

CIS 3362 Homework #5 Solution (Coding Group)

Search Strategy

Since Java has a built in HashMap, where we can map each long (hash value) we've encountered and store a string with that hash value, to save time, I ported the give function over to Java first. I figured this would be faster than coding up my own HashMap structure in C.

Next, I wrote a function to generate a unique alphabetic string from each integer. This basically worked like a base 26 converter. For ease of implementation, I just made strings of upper case characters. Then I ran a for loop up to ($1 \lll 30$), but it never got that far. In the for loop, I used the loop index to generate the next string. I calculated the hash value of that string and searched my hash map to see if I had previously generated that hash value. If I had, then I found a match! I would output the String value mapped to this Long key and this new string that created the same hash value. Otherwise, I'd add this hash value, string pair to my hash map and go generate the next string. Unfortunately, this crashed in between 12 million and 13 million iterations, due to running out of heap space.

My next strategy was writing a generic function that returned a random string of a given length (from an input parameter) where each character was either an upper or lower case letter. I then tried generating random strings of length 5. Again, my program ran roughly 13 million iterations before I stopped it. (I stopped it, thinking that a better strategy probably exists.)

My next attempt was trying strings of random length, from 1 to 13, instead of strings of a fixed length. Almost immediately, my program spit out the match "wy" and "wyf". A bit of further testing insinuates that you can add a lowercase f to any alphabetic message that has a length not divisible by 6 and there will be no change in the hash value!!!

Thus, my matching strings are "wy" and "wyf".

For parts B and C, my matching messages are:

I have the goods. Meet me in 1 hr, by the blue dumpster by the loading docks.

and

I have the goods. Meet me in 1 hr, by the blue dumpster by the loading docks.f

This isn't really in the spirit of part C because the content of the message hasn't been altered.

Attached Files

hmk5.java (used to generate the solution)

hash_sol.c (used to verify the solution generated by the Java code)