**Empirical Analysis of Linear Probing with Dynamic resizing**

**Ayesh Reddy Kothinti**

Hashing being the most optimal method to do searching on a given array, it has some limitation. This report will illustrate the clustering problem with linear probing technique. Performance analysis of linear probing techniques under different conditions like without rehash, rehash enabled hash table, incremental rehashing.

**Expectation:**

To find a break-even point for performance vs load factor in a hash table. determine the changes in performance with rehashing option and incremental rehashing.

**Observation:**

Obtaining results for various sizes of the hashtable and for different load factors its clearly visible that for approximately 70% of the hashtable the performance of search degrades exponentially.

**Statistical Observations:**

1. Linear Probing Without Hashing (With load factor step of 1%) for the array size of 100000, 500000, 1000000

A screenshot of a social media post

Description automatically generated

By Observing this line plot we can clearly see that the Search operation will increase exponentially after approximately 70% of the Load Factor.

1. Linear Probing With Hashing (With load factor step of 1% And Rehash at 75% ) for the array size of 100000, 500000, 1000000

A screenshot of a social media post

Description automatically generated

By Observing the above line plot, we can clearly see that the graph is linear its not increasing even after the 70% break point. This is due to the increase in the array size at the break point which is 75% and rehashing all the existing hashtable to the new resized array. But this is not optimal because rehashing the whole array at 75% mark will impact the put operation of the array

1. Linear Probing With Incremental Hashing in the background (With load factor step of 1% And Rehash at 75% ) for the array size of 100000, 500000, 1000000

A screenshot of a social media post

Description automatically generated

By Observing the above line plot, we can clearly see that the graph is linear its not increasing even after the 70% break point. This is due to the increase in the array size at the break point which is 75% and rehashing all the existing hashtable to the new resized array. This is the most optimal because the rehash happens in the background without impacting the put operation of the hashtable.

Plotting all observations in one graph.

Please note (NH states for No ReHashing, H states for ReHashing, BH states for Background Hashing)

A screenshot of a social media post

Description automatically generated

**Conclusion:**

From the observations it can be inferred that the breakpoint is 70 to 75%. The performance will decrease exponentially if the rehasing at 70% is not applied. By comparing all the three linear probing techniques the order of performance is Incremental ReHashing > Rehashing > No Rehashing.