

복잡도

자료 구조 (Data Structure)

효율적으로 데이터를 관리하고 수정, 삭제, 탐색, 저장할 수 있는 데이터 집합

시간 복잡도

문제를 해결하는데 걸리는 시간과 입력의 함수 관계

어떠한 알고리즘의 로직이 얼마나 오랜 시간이 걸리는지를 나타내는데 사용

보통 빅오 표기법으로 표현함

| 빅오 표기법

입력범위 n 을 기준으로 로직이 몇 번 반복되는지 나타내는 것

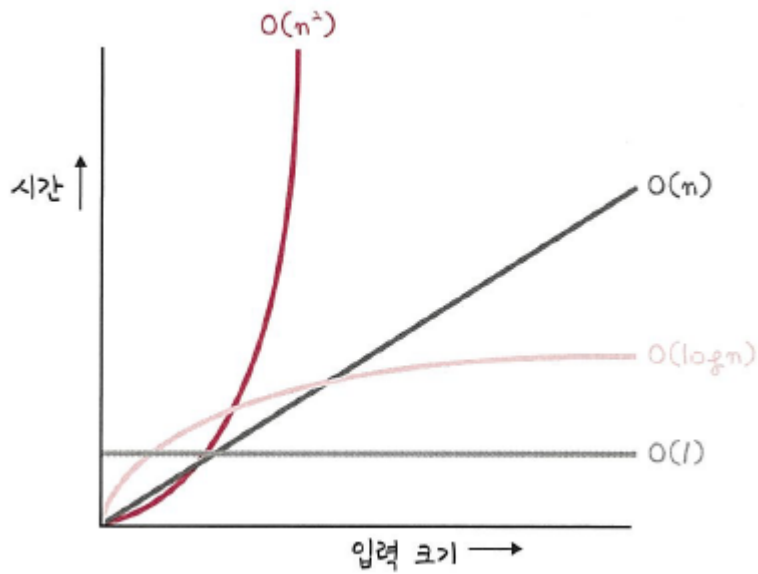
가장 영향을 많이 끼치는 항의 상수 인자를 빼고 나머지 항을 없앤 것

입력의 크기가 커질 수록 해당 항의 연산량이 가장 많이 커지고, 다른 항들은 그에 비해 미미하므로 신경 쓰지 않아도 된다는 이론

시간 복잡도의 존재 이유

효율적인 코드로 개선하는데 쓰이는 척도

시간 복잡도의 속도 비교



$O(1)$ 과 $O(n)$ 은 입력 크기가 커질 수록 차이가 많이 남
 $O(n^2)$ 은 훨씬 더 큰 차이를 보임

공간 복잡도

프로그램을 실행시켰을 때 필요로 하는 자원 공간의 양

정적 변수로 선언된 것 말고도 동적으로 재귀적인 함수로 인해 공간을 계속해서 필요로 할 경우도 포함

자료구조에서의 시간 복잡도

일반적으로 평균, 최악의 시간 복잡도를 고려하여 사용

- 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 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	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$