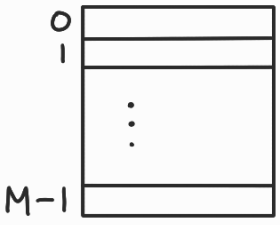
Ch 11. 해싱 (Hashing)

: 검색 (Search) 시간 O(1)을 위한 시도

⇒ 초대형 공간 (Hash Table T of size M = 주소공간: address space)

T[0..M-1]



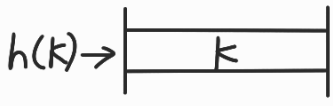
- 보통 prime number (소수)
- datasize N의 수천~수백만배
- 공간낭비가 크나 긴급 검색을 위해 필요

1. 배경

① data k가 저장될 위치를 k로부터 직접 구하는 방식

② h(k): Hash Function

T[h(k)] = k



2. 문제점

① collision (충돌)

for some $x \neq y$ $h(x) = h(y)$

(서로 다른 data가 같은 위치에 겹치는 현상

② clustering (군집화): data가 Hash Table에 고루 분산되지 않고 부분적으로 뭉치는 현상 ⇒ 좋은 hash function의 선택이 가장 중요함

3. Hash Function

① 조건: less collisions
고루 분산되어야 (clustering)

② 일반적: $\boxed{h(x) = f(x) \bmod M}$: 0~M-1 주소값

4. collision 해결

: data k가 저장될 위치 h(k)에 이미 다른 data가 존재

- probing (조사법): 다른 장소에 저장
  - ① linear probing
  - ② quadratic probing
  - ③ double hashing
- ④ chaining (체이닝): 한 위치에 여러 개 저장
  - linked list
  - bucket

① linear probing (선형조사법) 충돌 발생하면 옆자리 비어있는지 살펴보고, 비어있을 경우 그 자리에 대신 저장

: 장소 검색을 $(h(k)+i) \bmod M$, $i = 1 \sim M$ 로 한다.

장점: 매우 간단

단점: 1) 충돌 발생 시 그 하단에 1차 군집화 → 검색이 느려짐

2) 연뜩 저장된 data 중 하나가 삭제가 되면 아래쪽 data는 검색 불가

→ 빈자리의 분류: 사용중, 미사용, 삭제된 곳

② quadratic probing (2차 조사법)

: 1차 군집화의 감소를 위해 조사 위치 변경

$(h(k)+i^2) \bmod M$, $i = 1 \sim M$

$h(k)+1$, $h(k)+4$, $h(k)+9$, $h(k)+16$로 검사

(+3, +5, +7)

장점: 1차 군집화는 크게 약화됨

단점: 여전히 2차 군집화 가능성

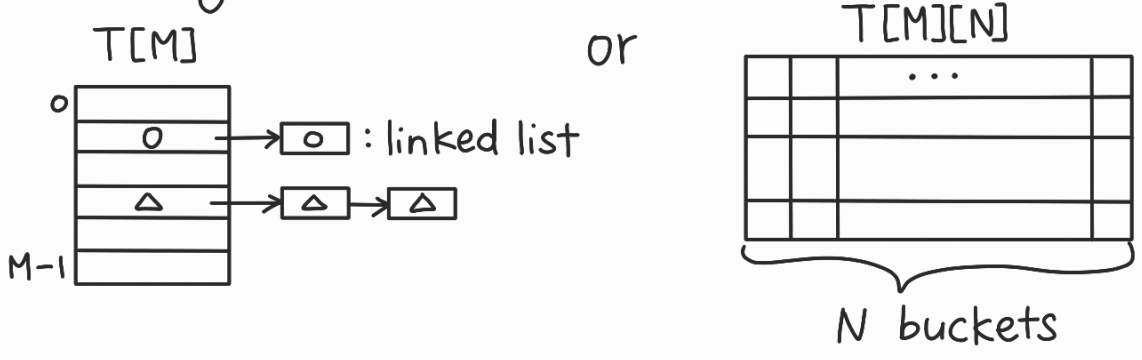
③ double hashing (이중해싱, rehashing)

$(h(k)+i*d) \bmod M$, $i = 1 \sim M$

↳ 간격 $d = C - [g(x) \bmod C]$ → M보다 약간 작은 prime (소수)

장단점: 군집화는 거의 없으나, not perfect

④ chaining



5. 성능 분석

삽입/삭제는 검색이 선행된다. → O(1): no collision / O(N): worst

let $\alpha = \frac{N}{M}$ = 적재 밀도 (loading density)

ex) linear probing: 성공 $= \frac{1}{2}[1+y(1-\alpha)]$ ⇒ $\alpha \approx 0.1$ 이어도 양호

실패 $= \frac{1}{2}[1+1/(1-\alpha)^2]$