DATA STRUCTURES AND ALGORITHMS

Hash and Hashing table

Instructor:
Zainab Malik

Introduction

- Introduction of hash and hashing table
- Basic Operations

1. Insertion
2. Searching
3. Deletion

- Collision and Collision Resolution
- Hash functions

1. Division Method
2. Multiplication Method
3. Mid-Square Method
4. Folding Method

Hash Table and Hashing

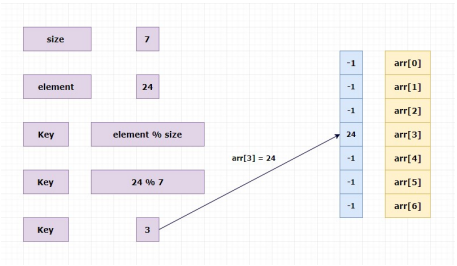
- **Hash table** is a type of data structure which is used for storing and accessing data very quickly.
- Insertion of data in a table is based on a key value. Hence every entry in the hash table is defined with some key.
- By using this key data can be searched in the hash table by few key comparisons. searching time is dependent upon the size of the hash table.
- Hashing is implemented in two steps:

1. An element is converted into an integer by using a hash function. This integer can be used as an index to store the original element.
2. The element is stored in the hash table where it can be quickly retrieved using hashed key.

- **Formula to calculate key is,**
- **key = element % size**

Insertion

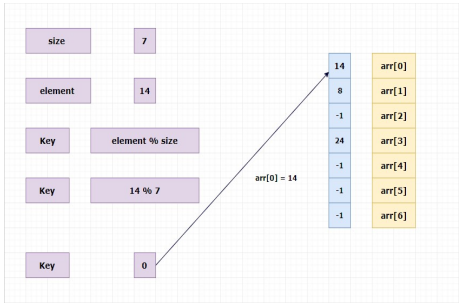
i)Insert 24



5

Example

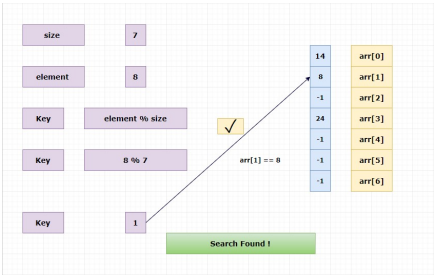
ii)Insert 14



6

Searching

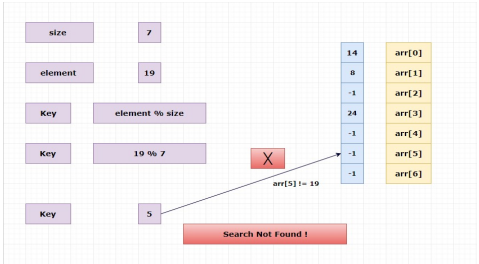
i)search 8



7

Example

ii)Search(19)

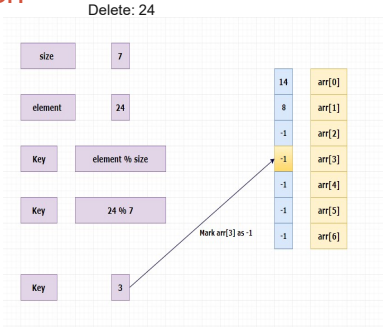


Incase we are searching a value that is not present in the array the process will halt if a null value is found.

8

Deletion

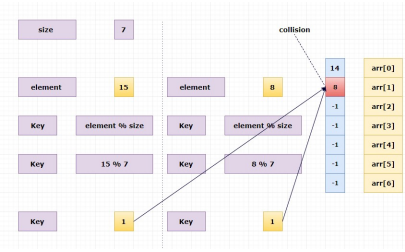
Delete: 24



9

Collision

Insert : 15
Key = element % key
Key = 15 % 7
Key = 1
But already arr[1] has element 8 !
Here, two or more different elements pointing to the same index under modulo size. This is called **collision**.



10

Collision Resolution

- A **collision** occurs when two keys hash to the same index in the array representing the hash table.
- Even if your hash table is larger than your dataset and you've chosen a good hash function, you need a plan for dealing with collisions if and when they arise.

Collision resolution technique:

- If there is a problem of collision occurs then it can be handled by apply some technique. These techniques are called as collision resolution techniques. There are generally four techniques which are described below.

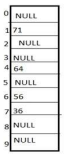
1) **Linear probing**
2) **Quadratic probing**
3) **Separate chaining**

11

Linear Probing

- **It is very easy and simple method to resolve or to handle the collision. In this collision can be solved by placing the second record linearly down, whenever the empty place is found.**

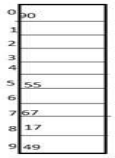
- **Example:** Let us consider a hash table of size 10 and hash function is defined as H(key)=key % table size.
- Consider that following keys are to be inserted that are 56,64,36,71. In this diagram we can see that 56 and 36 need to be placed at same bucket but by linear probing technique
- The records linearly placed downward if place is empty i.e. it can be see 36 is placed at index 7.



12

Quadratic probing

- In this method the hash function is defined by the H(key)=(H(key)+x*x)%table size.
- Let us consider we have to insert following elements that are:-67, 90,55,17,49.
- In this we can see if we insert 67, 90, and 55 it can be inserted easily but in case of 17 hash function is used in such a manner that :
- (17+0*0)%10=17 (when x=0 it provide the index value 7 only) by making the increment in value of x. let x =1 so (17+1*1)%10=8.
- in this case bucket 8 is empty hence we will place 17 at index 8.



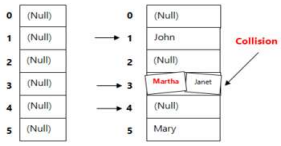
13

Separate chaining

Separate chaining

- To handle collisions, the hash table has a technique known as separate chaining. **Separate chaining** is defined as a method by which linked lists of values are built in association with each location within the hash table when a collision occurs.
- The figure shows incidences of collisions in different table locations.

Figure 1: Hash Table: Empty / Collision Occurrence

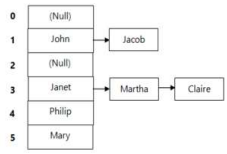


14

Separate chaining(continued)

- So, in place of the collision error which occurred in the figure(1), the table's cell now contains a linked list containing the string 'Janet' and 'Martha' as seen in this new figure.
- We can see in figure(2) how the subsequent strings are loaded using the separate chaining technique.

Figure 2: Separate Chaining



15

Hash functions

- A hash function is any function which is applied on a key by which it produces an integer, which can be used as an address of hash table.
- The values returned by a hash function are called hash values, hash codes, hash sums, or simply hashes
- To achieve a good hashing mechanism, It is important to have a good hash function with the following basic requirements:
- Easy to compute
- Uniform distribution: It should provide a uniform distribution across the hash table and should not result in clustering.
- Less collisions: Collisions occur when pairs of elements are mapped to the same hash value. These should be avoided.

16

Hash functions

The following are some of the Hash Functions −

Division Method

- This is the easiest method to create a hash function. The hash function can be described as −
- h(k) = k mod n
- Here, h(k) is the hash value obtained by dividing the key value k by size of hash table n using the remainder. It is best that n is a prime number as that makes sure the keys are distributed with more uniformity.
- An example of the Division Method is as follows −
  if the record 52,68,99,84 is to be placed in a hash table and let us take the table size is 10.

Then:

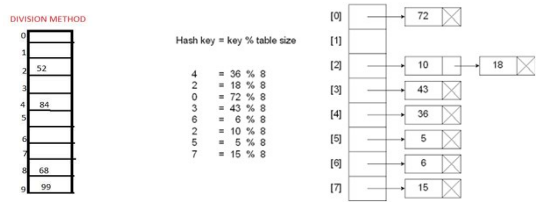
h(key)=record% table size.

2=52%10
8=68%10
9=99%10
4=84%10

17

Hash Functions(continued)

Division method



Inserting data while using linear probing

Inserting data while using Separate chaining technique.(collision avoiding technique)

18

Hash functions(continued)

- A disadvantage of the division method is that consecutive keys map to consecutive hash values in the hash table. This leads to a poor performance.

Multiplication Method

- The hash function used for the multiplication method is −
- h(k) = floor(n(kA mod 1))
- Here, k is the element and A can be any constant value between 0 and 1. Both k and A are multiplied and their fractional part is separated. This is then multiplied with n to get the hash value.
- An example of the Multiplication Method is as follows −

k=123
n=100
A=0.618033
h(123) = 100 (123 * 0.618033 mod 1)
= 100 (76.018059 mod 1)
= 100 (0.018059)
= 1(hash value)

19

Hash functions(continued)

- The hash value obtained is 1
- An advantage of the multiplication method is that it can work with any value of A, although some values are believed to be better than others.

Mid Square Method

- The mid square method is a very good hash function. It involves squaring the value of the key and then extracting the middle r digits as the hash value. The value of r can be decided according to the size of the hash table.
- An example of the Mid Square Method is as follows −
- Suppose the hash table has 100 memory locations. So **r=2** because two digits are required to map the key to memory location.

k = 50
k*k = 2500 (discarding 2 and 0)
h(50) = 50

- The hash value obtained is 50

20

Hash Functions(continued)

Mid-Square Method

K=	3205	7148	2345
K2=	10272025	51093904	5499025
H(K)=	72	93	99

Squaring key value and then using the digits as hash value that comes in the middle of the squared key(k).

21

Hash functions(continued)

Folding Method

- The folding method for constructing hash functions begins by dividing the item into equal-size pieces (the last piece may not be of equal size). These pieces are then added together to give the resulting hash value.
- Two types of folding are used, **shift folding** and **boundary folding**
- In **shift folding**, the parts are placed underneath each other and then processed (for example, by adding)
- Using a Social Security number, say 123-45-6789, we can divide it into three parts - 123, 456, and 789 – and add them to get 1368

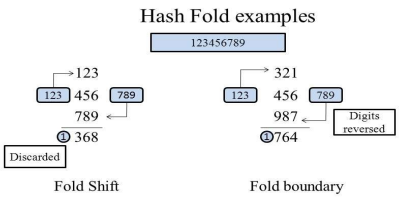
22

Hash functions(continued)

- With **boundary folding**, the key is visualized as being written on a piece of paper and folded on the boundaries between the parts.
- The result is that alternating parts of the key are reversed, so the Social Security number part would be 123, 654, 789, totaling 1566
- As can be seen, in both versions, the key is divided into even length parts of some fixed size, plus any leftover digits
- Then these are added together.

23

Hash functions(continued)



24

Applications

- Your Task