

자료구조

(DATA STRUCTURE)

데이터를 효율적으로 저장하고 관리하기 위한 구조

자료구조를 공부해야 하는 이유

실행 결과

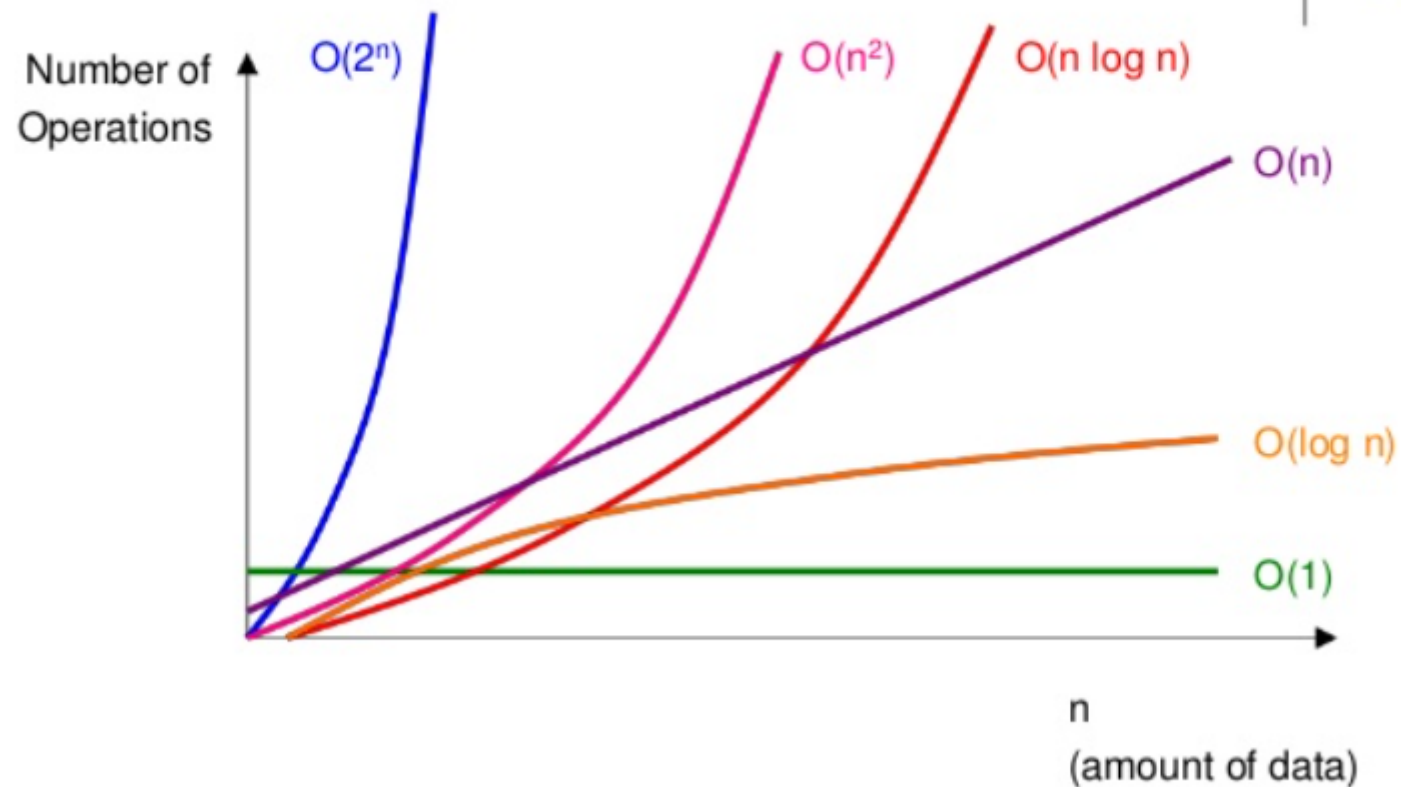
테스트 6 > 통과 (65.42ms, 10.7MB)
테스트 7 > 통과 (1799.37ms, 14MB)
테스트 8 > 통과 (6641.69ms, 18.6MB)
테스트 9 > 실패 (시간 초과)
테스트 10 > 실패 (시간 초과)
테스트 11 > 실패 (시간 초과)
테스트 12 > 실패 (시간 초과)
테스트 13 > 실패 (시간 초과)
테스트 14 > 통과 (0.01ms, 10.2MB)
테스트 15 > 통과 (0.00ms, 10MB)
테스트 16 > 통과 (0.01ms, 10.1MB)



홍종훈 (R반, 7조) GUEST



Comparing Big O Functions



대표적인 자료구조의 종류

- 배열(Array)
- 연결 리스트(Linked List)
- 스택(Stack)
- 큐(Queue)
- 트리(Tree)
- 그래프(Graph)
- 해시 테이블(Hash Table)
- 우선순위 큐(Priority Queue)
- 힙(Heap)

해시 테이블(HASH TABLE)

SET이란?

1. Set 객체는 값 컬렉션으로, 삽입 순서대로 요소를 순회할 수 있습니다.
2. Set 객체는 모든 값을 고유하게 유지하려는 경우에 유용합니다.
3. Set 객체는 iterable(반복 가능한 객체)를 받을 수 있으며, 삽입 순서대로 값의 컬렉션을 만듭니다.

SET 객체는 다음과 같은 기능을 제공합니다.

- `Set.prototype.add(value)` : 값(value)을 Set 객체에 추가합니다.
- `Set.prototype.clear()` : Set 객체에서 모든 요소를 제거합니다.
- `Set.prototype.delete(value)` : 값(value)을 Set 객체에서 제거합니다.
- `Set.prototype.has(value)` : 값(value)이 Set 객체에 존재하는지 확인합니다.
- `Set.prototype.size` : Set 객체의 요소 개수를 반환합니다.

NEW SET()의 기능

```
1  const mySet = new Set();
2  mySet.add(1);
3  mySet.add(2);
4  mySet.add(3);
5  mySet.add(2);
6
7  console.log(mySet.has(2)); // true
8  console.log(mySet.size); // 3
9
10 mySet.delete(2);
11 console.log(mySet.has(2)); // false
12 console.log(mySet.size); // 2
13
14 mySet.clear();
15 console.log(mySet.size); // 0
```


MAP0|란?

1. Map 객체는 키-값 쌍의 컬렉션입니다. 키는 값에 매핑됩니다.
2. Map 객체는 객체의 키가 문자열인 경우와 달리, 어떤 자료형도 키로 사용할 수 있습니다.

MAP 객체는 다음과 같은 기능을 제공합니다.

- `Map.prototype.set(key, value)` : 키(key)와 값(value)을 Map 객체에 추가합니다.
- `Map.prototype.get(key)` : 키(key)에 해당하는 값(value)을 반환합니다.
- `Map.prototype.delete(key)` : 키(key)와 해당하는 값(value)을 Map 객체에서 제거합니다.
- `Map.prototype.has(key)` : 키(key)가 Map 객체에 존재하는지 확인합니다.
- `Map.prototype.size` : Map 객체의 키-값 쌍 개수를 반환합니다.

NEW MAP()의 기능

```
1  const myMap = new Map();
2  myMap.set('a', 1);
3  myMap.set('b', 2);
4  myMap.set('c', 3);
5
6  console.log(myMap.get('b')); // 2
7  console.log(myMap.size); // 3
8
9  myMap.delete('b');
10 console.log(myMap.has('b')); // false
11 console.log(myMap.size); // 2
12
13 myMap.clear();
14 console.log(myMap.size); // 0
```

THE END

THANK YOU FOR WATCHING