

## Lab#14 Hashing

### 1. Data Structure

```
class Node { private: int data; Node *link; friend class Htable; };
```

### 2. ADT 함수 : 1) findkey, 2) insertKey 3) deleteKey 4) printable

### 3. 메뉴로 구성할 것 (1. Find, 2. Insert 3. Delete 4. Quit)

### 4. Hashing Function: division method ex) Print Table: table size 7

### 5. Algorithm : **Chaining** Method

#### ● Main Program:

```
do { Enter command (i,f,d,q): read(ch);
    if (ch != 'q') {
        print("Enter key "); read(inkey);
        switch (ch){
            case 'i': check=insertkey(inkey);
                if (check==false) print("Cannot insert key"); break;
            case 'f': check=findkey(inkey, index);
                if (check==false) print("key not found"); break;
            case 'd': check=deletekey(inkey);
                if (check==false) print("key not found "); break;
            default : print("Bad Command");
        }
    }
    printtable(); // insert, delete, find 후에 매번 테이블의 내용출력.
}while (ch == 'q');
```

#### ● FindKey

```
- Get index value for Key
- Get head node from HashTable
- If (p = NULL)
    return false;
else {
    Search the table for the Key // q= head;
                                // while ((q!=null)&&(q->data!=key))
                                // q= q->link
    If (q = null) return false;
    else return true
}
```

#### ● Insert Key

```
- Get Index for the Key
- Check = findkey(key);
- If (check == true) return false;
- Get head node for the Key
```

- if (head node = NULL) insert new node
- else find place to insert the key      // singly linked list 와 유사함
  
- Delete Key
  - Get index for the key
  - Get head node for the Key
  - if (head = key) delete head node & move head= head->next; (update head node)
  - else {
  - find node and delete the node for the key    // same as singly linked list

- Print Table
 

```
for(i= 0; i < maxsize; i++) {
    Print "Htable [i]"
    Get head node for "i"
    For (head; head!= NULL; head= head->link)
        Print "node"
```

- Testing    (아래 screen shot 과 같이 매번 HASH Table 출력)
  - ➔ Table size: 7

Insert 7➔insert 36 ➔ delete 15 ➔   delete 36 ➔ insert 22➔ find 25 ➔ find 22  
 ➔ insert 8 ➔ insert 17 ➔ insert 3

(최종 출력화면)

Htable[0]: 7  
 Htable[1]: 22➔ 8  
 Htable[2]:  
 Htable[3]: 17➔3  
 Htable[4]:  
 Htable[5]:  
 Htable[6]:

## ● Screen Shot

```
Enter command (i,f,d,q)i
Enter key 7
LinkT[0]: 7 LinkT[1]: LinkT[2]: LinkT[3]: LinkT[4]: LinkT[5]: LinkT[6]:
Enter command (i,f,d,q)i
Enter key 36
LinkT[0]: 7 LinkT[1]: 36 LinkT[2]: LinkT[3]: LinkT[4]: LinkT[5]: LinkT[6]:
Enter command (i,f,d,q)d
Enter key 15
key not found
LinkT[0]: 7 LinkT[1]: 36 LinkT[2]: LinkT[3]: LinkT[4]: LinkT[5]: LinkT[6]:
Enter command (i,f,d,q)d
Enter key 36
LinkT[0]: 7 LinkT[1]: LinkT[2]: LinkT[3]: LinkT[4]: LinkT[5]: LinkT[6]:
Enter command (i,f,d,q)i
Enter key 22
LinkT[0]: 7 LinkT[1]: 22 LinkT[2]: LinkT[3]: LinkT[4]: LinkT[5]: LinkT[6]:
Enter command (i,f,d,q)f
Enter key 25
key not found
LinkT[0]: 7 LinkT[1]: 22 LinkT[2]: LinkT[3]: LinkT[4]: LinkT[5]: LinkT[6]:
Enter command (i,f,d,q)f
Enter key 22
Key is found
LinkT[0]: 7 LinkT[1]: 22 LinkT[2]: LinkT[3]: LinkT[4]: LinkT[5]: LinkT[6]:
Enter command (i,f,d,q)i
Enter key 8
LinkT[0]: 7 LinkT[1]: 22 8 LinkT[2]: LinkT[3]: LinkT[4]: LinkT[5]: LinkT[6]:
Enter command (i,f,d,q)i
Enter key 17
LinkT[0]: 7 LinkT[1]: 22 8 LinkT[2]: LinkT[3]: 17 LinkT[4]: LinkT[5]: LinkT[6]:
Enter command (i,f,d,q)i
Enter key 3
LinkT[0]: 7 LinkT[1]: 22 8 LinkT[2]: LinkT[3]: 17 3 LinkT[4]: LinkT[5]: LinkT[6]:
;
Enter command (i,f,d,q)q
LinkT[0]: 7 LinkT[1]: 22 8 LinkT[2]: LinkT[3]: 17 3 LinkT[4]: LinkT[5]: LinkT[6]:
```