

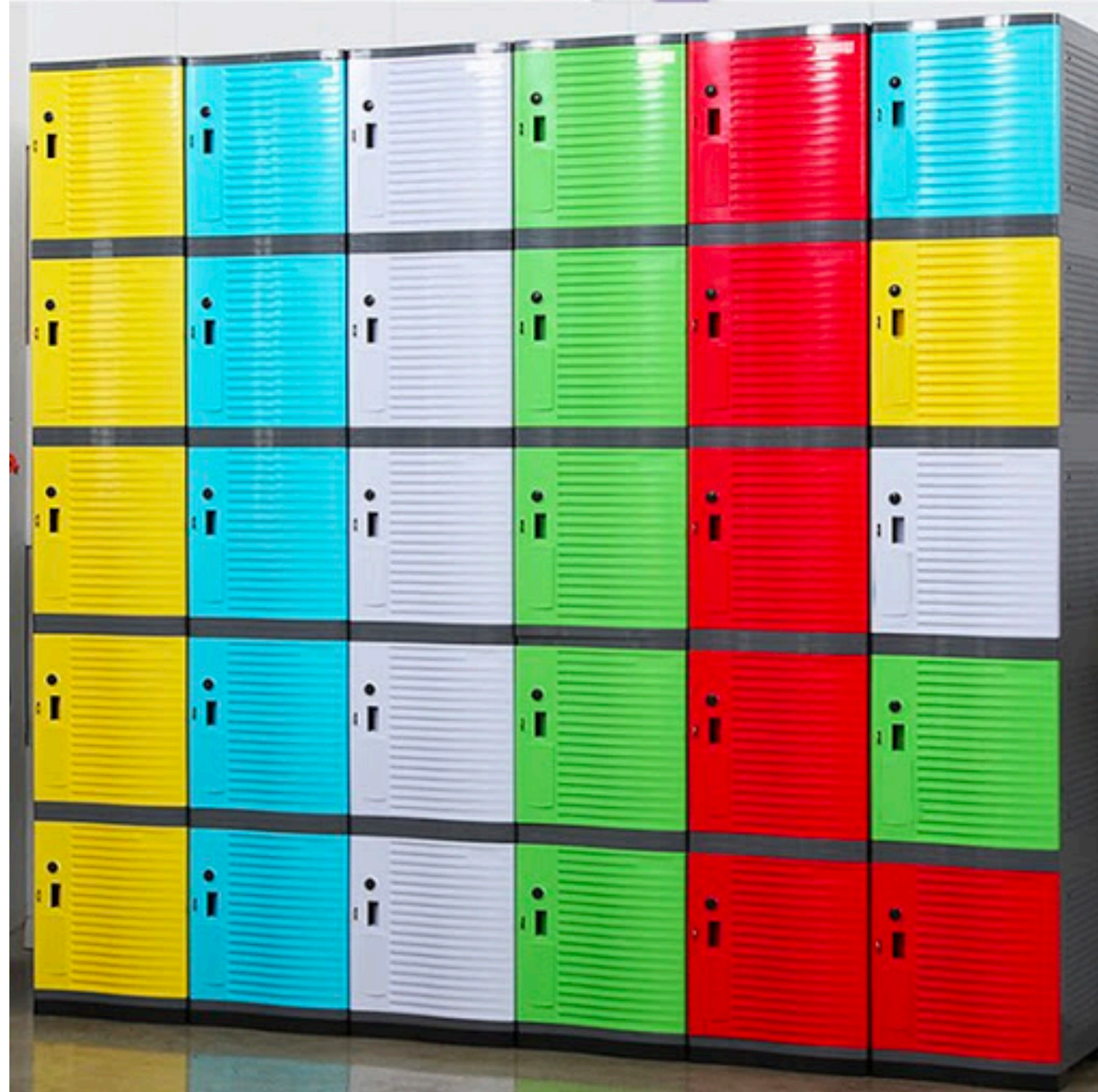
---

# 해시 테이블

---

코딩테스트 광탈방지 A to Z : JavaScript - 이선희 @kciter

# JS

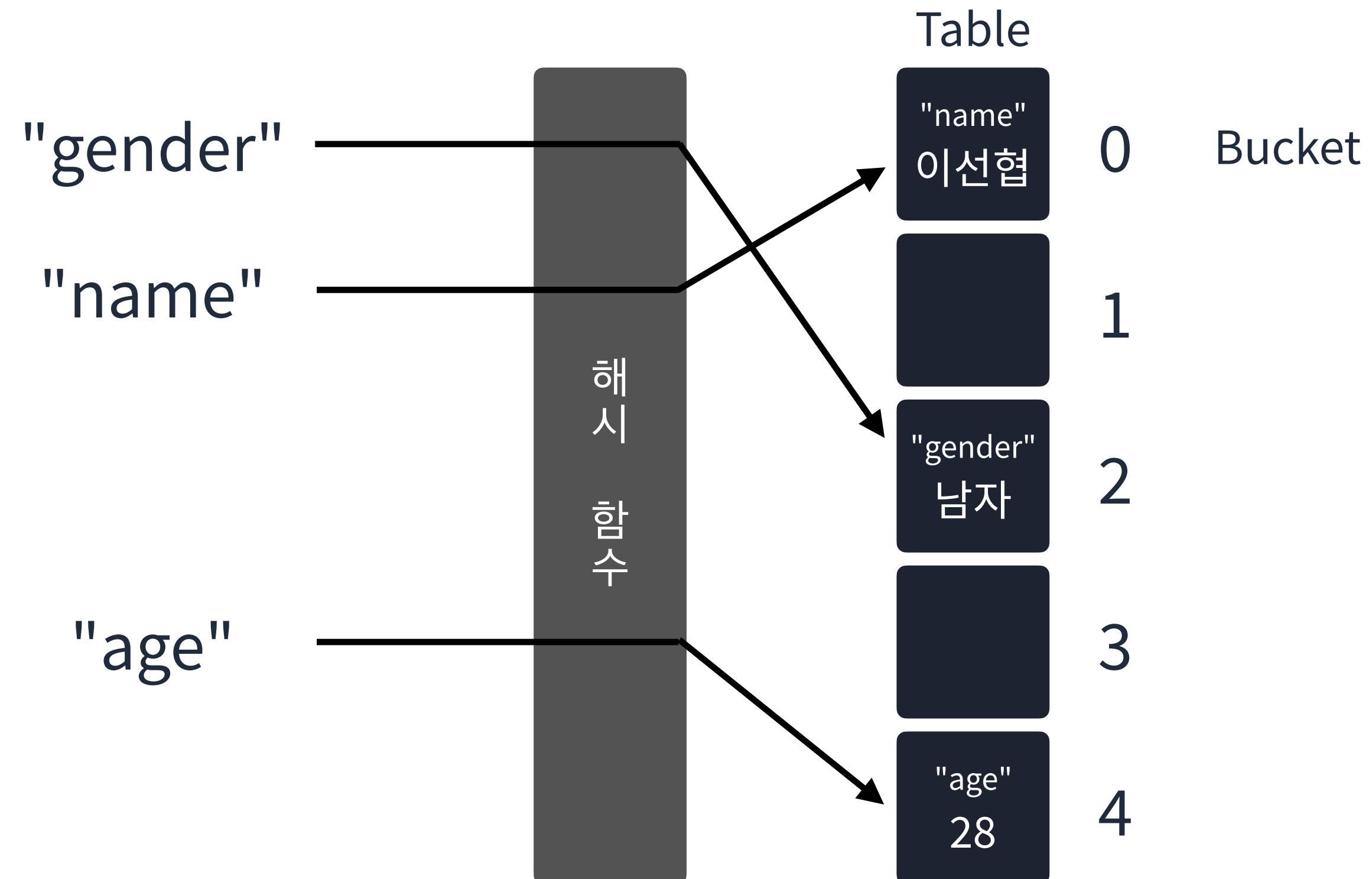


해시 테이블은 한정된 배열 공간에 key를 index로 변환하여 값들을 넣게된다.  
그럼 index는 어떻게 구할까?



## 해시 테이블

키와 값을 받아 키를 해싱(Hashing)하여 나온 index에 값을 저장하는 선형 자료구조  
삽입은  $O(1)$ 이며 키를 알고 있다면 삭제, 탐색도  $O(1)$ 로 수행한다.





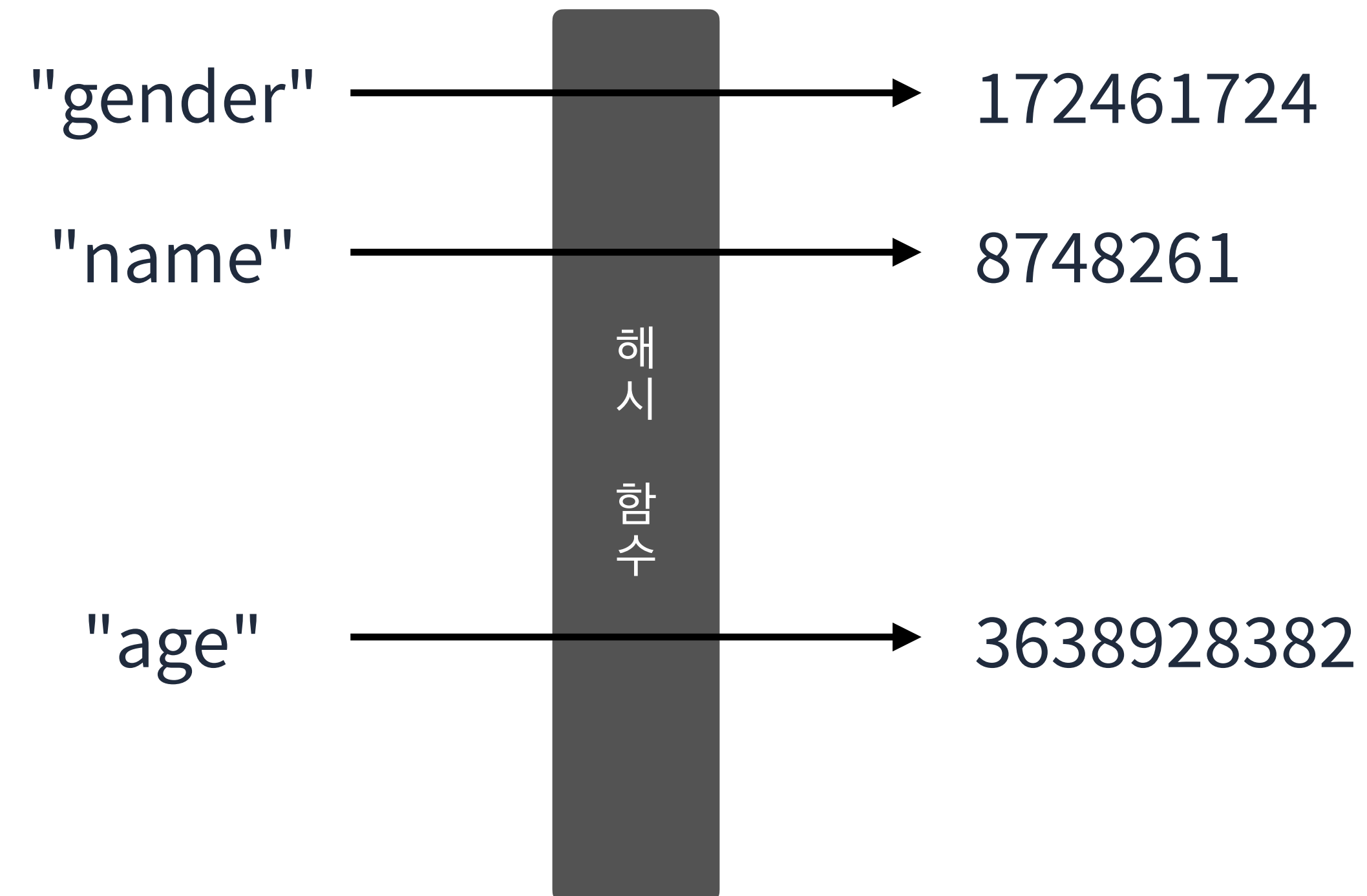


## Hash

고기와 감자를 잘게 다져 요리한 것

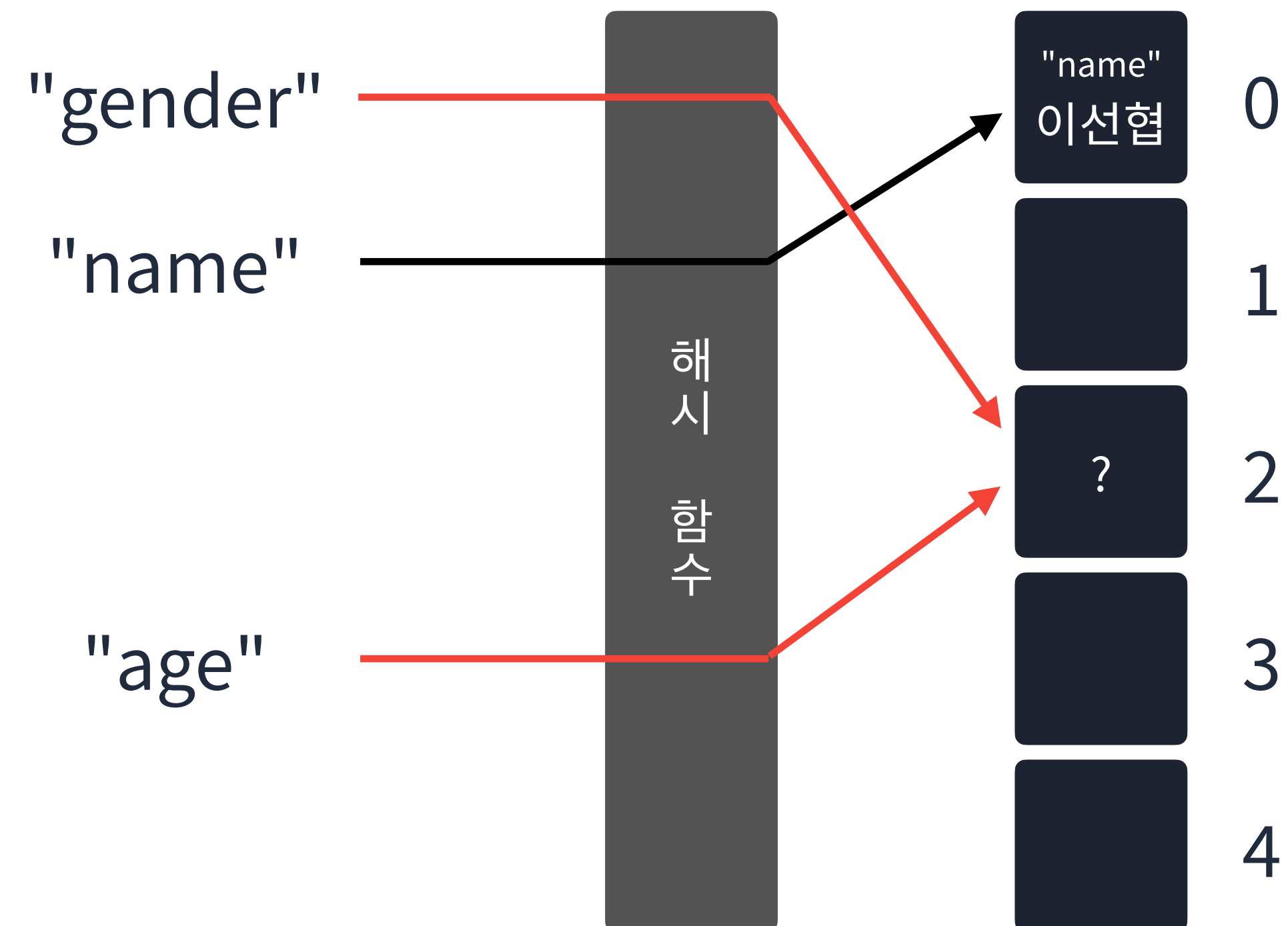
## 해시 함수

입력받은 값을 특정 범위 내 숫자로 변경하는 함수



## 해시 테이블의 문제점

만약 해시 함수의 결과가 동일하여 겹친다면?

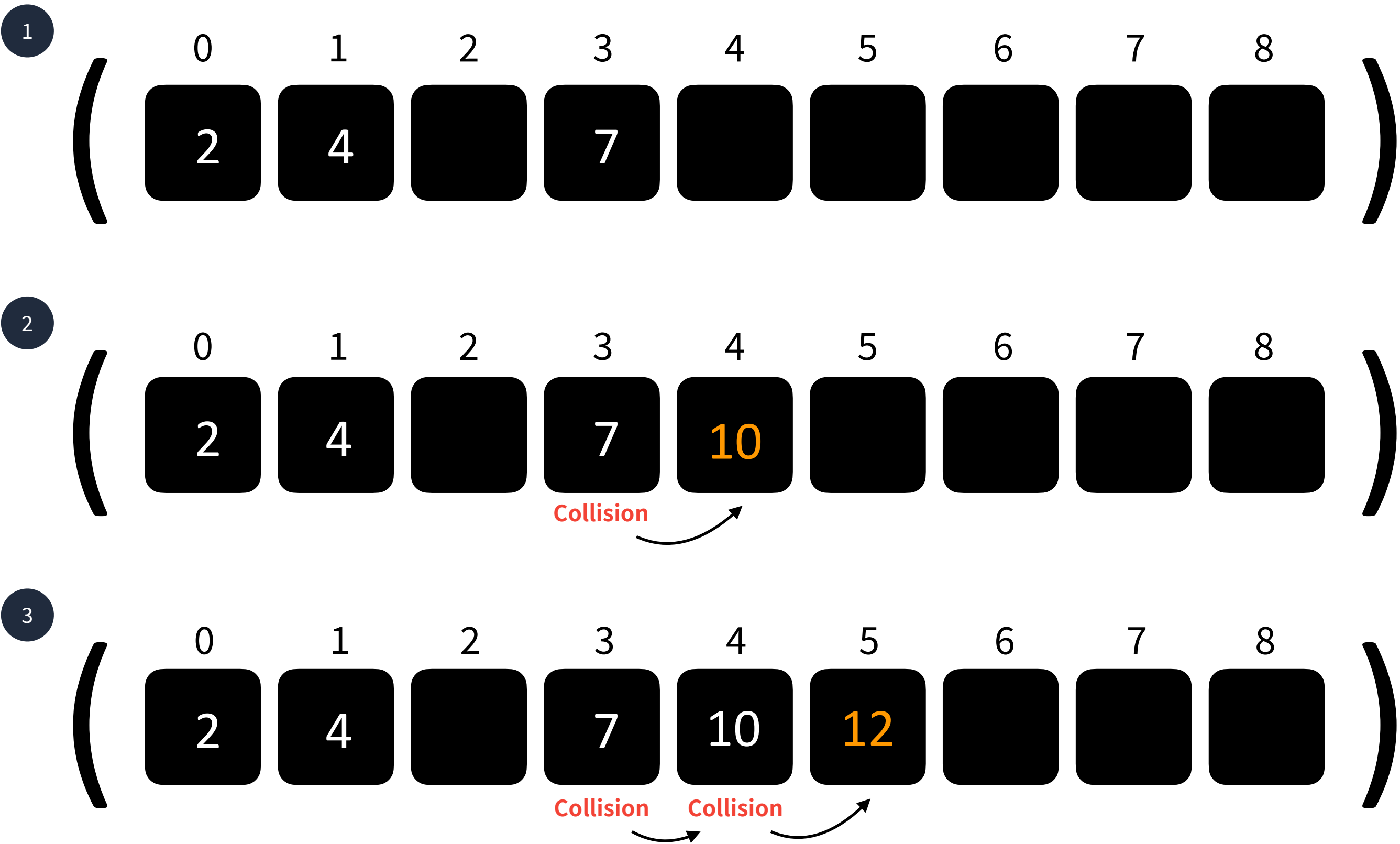


# Hash Collision



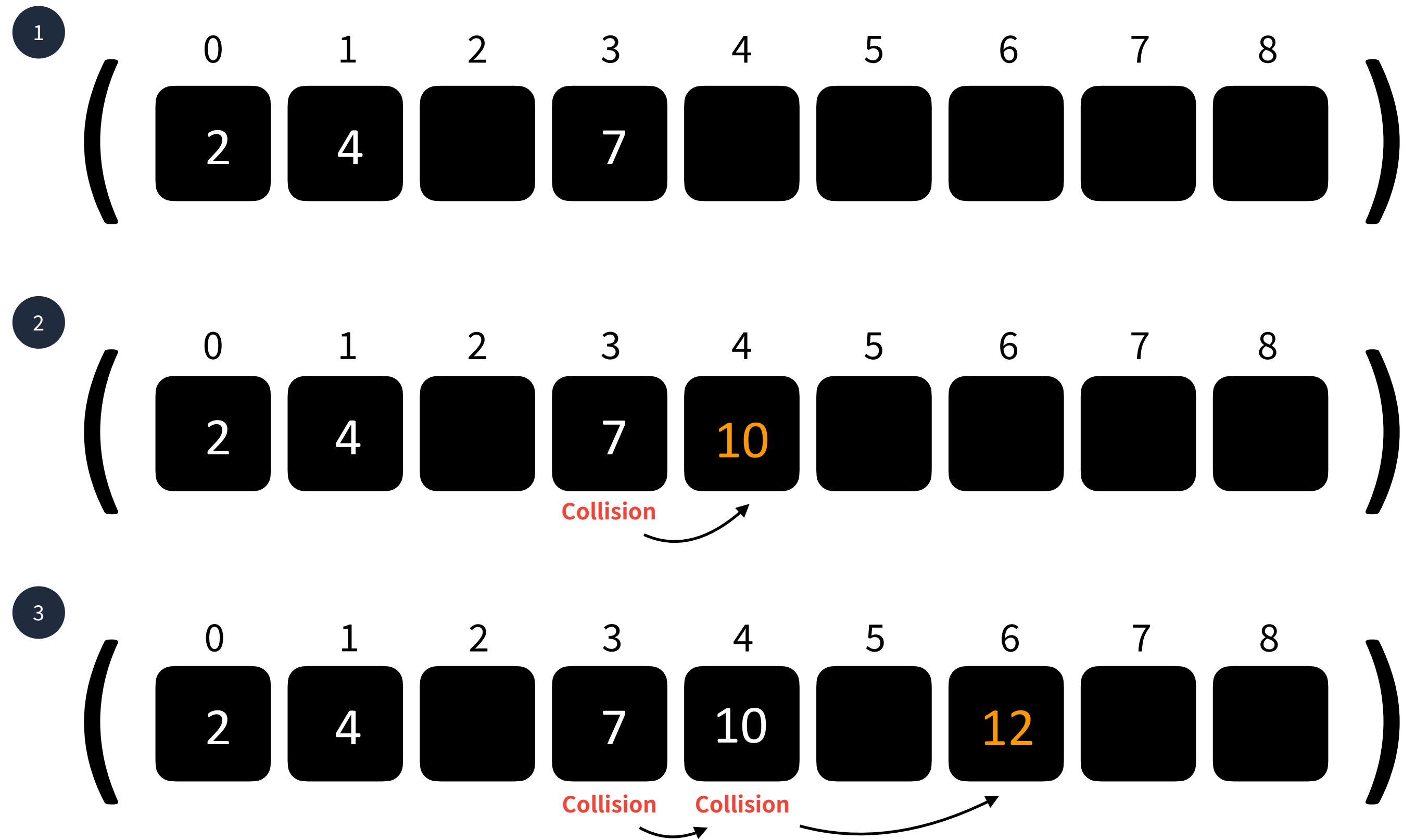
# 선형 탐사법

충돌이 발생하면 옆으로 한 칸 이동한다.



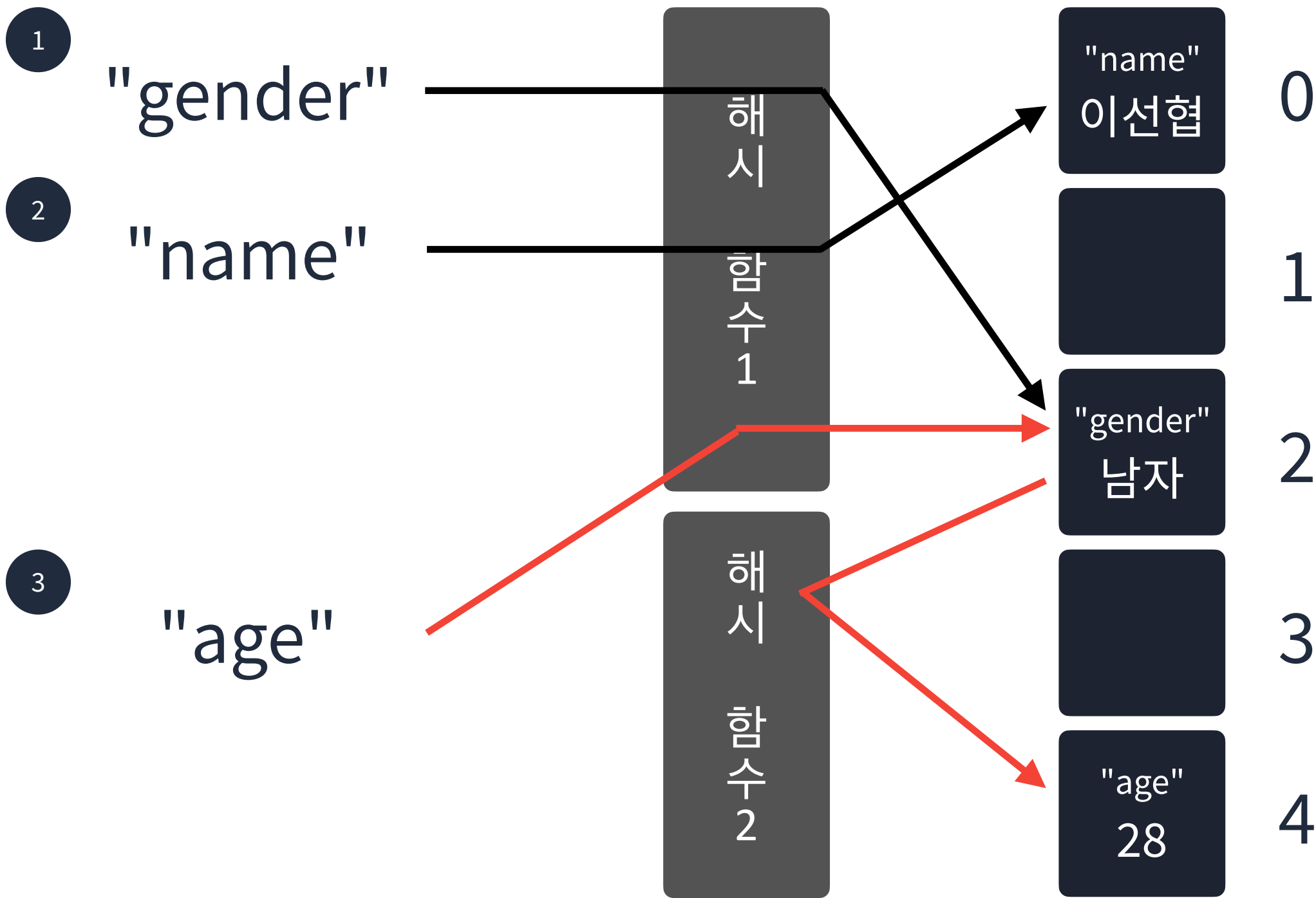
# 제공 탐사법

충돌이 발생하면 충돌이 발생한 횟수의 제공만큼 옆으로 이동한다.



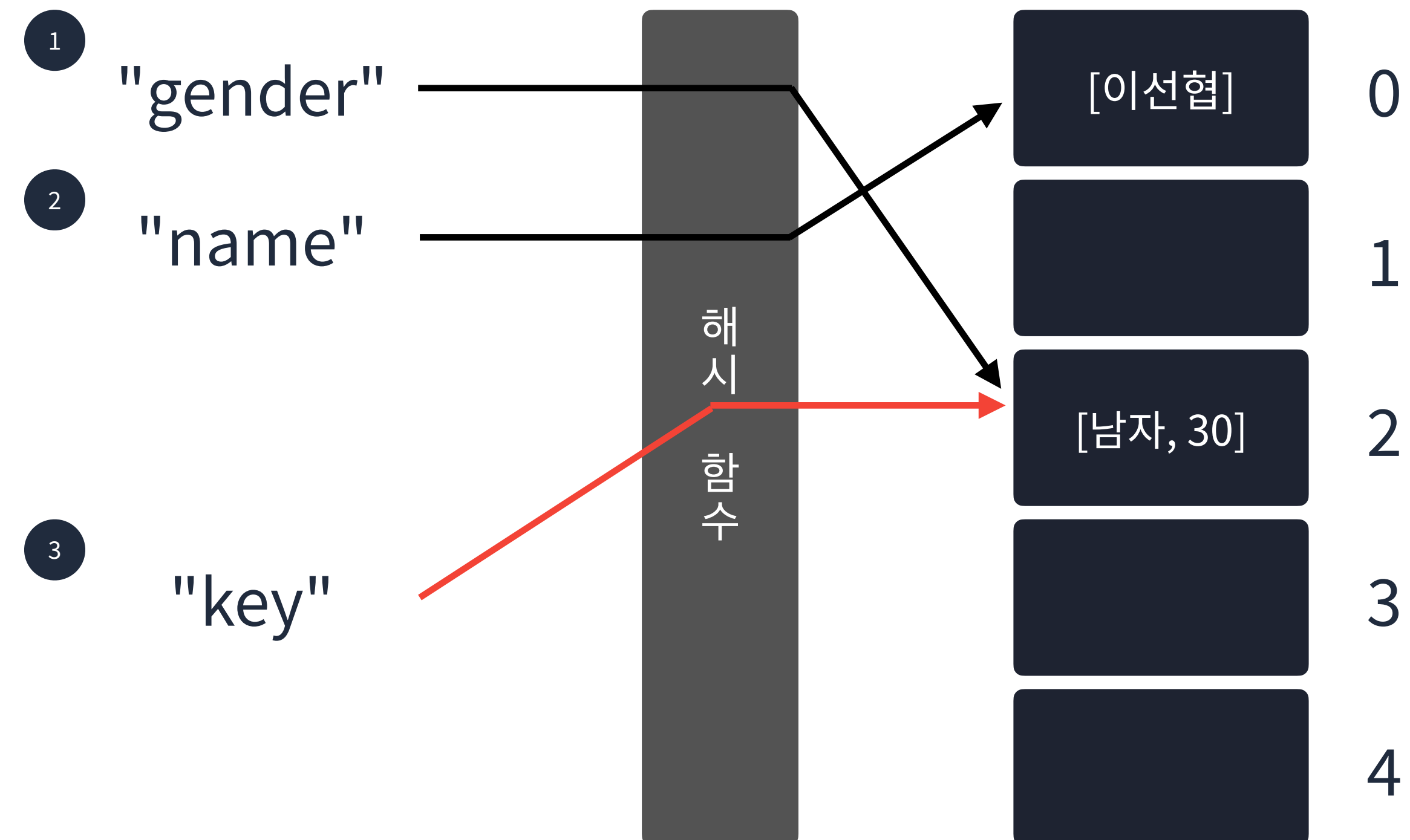
# 이중 해싱

충돌이 발생하면 다른 해시 함수를 이용한다



## 분리 연결법

버킷의 값을 연결 리스트로 사용하여 충돌이 발생하면 리스트에 값을 추가한다.



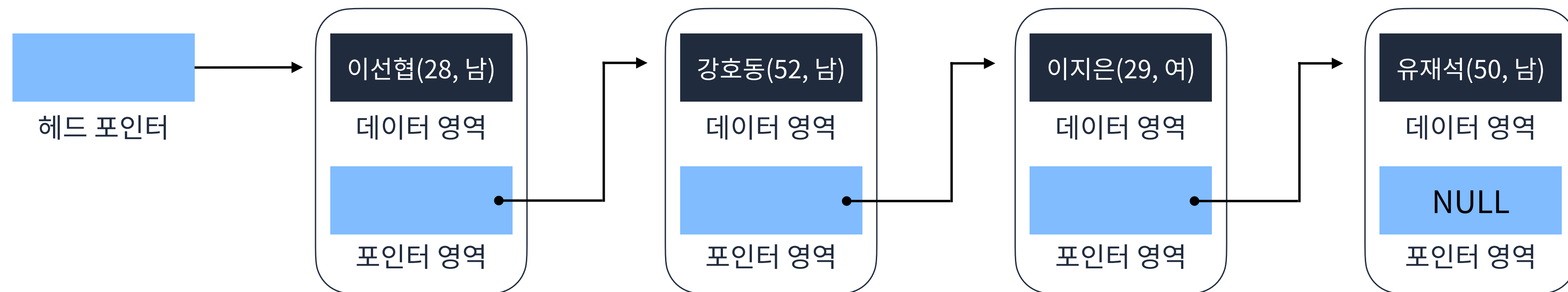
어디에 사용하는가?



학생 정보를 어떻게 관리할 것인가?



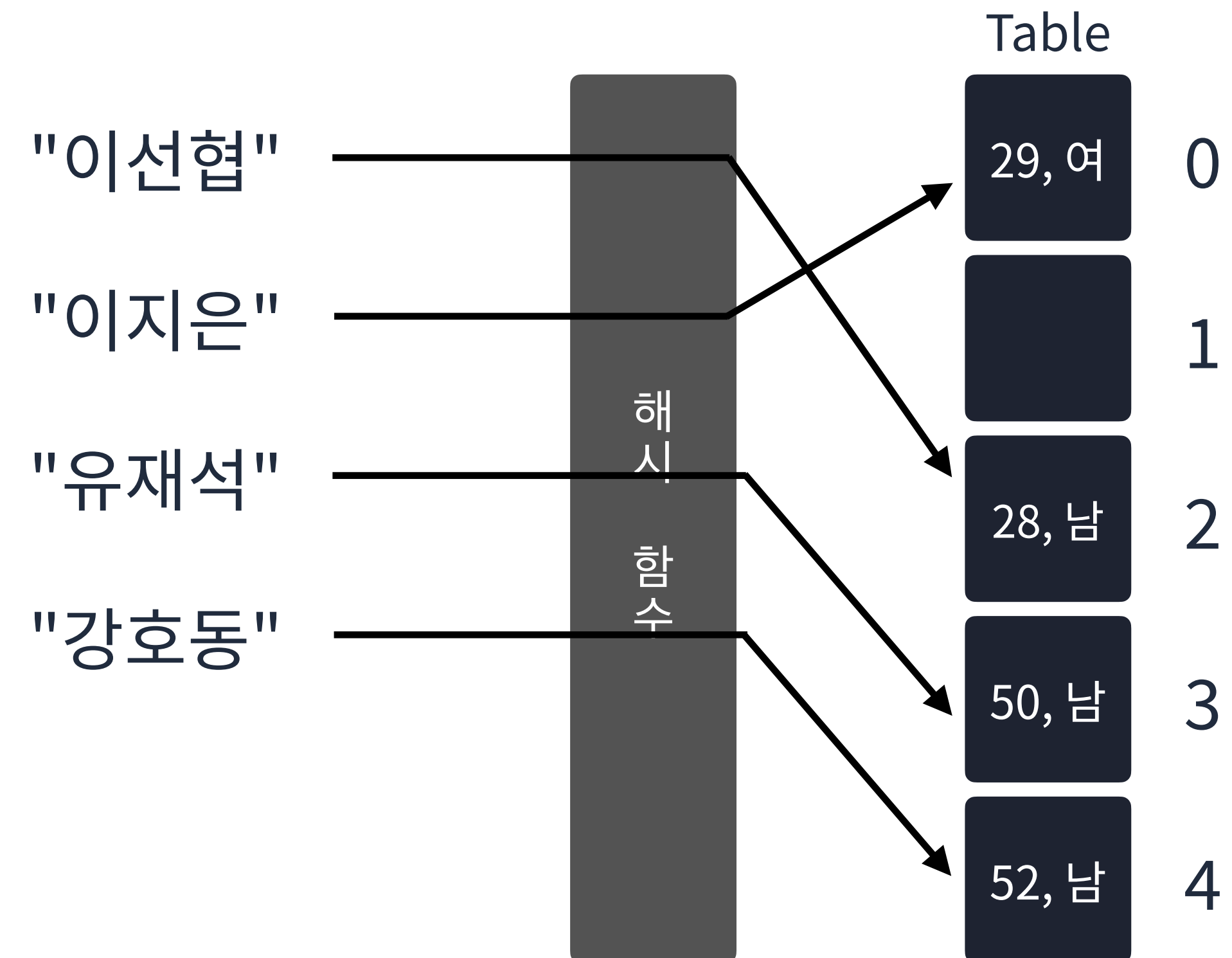
연결 리스트를 사용하면 학생 정보가 알고 싶을 때  $O(n)$  시간복잡도가 걸린다.



배열은 인덱스를 모를 경우 탐색에  $O(n)$ 이 걸린다.



반면 해시 테이블을 사용하면  $O(1)$ 에 찾을 수 있다.  
따라서 빠르게 값을 찾아야하는 경우 해시 테이블을 사용하는 것이 좋다.



# JavaScript에서 사용법



# JavaScript Array $\approx$ Hash Table

```
const table = [];  
table["key"] = 100;  
table["key2"] = "Hello";  
console.log(table["key"]); // 100  
table["key"] = 349;  
console.log(table["key"]); // 349  
delete table["key"];  
console.log(table["key"]); // undefined
```

# JavaScript Object $\approx$ Hash Table

```
const table = {};  
table["key"] = 100;  
table["key2"] = "Hello";  
console.log(table["key"]); // 100  
table["key"] = 349;  
console.log(table["key"]); // 349  
delete table["key"];  
console.log(table["key"]); // undefined
```

# Map

```
const table = new Map();
table.set("key", 100);
table.set("key2", "Hello");
console.log(table["key"]); // undefined
console.log(table.get("key")); // 100
const object = { a: 1 };
table.set(object, "A1"); // Map은 Object도 Key로 쓸 수 있다
console.log(table.get(object)); // A1
table.delete(object);
console.log(table.get(object)); // undefined
console.log(table.keys()); // { 'key', 'key2' }
console.log(table.values()); // { 100, 'Hello' }
table.clear();
console.log(table.values()); // { }
```

# Set

```
const table = new Set();
table.add("key"); // Key와 Value가 동일하게 들어간다
table.add("key2");
console.log(table.has("key")); // true
console.log(table.has("key3")); // false
table.delete("key2");
console.log(table.has("key2")); // false
table.add("key3");
console.log(table.size); // 2
table.clear();
console.log(table.size); // 0
```

---

# 해시 테이블

---

코딩테스트 광탈방지 A to Z : JavaScript - 이선희 @kciter

# JS