Eric Furukawa & Jimmy Zheng

PA 6 Report

Section A: Problem Statement

The goal of this programming assignment is to compare the performance of different hash functions and the different collision resolution implementations that each of the hash functions has to offer.

Section B: Experimental Setup

CPU: Intel Core i5

Clock Speed: 2.8GHZ

RAM: 8 GB

How many experiments were performed and averaged, to determine each point in your table?

We performed about 3 experiments to get the average for each of the values on our table.

Section C: Experimental Results (All times in microseconds)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Collision |  | Insert |  | Search |  |
| Strategy | Total time | Avg. time | Total # of collisions | Total time | Avg. time |
| Chaining | 33575 | 3.24 | 6882 | 338 | 0.225 |
| Linear | 5099 | 0.49 | 5693 | 212 | 0.141 |
| Quadratic | 4523 | 0.44 | 4901 | 323 | 0.215 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Hash |  | Insert |  | Search |  |
| Function | Total time | Avg. time | Total # of collisions | Total time | Avg. time |
| Simple | 3878 | 0.37 | 39693 | 353 | 0.235 |
| Prefix | 658 | 0.06 | 0 | 191 | 0.127 |
| Full-length | 4523 | 0.44 | 4901 | 323 | .215 |

Observations

For most of the observations we made they seem to be pretty close to our theoretical expectations expect for the total number of prefix collisions which ended up being zero which seemed kinda odd. Even a very efficient hash function should have a couple collisions for such a great pool of data. It seems to be justified by the time it takes which was about 6 times faster than Simple and about 7 times faster than the Full-length in the average insert time. It was basically twice as fast in the search compared to the other two hash functions. Other then that everything else seemed to be what we had theoretically predicted. The simple hash function by nature would be less efficient by both its naming convention and the simplicity of the formula. The results reflected that assumption. We were not surprised chaining ended up being very slow as linked lists were needed for every location in the table. Linear time also ended up being slightly longer than quadratic. As we discussed in class, the quadratic algorithm would be faster than the linear one due to less bunching of data. I was a little surprised the improvement wasn’t greater than it was.