

# 자료구조응용

## 22. Hashing: Linear Probing (10점)

2020/6/1 (월)

1. [ Linear Probing ] 다음과 같은 해시함수와 Linear Probing을 사용하는 해시테이블에 대해 search, insert 함수를 작성하고 그 결과를 출력하는 프로그램을 작성하라.

<해싱조건>

입력파일(input.txt) :

acos atoi char define exp ceil cos float floor ctime

※ 입력문자열의 최대 크기는 10임을 가정한다.

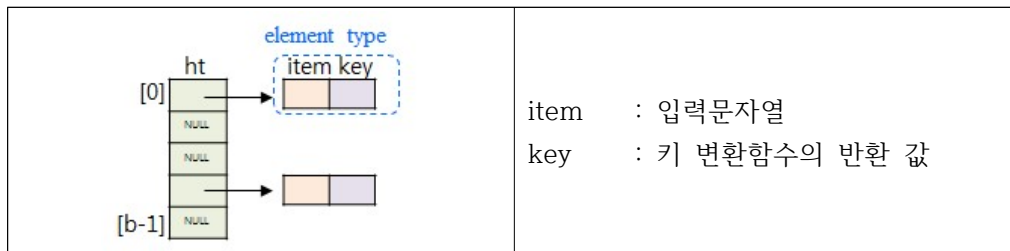
키 변환함수 : 각 입력문자열을 0 이상의 정수로 바꿈 (Program 8.1)

반환된 정수를 해싱함수의 입력 k로 사용

해싱함수(  $h(k)$  ) :  $k \% b$  연산 결과를 반환함

버킷 수 (b) : 11

슬롯 수 (s) : 1



<실행순서>

- ① 전역변수로 해시테이블(ht)을 초기화하고, MAX 기호상수를 11로 정의한다.
- ② 입력파일로부터 읽은 각 문자열과 key값은 element 타입의 구조체를 동적으로 할당받아 저장하고 그 주소를 해당 버킷에 저장한다.
- ③ 해시테이블을 최종 생성한 후 인덱스 순서대로 (item, key)를 출력한다.
- ④ 사용자로부터 문자열을 입력받아 탐색 후 (item, key, 비교횟수)를 출력한다.

<함수정의>

insert 함수

- search(Program 8.3) 함수를 참고하여 구현
- 해시테이블(ht)에 더 이상 추가할 수 없을 때는 적절한 메시지를 출력하고 종료한다.
- 중복된 key가 있을 경우에는 적절한 메시지를 출력하고 종료한다.

---

```

unsigned int stringToInt(char *key)
{
    /* simple additive approach to create a natural number
       that is within the integer range */
    int number = 0;
    while (*key)
        number += *key++;
    return number;
}

```

---

**Program 8.1:** Converting a string into a non-negative integer

---

```

element* search(int k)
{
    /* search the linear probing hash table ht (each bucket has
       exactly one slot) for k, if a pair with key k is found,
       return a pointer to this pair; otherwise, return NULL */
    int homeBucket, currentBucket;
    homeBucket = h(k);
    for (currentBucket = homeBucket; ht[currentBucket]
        && ht[currentBucket]->key != k;) {
        currentBucket = (currentBucket + 1) % b;
        /* treat the table as circular */
        if (currentBucket == homeBucket)
            return NULL; /* back to start point */
    }
    if (ht[currentBucket]->key == k)
        return ht[currentBucket];
    return NULL;
}

```

---

**Program 8.3:** Linear probing

---

※ 밑줄친 부분 : hash table에 있는 key 항목에 대한 탐색만 고려할 때 OK!  
 hash table에 없는 key항목에 대한 탐색까지 고려하면 어떻게 수정되어야 할까?

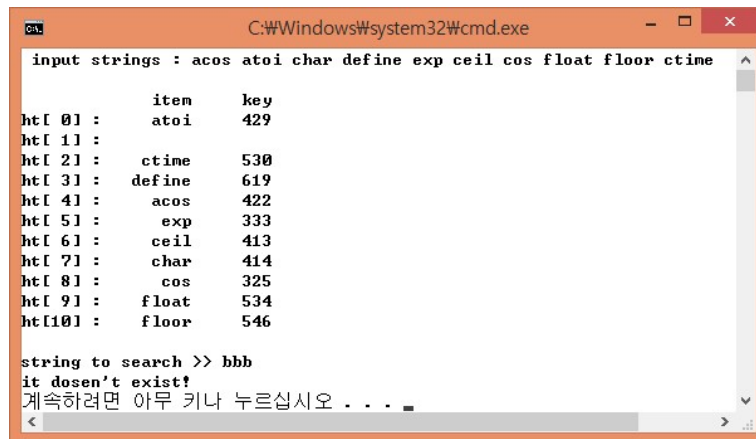
<실행결과>

```

C:\Windows\system32\cmd.exe
input strings : acos atoi char define exp ceil cos float floor ctime
               item    key
ht[ 0] :      atoi    429
ht[ 1] :
ht[ 2] :      ctime    530
ht[ 3] :      define   619
ht[ 4] :      acos     422
ht[ 5] :      exp      333
ht[ 6] :      ceil     413
ht[ 7] :      char     414
ht[ 8] :      cos      325
ht[ 9] :      float    534
ht[10] :      floor    546

string to search >> floor
item: floor, key : 546, the number of comparisons : 4
계속하려면 아무 키나 누르십시오 . . .

```



```
C:\Windows\system32\cmd.exe

input strings : acos atoi char define exp ceil cos float floor ctime

      item      key
ht[ 0] :      atoi      429
ht[ 1] :
ht[ 2] :      ctime      530
ht[ 3] :      define      619
ht[ 4] :      acos      422
ht[ 5] :      exp      333
ht[ 6] :      ceil      413
ht[ 7] :      char      414
ht[ 8] :      cos      325
ht[ 9] :      float      534
ht[10] :      floor      546

string to search >> bbb
it dosen't exist!
계속하려면 아무 키나 누르십시오 . . .
```

■ 제출 형식

- 솔루션/프로젝트 이름 : DS 22
- 소스파일 이름 : 1.c
- 각 소스파일에 주석처리  
“학번 이름”  
“본인은 이 소스파일을 복사 없이 직접 작성하였습니다.”
- 솔루션 정리 후 솔루션 폴더를 “학번.zip”으로 압축하여 과제 게시판에 제출

■ 주의

- 소스 복사로는 실력향상을 기대할 수 없습니다!!!
- 1차 마감 : 6월 2일(화) 자정
- 2차 마감 : 6월 3일(수) 자정(만점의 80%)