Lab 5 COSC 3319 Fall 2017 Burris

For your convenience a file of 200+ random words and names (Words200D16) has been provided on Blackboard. Each line of text in the file is exactly 16 characters in length with the word left justified.

**"C" Option:**

Assume a main memory hash table of size 128

The hash address is to be calculated using folding combined with the division remainder technique

The hash address must be computed by adding the ((adding characters 1-2 of the key to characters 6-7 of the key) + 101) plus the 15th character of the key) modulo 128.

*All characters must be treated as positive integers in all steps of the calculation*!

Remember all characters and hash addresses must be treated as positive numbers

Use the remainder as the hash address

For example, it the string contains the value “ABCDEFGHIJK45678” then the hash address would be HA 🡨 ( ( “AB” + “GH” + 101 + ‘7’ ) modulo 128

Anticipate this function will be substantially less than optimal generating a plethora of collisions.

Use a long integer to hold the intermediate results of the hash calculation.

Remember a “slice” in Ada allows the user to treat a substring as a single unit

Given str has the value “ABCD” str(2..3) refers to the characters in positions 2 through 3) as a group from the string str, i.e., ”BC.”

I am assuming you have instantiated a function which converts a two character (8 bits per ASCII character) to a long integer (32 bits) so overflow will not occur in the sum.

**Each entry placed in the hash table must be a record with 3 fields**

**The first filed is the key**

**second the initial hash address**

**third field is the number of probes to insert the key in the table counting the initial hash address as the first probe.**

A)

Create a hash table in main memory and fill it 50% full.

Use the linear probe technique developed in class to handle collisions

Look up the first 30 entries placed in the table

Print the min, max, and avg number of probes required to locate the first 30 keys placed in the table

Now search for the last 30 keys placed in the table.

Print the min, max, and avg number of probes required to locate the last 30 keys placed in the table

Print the contents of the hash table clearly indicating open entries (see primary/secondary clustering effect)

Calculate and print the theoretical expected number of probes to locate a random item in the table.

Explain your empirical results in light of the theoretical results

**Your grade points will be highly dependent on your explanation!**

B)

Create a hash table in main memory and fill it 90% full

Use the linear probe technique developed in class to handle collisions

Now look up the first 30 entries placed in the table

Print the min, max, and avg number of probes required to locate the first 30 keys placed in the table

Now search for the last 30 keys placed in the table (physically search for each key to calculate the statistics)

Print the min, max, and avg number of probes required to locate the last 30 keys placed in the table

Start filling the table with the same keys in the same order as used when only filling the table 40%(50%) full

Print the contents of the table clearly indicating open entries (allow you to see the primary clustering effect)

Calculate and print the theoretical expected number of probes to locate a random item in the table

Discuss your empirical results in light of the theoretical results.

**Your grade points will be highly dependent on your explanation!**

C)

Repeat A and B above but use the random probe for handling collisions as developed in class.

When you print the contents of the table you should be able to visually see the secondary clustering affect.

Calculate and print the theoretical expected number of probes to locate a random item in the table.

Discuss your empirical results in light of the theoretical results.

**Your grade points will be highly dependent on your explanation!**

D)

***Present results in tabular format (typed naturally). Compare your results to the theoretical results.***

***If there are differences, please explain why***.

**I expect you to physically calculate the theoretical values for comparison to your empirical results for**

**both the linear and random probes**

E)

My required hash function has multiple weaknesses. Criticize the hash function on a technical basis.

**To receive full credit, you must state clearly and explicitly state why my hash function should fail!**

Based on your criticism, write a better hash function.

**Explain why your hash function should be better from both a theoretical and empirical standpoint.**

**You will not receive credit for simply writing a hash function that performs better**.

Implement the hash function and generate the same results as required for parts A thru D presented in tabular

format for comparison.

Formally evaluate the results as part of your lab (typed evaluation)

***The results for all parts of the lab should appear in a single table***

**"B" Option:**

You need not complete the "C" option using a main memory table.

Rather, complete the "C" option **using a random access (relative) file for the hash table**

The **hash table must never appear in main memory**!

Be sure to address all parts the “C” option.

A relative file is preferred for this option.

You may however use a commercial database product accessed from your Ada program if desired rather than a relative file. If a database is utilized, the primary key must be the hash address calculated as specified above.

**"A" Option:**

Choice (do one of the following only):

A) Do both the "C" and "B" options. **Compare the results. Main mem vs relative file**

B)

Enter 150 keys into a hash table of size 50 using main memory.

Place all keys that collide on a linked list treated as a circular queue, i.e.,

use the 50 entries in the hash table as list heads to linked lists of all keys that hash to that address

Repeat this process using a relative file for the hash table

The hash table must never appear in main memory!

Now look up the first 60 entries placed in the table

Print the minimum, maximum, and avg num of probes required to locate the first 60 keys placed in the table

Now search for the last 60 items placed in the table

Print the minimum, maximum, and avg num of probes required to locate the last 60 keys placed in the table. Exhibit your results in a tabular format.

Evaluate this strategy for handling collisions

What are its strengths and weaknesses?

Compare your results to the linear and random probe techniques.

You must complete all requirements for the “C” and “B” options to receive credit.