자료구조 (Data Structure)
11주차: 해시 테이블

지금까지 한 내용 정리

- 일차원 배열로 표현 가능한 구조
  - 리스트, 스택, 큐, 이진 트리, 힙
  - 데이터 추가 / 삭제

- 이차원 배열로 표현 가능한 구조
  - 그래프
  - 단절점, 최단 경로, 최단 길이 연결

이번 시간 목차

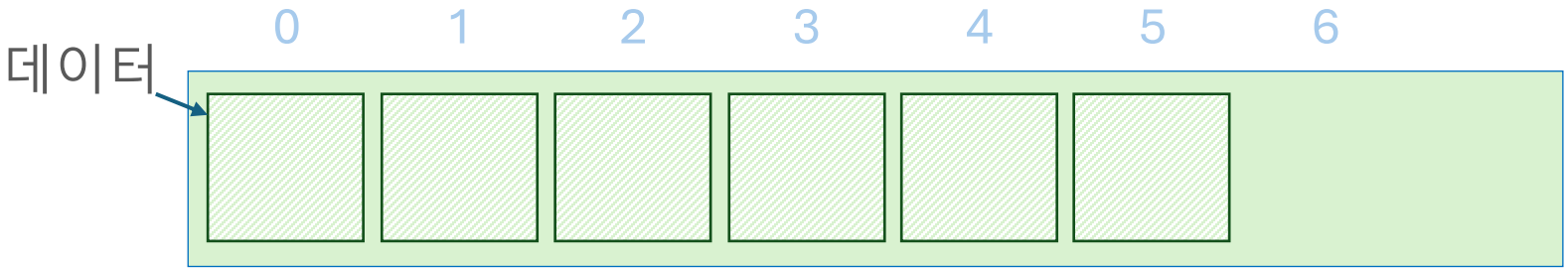
- 검색을 쉽게 하는 방법
- 검색을 쉽고 효율적으로 하는 방법
- 최악의 경우 고려하기

검색을 하는 방법

- 건물, 방번호, 내선번호를 저장하는 방법은?

검색을 하는 방법

- 내선번호로부터 건물과 방번호를 찾으려면?

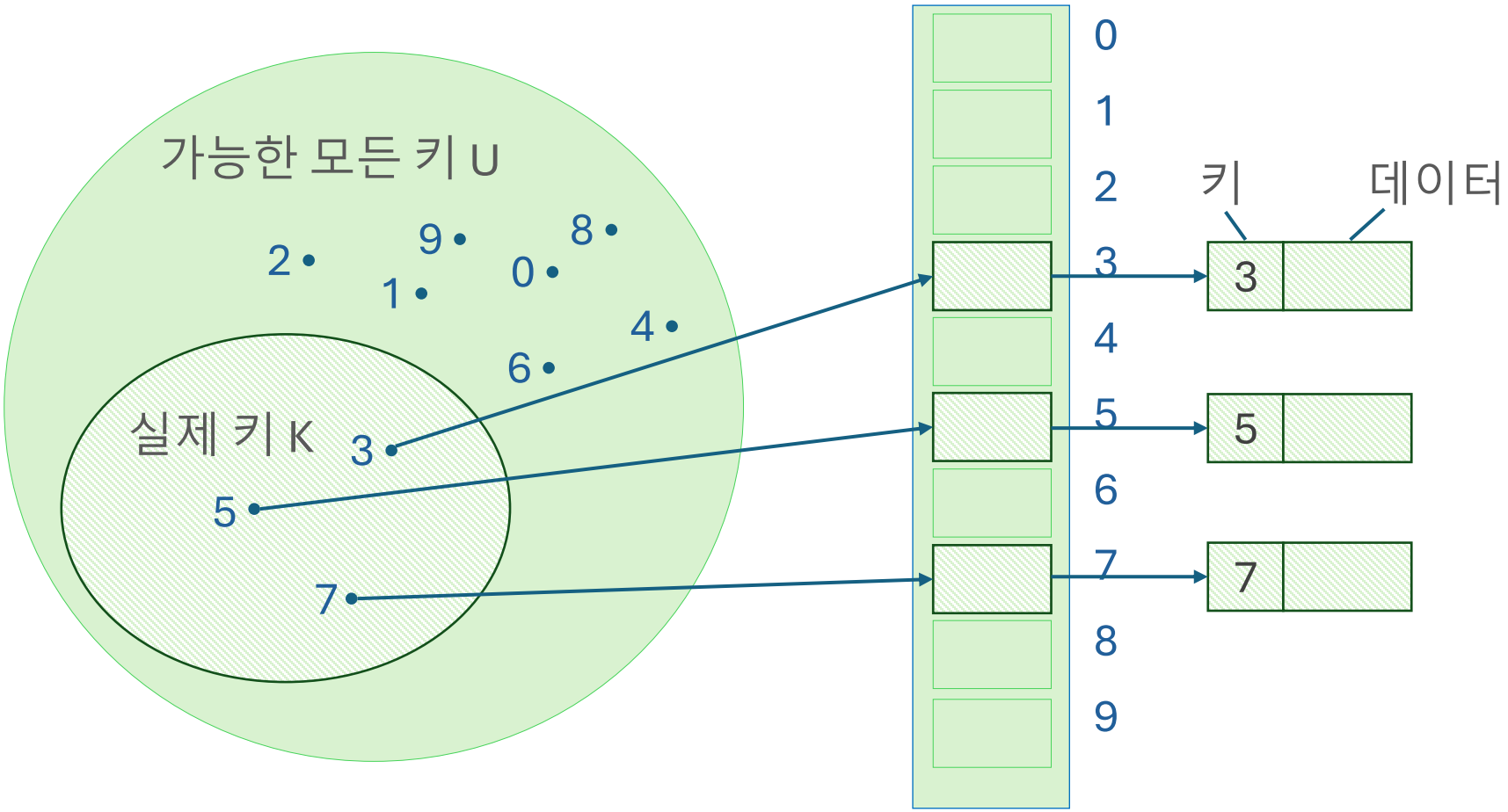


검색을 쉽게 하는 방법

- 내선번호로부터 건물과 방번호를 쉽게 찾으려면?

직통 주소 표

- 키를 배열에서의 인덱스로 바꿔서 쓴다



직통 주소 표 - 데이터 추가

함수 DIRECT_ADDRESS_INSERT(주소표 T, 데이터 x):

- T[x->key] = x

직통 주소 표 - 데이터 삭제

함수 DIRECT_ADDRESS_DELETE(주소표 T, 데이터 x):

- T[x->key] = 없음

직통 주소 표 - 데이터 찾기

함수 DIRECT_ADDRESS_SEARCH(주소표 T, 키 k):

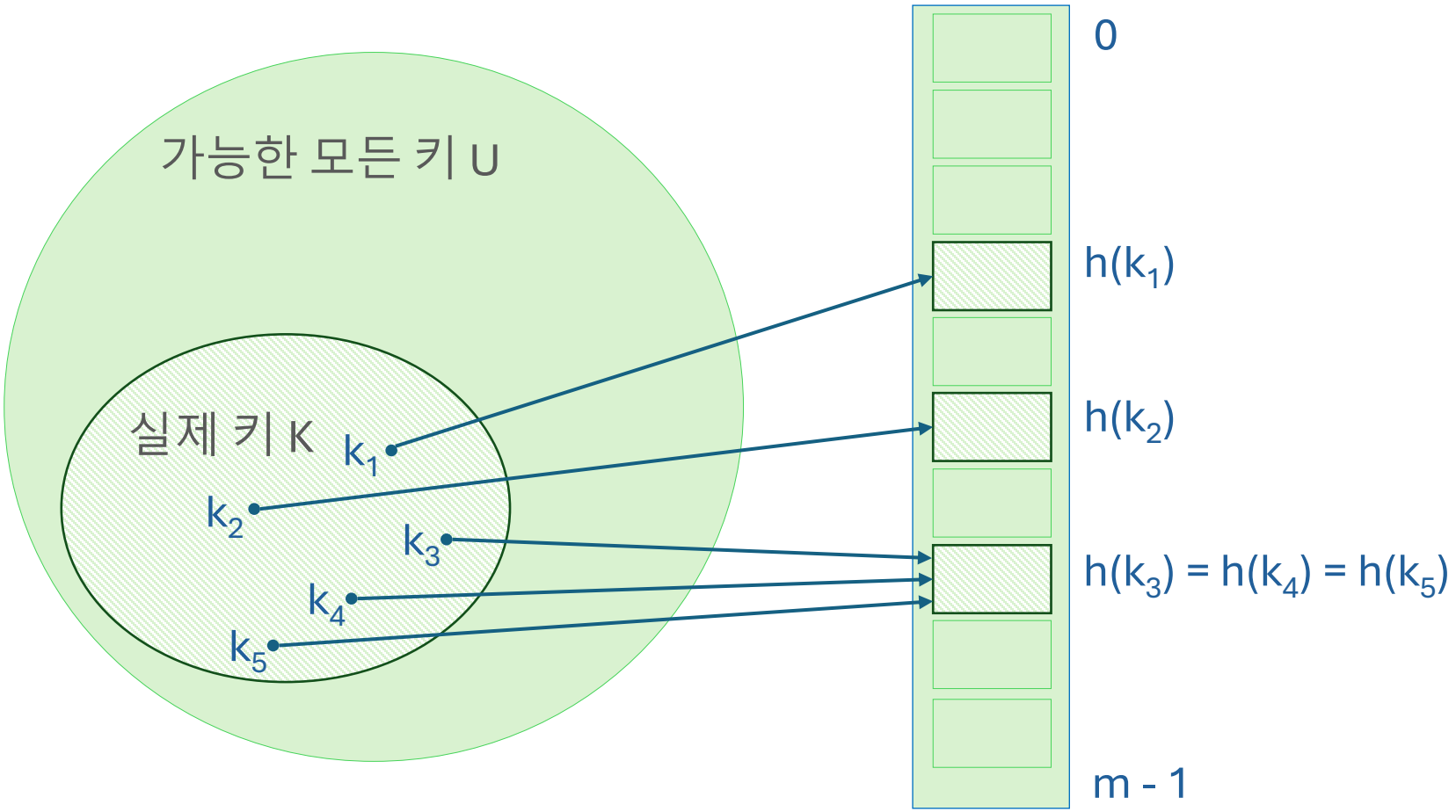
- 반환: T[k]

직통 주소 표

- 가능한 키의 경우의 수가 너무 많다면?

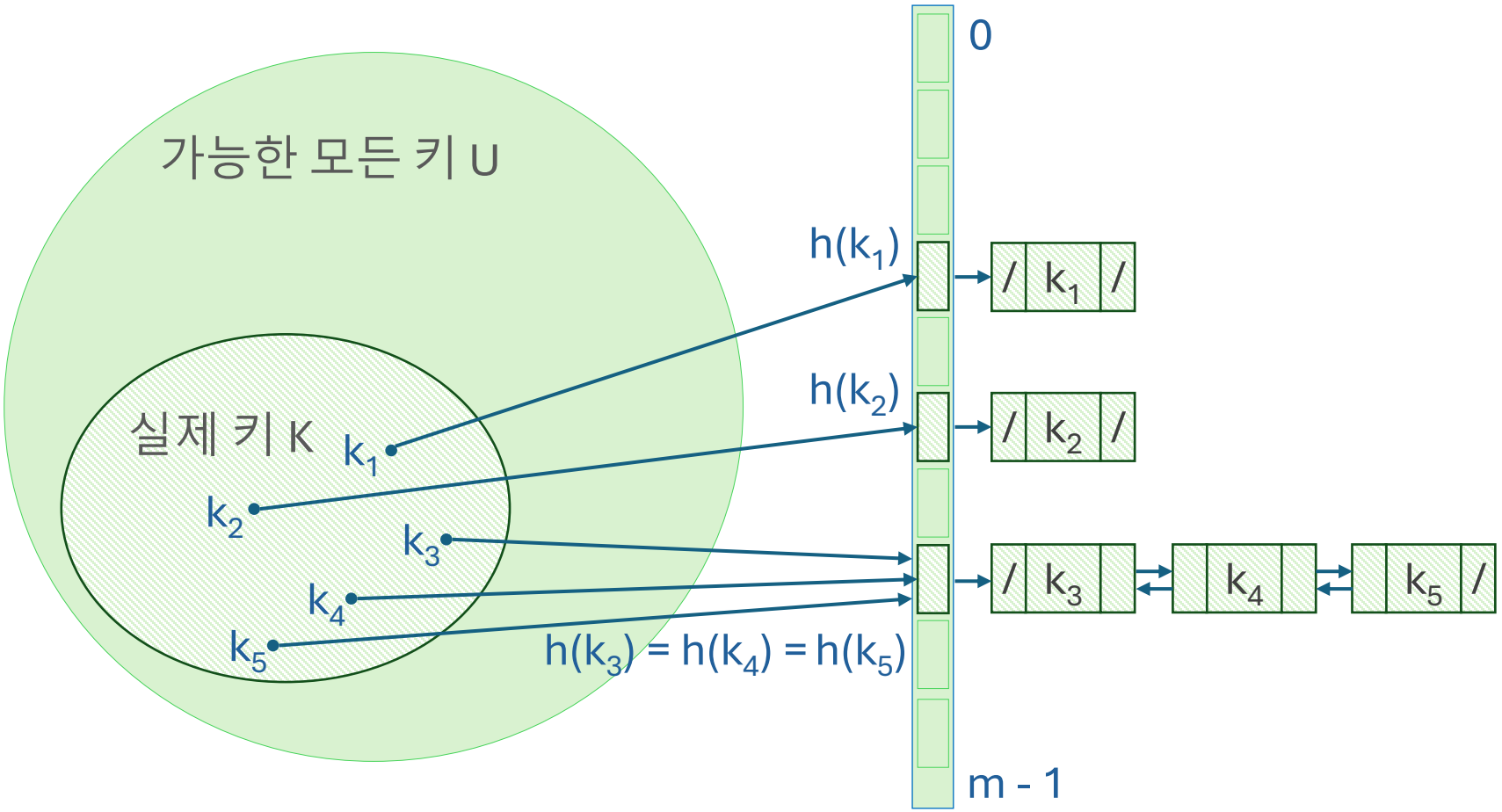
해시 표

- 해시 함수 $h: \mathrm{U} \rightarrow \{0, 1, \ldots, m-1\}$



해시 표

- 해시값 충돌 해결을 위해 링크드 리스트 사용



해시 표 - 데이터 추가

함수 CHAINED_HASH_INSERT(주소표 T, 데이터 x):

- 리스트 T[h(x->key)]의 앞에 x 추가

해시 표 - 데이터 삭제

함수 CHAINED_HASH_DELETE(주소표 T, 데이터 x):

- 리스트 T[h(x->key)]에서 x 삭제

해시 표 - 데이터 찾기

함수 CHAINED_HASH_SEARCH(주소표 T, 키 k):

- 리스트 T[h(x->key)]에서 키가 k인 노드를 찾기

해시 표 - 데이터 찾기 - 시간 분석

없는 키 k를 찾을 때:

- $1 + size_{h(k)}$ (리스트 T[h(k)]의 길이)

  $=$ 평균 $1 + \frac{n}{m}$

해시 표 - 데이터 찾기 - 시간 분석

i번째 추가된 키 k를 찾을 때:

- $1 + size_{h(k),i}$까지 (리스트 T[h(k)]의 i까지 길이)

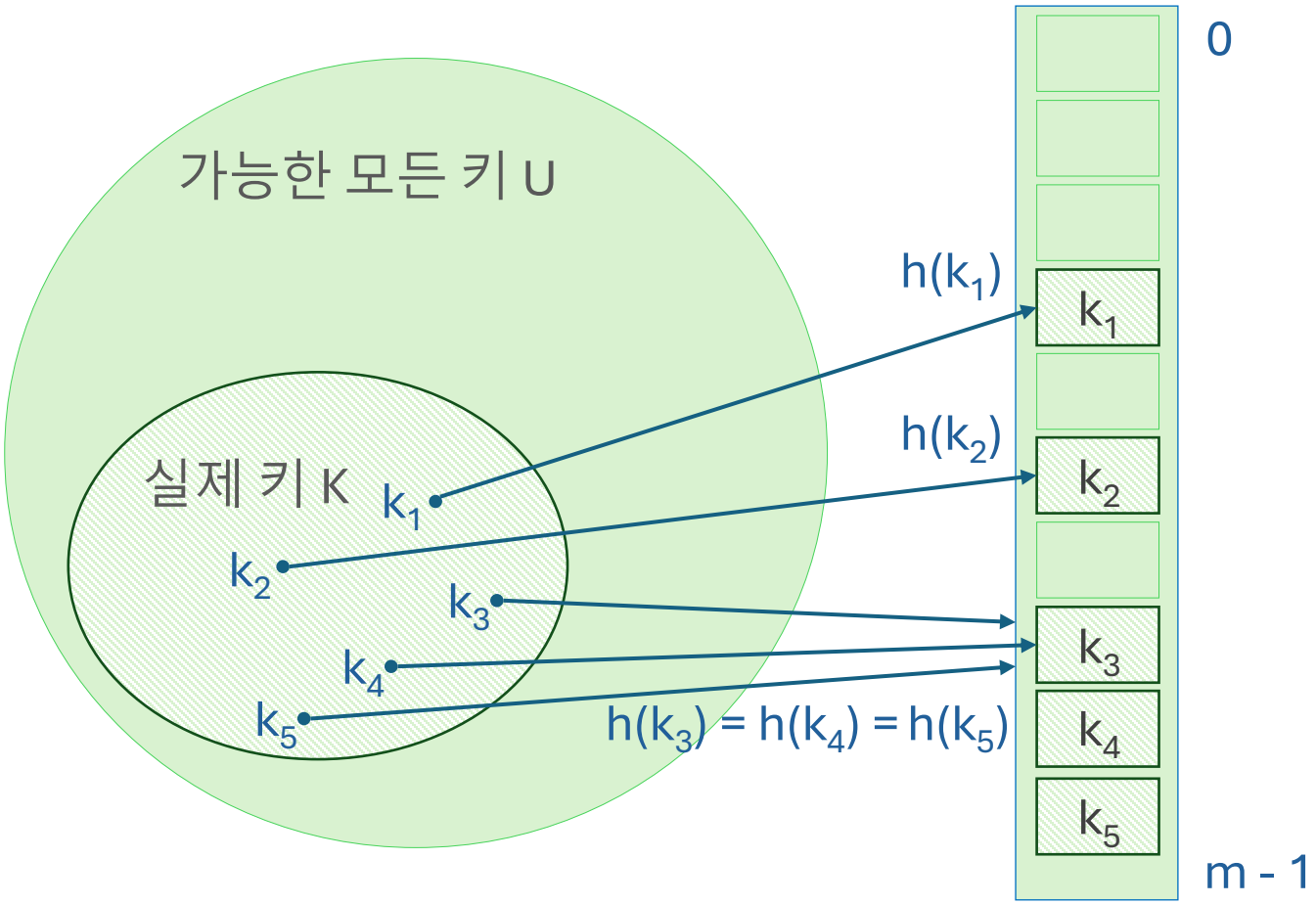
  $=$ 평균 $1 + \frac{n-1}{2m}$

해시 표

- 해시 표의 자료 구조를 더 간단하게 하려면?

단순한 해시 표

- 찾아간 주소에 데이터가 있으면 다음 칸을 가본다

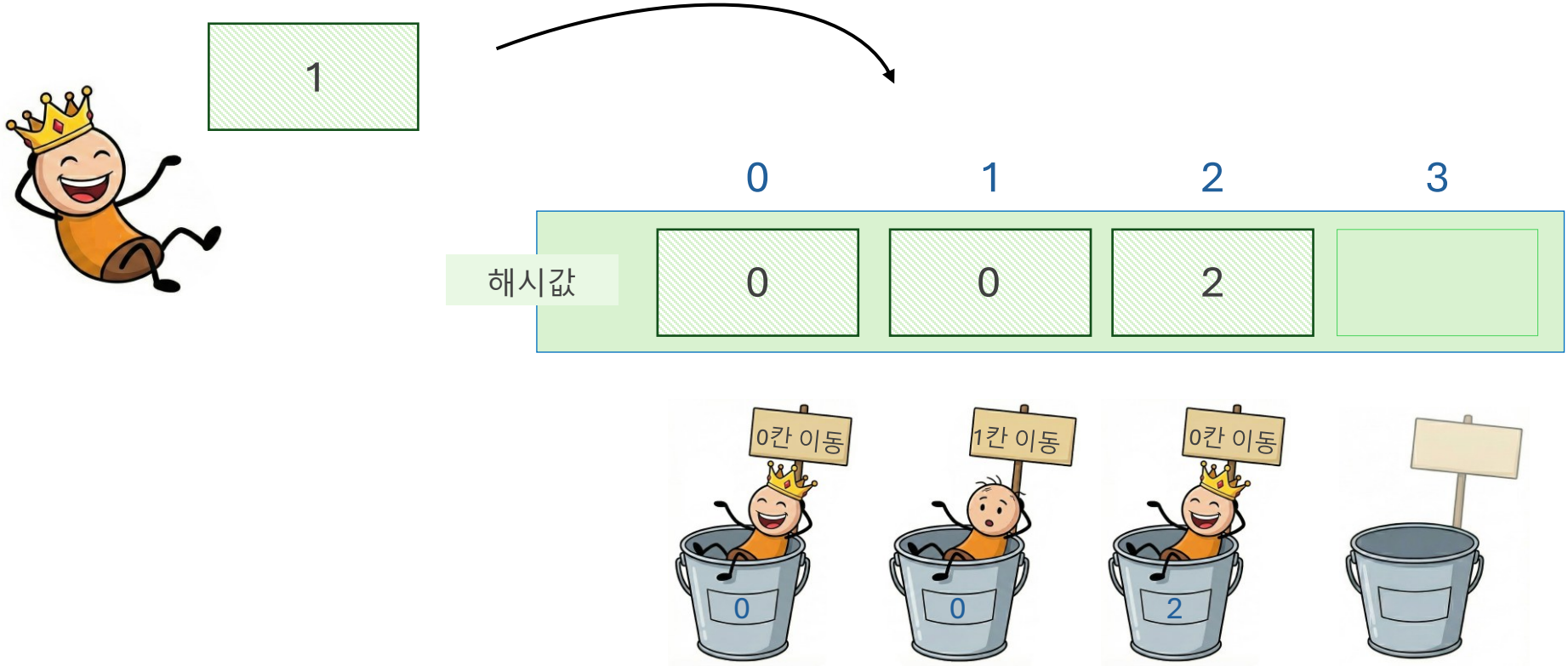


단순한 해시 표

- 찾아간 주소마다 계속 다른 데이터가 있으면?

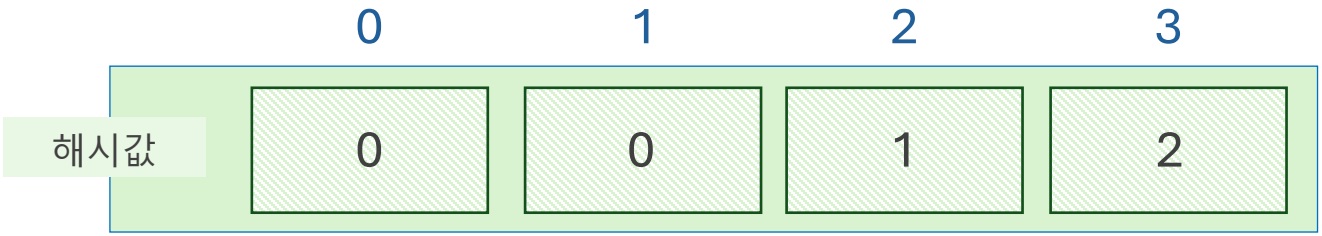
로빈후드식 단순한 해시 표

- 더 간절한 (멀리서 온) 데이터에 자리를 양보한다



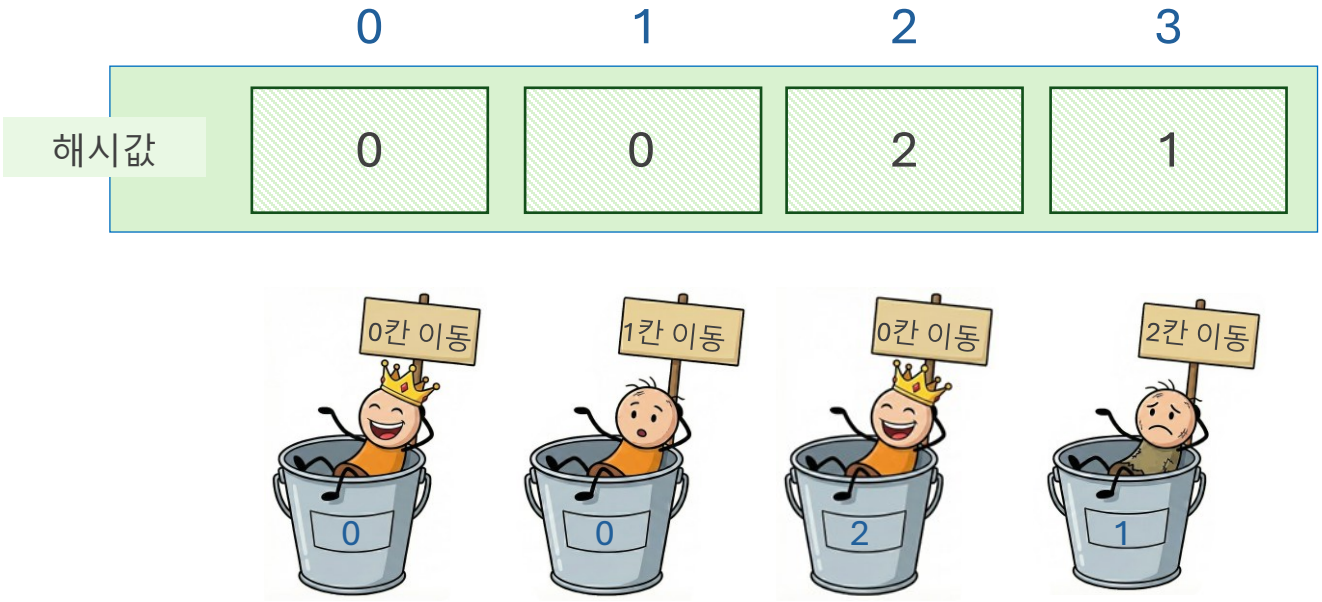
로빈후드식 단순한 해시 표

- 연속된 데이터들에서 해시 값은 작아지지 않는다
  - 없는 키를 찾을때 효율적



단순한 해시 표

- 연속된 데이터들에서 해시 값에 규칙이 없다
  - 없는 키를 찾을때 비효율적



단순한 해시 표

- 연속된 데이터 (클러스터)의 규모가 커질 확률은?

단순한 해시 표

- 서버의 해시 표를 공격해서 서비스 속도를 현저하게 늦추는 방법은?

개선된 주소 계산법

- 이동할 때마다 이동하는 칸수를 1 증가
- 클러스터 안에 있을 때 더 빠르게 벗어난다

최신 해시 표

- 키의 크기가 큰 경우에 빠르게 비교하기 위해, 비교를 하기 위한 해시값 생성
- 최신 CPU 기계어로 병렬처리

단순한 해시 표 - 시간 분석

없는 키 k를 찾을 때:

- $1 + h(k)$에 데이터가 있을 확률
  $+(h(k) + 1)$에도 데이터가 있을 확률
  $+ \cdots$

$$\leq 1 + \frac{n}{m} + \left(\frac{n}{m}\right)^2 + \cdots$$

$$\approx \frac{1}{1-\frac{n}{m}} = \frac{m}{m-n}$$

단순한 해시 표 - 시간 분석

키 k를 추가할 때:

- $1 + h(k)$에 데이터가 있을 확률
  $+(h(k) + 1)$에도 데이터가 있을 확률
  $+ \cdots$

$$\leq 1 + \frac{n}{m} + \left(\frac{n}{m}\right)^2 + \cdots$$

$$\approx \frac{1}{1-\frac{n}{m}} = \frac{m}{m-n}$$

단순한 해시 표 - 시간 분석

i번째에 추가된 키 k를 추가할 때:

- $\frac{m}{m-i}$

평균

- $\frac{1}{n}\sum\frac{m}{m-i}$

$\leq \frac{m}{n}\int\frac{1}{x} = \frac{m}{n}(\ln m - \ln m - n) = \frac{m}{n}\ln\frac{m}{m-n}$

단순한 해시 표 - 데이터 삭제

방법 1: 삭제됨이라고 표시한다.

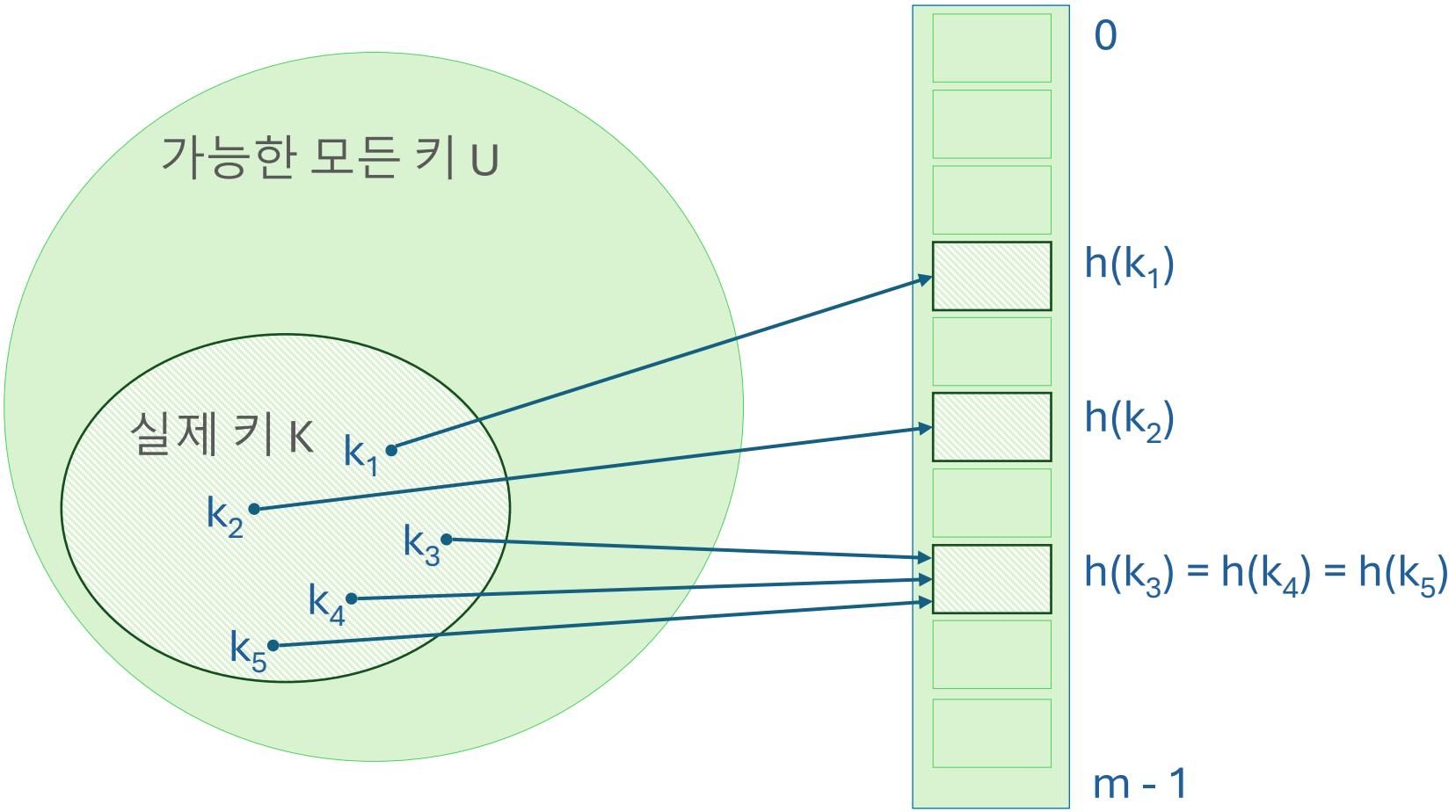
- 추가할 때, 삭제된 자리에 데이터 추가 가능
- 검색할 때, 데이터가 있다고 취급

단순한 해시 표 - 데이터 삭제

방법 2: 다음 자리에 있는 데이터를 이동해서 메꾼다.

해시 함수

- 해시 함수 $h: \mathrm{U} \rightarrow \{0, 1, \ldots, m-1\}$



해시 함수

- 1단계: 키를 음이 아닌 정수로 바꾼다.

    - [0, 1) 구간의 실수: 큰 수를 곱하기

    - 문자열: 자리수 정하기

해시 함수

- 2단계: 결과값을 [0, m - 1] 범위로 만든다

  - m으로 나눈 나머지

  - m이 2의 지수면, & 연산으로 가능

해시 함수

- 1.5단계: 적절한 수를 곱해서 섞는다

  - 곱하는 수의 예시: 황금비

해시 함수

- 해시 값을 예측할 수 없게 하는 방법은?

보안에 쓰이는 해시 함수

- 비밀번호를 저장하는 방법

  - 모든 비밀번호의 해시 값이 달라야 한다

  - 계산이 복잡해야 한다

요약

- 해시 테이블: 해시값을 인덱스로 써서 저장
- 체이닝 방법: 해시값이 같은 데이터를 리스트로
- 개방 주소법: 해시값부터 시작해서 주소를 찾기
- 해시 함수: 해시값을 계산하는 함수

다음 시간

- 해시 테이블을 C로 구현