Jennifer Jozefiak
Honors Project: Hashing: Linear Probing Versus Double Hashing Cluster Size
April 28, 2014

Introduction:

For my honors project I created two different hash classes, one that performed linear probing and one that performed double hashing. The goal was to calculate cluster sizes using the different hashing methods to find out if double hashing really does make a difference in cluster sizes.

Experiment:

I began by creating my hash classes. The capabilities each one had was to create an array, populate it, double the size if necessary, hash the values passed in (either linear or double), place the values to the position they were hashed, delete a value, and get a value. My linear probing hash class has six functions:

constructor: creates an empty array of the size passed in

_hash: generates the hash value

doubleSize: takes the existing array, doubles the size, then rehashes each value to the new array size

setitem: Places the key into the hashed index value of the array. calls double hash if the number of keys in an array is greater than half the array size.

delItem: Deletes a specified key, replacing it with a '*'. This value is considered None when calculating largest cluster.

getItem: Returns the position of the desired key

The Double Hash class's main difference is how a key is hashed, using x - H%x where x is a prime number and H is the original hash value. Double Hash has the same 6 function as above in addition to:

primeNumber: generates a prime number less than the size of half the array size to be used in the hashing value.

In the driver file I calculated the cluster sizes. To compare the cluster sizes I generated 5 different trials. For each trial I began with an array size of five and randomly assigned values into the array (number of values equaled the array size divided by two). After each time through I would add more random values (which would also automatically increase my array size through the doubleSize function). I would do this five times for a total array size of 80. I then recorded the results of the cluster size for each array size, seen in the tables below.

Trial: 1	Size of Array	Largest Cluster: Linear Probing	Largest Cluster: Double Hashing
	5	1	1
	10	2	2
	20	5	6
	40	6	5
	80	8	6

Trail: 2	Size of Array	Largest Cluster: Linear Probing	Largest Cluster: Double Hashing
	5	1	1
	10	3	4
	20	4	4
	40	4	5
	80	12	10

Trial: 3	Size of Array	Largest Cluster: Linear Probing	Largest Cluster: Double Hashing
	5	1	1
	10	3	3

Trial: 3	Size of Array	Largest Cluster: Linear Probing	Largest Cluster: Double Hashing
	20	3	2
	40	7	4
	80	21	15

Trail: 4	Size of Array	Largest Cluster: Linear Probing	Largest Cluster: Double Hashing
	5	1	1
	10	3	2
	20	3	3
	40	6	7
	80	7	7

Trail: 5	Size of Array	Largest Cluster: Linear Probing	Largest Cluster: Double Hashing
	5	1	1
	10	4	4
	20	4	3
	40	3	6
	80	17	17

Results:

Surprisingly, my results found that double hashing did not make much difference in cluster size. I calculated the results by adding up the cluster sizes that were equal to or greater than its counterpart, and also, the number of all those that were even. After totaling up the results for each type of hashing I found these results:

Linear Probing: 18/25

Double Hashing: 17/25

Number even: 12/25

Conclusion:

Double hashing barely came out as the smaller cluster size generator, making it not worth while to execute over linear probing. However, I do still think there is some value in double hashing. In order to verify if this is true, some possible future experiments might include a more complex hashing equation for double hashing, or perhaps expanding the size of the array to much greater numbers to see if that makes a more dramatic difference.