

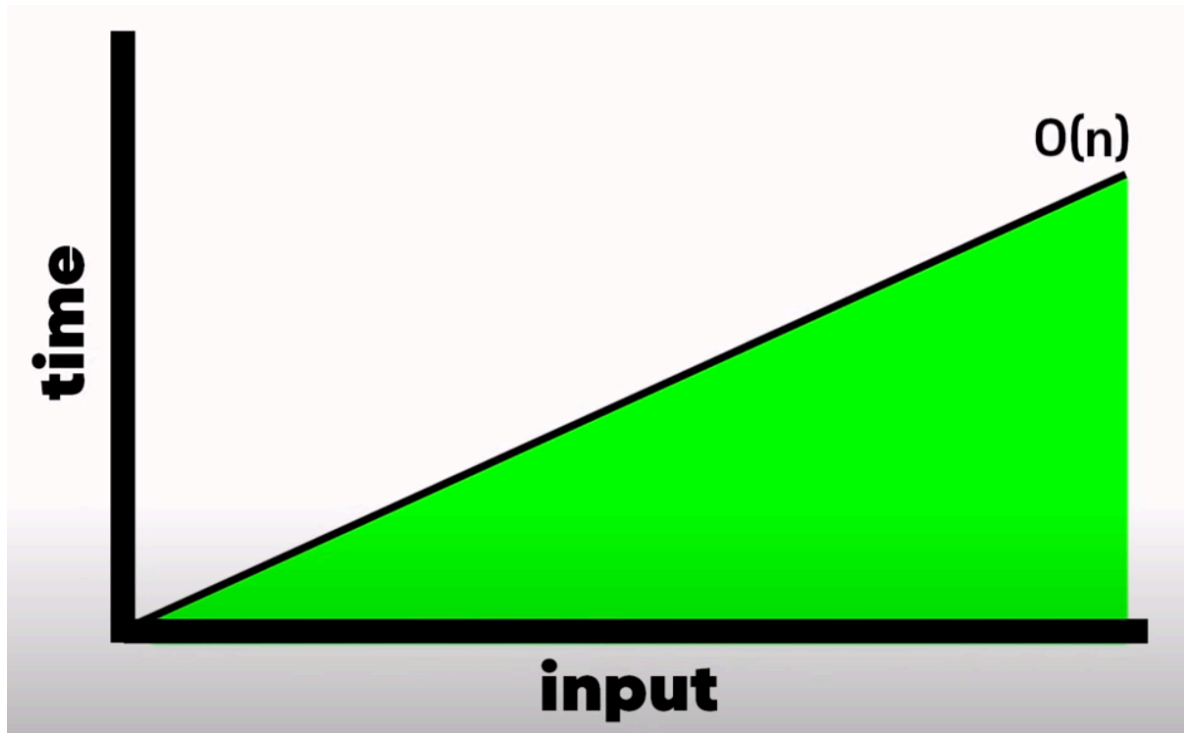
해시 자료 구조(Hash Tables)

- 영상 정보. <https://www.youtube.com/watch?v=HraOg7W3VAM>
 - 영상 구성 Part 1. Hash Table 정의와 활용
 - 영상 구성 Part 2. Hash Table의 작동 원리와 컴퓨터 레벨에서의 동작 순서
- 해시 테이블(Hash Tables)이란?
 - Key:Value System (ex. 사전 - 단어: Key / 뜻,설명: Value)
 - JS : Object / Python : Dictionary / Go : map ...
- 해시 테이블(Hash Tables) / Array(배열) 의 비교
- 배열(Array)에서 Value 검색 시 선형 검색(Linear Search)을 통해 원하는 Key의 Value 값을 가져올 수 있음.

```
menu = [  
    { name: "coffee", price: 10 },  
    { name: "burger", price: 15 },  
    { name: "tea", price: 5 },  
    { name: "pizza", price: 10 },  
    { name: "juice", price: 5 },  
];
```

-> ex. pizza의 가격을 알기 위해 coffee부터 차례대로 데이터를 체크하면서 pizza를 찾는다.(선형 검색)

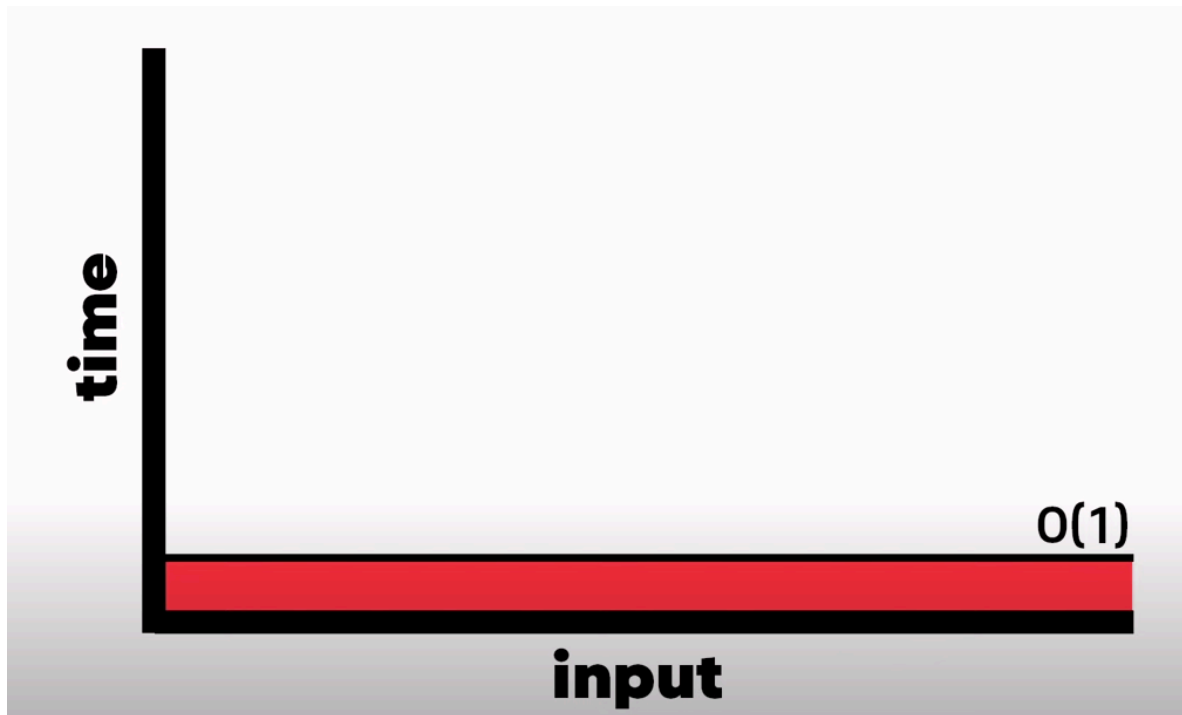
- 선형 검색이란?
 - 각각의 아이템을 처음부터 끝까지 체크하는 방법 -> 하나하나씩 체크하기 때문에 시간이 오래 소요된다는 단점이 있다.
- 배열에서의 시간복잡도 : $O(n)$



- Linear Time(선형 시간) : 데이터가 많아질 수록 찾는 시간도 증가
- 해시 테이블(Hash Table)에서는 Key 값을 검색하여 Value값을 가져온다.

```
menu = {  
    coffee: 10,  
    burger: 15,  
    tea: 5,  
    pizza: 10,  
    juice: 5,  
};
```

- 해시 테이블에서의 시간 복잡도 : $O(1)$



- Constant Time(상수 시간) : 어떤 데이터의 값을 찾거나 추가, 삭제 시에도 $O(1)$ 이기에 배열과 비교 시 속도라 빠르다는 장점이 있다.
- 더 빠르게도 가능한가? Yes!
 - Key, Value를 저장하지 않고, Value만 저장함으로써 시간을 단축 시킬 수 있다.

Part 2. 해시 테이블의 원리

Hash? -> array구조로 되어 있으며 모든 해시 테이블의 값들은 배열내에 존재한다.

배열에서는 index를 통해 값에 접근이 가능하다.

해시 테이블은 Hash Function이 있기 때문에 좀 더 빠른 값을 찾아올 수 있다.

- 해시 함수(Hash Function)
 - 내가 저장하고자 하는 Key를 해시함수에 넣어 숫자로 바꿔버리고 그 숫자가 인덱스(index)가 되어 그 안에 Value가 저장된다.
 - 다른 Key에 대해 해시함수가 동일한 숫자를 준 경우 해시 충돌(Collision)이 발생하게 된다.
- 해시 테이블 검색
 - 충돌이 발생할 수 있고, 그 경우 선형 검색을 해야 하기 때문에 해시 테이블의 검색은 항상 언제나 상수시간($O(1)$)인 것은 아니다.
- Collision(해시 충돌)
 - 1. 같은 index 공간에 또 다른 배열을 넣어준다.
 - Cake=key -> 해시함수입력 -> 4 출력 -> index 4로 이동 -> . Cake의 가격을 찾을 때까지 선형검색