

Hash – Open Addressing

[code] DS1901-HW08

Problem Open Addressing 방법 중 linear probing과 quadratic probing의 효율성의 차이를 비교하기 위해 주어진 key들을 차례대로 insert할 때, collision이 발생하는 빈도수의 차이를 리턴하는 int compare_hash() 함수를 구현하시오.

[제약조건]

- 함수의 prototype은 다음과 같다.
int compare_hash(int M, int *key, int N);
M : Hash table의 크기
key : Insert할 key값들의 배열 (단, 모든 key값은 자연수)
N : key 배열의 크기
리턴값 : (linear-probing으로 insert할 때 collision횟수)
 - (quadratic-probing으로 insert할 때 collision횟수)
- Overflow 등의 예외는 고려하지 않는다.
- 오른쪽과 같은 linear/quadratic hash function을 구현한다.
- 주의 제출하는 소스파일에는 main 함수는 포함하지 않는다.
(다음 페이지 예처럼 mycode.c만 제출한다.)

[Linear Probing]

$$h(key, i) = (key + i) \% M$$

[Quadratic Probing]

$$h(key, i) = (key + i + i * i) \% M$$

HASH-INSERT(T, k)

$i = 0$

repeat

$j = h(k, i)$

if $T[j] == \text{NIL}$

$T[j] = k$

return j

else $i = i + 1$

Collision

until $i == m$

error “hash table overflow”

Submission

Due: 6월 7일 (금) 23시 59분 59초

19950001@ubuntu:~/DS\$ submit DS1901-HW08 mycode.c

Self Test

Example)

M=20, N=15

Key= { 432025320 1470895010 1483763984 1982794092 776711393 1990779951 643445809
1416446822 1058311823 1159981259 668521661 1643697457 128634380 142633774 400658964 }

of collision = 7 (in linear-probing)

of collision = 5 (in quadratic-probing)

```
19950001@ubuntu:~/DS$ sftest DS1901-HW08 mycode.c
```

```
MSG> your source file was successfully compiled.
```

```
1 th. Testing 20 15 987 ---
```

```
Your answer is 2 : success.
```

```
2 th. Testing 53 40 987 ---
```

```
Your answer is 12 : success.
```

```
3 th. Testing 53 40 123 ---
```

```
Your answer is 32 : success.
```