Albert Wu

CptS 223

PA6 Report

1. Problem Statement

The goal of this exercise is to compare the performance of different collision resolution techniques, with the hash function held constant, and the performance of different hash functions, with the collision resolution technique held constant, in hash tables.

1. Experimental Setup

Tested on a MacBook Air with a 2.2 GHz Intel Core i7 processor.

Only one full experiment was used for the data points, but involved averaging across many insertions and queries. Additionally, the total collisions were constant across experiments. Times only changed slightly, with overall patterns unchanging.

1. Experimental Results (all times in milliseconds)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Insert |  |  | Search |  |
| Collision Strategy | Total time | Average time | Total Collisions | Total time | Average time |
| Chaining | 10076 | 0.970994 | 14525 | 347 | 0.231333 |
| Linear Probing | 7275 | 0.70107 | 5166 | 304 | 0.202667 |
| Quadratic Probing | 5974 | 0.575696 | 4166 | 275 | 0.183333 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Insert |  |  | Search |  |
| Collision Strategy | Total time | Average time | Total Collisions | Total time | Average time |
| Simple | 16360 | 1.57656 | 239016 | 317 | 0.211333 |
| Prefix | 15843 | 1.52674 | 249521 | 248 | 0.165333 |
| Full-Length | 15719 | 1.51479 | 238543 | 303 | 0.202 |

The observations made when varying the collision resolution techniques were as per my theoretical expectations. When counting the collisions using the chaining resolution technique, I used the length of the whole list at the index given by the hash function as the number of collisions. Thus, it is expected that chaining has the most collisions, as it makes use of the least of the hash table’s array size. Our probing implementations perform rehashing when the table gets too full. Additionally, quadratic probing resulting in fewer collisions than linear probing also matches the mathematically calculated theoretical bounds. The times for both insertion and querying are inversely proportional to the number of collisions, which is expected.

The observation made when varying the hashing functions matched expectations to some extent, but it was also surprising by how little the total number of collisions varied across hash functions. While, as expected, the full-length hash function resulted in fewer collisions than the prefix hash function, the full-length hash function only had a tiny amount of fewer collisions than the simple hash function. Times for insertion were similar, with the simple hashing function pulling ahead slightly. Both the simple and full-length hash functions resulted in significantly longer search times than the prefix hash function, suggesting that, in our particular case, the speed of the hashing function was more important to overall search performance than number of collisions.