile:  
    Instructions to run:  
*/
```

### Submission

Upload only the source files (.h or .hpp or .cpp or .cc files) and the report (Word or PDF file) to Blackboard in the assignment submission section for this homework. Do not upload zip/tar files to Blackboard, upload individual source files (no object files or executables) and the Word or PDF file for the report.

### Grading Rubrics

The program will be graded not only for correctness with the sample input provided but also for the most efficient implementation. The following rubrics is used for grading:

| Description   | Points |
|---|--------|
| 1. Correct implementation of the solution   | 130    |
| 2. Efficient implementation   | 40     |
| 3. Report with table and justification for the choice of the data structure and algorithm | 30     |