O COMP 215 Algorithms Θ

Lab #10

This work should be submitted electronically before 11:59pm Thursday. There is comparatively little you need to code this time, so please submit a single file lab10.cpp containing all your code, with the answers to all the questions I ask here written in comments at the beginning of the file. The code in the file should allow me to recompute all these answers easily.

In this lab, you will experiment with the birthday paradox, and see how quickly collisions occur as we throw stuff in a hash table (we will not really be implementing a hash table, just hashing keys into an array and see what happens).

We will be using the `rand()` function a lot to generate random keys, please do not reseed the function anywhere in your code (that is, *do not* use the function `srand()` anywhere in your code), that way, the answers should be the same for everybody.

1. Implement a function `generateRandomKey` which does not take any input, but returns `rand()` mod 90000.
2. Implement a function `hashFunction` that takes a long integer input `key` and returns $((11057 * key) \bmod 179999) \bmod 90000$. We will see in the next course that this is a "good" hash function.

So basically, we will be generating random keys and hashing them into a great big array of 90000 elements and studying how long it takes before we get a collision, how long it takes before we have an element in every square of the array, how long it takes before we have multi-collisions, etc.

3. First, allocate an array `table` of 90000 long integers in your main function.
4. Initialize all the positions in this array to zero.
5. Inside a loop, repeatedly do the following operations:
   (a) Generate a random key and assign the result of the hash to a variable `key`.
   (b) Hash this key value and assign it to a variable `hashedKey`.
   (c) Add one to `table[hashedKey]`.
6. Use this to answer the following questions:
   (a) How many keys do you have to generate before you have your first collision? (you will find your first collision the first time that you get `table[k]` $= 2$ for some `k`)
   (b) How many keys do you have to generate before you have your first 5-fold collision? (that is, the first time you get `table[k]` $= 5$ for some `k`).

(c) How many collisions have occurred (that is, how many keys have been inserted in a location in table with `table[hashedKey]` >= 1) after inserting 90000 keys?

(d) How many spaces in the table are empty after inserting 90000 keys?

(e) What is the highest number of collisions in a space of the table (that is, what is the maximum value in `table`) after inserting 90000 elements

(f) How many random keys do you have to insert in the table before each space in the table has at least one key?

(g) After having inserted this minimum number of keys in the table so that each space in the table has at least one key, what is the maximum value in the table?