

# 복잡도 (Program Complexity)

## 복잡도란?

- 입력 N에 대해서 프로그램이 실행되어 연산하는 데에 사용되는 시•공간적 자원
- 당연히 N이 커질수록 복잡도는 커진다.
- 프로그램에 따라서 복잡도는 완전히 달라진다.
- 이를 줄이기 위해 사용하는 것이 최적화, 알고리즘이다.

## 공간복잡도 Space Complexity : S(P)

- $S(P) = \text{Fixed space}(c) + \text{Variable space}(Sp(I))$  === 정적 공간과 동적 공간
- **Fixed space(c):** 입출력에 상관없는 공간 (==Auxiliary Space: 보조 공간, 임시 공간 등 동치)
  - 코드의 자체적인 공간
  - 프로그램 내의 선언된 변수들의 크기
  - 크기를 미리 알 수 있는 자료형 등의 공간(ex. int, char, double...)
- **Variable space (Sp(I)):** 입출력에 영향을 받는(종속된) 공간
  - run time 에서 instance에 의해 결정되는 공간 (instance: 쉽게 말해 선언된 실질적 객체들)
  - instance가 어떻게 되었냐에 따라 변화
  - ex. function call이 많을수록 공간 소모가 많이 발생
  - 메모리 제한에 영향을 주는 요소

## 시간 복잡도(Time Complexity)

- $T(P) = \text{Compile time}(T_c) + \text{Run time}(T_p)$  === 컴파일 시간과 실행 시간
- 중요한 것은 Run time!!
- 실행 시간은 입력 값에 따라 연산 수행 시간이 달라진다.

## 환경에 따라 프로그램 수행 시간은 달라질 수 있으므로 기본 연산의 실행 횟수로 시간을 평가함.

기본 연산의 종류

1. 데이터 입출력 - copy, move, swap, etc.
2. 산술 연산 - add, subtract, multiply, divide, etc.
3. 제어 연산 - if, while, for, etc.

## 시간 복잡도의 표현 방법

- 기본적으로 점근적 표기법을 따름 (최고차항만을 표기하는 것)
- 최선의 경우:  $\Omega(g(n))$ , 빅-오메가 표기법을 사용. 최소 이 정도의 시간이 걸린다.
- 평균의 경우:  $\Theta(g(n))$ , 빅-세타 표기법을 사용, 평균적으로 이 정도의 시간이 걸린다.
- 최악의 경우:  $O(g(n))$ , 빅-오 표기법을 사용, 최대 이 정도의 시간이 걸린다.

제일 중요한 것은?  $\Rightarrow$  **big-O notation!!**

$\Rightarrow$  프로그래밍 시, 최악의 경우를 상정하여 프로그래밍 해야 한다.

프로그램에 어떤 입력이 들어와도 어느 한계 이상의 연산 수행을 하진 않는다는 보장 가능 (물론 오류 나거나, 환경적 제약이 생기면 당연히 안 되는거고)

$$T(n)=4n^2+20n+100 \Rightarrow O(N)=O(n^2)$$

시간 복잡도의 종류 (N is input)

$O(1)$ : (상수 시간) 상수 시간에 해결할 수 있음.

- 배열의 인덱스 접근, 다항함수의 함수값 구하기 등

$O(N)$ : (선형 시간) 데이터 수 만큼 걸림.

- 계수 정렬, 합계, 평균, 연결 리스트 탐색 (선형 탐색•순회)

$O(\log N)$ : (로그 시간) 연산할 횟수가 점점 절반으로 줄어듦.

- 이분 탐색 (Binary search), 숫자 업다운 게임 (이것도 결국 이분 탐색)

$O(N \log N)$ : (선형로그 시간) 선형 연산과 로그 연산이 데이터에 적용됨.

- 퀵 정렬, 병합 정렬,

$O(N^2, N^c)$ : (다차 시간) 입력에 몇제곱 꼴로 나타남. 반복문 중첩시 차수 증가

- 버블 정렬, 선택 정렬, 삽입 정렬, 행렬곱,

$O(c^N)$ : (지수 시간) 데이터가 지수형태로 나타남. 재귀함수가 이같은 꼴을 나타낸다.

- 피보나치 재귀 함수,

$O(N!)$ : (팩토리얼, 계승 시간) 데이터를 지속적으로 사용하는 경우가 곱해진 경우. 한마디로 복잡.

- bogo sort, stupid sort

기본적으로 이정도, 이들을 적절히 조합한 여러 복잡도가 존재한다.

아래는 입력과 수행시간  $N$ ,  $O(N)$ 의 그래프

