

CSAI 230 - Fall 2024

Assignment 4: Hashing

Assigned: Sunday, December 15th in Class

Due: Sunday, December 29th at 17:00

Delayed submission with penalty until Thursday, December 31st at midnight.

Goals

This assignment is an individual assignment and you will work on it on your own. The goal of this assignment is to implement hashing. This assignment is a continuation of assignment 3. The whole idea is that instead of indexing the web server log records we used in assignment 3 into an AVL tree, we will use hashing to hash the IP address into a hash table. Each entry in the hash table will point to an AVL tree that contains all the records of the corresponding IP indexed by date and time.

Details

As mentioned above, this assignment depends on assignment 3. In this assignment, you are required to use separate chaining hashing to index IP addresses into a hash table; so your key is the IP address. Each hash table entry will contain a pointer to an AVL tree that index all the records associated with the corresponding IP addresses by date and time.

With careful look at this assignment, one can realize that all the building blocks are already available for you, and what need to be done is to assemble them in the right order and structure. Although you already have a linked list implementation that you can use with the separate chaining hash table, nevertheless you are required to use the AVL tree implementation to construct the separate chaining hashing for better performance. You are also required to use the same AVL tree implementation for indexing records by date and time. Finally, we have presented in class the source code of multiple hash functions, and you can choose one for your implementation or come up with your own if you think that for this specific problem you can provide a better hash function; make sure to choose an efficient hash function and that you can justify your choice as this will be part of our grading criteria.

You are required to provide methods that can search for records with a specific IP address within a range of date and time, and this is already supported by your AVL tree implementation in assignment 3.

You are required to conduct the Big-O analysis for your hash table and show that it provides the best running time for the intended functionality.

Dataset

You will use the same dataset as in assignment 3.

What to submit

1. Your full in-line documented source code; it should include a main program that has some test cases that shows how your ADT works.
2. A PDF report that includes:
 - a. Any assumptions you have made.
 - b. Your design approach to solve this problem and why do you think it is the most optimum approach.
 - c. Your Big-O Analysis with enough reasoning on how you reached such analysis.
3. A readme file that explains how to compile and run your program.
4. Your code should compile using the GNU C/C++ compiler and run on Linux; I advice you to use a Dockers image for that. Docker is a very important technology that utilizes process level virtualization and the GIT repository management system and can integrate with github. The benefit of using dockers to submit your assignment is mainly the ability of the TA to reproduce your work easily, and put you in the safe side of any assumption about your work. (Let me know if you need help and if you are interested in learning how to use dockers and I can help you with that during my office hours)

How to submit:

Compress all your work: source code, report, readme file, and any extra information into a zip archive. You should name your archive in the specific format <Student_ID>_<Name>_Assignment4.zip. Finally, upload your code to moodle.

If you were able to package your work into a docker environment, which is optional, you can mention your docker repository in your readme file, but it is a must to upload all your source code assignment material to moodle.

Grade

This assignment is worth 15% of the overall course grade. The assignment will be graded on a 100% grade scale, and then will be scaled down to the 15% its worth. The grading of the assignment will be broken down as follows:

1. 10 % for just submitting a meaningful assignment before or on the due date. This 10% does not account for the correctness of your assignment but submitting an empty assignment without code will definitely results in loosing this 10% and consequently the whole grade of this assignment.
2. 50 % for the correctness of your code.
3. 15 % for the quality of the inline documentation.
4. 25 % for the report and the analysis.

Delays

You have up to 2 working days of delay, after which the assignment will not be accepted and your grade in that case will be ZERO. For every day (of the 2 allowed days), a penalty of 10% will be deducted from the grade. And of course you will lose the 10% mentioned in point 1 above under the “Grade” section.