제 8 장 해싱(Hashing)

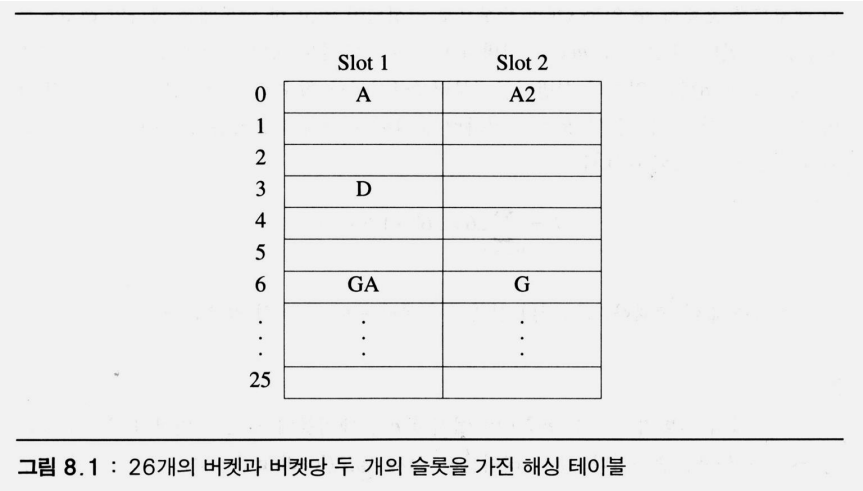
8.1 개요

- 사전(dictionary)의 예: 철자 검사기, 시소러스(thesaurus), 로더, 어셈블러, 컴파일러 등의 심볼 테이블
- 심볼 테이블은 (이름, 속성) 쌍의 집합으로 구성되어 있으며, 주로 이름에 대한 검색, 삽입, 삭제 등의 연산이 이루어 진다.
  - 특정 이름이 테이블 내에 존재하는지 검사
  - 주어진 이름의 속성을 검사
  - 주어진 이름의 속성을 변경
  - 새로운 이름과 그 속성을 삽입
  - 이름과 해당 속성의 삭제
- 심볼 테이블 구현 방법: 해싱 또는 트리

8.2 정적 해싱

(1) 해시 테이블

- 일정한 크기의 테이블(해싱 테이블)에 식별자를 저장하며, 식별자 x의 주소를 결정하기 위해 산술함수 h를 사용한다.
- 이때 사용되는 산술함수 h를 해시 함수(hash function)라 한다.
- h(x)는 x의 주소에 해당하며, 이 주소를 해시 주소 또는 본 주소(home address)라 한다.
- 해시 테이블을 ht라 할 때, b 개의 버켓 ht[0], ..., ht[b-1] 중 첨자를 h(x)로 갖는 버켓에 x와 속성들을 저장한다.
- 버켓은 한 개 이상의 슬롯(slot)으로 구성될 수 있다.



	Slot 1	Slot 2
0	A	A2
1		
2		
3	D	
4		
5		
6	GA	G
.	.	.
.	.	.
.	.	.
25		

그림 8.1 : 26개의 버켓과 버켓당 두 개의 슬롯을 가진 해싱 테이블

해싱의 용어와 성능

- 용어:

  적재 밀도(또는 적재 인수): $\alpha = n / (b \cdot s)$, n=식별자의 수

  b=버켓의 수, s=슬롯의 수

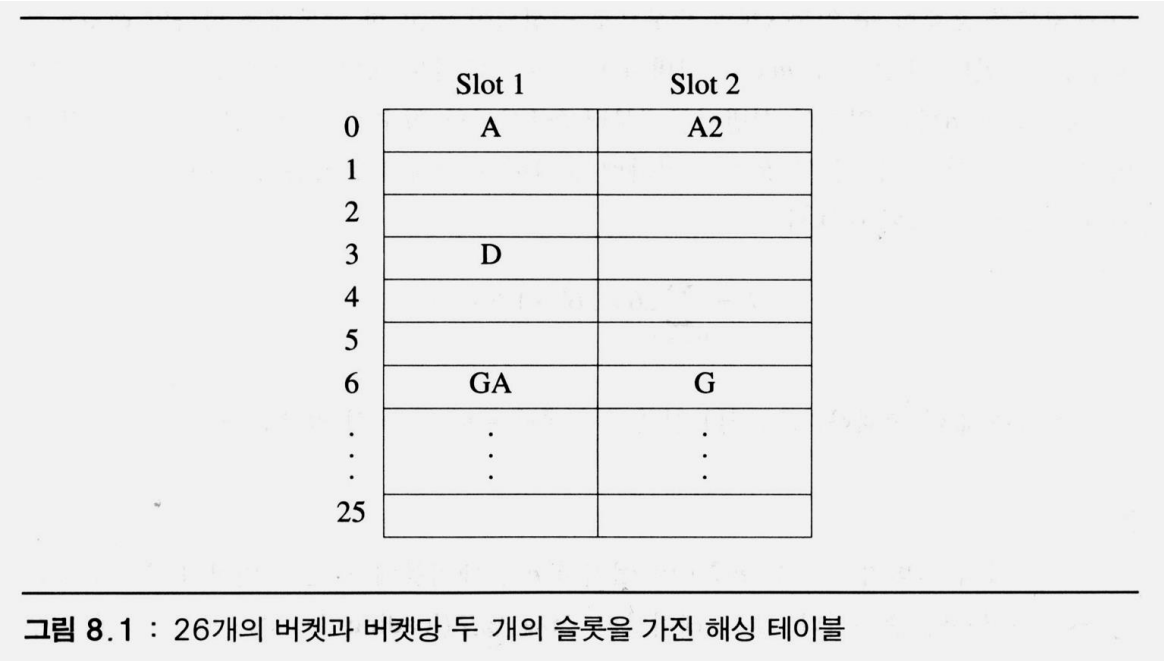
  충돌(collision): 두개 이상의 식별자가 같은 해시 주소를 가질 때

  이때 두 개의 식별자를 동거자(synonym)라 한다.

  오버플로우(overflow): 식별자를 저장할 더 이상의 슬롯이 없을 때
- 해싱의 성능은 적재 밀도, 해싱함수, 적절한 오버플로우 처리 방법에 따라 결정된다.

예제 8.1

- b=26, s=2,
- h(x)= x의 첫문자의 알파벳 순서(즉 A ~ Z를 0 ~ 25의 수로 맵핑)
  GA, D, A, G, L, A2, A1, A3, A4, E 는 각각
  6, 3, 0, 6, 11, 0, 0, 0, 0, 4 의 해시 주소를 가진다.



	Slot 1	Slot 2
0	A	A2
1		
2		
3	D	
4		
5		
6	GA	G
.	.	.
.	.	.
.	.	.
25		

그림 8.1 : 26개의 버켓과 버켓당 두 개의 슬롯을 가진 해싱 테이블

(2) 해싱 함수

- 해싱 함수는 계산이 쉽고 충돌이 적어야 한다.
- 균일 해싱함수(uniform hash function): 각 버켓에 맵핑될 확률이 같은 함수
- 4가지 해싱 함수

1) 중간 제곱**(mid-square)** 함수**: h_m**

식별자를 제곱한 후에 그 결과의 중간에 적당한 수의 비트를 취하여 버켓주소로 한다. 비트 수는 버켓의 크기에 따라 결정된다. (r 비트는 2^r 버켓 크기)

2) 제산**(division)** 함수

$h_D(x) = x \% D$

버켓의 범위는 **0**에서 **(D-1)**까지 이며**, D**는 **20**보다 작은 소수로 나누어지지 않으면 충분하다고 실험적으로 입증되었다.

해싱 함수(계속)

3) 접지**(folding)** 함수

식별자 **x**를 여러 부분으로 나눈 후 나누어진 부분들을 더해서 **x**에 대한 주소를 얻는다.

4) 숫자 분석 함수

식별자 **x**를 어떤 기수 **r**을 이용하여 숫자로 바꾼 다음, 필요한 부분의 숫자들을 추출하여 주소로 사용한다.

이 함수는 테이블에 있는 모든 식별자들을 미리 알고 있는 정적 파일과 같은 경우에 유용하다.

(3) 오버플로우 처리

- 개방 주소법(open addressing)과 폐쇄 주소법(closed addressing)이 있다. 폐쇄 주소법을 체인법(chaining)이라고 부른다.

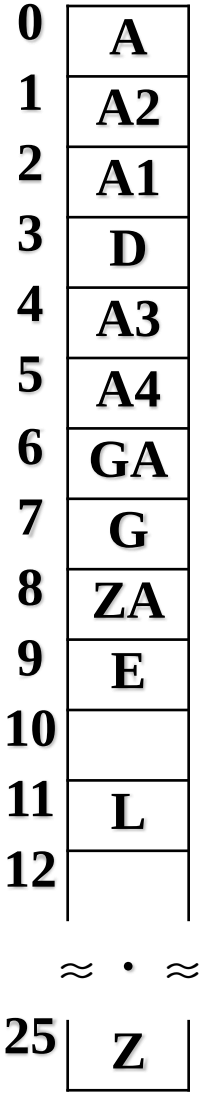
1) 개방 주소법

- 오버플로우가 발생했을 때 가장 가까운 빈 버켓을 찾아 식별자를 저장한다. 즉 f(x), f(x)+1, f(x)+2, ..., f(x)+j 의 순서로 찾아 간다.

  ➔ 선형조사법(linear probing) 또는 선형 개방 주소법

예제 8.3:

GA, D, A, G, L, A2, A1, A3, A4, Z, ZA, E



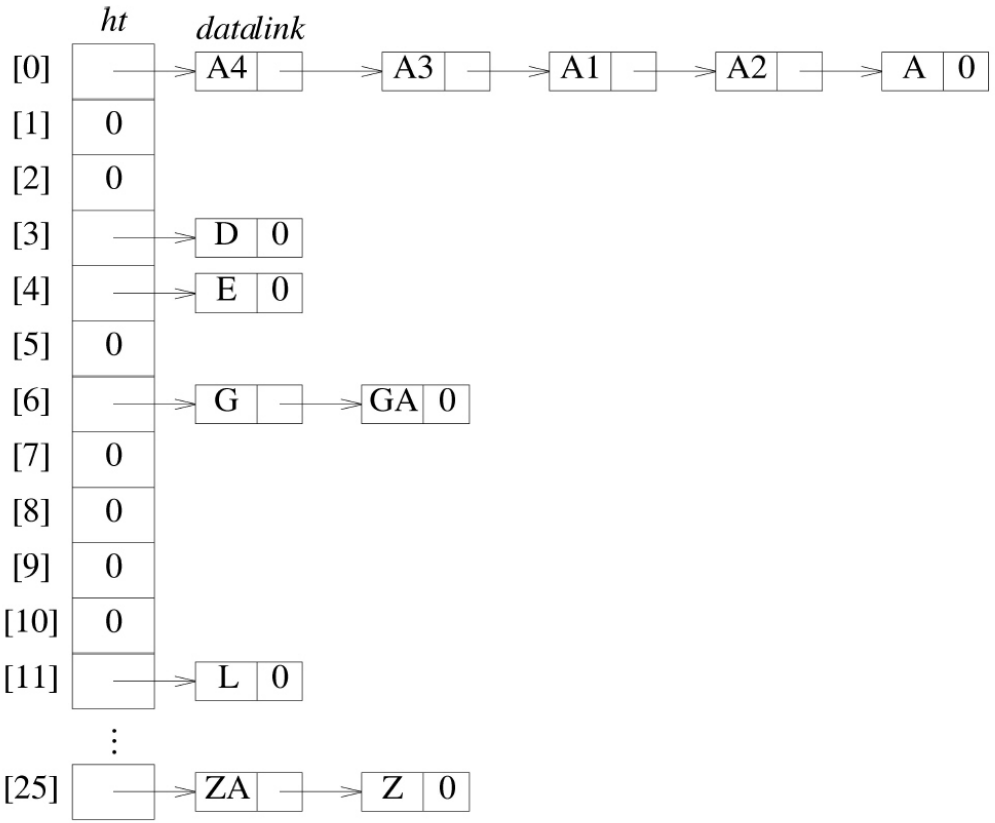
선형 조사법에 의한 해시 테이블
(26개의 버킷, 버킷당 하나의 슬롯)

이차 조사법

- 선형 조사법의 가장 큰 문제점은 군집화 현상이 발생하는 것이므로, 이를 완화시키기 위해 이차 조사법(quadratic probing)을 사용할 수 있다.
- $1 \le i \le (b-1)/2$ 에 대해, $f(x)$, $(f(x) + i^2)\%b$, $(f(x) - i^2)\%b$, 로 탐색한다.
- 군집화를 줄이기 위한 다른 방법은 여러 해싱 함수를 사용하는 것으로, 이를 재해싱(rehashing)이라 한다.

2) 체인법

- 오버플로우가 발생하면 연결리스트에 추가한다(폐쇄주소법).
- 예:



26버킷으로 된 해시 테이블.

그림 8.5 그림 8.3에 해당하는 해시 체인