

Testing two Databases

Analysis by Diego Gonzalez Reyes

The program tested two databases for a small firm that runs a credit card database. The databases are implemented in two different ways: Sorted Arrays and Hash Tables. The program tested databases of 20; 200; 2,000; 20,000; and 200,000 sizes.

The size of the Sorted Array is of a maximum 200,000 as it is the biggest database, but the program keeps counts of the accounts added to the database. For the Hash Table, the size is 9999 as it is the maximum number of possible values, the program calculates the size this way, 9999 % capacity. This database also keeps count of the accounts added.

This suggests that Hash Tables are faster than Binary Search. This is an expected result as Binary Search is implemented to look through the database in sections, splitting the search in half until it finds the element or it was not found. Whereas Hash Tables use hashes to find the element. This is direct access to the elements in the database. However, as the Size of the Database was increasing, Binary Search was improving its number. An astonishing example of this is running a test for a database of size 2,000,000 and the results followed the trend: The Sorted Array took 0.313 seconds to run through the test, and the hash table took 2.670 seconds. My belief is that if we were to increase the dataset Binary Search would improve its numbers. As for the Hash Table, the trend would be around the same time.

The following graph illustrates the difference between the two Databases. The Y-axis represents the time that it took to run the test for each database. The X-axis represents the Database sizes. The Blue line represents the Sorted Array that implements a Binary Search, and the red line represents the Hash table that implements that Hash Function. This graph does not include the 2,000,000 tests.

Φ	A	B	C	D	E
1	20	200	2000	20000	200000
2	3.121	8.985	22.476	11.141	9.469
3	2.872	2.982	3.54	3.324	3.889

