**LAB SESSION 11: Hashing and Searching**

**AIM**: To implement hashing and searching methods.

**PROBLEM DEFINITION:**

1. Implement linear search and binary search using an employee structure (structure should have id, name and address).
2. Implement hashing using following collision resolution techniques

* Linear Probing
* Quadratic probing
* Double hashing

1. Implement hashing using separate chaining.

**THEORY**:

**Linear search:**

In Linear Search Algorithm,

* Every element is considered as a potential match for the key and checked for the same.
* If any element is found equal to the key, the search is successful and the index of that element is returned.
* If no element is found equal to the key, the search yields “No match found

**Binary search:**

In this algorithm,

* Divide the search space into two halves by [finding the middle index “mid”](https://www.geeksforgeeks.org/problem-binary-search-implementations/).
* Compare the middle element of the search space with the key.
* If the key is found at middle element, the process is terminated.
* If the key is not found at middle element, choose which half will be used as the next search space.
  + If the key is smaller than the middle element, then the left side is used for next search.
  + If the key is larger than the middle element, then the right side is used for next search.
* This process is continued until the key is found or the total search space is exhausted

**Linear probing:**

In linear probing, the hash table is searched sequentially that starts from the original location of the hash. If in case the location that we get is already occupied, then we check for the next location.The main disadvantage of linear probing is primary clustering. Clustering increases the number of probes to search or insert a key and hence the search and insertion times of the records also increase.

## ****Quadratic Probing:****

Quadratic probing is an [open-addressing](https://www.geeksforgeeks.org/hashing-set-3-open-addressing/) scheme where we look for the i2‘th slot in the i’th iteration if the given hash value x collides in the hash table.

## **How Quadratic Probing is done?**

Let hash(x) be the slot index computed using the hash function.

* If the slot hash(x) % S is full, then we try (hash(x) + 1\*1) % S.
* If (hash(x) + 1\*1) % S is also full, then we try (hash(x) + 2\*2) % S.
* If (hash(x) + 2\*2) % S is also full, then we try (hash(x) + 3\*3) % S.
* This process is repeated for all the values of i until an empty slot is found

**Double hashing:**

Double hashing is a collision resolution technique used in hash tables. It works by using two hash functions to compute two different hash values for a given key. Double hashing has the ability to have a low collision rate, as it uses two hash functions to compute the hash value and the step size. This means that the probability of a collision occurring is lower than in other collision resolution techniques such as linear probing or quadratic probing.

However, double hashing has a few drawbacks. First, it requires the use of two hash functions, which can increase the computational complexity of the insertion and search operations. Second, it requires a good choice of hash functions to achieve good performance. If the hash functions are not well-designed, the collision rate may still be high.

**Advantages of Double hashing**

* The advantage of Double hashing is that it is one of the best forms of probing, producing a uniform distribution of records throughout a hash table.
* This technique does not yield any clusters.
* It is one of the effective methods for resolving collisions.

Double hashing can be done using:   
**(hash1(key) + i \* hash2(key)) % TABLE\_SIZE**   
Here hash1() and hash2() are hash functions and TABLE\_SIZE is size of hash table.

## ****Separate Chaining:****

The idea behind separate chaining is to implement the array as a linked list called a chain. Separate chaining is one of the most popular and commonly used techniques in order to handle collisions.

The **linked list**data structure is used to implement this technique.When multiple elements are hashed into the same slot index, then these elements are inserted into a singly-linked list which is known as a chain.

Here, all those elements that hash into the same slot index are inserted into a linked list. Now, we can use a key K to search in the linked list by just linearly traversing. If the intrinsic key for any entry is equal to K then it means that we have found our entry. If we have reached the end of the linked list and yet we haven’t found our entry then it means that the entry does not exist. Hence, the conclusion is that in separate chaining, if two different elements have the same hash value then we store both the elements in the same linked list one after the other.

**ALGORITHMS**:

* 1. Linear Search
  2. Binary Search
  3. Linear Probing
  4. Quadratic probing
  5. Double hashing

**PROGRAM AND OUTPUT:**

**CONCLUSION**: