DATA HOMEWORK

I used Double Hashing, Linear probing, SSF, PAF, Hash Table, Array and SLL of data structures in my homework. The read words was thrown into the array and then the read words was processed at DH, LP, SSF and PAF Operations before it was thrown into the hash table. I used the SLL structure to hold the value variables. I made a resize when there was a certain load factor with prime number.The structure of my code has Abstract Data Type, Object Oriented Programming and Generic structures. There is also Try-Catches for error messages.

Load Factor	Hash Function	Collision Handling	Collision Count	Indexing Time	Avg. Search Time	Min. Search Time	Max. Search Time
α=50%	SSF	LP	2480074606	51.06 s	0.000397	0.0001 s	0.016 s
		DH	259511857	17.62 s	0.000388	0.0001 s	0.032 s
	PAF	LP	348159922	18.06 s	0.000249	0.0001 s	0.005 s
		DH	34021616	9.68 s	0.000688	0.0001 s	0.023 s
α=80%	SSF	LP	2753329129	54.51 s	0.000390	0.0001 s	0.004 s
		DH	288482481	18.57 s	0.000332	0.0001 s	0.015 s
	PAF	LP	524227173	21.63 s	0.000337	0.0001 s	0.057 s
		DH	71373780	12.43 s	0.000637	0.0001 s	0.014 s

Performance matrix

Polynomial Accumulation Function and Double Hashing have indexing time better than other operations with 0.5 load factor.

Simple Summation Function and Linear Probing have index timing worse than other operations with 0.8 load factor.

Sorting of operations;

1. *LoadFactor(0.5)-PAF-DH*
2. *LoadFactor(0.8)-PAF-DH*
3. *LoadFactor(0.5)-SSF-DH*
4. *LoadFactor(0.5)-PAF-LP*
5. *LoadFactor(0.8)-SSF-DH*
6. *LoadFactor(0.8)-PAF-LP*
7. *LoadFactor(0.5)-SSF-LP*
8. *LoadFactor(0.8)-SSF-LP*

Due to the fact that the value was "0.0 s", it was rounded to "0.0001 s" at Min Search Time.

My assumptions;

- *If the number of resizes is high, the time increases.*
- *If the number of collisions is high, the time increases.*
- *If the high prime number is large, the time increases.*
- *The prime number in the double hashing really effects time.*

Mesut Selim Serbes

2017510100