**Assignment 5**

Empirical analysis of Linear Probing

There are 3 types of searching algorithm:

1. Linear Search
2. Binary Search
3. Hashing Search

Time complexity of Linear search in worst case is O(n). While Binary Search has runtime complexity of O(logn). Hashing Search algorithm takes constant time(O(1)) to search an element in an array.

Hashing search using linear probing calculates the index value using following function:

***HASH(X) = (X+i)%TABLE\_SIZE***

In case of collision, i is incremented evertytime.

Empirical analysis of Hashing Search algorithm using linear probing technique is as follow:

To find efficient performance of hashing search using linear probing, Average no of searching is calculated for different load factor. It is shown in below table:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Average search time to find element in hash table** | | | | | | | | | | | | |
|
|  | **Array size** | | | | | | | | | | | |
|  |  | **15** | **25** | **35** | **45** | **55** | **65** | **75** | **85** | **95** | **105** | **Average** |
| **Load factor** | **0.6** | 1.46 | 1.76 | 1.37 | 1.48 | 1.56 | 1.33 | 1.46 | 1.61 | 1.62 | 1.82 | 1.547 |
| **0.65** | 1.46 | 1.2 | 1.6 | 1.42 | 1.8 | 1.53 | 1.81 | 1.92 | 1.9 | 1.81 | 1.645 |
| **0.7** | 2.46 | 1.68 | 1.62 | 1.66 | 1.89 | 2.01 | 2.37 | 2.6 | 1.97 | 1.72 | 1.998 |
| **0.75** | 2.13 | 1.6 | 1.54 | 1.66 | 1.63 | 1.63 | 3.02 | 1.97 | 1.68 | 2 | 1.886 |
| **0.8** | 2.53 | 1.96 | 2.45 | 2.31 | 1.8 | 1.95 | 2.49 | 1.83 | 2.48 | 2.03 | 2.183 |
| **0.85** | 2.2 | 2.92 | 2.48 | 1.91 | 2.43 | 3.55 | 3.5 | 2.51 | 2.33 | 2.78 | 2.661 |
| **0.9** | 2.6 | 3.6 | 2.85 | 3.4 | 3.72 | 2.8 | 6.16 | 3.71 | 2.87 | 3.11 | 3.482 |
| **0.95** | 3.93 | 3.2 | 5.54 | 2.51 | 4.96 | 2.63 | 7.05 | 5.25 | 2.45 | 5.03 | 4.255 |
| **1** | 3.93 | 3.36 | 4 | 6.11 | 4.14 | 2.3 | 4.26 | 8.11 | 6.95 | 7.52 | 5.068 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Efficient load factor from above readings: | | | | | **0.75** |  |  |  |  |  |

The efficient load factor from above table is 0.75. It means whenever the hash table is more than 75% full it will resize the hash table twice and rehash all the values in newer hash table.

Other values of load factor can also be considered but hash search is a combination of linear search and binary search in runtime and space complexity perspective. So if we take value of load factor as 0.6 from above table keeping in mind that average searching is less compare to others, in this case it requires more hashing compare to the load factor value of 0.75. More space is requires every-time rehashing is done.

Graph for the above observations is as follow:

**Conclusion:**

From the above table and graph, Average searching is constant for load factor of 0.75 compare to other values of load factor. So for the load factor value of 0.75 hashing using linear search will give efficient performance in space and runtime perspective.