
Instructor: Michael Hackett
Department: Computer Science
Email: mhackett@ccp.edu

J Number Hashing

Each student at CCP is assigned an ID called a “J Number”. Every J Number consists of the letter J followed by 8 digits. For example, “J12345678”.

Your assignment is to create a custom hash function for J Numbers. You’re only interested in the function’s performance: how well it avoids collisions without any collision resolution. You do not actually need to create a hash table, only create the hash function and test it in the following four ways:

- 100 randomly generated J Numbers hashed for an array with a length of 127.
- 500 randomly generated J Numbers hashed for an array with a length of 512.
- 1000 randomly generated J Numbers hashed for an array with a length of 1024.
- 25000 randomly generated J Numbers hashed for an array with a length of 27000.
 - Roughly the current enrollment at CCP

To establish a baseline, complete all four tests with the simple string hash function shown in class. Compare the results of your hash function’s performance and the simple string hash function’s performance. If necessary, tweak your hash function so that it performs better than the simple hash function.

You will need to keep track of how many times a collision occurred in both cases. You could create arrays of the appropriate length and increment each index that is the result of the hash function. If, at the end of the test, index 4 contained a 3, it means there were **two** collisions for that index. (See the Test.cpp source code files in the HashingComparisons folder of the Module download for inspiration.)

Your hash function should expect a string containing a J Number, not an int or other data type.

In a separate document, you will explain:

- How your hash function works: its complexity, how you came up with the idea for the implementation of your hash function, how you tweaked/improved your original idea, etc.
- Its performance in the four tests: how well did the hash function do at distributing indexes, how well did it avoid collisions, what was the ratio of successful hashes to collisions, etc.
- How your hash function performed compared to the simple string hash function.

Submit your document and all related source code files in the Assignment 7 submission link.

Grading

See Assignment Rubric in Canvas.