Yi Li

CSE 520P

HW 2

1. For this problem, I decided to implement the multiply-mod-prime hash function and the multiply-shift hash function. For my favorite hash function, I decided combine these two algorithms; I applied a multiply-shift hash function after doing a multiply-mod-prime hash function on the input.

I used n = since this would be effective in observing a full spectrum of the data.

To compare the times it took to hash all the elements, I experimented hashing 100, 200… 100n elements for all three algorithms. Here is the Graph below:

// Graph

As observed, the multiply-shift hash function is the fastest.

I also looked at largest bucket size and the sum of the squared bucket sizes against the number of elements.

// Graph

As observed, the multiply-shift hash function has very little difference from either the multiply-mod-prime and the favorite algorithm.

Thus, we can conclude that the multiply-shift hash function as efficient as multiply-mod-prime as a universal hash function while have a significant performance benefit.

After generating a set of n keys that had a lot of collisions for my favorite hash function, I ran this values against but the multiply-mod-prime hash and the multiply-shift hash function. As seen, these two algorithms are very effective at giving a universal hash function.

To compare the times it took to hash all the elements for linear probing, I experimented hashing 100, 200… 100n elements for all three algorithms.

I used n = .

Here is the Graph below:

// Graph

As observed, linear probing is much slower than the open chaining. This is due to the necessity that linear probing needs to find a spot.

I also looked at probe length and the sum of the squared probe lengths against the number of elements.

// Graph

As observed, as the number of elements increase, the probe length and sum of the squared probe lengths increase exponentially.

Note that the different set of algorithms do not have a large impact on this.

Next, I experimented with the size of the has table. Below are the Graphs for this.

// Graph

// Graph

// Graph

As observed, the time and probe lengths significantly decrease when the size of the hash table is increased.