DSAL

Shrikrushna S Zirape
21286

PICT, PUNE

Assignment - 04

Title :

Consider a Telephone book database of N clients. make use of hashtable implimentation to quickly look up clients telephone number. make use of two collection handling technique and compare them using no of Collisions require to find telephone no

Objective :

- To understand Concept of hashing.
- To understand find record quickly using hash function.
- To understand concept of OOP

Outcome :-

- Learnt the OOP features.
- Understood and implimented Concept of hash table.

Theory :-

Hashtable are an efficient implimentation of keyed array of datastructure. a structure something known as associative array or map.

In hash table key is used to find an element instead of index number. Since hash table has to be coded using an indexed array, there has to be some way of transform key into number.

HashFunction:-

Generally hashing function returns value based on a key and the size of array of table.

Basic Operations:-

① Search
② Insert with Replacement
③ Insert without Replacement
④ Delete.
⑤ Print hast table.

DataItem :-

```
class Person {
    int number
    str name . bool status = false .
};
```

Hash Function.

Defines a hashing method to compute hash code of key

```
int hashcode (int key) {
    int k;
    For(int i=0 i<4 ;i++)
        k += (no%010)*i
        no = no/10
    return (k% size)
```

① Search:

```
def Search (~~self~~ int key):
    index= Self.hash function (key)
    if (self.directory[index].number == key)
        print ("found")
        return index.
    else: int count = 0
        (While (self.directory[index].NO != 0)
            if (self.directory [index].NO == key)
                print ("found")
                print (count)
                return index.
            index = (index+1) % 10 .
            count += 1
    print ("key Not present")
    return -1
```

② insert with replacement.

```
def insertion (~~self~~ name, no) :
    if (self.currsize == size)
        print ( "No space")
        return.
    index = self.hasfunction (no);
    if (self.directory [index]. number == 0) :
        self.directory [index].~~numbe~~ no
        self.directory [index].name=name;
        self.directory [index] . status= True;
        self. currsize += 1
        return .
```

```
elif (self directory [index] .staus==fulse)
    temp=index.
    index +=1
    while(index!=temp and directory [index].staus=false)
        self directory[index].number==0]:
            self.directory [index]number=no
            self.directory[index] .name=name.
            self.currsize +=1
            return
    index =(index+1)%10
```

③ Insert without Replacement.

```
def IWOR (self, name, no):
    index = self.hasfunction(no)
    if(self.directory [index].number ==0 ):
        self.directory [index]name = nane
        self.directory [index].number= no
        self.directory [index].Status =True.
        return.
    else:
        While(self.directory [index].number !=0):
            index =(index+1)%10
            self.directory [index].number=no
            self.directory [index].name=name.
            self.currsize +=1
            return.
```

```
4  def Delete (self, key )
       index = self.search (key
       if (index = = -1)
           print ("Not present")
           return
       else:
           temp= index.
           self.directory [index ]. number=0
           self directory [index]. name = ''
           self. directory [index]. status= false .
           index += 1
           while (self . hasfunction (self .directory [index ] .number == temp)
                   self.directory [temp] .number = self.director[index] .no
                   self.directory [temp]. name = self directory [index] .name
                   print ("successfully Replaced"
                   return .
               index = (index+1)% size
```

NO.	description	expected O/P	Result
	1. Insert HR	inserted	
	2. Insert LIOR	inserted	
	3. Delete.	inserted	
	4. search	0 abc 12345	Pass .
	5. display.	1 bcd 34567	
		2 lmn 78910	
	1 → name: abc	3	
	no : 123456	4	
	2 → name: bcd	5	
	no : 34567	6	
	4 → name: lmn	7	
	no : 78910	8	
	5 →	9	

Conclusion:-

In this way We have successfully implimented hash table for quickly look up.