

CS 302 Homework 3: Lost assignments

Description

An instructor for a class decided to clean his/her office and noticed that he/she lost the students assignments (I guess the instructor forgot to record the grades earlier in the semester). Luckily the instructor found a pile of the graded assignments but looks like a few students' assignments are lost and somehow some of them are duplicated. The instructor wants to determine how many submissions was lost for each student by simply tallying up each assignment for each student without counting any duplicates. Each student has turned in 9 assignments so far thus if any student tally amount is below 9 then the instructor lost at least one assignment for that student. The instructor wants this task done quickly since he/she is often busy, thus a hash map will be used to tally up each assignment for each student (without counting duplicates) since a brute force search would take quite some time especially if there are a lot of assignments and/or a lot of students in the class.

Input

You will be given two files, one that has all the students' names and the other with a set of assignments, each assignment will contain the following string `studentName_A#` where # will be a number from 1-9

Output

Output the amount of missing assignments for each student

Contents of main

You will prompt for a names file and store all the students names into a `vector of string` objects. You then will read in a file with the assignment submissions, and determine if this file has already been tallied, if it has not been then tally into the amount of assignments tallied for a student. This will require a few `unordered_map objects` (I will not tell you what the key and value types will be, you'll need to figure that out)

Specifications

- Comment your code and your functions
- You must use the `unordered_map` to find student names or assignments
- You can traverse through elements in a list but all searches must use a hash, no linear or binary searches can be used

Written portion

Assume the amount of student names is some large constant and the amount of assignments is n

1. Assume every hashes look up takes $O(1)$ time, what would the runtime of your algorithm you wrote be? (In terms of Big O using n)

2. What if you sorted the list of students and the assignments, what would the runtime of this algorithm be? (In terms of Big O using n)
3. What if you maintained an unsorted list of students names and assignments, what would the runtime of this algorithm be? (In terms of Big O using n)

Extra Credit

You can get 20 extra credit points for the assignment total if you can answer the following:

Assume the amount of names is n and the amount of assignments is m

1. Assume every hash look up takes $O(1)$ time, what would the runtime of your algorithm you wrote be? (In terms of Big O using n and/or m)
2. What if you sorted the list of students and the assignments, what would the runtime of this algorithm be? (In terms of Big O using n and/or m)
3. What if you maintained an unsorted list of students names and assignments, what would the runtime of this algorithm be? (In terms of Big O using n and/or m)

```
Sample Run

$ g++ Assignment06.cpp
$ ./a.out

Enter students file: Students.txt
Enter students file: students.txt
Enter students file: Names.txt

Enter assignments file: Programs.txt
Enter assignments file: Assignments.txt

NAME           MISSING ASSIGNMENTS
Ora             0
Wyatan         5
Aubrey         7
Delinda        8
Ruttger        8
Blanche        0
Ford           7
Doretta        5
Lu             1
Annamaria      8
Ida            7
Andie          0
Rosina         7
Chantalle      8
Hilary         4
Glyn           6
Pacorro        1
Matthew        8
Kassi          0
Jilly         5
Bentlee        4
Jolie          5
Lorena         8
Tasia          8
Sebastien      8
```

Submission

Upload a zip file with your source code, the input files, and the write up.