Algorithms

CMPT 435

Assignment 2 - 75 points

Goals

- to implement searching and hashing, and to understand their performance.

Requirements and Notes

- Download the the `magicitems.txt` file from our web site again.
- Read it line-by-line into an array, reusing (and improving) your code from assignment #1 (unless it was already perfect, in which case: yay!).
- **Sort the array** using one of your sort implementations from assignment #1. Include a copy of your sorting code in this assignment's directory so that your project remains easy to compile.

- Develop **your own** implementation of *linear* and *binary* search. [15 points]
  - Randomly pick 42 items from the array of magic items.
  - Perform a *linear search* of the entire (sorted) array for each of those 42 randomly selected items. Print the number of comparisons used for each search and compute the overall average to two decimal places. Be careful about counting comparisons.
  - Perform a *binary search* on the entire (sorted) array for the same 42 items as before. Print the number of comparisons used for each search and compute the overall average. Remain careful about counting comparisons.
  - Record your results in a table in a LaTeX document along with your code listings and documentation. Note the asymptotic running time of each search and **explain why it is that way**. [25 points]

- Develop **your own** implementation of a hash table with chaining, of size 250. Use the hash function we spoke about in class (and in the example code on our web site at https://www.labouseur.com/courses/algorithms/Hashing.java.html). [10 points]
  - Load your hash table with all of the the magic items.
  - Retrieve the **same** 42 (no longer random) items from your hash table. Print the number of (*get* + comparisons) for each item and compute the overall average to two decimal places. (Every *get* is one compare, then count the comparisons needed to handle chaining.)
  - Add these results to your LaTeX document, including the asymptotic running time of hashing with chaining and **explain why it is that way**. [25 points]

As usual, your code must separate structure from presentation, be professionally formatted yet uniquely yours (show some personality), use and demonstrate best practices, and make me proud to be your teacher. [−∞ if not]

Resources

- Linear and binary search are described in the 3rd edition of our text in 10.2 and 27.3.
- Hash tables with chaining are described in the 3rd edition of our text in section 11.2.

Submitting Your Work

In addition to your source code, commit your LaTeX document in both .tex and .pdf forms to your GitHub repository. For your code, make **many** commits to GitHub. If you don't make enough commits, I will not accept your work. Be sure that you make your final commit for this assignment on or before the due date. (See our syllabus for those details.)