



Section1. 복잡도

5.1.1 시간 복잡도

시간복잡도란?

- 문제를 해결하는데 걸리는 시간과 입력의 함수 관계
- 어떠한 알고리즘 로직이 ‘얼마나 오랜시간’이 걸리는지 나타내는데 쓰임.

빅오 표기법

- 범위 n 을 기준으로, 로직이 몇번 반복되는지 나타내는 것.
- 가장 영향을 많이 끼치는 항의 상수인자를 빼고, 나머지항을 없앤 것.
- 입력 크기가 커질수록 연산량이 가장 많이 커지는 항은 n 의 제곱 항.

시간 복잡도의 존재이유?

- 효율적인 코드로 개선하는 척도이기 때문에.

5.1.2 공간 복잡도

공간복잡도란?

- 프로그램을 실행시켰을 때 필요로 하는 자원 공간의 양.
- 정적 변수로 선언된 것 말고도 동적으로 재귀적인 함수로 인해 공간을 계속해서 필요로 할 경우도 포함.

5.1.3 자료 구조에서의 시간복잡도

- 자료구조 사용시 시간복잡도를 잘 생각해야 함.

▼ 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	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$

▼ 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	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$