

# 5.1 복잡도

## 5.1.1 시간 복잡도

### 빅오 표기법

- 시간 복잡도란 “문제를 해결하는데 걸리는 시간과 입력의 함수 관계”를 가리킨다.
- 어떠한 알고리즘의 로직이 “얼마나 오랜 시간” 이 걸리는지 나타내는데 쓰이며, 보통 빅오 표기법으로 나타낸다.
- 입력 범위  $n$ 을 기준으로해서 로직이 몇 번 반복되는지 나타내는 것이다.
- “가장 영향을 많이 끼치는” 항의 상수 인자를 빼고 나머지 항을 없앤 것이다. 다른 항들이 신경 쓰일수도 있지만 증가 속도를 고려하면 그렇지 않다.
- 입력 크기가 커질수록 연산량이 가장 많이 커지는 항은  $n$ 의 제곱함이고 다른 것은 그에 비해 미미하기 때문에 이것만 신경 쓰면 된다

### 시간 복잡도의 존재 이유

- 효율적인 코드로 개선하는 데 쓰이는 척도가 된다.
- 버튼을 누르고 화면이 나타내는 로직이  $n$ 제곱 시간 복잡도를 가지고 9초가 걸리면  $n$  시간 복잡도를 가지는 알고리즘으로 개선하면 3초가 걸린다.

## 5.1.2 공간 복잡도

- 공간 복잡도는 프로그램을 실행시켰을 때 필요로 하는 자원 공간의 양을 말한다.
- 정적 변수로 선언된 것 말고도 동적으로 재귀적인 함수로 인해 공간을 계속해서 필요로 하는 경우도 포함된다.

## 5.1.3 자료 구조에서의 시간 복잡도

### 평균

▼ 표 5-1 자료 구조의 평균 시간 복잡도

자료 구조	접근	탐색	삽입	삭제
배열(array)	$O(1)$	$O(n)$	$O(n)$	$O(n)$
스택(stack)	$O(n)$	$O(n)$	$O(1)$	$O(1)$
큐(queue)	$O(n)$	$O(n)$	$O(1)$	$O(1)$
이중 연결 리스트(doubly linked list)	$O(n)$	$O(n)$	$O(1)$	$O(1)$
해시 테이블(hash table)	$O(1)$	$O(1)$	$O(1)$	$O(1)$
이진 탐색 트리(BST)	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$
AVL 트리	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$
레드 블랙 트리	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$

## 최악

▼ 표 5-2 자료 구조 최악의 시간 복잡도

자료 구조	접근	탐색	삽입	삭제
배열(array)	$O(1)$	$O(n)$	$O(n)$	$O(n)$
스택(stack)	$O(n)$	$O(n)$	$O(1)$	$O(1)$
큐(queue)	$O(n)$	$O(n)$	$O(1)$	$O(1)$
이중 연결 리스트(doubly linked list)	$O(n)$	$O(n)$	$O(1)$	$O(1)$
해시 테이블(hash table)	$O(n)$	$O(n)$	$O(n)$	$O(n)$
이진 탐색 트리(BST)	$O(n)$	$O(n)$	$O(n)$	$O(n)$
AVL 트리	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$
레드 블랙 트리	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$