

자료구조응용

24. Static Hashing - Linear Probing (10점)

2025.12.3

1. [Linear Probing] 다음과 같은 해시함수와 Linear Probing을 사용하는 해시테이블에 대해 search, insert 함수를 작성하고 그 결과를 출력하는 프로그램을 작성하라. (10점)

<해싱 조건>

입력 파일(input.txt) :

```
acos atoi char define exp ceil cos float floor ctime
```

※ 입력문자열의 최대 크기는 10임을 가정한다.

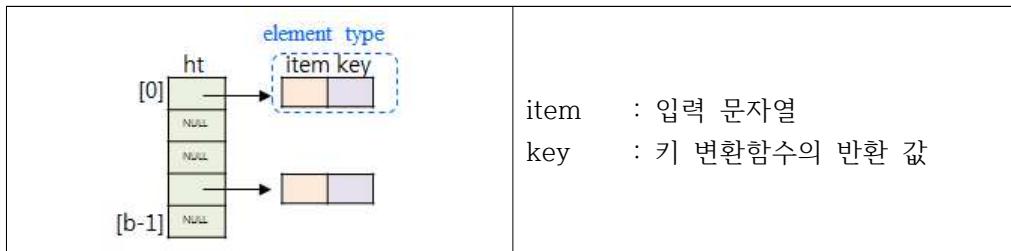
키 변환함수 : 각 입력 문자열을 0 이상의 정수로 바꿈 (Program 8.1)

반환된 정수를 해싱함수의 입력 k로 사용

해싱함수(h(k)) : $k \% b$ 연산 결과를 반환함

버킷 수 (b) : 11

슬롯 수 (s) : 1



<실행 순서>

- ① 전역변수로 해시테이블(ht)을 초기화하고, MAX 기호상수를 11로 정의한다.
- ② 입력 파일로부터 읽은 각 문자열과 key값은 element 타입의 구조체를 동적으로 할당받아 저장하고 그 주소를 해당 버킷에 저장한다.
- ③ 해싱 테이블을 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

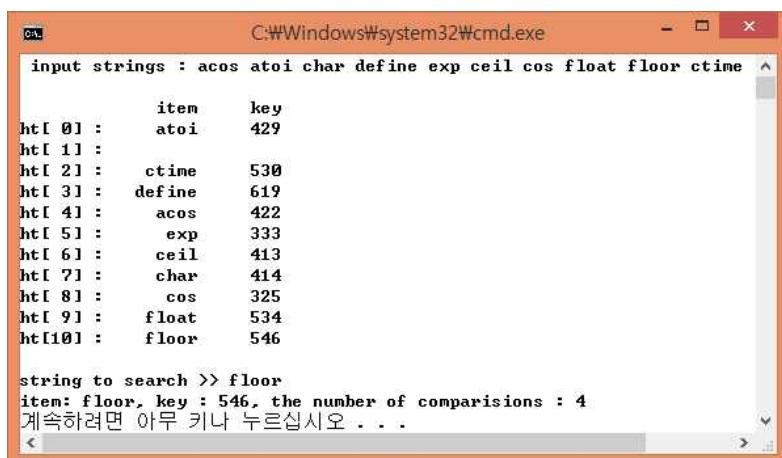
```

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] && ht[currentBucket]->key == k)
    return ht[currentBucket];
return NULL;
}

```

Program 8.3: Linear probing

<실행 결과>



	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 comparisions : 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 doesn't exist!
계속하려면 아무 키나 누르십시오 . . .

```

■ 제출 형식

- 솔루션 이름 : DS 24

- 프로젝트 이름 : 1

- 각 소스파일에 주석처리

“학번 이름”

“본인은 이 소스파일을 다른 사람의 소스를 복사하지 않고 직접 작성하였습니다.”

- 제출 파일

① 소스코드와 실행 결과가 보이도록 화면을 캡쳐한 보고서 파일(“학번.pdf”)

※ 한글 [파일 → pdf로 저장하기...] 메뉴 사용

② C 소스 파일을 하나의 디렉터리에 모아 압축한 파일 (“학번.zip”)

※ “학번.pdf”와 “학번.zip”을 하나로 압축하지 말고 별도 파일로 제출

■ 주의

- 소스 복사로는 실력 향상을 기대할 수 없습니다!!!

- 1차 마감 : 수업일 자정

- 2차 마감 : 수업 익일 자정(만점의 60%, 반올림)

- 문항 별로 1차 2차 나눠서 제출할 수 없으며, 최종 제출 시간에 따라 1차, 2차로 구분함

- 4시 40분에 제출 상황 체크함

- 완료된 과제를 제출하지 않고 일찍 퇴실한 학생은 결석 처리하겠음