

1 | 해상의 개념

1 해싱(Hashing)

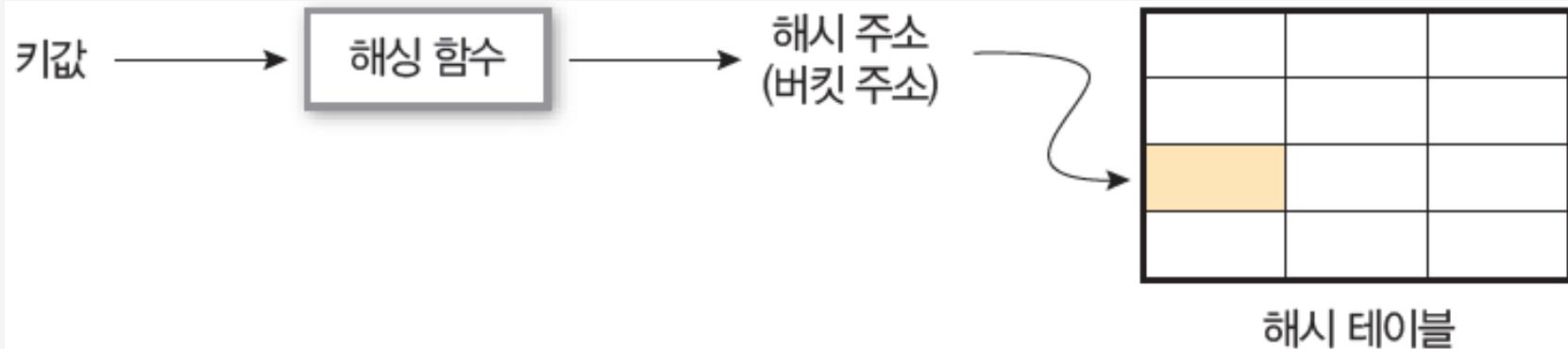
- ▶ 키값을 비교하여 찾는 검색 방법이 아니라
산술적인 연산을 이용하여 키가 있는 위치를 계산하여
바로 찾아가는 계산 검색 방식
- ▶ 검색 방법
 - 키 값에 대해서 해시 함수를 계산하여 주소를 구하고,
 - 구한 주소에 해당하는 해시 테이블로 바로 이동
 - 해당 주소에 찾는 항목이 있으면 검색 성공,
없으면 검색 실패

1 해싱(Hashing)

- ▶ 해시 함수(Hashing function)
 - 키 값을 원소의 위치로 변환하는 함수
- ▶ 해시 테이블(Hash table)
 - 해시 함수에 의해 계산된 주소의 위치에 항목을 저장한 표

2 해싱 검색 수행 방법

- ▶ 키 값에 대해서 해시 함수를 계산하여 주소를 구하고, 구한 주소에 해당하는 해시 테이블로 바로 이동하여 찾는 항목이 있으면 검색 성공이 되고 없으면 검색 실패가 됨

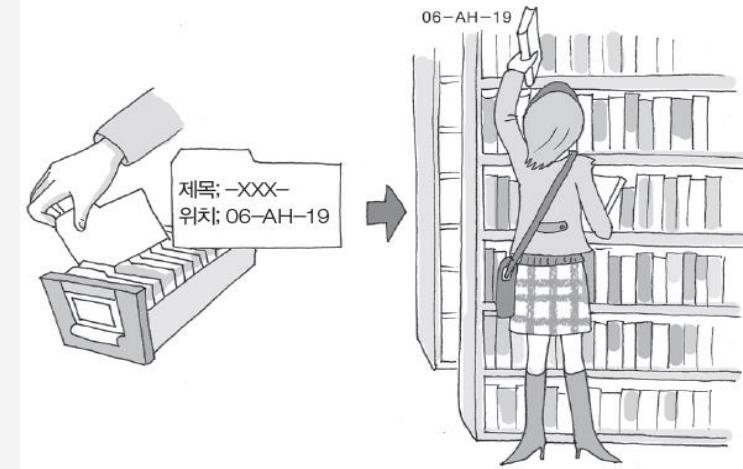


※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 해싱의 예

- ▶ 도서관에서의 도서 검색
 - 도서관에서 책을 찾을 때 첫 번째 책꽂이부터 책제목을 비교하면서 찾는 사람은 없을 것임, 도서관에 꽂힌 수 많은 책을 일일이 비교하면서 찾는다는 건 무모함

[위치 번호를 보고 책 찾기]



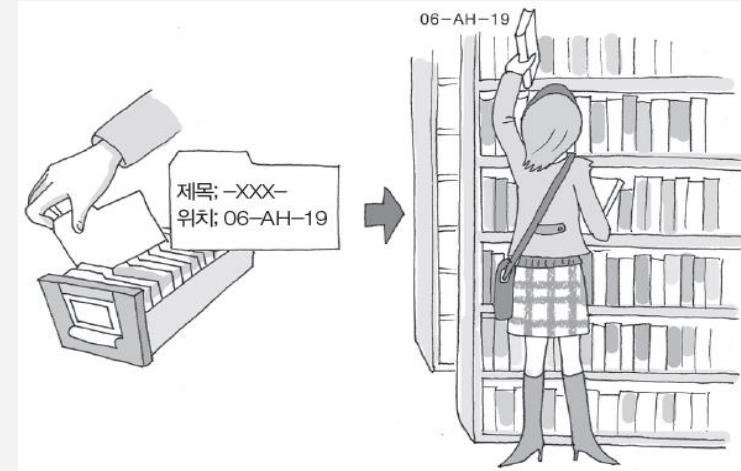
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 해싱의 예

▶ 도서관에서의 도서 검색

- 도서관에서는 도서 목록 카드를 이용하도록 권하는데, 원하는 책의 도서 목록 카드를 보면 책이 있는 위치가 적혀 있고, 그 위치로 바로 가서 책을 찾음, 전산화되어 있는 도서 시스템에서 제목을 입력하여 책의 위치를 찾는 것 역시 해싱의 예임

[위치 번호를 보고 책 찾기]



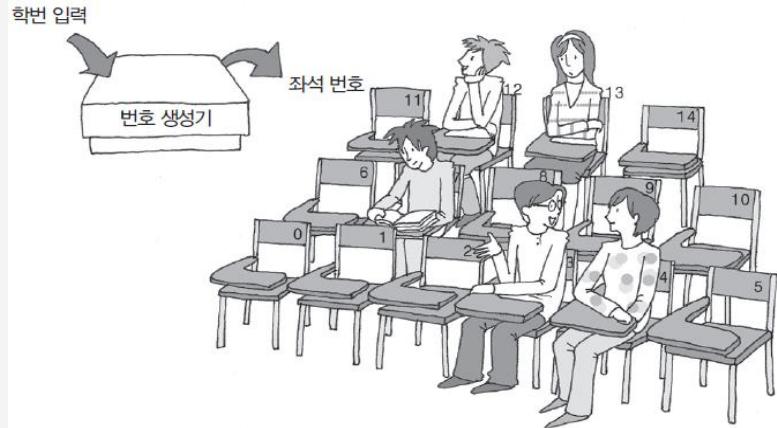
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 해싱의 예

▶ 강의실 좌석 배정

- 수업을 듣는 학생 열다섯 명에게 자리를 지정해 주기 위해 학번을 입력하면 0~14 자리 번호를 생성하는 번호 생성기를 사용한다고 생각해 보자

[학번에 따라 강의실 좌석 배정]



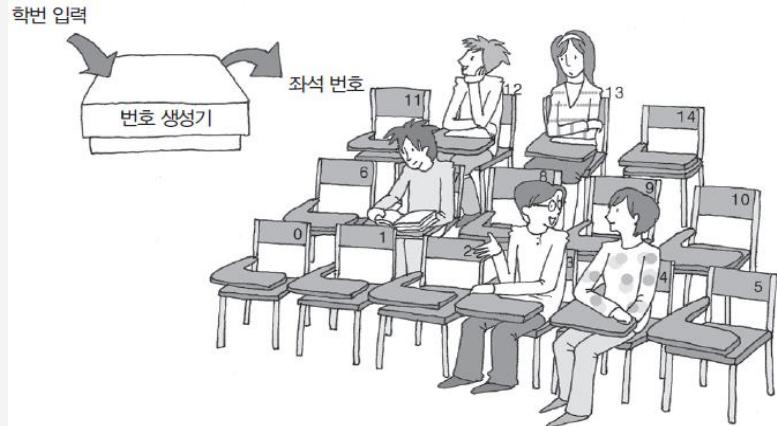
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 해싱의 예

▶ 강의실 좌석 배정

- 각자 자기 학번에 따라 생성된 번호의 자리에 앉으면 되고, 빈자리가 있으면 그 자리에 해당하는 학번의 학생은 결석이 됨, 여기에서 학번은 키 값이 되고, 자리번호를 생성하는 번호 생성기는 해시 함수가 됨

[학번에 따라 강의실 좌석 배정]



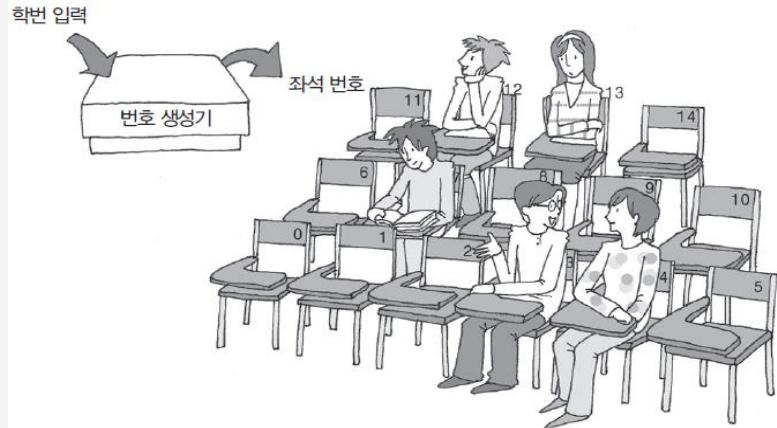
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 해싱의 예

▶ 강의실 좌석 배정

- 학생들이 앉은 지정자리가 있는 강의실은 해시 테이블이 되며, 강의실에 있는 학생은 검색 성공이 됨, 결석하여 자리에 없는 학생은 검색 실패가 됨

[학번에 따라 강의실 좌석 배정]



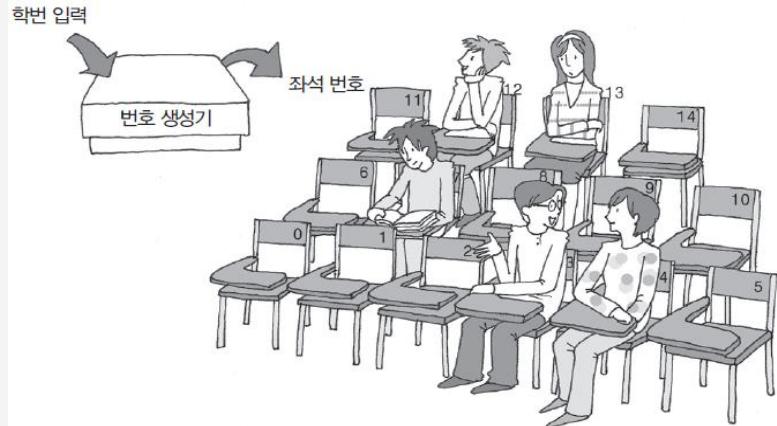
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 해싱의 예

▶ 강의실 좌석 배정

- 해시 테이블은 버킷 n 개와 슬롯 m 개로 구성하는데, 해시함수에 의해 계산된 주소가 버킷 주소가 됨,
이때 사용하는 해시 함수는 0과 $n-1$ 사이의 버킷 주소만 만들어야 함,
해시 함수로 알아낸 버킷에 키 값이 저장된 슬롯이 여러 개 있는 경우에는
순차 검색을 하여 해당 슬롯을 검색

[학번에 따라 강의실 좌석 배정]



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 해싱 관련 용어

- ▶ 모든 키 값이 고르게 사용되지는 않음, 많이 사용하는 키 값이 있을 수 있고 거의 사용되지 않는 키 값이 있을 수 있음, 따라서 키 값의 수 만큼 버킷을 많이 만들면 메모리 공간이 낭비됨
→ 해시 테이블에서 버킷 수를 줄이고 같은 버킷 안에 슬롯을 여러 개 두어 해시 함수로 만든 주소가 같은 키 값을 같은 버킷에 저장함

4 해싱 관련 용어

▶ 동거자(Synonym)

- 서로 다른 키 값을 가지지만 해싱 함수에 의해 같은 버킷에 저장된 키 값들
- 버킷 하나에 슬롯을 여러 개 두는 것은 좌석 배정 예에서 좌석을 1인용 좌석이 아닌 다인용으로 두는 것과 같음, 번호 생성기에서 같은 번호를 받은 학생은 같은 좌석에 함께 앉은 짹꿍이 되는데, 이들이 같은 버킷에 저장된 동거자임

[같은 좌석에 앉은 짹꿍]

짜꿍



1인용 좌석



3인용 좌석

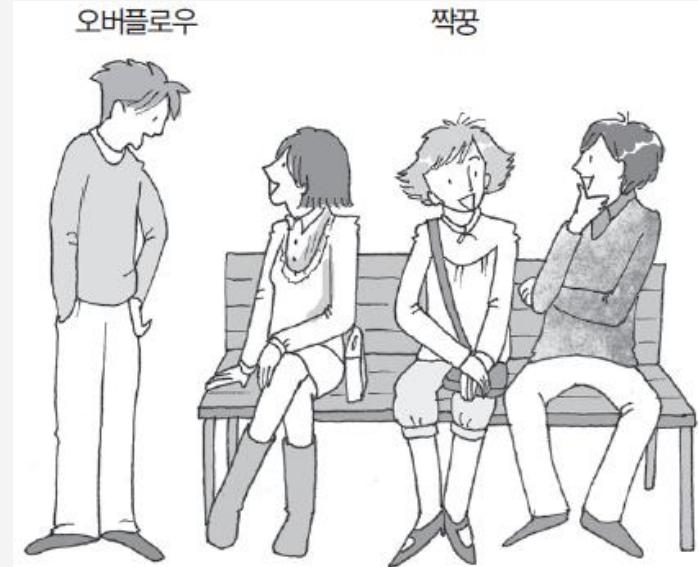
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 해싱 관련 용어

▶ 충돌(Collision)

- 서로 다른 키 값에 대해서 해싱 함수에 의해 주어진 버킷 주소가 같은 경우
- 충돌이 발생한 경우에 비어있는 슬롯에 동거자 관계로 키 값 저장하면 되지만 비어 있는 슬롯이 없으면 문제가 생김

[오버플로의 예]



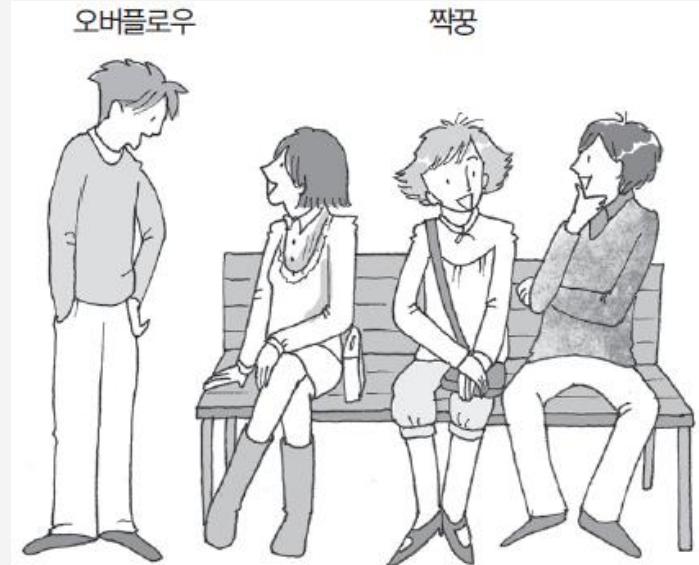
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

4 해싱 관련 용어

▶ 충돌(Collision)

- 포화 버킷 상태
: 버킷에 비어 있는 슬롯이 없는 상태
- 오버플로
: 버킷에 비어있는 슬롯이 없는
포화 버킷 상태에서 충돌이 발생하여
해당 버킷에 키 값을 저장할 수 없는 상태
- 3인용 좌석을 네 번째로 배정받은 학생은
앉을 자리가 없으므로 오버플로우가 됨

[오버플로의 예]



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

5 키 값 밀도와 적재 밀도

▶ 키 값 밀도

- 사용 가능한 전체 키 값들 중에서 현재 해시 테이블에 저장되어서 실제 사용되고 있는 키 값의 개수 정도
- 키값 밀도 = 실제 사용 중인 키값의 개수 / 사용 가능한 키값의 개수

5 키 값 밀도와 적재 밀도

▶ 적재 밀도

- 해시 테이블에 저장 가능한 키 값의 개수 중에서 현재 해시 테이블에 저장되어서 실제 사용되고 있는 키 값의 개수 정도
- 적재 밀도
 - = 실제 사용 중인 키값의 개수 / 해시 테이블에 저장 가능한 전체 키값의 개수
 - = 실제 사용 중인 키값의 개수 / (버킷 개수 × 슬롯 개수)

2 | 해시 함수

1 해시 함수

- ▶ 키 값을 원소의 위치로 변환하는 함수
- ▶ 해시 함수의 조건
 - 해시 함수는 계산이 쉬워야 함
 - 비교 검색 방법을 사용하여 키 값의 비교 연산을 수행하는 시간보다 해싱 함수를 사용하여 계산하는 시간이 빨라야 해싱 검색을 사용하는 의미가 됨

1 해시 함수

▶ 해시 함수의 조건

- 해시 함수는 충돌이 적어야 함
 - 충돌이 많이 발생한다는 것은 같은 버킷을 할당 받는 키 값이 많다는 것이므로 비어있는 버킷이 많은데도 어떤 버킷은 오버플로가 발생할 수 있는 상태가 되므로 좋은 해시 함수가 될 수 없음
 - 해시 테이블에 고르게 분포할 수 있도록 주소를 만들어야 함

2 중간 제곱 함수

- ▶ 키 값을 제곱한 결과 값에서 중간에 있는 적당한 비트를 주소로 사용
- ▶ 제곱한 값의 중간 비트들은 대개 키의 모든 값과 관련이 있기 때문에 서로 다른 키 값은 서로 다른 중간 제곱 함수 값을 갖게 됨

2 중간 제곱 함수

▶ 예)

키 값 00110101 10100111에 대한 해시 주소 구하기

00110101 10100111
X 00110101 10100111

000010110011**11101001**00101110001

해시 주소

3 제산(Division) 함수

- ▶ 함수는 나머지 연산자 mod(C에서의 %연산자)를 사용하는 방법
- ▶ 키 값 k 를 해시 테이블의 크기 M 으로 나눈 나머지를 해시 주소로 사용
- ▶ M 으로 나눈 나머지 값은 $0 \sim (M-1)$ 이 되므로 해시 테이블의 인덱스로 사용
- ▶ 해시 주소는 충돌 발생 없이 고르게 분포하도록 생성되어야 하므로 키 값을 나누는 해시 테이블의 크기 M 은 적당한 크기의 소수(Prime number) 사용

$$h(k) = k \bmod M$$

4 승산 함수

- ▶ 곱하기 연산을 사용하는 방법
- ▶ 키 값 k 와 정해진 실수 α 를 곱한 결과에서 소수점 이하 부분만을 테이블의 크기 M 과 곱하여 그 정수 값을 주소로 사용

5 접지 함수

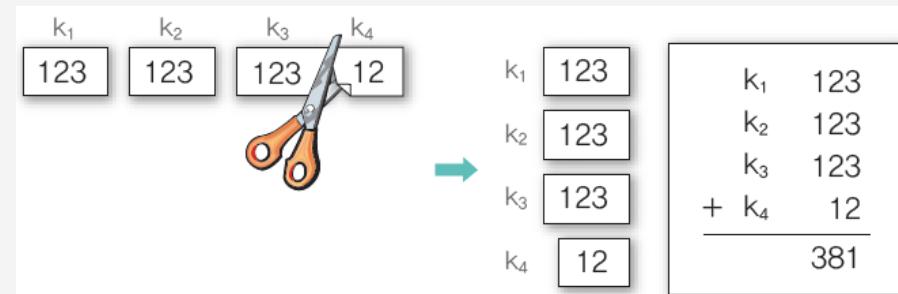
- ▶ 키의 비트 수가 해시 테이블 인덱스의 비트 수보다
큰 경우에 주로 사용

5 접지 함수

▶ 이동 접지 함수

- 각 분할 부분을 이동시켜서 오른쪽 끝자리가 일치하도록 맞추고 더하는 방법
- 예) 해시 테이블 인덱스가 3자리이고 키 값 k가 12312312312인 경우
 - 키 값 k를 3자리 크기로 분할하고 구한 키 값들을 오른쪽 끝자리가 일치하도록 옮겨서 더한 값인 381이 인덱스 테이블의 주소가 됨

[이동 접지 함수의 사용 예]



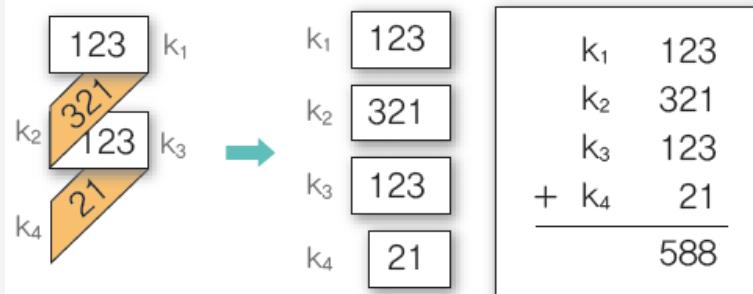
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

5 접지 함수

▶ 경계 접지 함수

- 분할된 각 경계를 기준으로 접으면서 서로 마주보도록 배치하고 더하는 방법
- 예) 해시 테이블 인덱스가 3자리이고 키 값 k가 12312312312인 경우
 - 키 값 k를 3자리 크기로 접으면서 분할하고 구한 키 값들을 오른쪽 끝자리가 일치하도록 옮겨서 더한 값인 588이 인덱스 테이블의 주소가 됨

[경계 접지 함수의 사용 예]



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

6 숫자 분석 함수

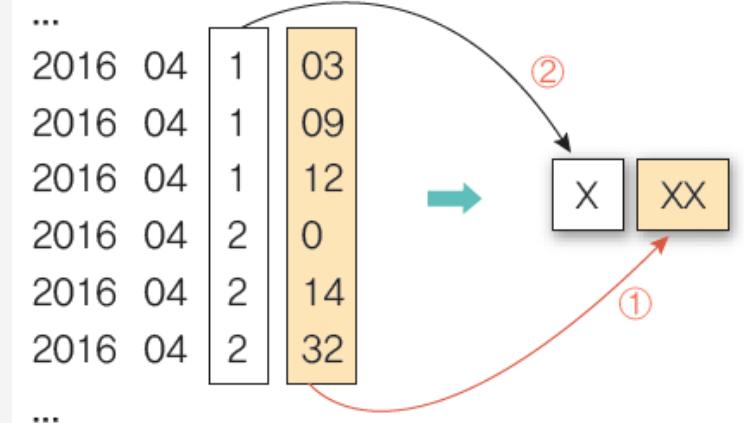
- ▶ 키 값을 이루고 있는 각 자릿수의 분포를 분석하여 해시 주소로 사용
- ▶ 각 키 값을 적절히 선택한 진수로 변환한 후에 각 자릿수의 분포를 분석하여 가장 편중된 분산을 가진 자릿수는 생략하고, 가장 고르게 분포된 자릿수부터 해시 테이블 주소의 자릿수만큼 차례로 뽑아서 만든 수를 역순으로 바꾸어서 주소로 사용

6 숫자 분석 함수

▶ 예) 키 값이 학번이고 해시 테이블 주소의 자릿수가 3자리인 경우

- 학번에서 2016은 입학 년도를
04는 학과, 1과 2는 주간과 야간,
마지막 두 자리는 학생번호를 나타냄
- 가장 고르게 분포된 번호는 학생번호,
다음으로 고르게 분포된 번호는
주간과 야간을 구분하는 숫자이므로
합쳐서 세 자리의 해시 테이블 주소를
만듦

[숫자 분석 함수의 사용 방법]



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

7

진법 변환 함수

- ▶ 키 값이 10진수가 아닌 다른 진수일 때,
10진수로 변환하고 해시 테이블 주소로 필요한
자릿수만큼만 하위자리의 수를 사용하는 방법

8

비트 추출 함수

- ▶ 해시 테이블의 크기가 2^k 일 때 키 값을 이진 비트로 놓고 임의의 위치에 있는 비트들을 추출하여 주소로 사용하는 방법
- ▶ 이 방법에서는 충돌이 발생할 가능성이 많으므로 테이블의 일부에 주소가 편중되지 않도록 키 값들의 비트들을 미리 분석하여 사용해야 함

3 | 해싱에서 오버플로우 처리 방법

1 해싱에서 오버플로우를 처리 방법

- ▶ 해시 테이블과 해시 함수를 잘 선택하여 오버플로가 발생하지 않도록 해야 하지만 오버플로가 발생하더라도 문제를 효율적으로 해결해야 함
- ▶ 오버플로를 해결하는 방법에는 선형 개방 주소법과 체이닝 방법이 있음

1 해싱에서 오버플로우 처리 방법

- ▶ 선형 개방 주소법 (선형 조사법 Linear probing)
 - 해싱 함수로 구한 버킷에 빈 슬롯이 없어서 오버플로우가 발생하면, 그 다음 버킷에 빈 슬롯이 있는지 조사
 - 빈 슬롯이 있으면 - 키 값을 저장
 - 빈 슬롯이 없으면 - 다시 그 다음 버킷을 조사
 - 이런 과정을 되풀이하면서 해시 테이블 내에 비어있는 슬롯을 순차적으로 찾아서 사용하여 오버플로우 문제를 처리하는 방법

1 해싱에서 오버플로를 처리 방법

3 | 해싱에서 오버플로우 처리 방법

1 해싱에서 오버플로우 처리 방법

▶ 선형 개방 주소법을 이용한 오버플로 처리 예

- ① $h(45) = 45 \bmod 5 = 0$
→ 해시 테이블 0번에
키 값 45 저장



※출처: 이지영(2016).

IT CookBook, C로 배우는 쉬운
자료구조(개정3판). 한빛미디어

3 | 해싱에서 오버플로우 처리 방법

1 해싱에서 오버플로우 처리 방법

▶ 선형 개방 주소법을 이용한 오버플로 처리 예

② $h(9) = 9 \bmod 5 = 4$
→ 해시 테이블 4번에
키값 9 저장



※출처: 이지영(2016).

IT CookBook, C로 배우는 쉬운
자료구조(개정3판). 한빛미디어

3 | 해싱에서 오버플로우 처리 방법

1 해싱에서 오버플로우를 처리 방법

▶ 선형 개방 주소법을 이용한 오버플로우 처리 예

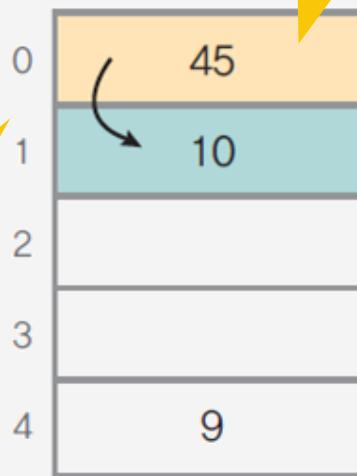
③ $h(10) = 10 \bmod 5 = 0$

→ 충돌 발생!

→ 다음 버킷 중에서
비어 있는 버킷 1에
키값 10을 저장

다음 버킷
조사하기

충돌 발생!



※출처: 이지영(2016).

IT CookBook, C로 배우는 쉬운
자료구조(개정3판). 한빛미디어

3 | 해싱에서 오버플로우 처리 방법

1 해싱에서 오버플로우를 처리 방법

▶ 선형 개방 주소법을 이용한 오버플로우 처리 예

④ $h(96) = 96 \bmod 5 = 1$

→ 충돌 발생!
→ 다음 버킷 중에서
비어 있는 버킷 2에
키값 96을 저장

다음 버킷
조사하기



※출처: 이지영(2016).

IT CookBook, C로 배우는 쉬운
자료구조(개정3판). 한빛미디어

3 | 해싱에서 오버플로우 처리 방법

1 해싱에서 오버플로우를 처리 방법

▶ 선형 개방 주소법을 이용한 오버플로우 처리 예

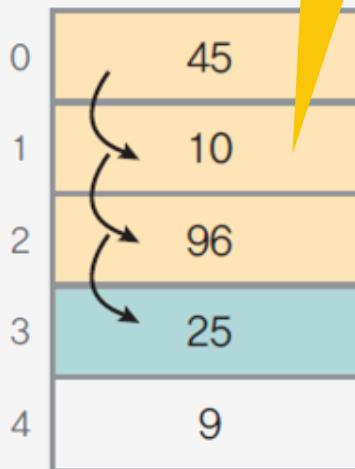
⑤ $h(25) = 25 \bmod 5 = 0$

→ 충돌 발생!

→ 다음 버킷 중에서
비어 있는 버킷 3에
키값 25를 저장

다음 버킷
조사하기

충돌 발생!



※출처: 이지영(2016).

IT CookBook, C로 배우는 쉬운
자료구조(개정3판). 한빛미디어

2 체이닝

- ▶ 해시 테이블의 구조를 변경하여 각 버킷에 하나 이상의 키 값을 저장할 수 있도록 하는 방법
- ▶ 버킷에 슬롯을 동적으로 삽입하고 삭제하기 위해서 연결 리스트 사용
 - 각 버킷에 대한 헤드노드를 1차원 배열로 만들고 각 버킷에 대한 헤드노드는 슬롯들을 연결 리스트로 가지고 있어서 슬롯의 삽입이나 삭제 연산을 쉽게 수행 가능
 - 버킷 내에서 원하는 슬롯을 검색하기 위해서는 버킷의 연결 리스트를 선형 검색

2 체이닝

▶ 체이닝을 이용한 오버플로 처리 예

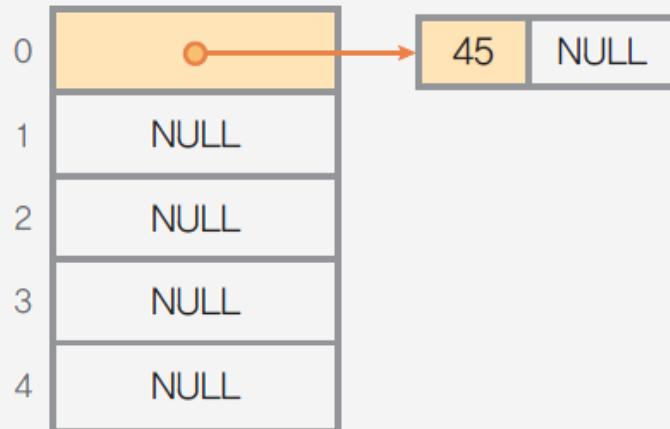
- 해시 테이블의 크기 : 5
- 해시 함수 : 제산함수 사용
 해시 함수 $h(k) = k \bmod 5$
- 저장할 키 값 : {45, 9, 10, 96, 25}

3 | 해싱에서 오버플로우 처리 방법

2 체이닝

▶ 체이닝을 이용한 오버플로우 처리 예

- ① $h(45) = 45 \bmod 5 = 0 \rightarrow$ 해시 테이블 0번에 노드를 삽입, 키값 45를 저장



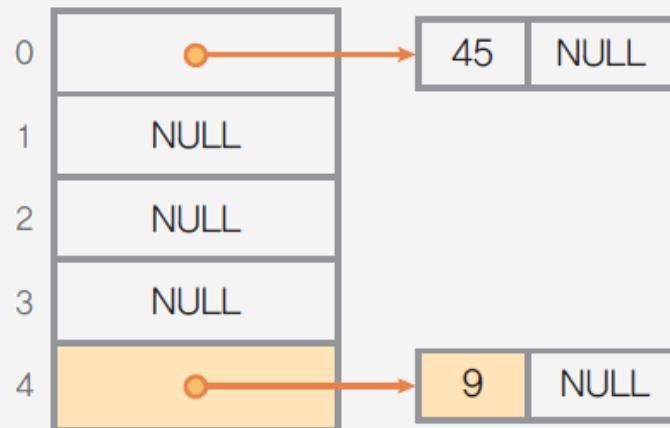
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 | 해싱에서 오버플로우 처리 방법

2 체이닝

▶ 체이닝을 이용한 오버플로우 처리 예

- ② $h(9) = 9 \bmod 5 = 4 \rightarrow$ 해시 테이블 4번에 노드를 삽입, 키값 9를 저장



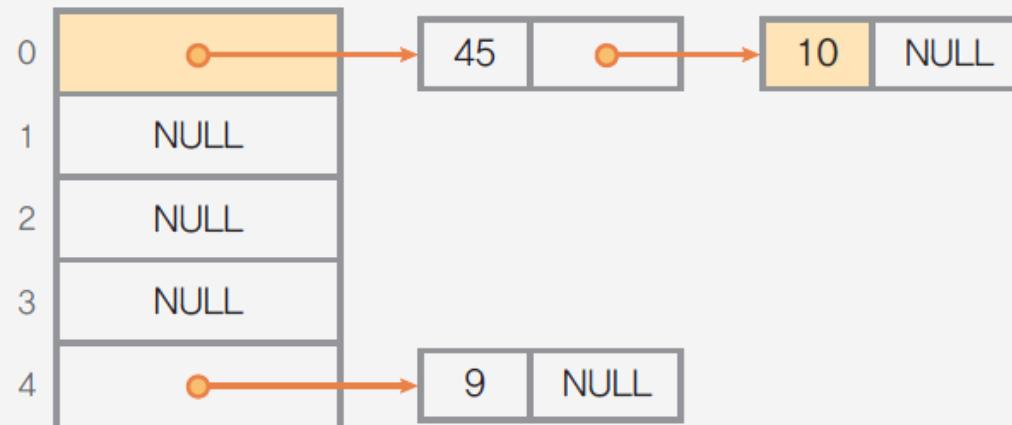
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 | 해싱에서 오버플로우 처리 방법

2 체이닝

▶ 체이닝을 이용한 오버플로우 처리 예

- ③ $h(10) = 10 \bmod 5 = 0 \rightarrow$ 해시 테이블 0번에 노드를 삽입, 키값 10을 저장



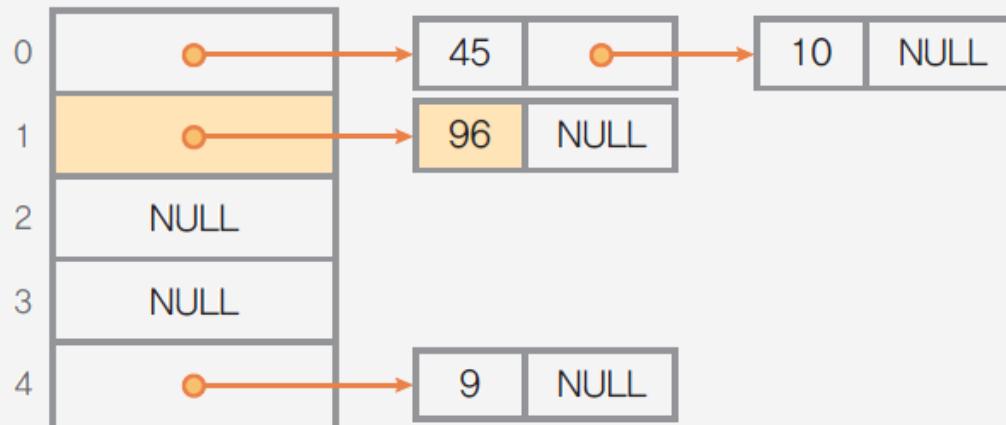
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어

3 | 해싱에서 오버플로우 처리 방법

2 체이닝

▶ 체이닝을 이용한 오버플로우 처리 예

- ④ $h(96) = 96 \bmod 5 = 1 \rightarrow$ 해시 테이블 1번에 노드를 삽입, 키값 96을 저장



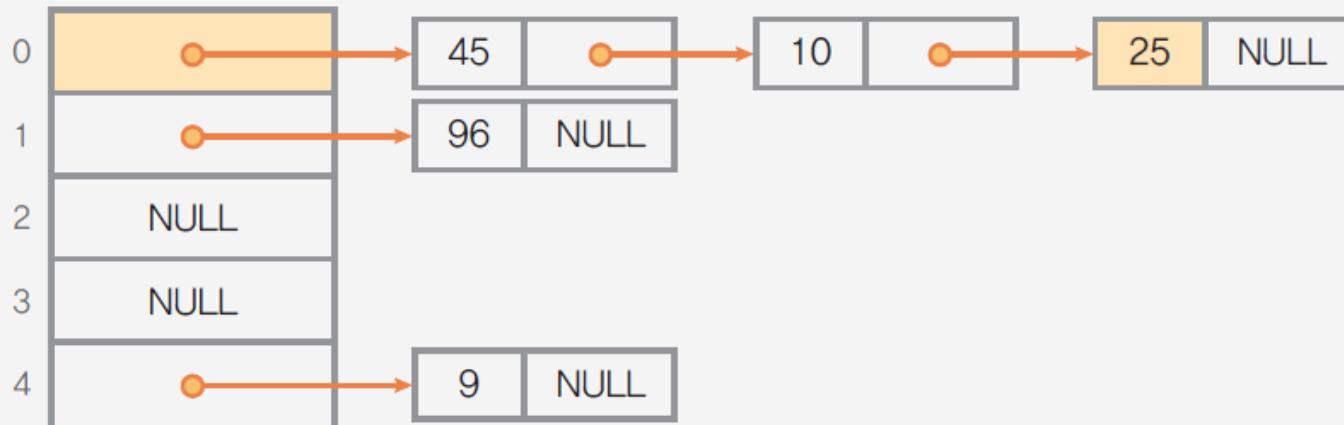
※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어



2 체이닝

▶ 체이닝을 이용한 오버플로우 처리 예

⑤ $h(25) = 25 \bmod 5 = 0 \rightarrow$ 해시 테이블 0번에 노드를 삽입, 키값 25를 저장



※출처: 이지영(2016). IT CookBook, C로 배우는 쉬운 자료구조(개정3판). 한빛미디어