

Tsung Yu Ho

2911531

Dhwani Pandya

Due: 02/27/2019

1. For Experimental profiling. My Executive file is for user menu. For three different implementation they have their own class DoubleHash, Quadratic, Openhash.

Openhash is more special because it has Node and Linkedlist file as well. Since the size of the hashtable for this experimental profile is told. So I basically construct the three Hashtable in their own constructor. In my Print1 function, at first it insert 0.1m number and find 0.01m number for five times to take average time. And then insert 0.2m number and find 0.02 m number for five times to take average time. We will repeat the steps for 0.3m,0.4m and 0.5m as well.

2. For this lab I used random number generator to find some number to test. The range of the number is $\text{rand()} \% 5000000 + 1$. At first, for build I put random number generator under for loop to run it 0.1m times, and then found and not found I put it under for loop to run it 0.01m times . For the rest of them just following this rule.
3. and 4.

Performance (Open Hashing):

	Build	Found	Not Found
100000	32.797 ms	2.582 ms	7.713 ms
200000	37.193 ms	8.453 ms	23.854 ms
300000	45.73 ms	17.369 ms	46.422 ms
400000	56.066 ms	29.849 ms	76.473 ms
500000	65.495 ms	46.224 ms	113.505 ms

Performance (Quadratic Probing):

	Build	Found	Not Found
100000	22.173 ms	48.356 ms	122.199 ms
200000	45.192 ms	53.091 ms	140.578 ms
300000	82.503 ms	58.85 ms	163.666 ms
400000	103.058 ms	67.191 ms	195.667 ms
500000	127.932 ms	77.582 ms	234.468 ms

Performance (Double Hashing):

	Build	Found	Not Found
100000	65.511 ms	80.051 ms	243.925 ms
200000	119.065 ms	84.891 ms	262.958 ms
300000	174.428 ms	92.097 ms	291.687 ms
400000	230.464 ms	102.062 ms	330.568 ms
500000	291.198 ms	114.237 ms	378.134 ms

5.

For all of the hashtable not found time is more than found which makes sense, because not found might go through $k=20$ 20 times to look for the value, found will stop when it finds the value. For build, I think open hashing is $O(n \log n)$ based on the data, and for the double hashing and quadratic probing I think is $O(n)$, since it goes through the whole for loop to build. For found and not found I think quadratic probing and double hashing is still $O(n)$ size it needs to go through for loop n 20 times to find the value. For open hashing complexity find and not found I felt like it's $O(n)$, since we can use mod function to find the array index immediately, and then go through the whole linkedlist to look for the value, it should be $I * n$ so it's $O(n)$. However, the time increases like $n \log n$, it's probably wrong analysis for myself.

6.

I think this a very complicated lab for me. Because I created two different arrays and two different functions for first and second option. It took me lots of time to debug. Since for last lab I used sequential search to find the value, it's not working for this lab at first. It will take lots of time to print it out. After I changed to hash function to find the position, everything works for me.