Ewan Seaders

Assignment 7 readme

File path: g++ test.cpp smart\_hash.cpp -o test

1. Create a design **before** you begin to code that describes or shows how we can store data in a hash table and what kind of problem we could solve with a hash table.

A screenshot of a computer

Description automatically generated

1. Create some tests (at least one per piece of functionality) **before** you begin coding that you want your hashtable to pass before you start coding.

The design also includes the tests

1. Create a hashtable that resolves collisions by simply overwriting the old value with the new value, including at least:
   1. Describe the way that you decide on hashing a value  
      (this can be simple or complex based on how interesting you find the topic)

A screen shot of a computer code

Description automatically generated with low confidence

* 1. An insert function that places the value at the appropriate location based on its hash value

A screenshot of a computer program

Description automatically generated with low confidence

* 1. A contains function that returns whether the value is already in the hashtable

A picture containing text, screenshot, software, multimedia software

Description automatically generated

* 1. (optional) A delete function that removes a value based on its hash and then returns that value…

A screen shot of a computer program

Description automatically generated with low confidence

1. Then create a smarter hashtable (double hashing or chaining) including at least the same functions as the simple hashtable

The photos above are of the functions from the smart hash table because I used the one we made in class for the dumb hashtable

1. Compare some information relating to collisions (frequency) and their effect on complexity (of insert and contains methods)

The collision frequency was much better using my has method in the smart hash table especially when it came to permutations of the same word. In fact, with such a small table it was more certain that those would not collide than anything else.