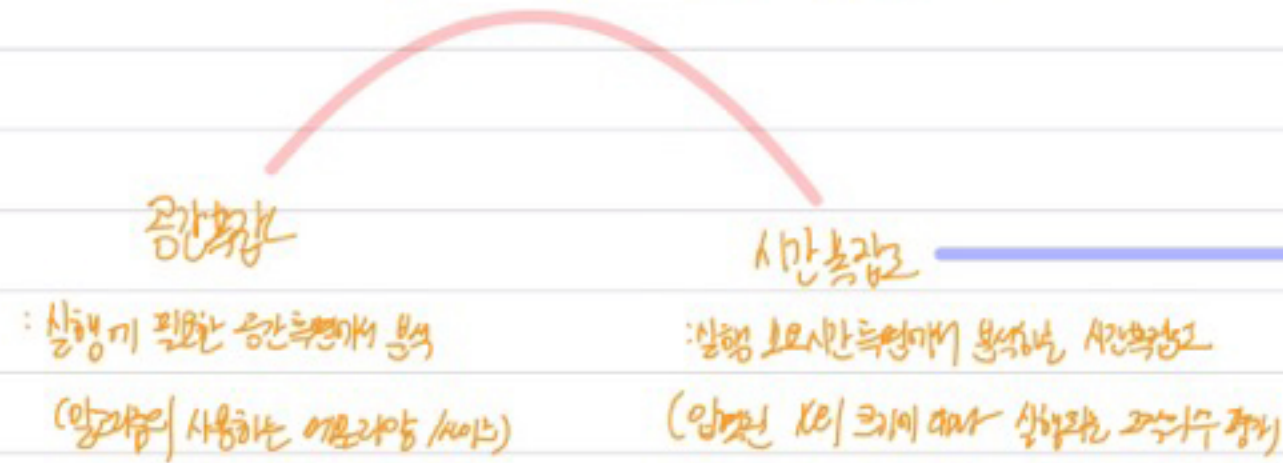


알고리즘: 어떤 문제나 목적을 달성하기 위해 가져야 하는 여러 과정

↳ 알고리즘의 실행시간은 컴퓨터가 알고리즘을 실행하는 데 필요한



최소 시간 복잡도

① 최선: 최선이지만  $f(n)$ 이  $g(n)$ 보다 항상 작거나  $f(n)$ 이  $g(n)$ 보다 느리게 증가  
(최소 이항 시간 복잡도)

② 최악: 최악 입력 크기  $n$ 에  $f(n)$ 은  $g(n)$ 에 비례하므로  $g(n)$ 만큼 빠름  
(시간이 최대 이항을 함)

③ 평균: 최악  $f(n)$ 이  $g(n)$ 에 대하여 양쪽으로 제곱되고  $f(n)$ 이  $g(n)$ 과 비례  
(시간이 평균적으로 빠름 함)

효율성 ↑ 그래프 ↑ = 비효율.  
↳ 최악은 최악의 시간만 알려  
알고리즘이 실행이 잘 될 것인지  
시작한 평균 분석이 있어야  
⇒ 그래서 '최소 사용'

비율 표기법

$O$  (증가함수)

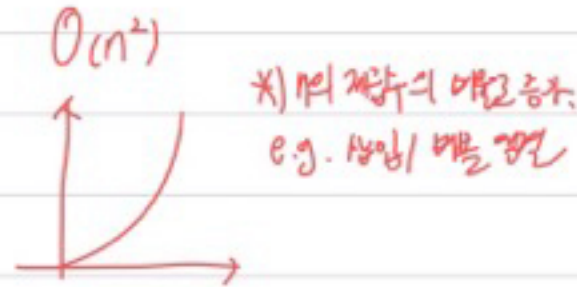
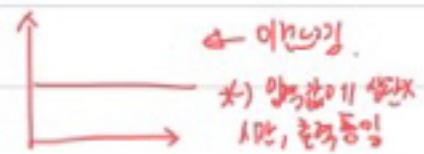
↳ 입력 데이터의 크기 세례로 증가하여 나타낼 수 있는 가장 낮은

↳ 연산 횟수가 증가하면 최상한 것을 제외한 모든 것들이 = 필요한 연산에

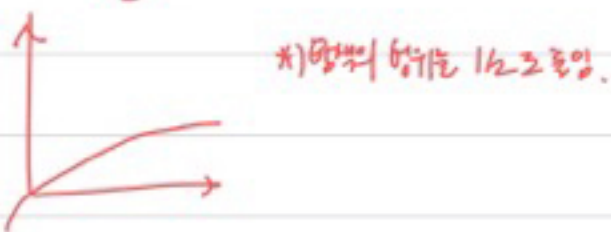
시간 복잡도 ↑

$O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(2^n) < O(n!)$

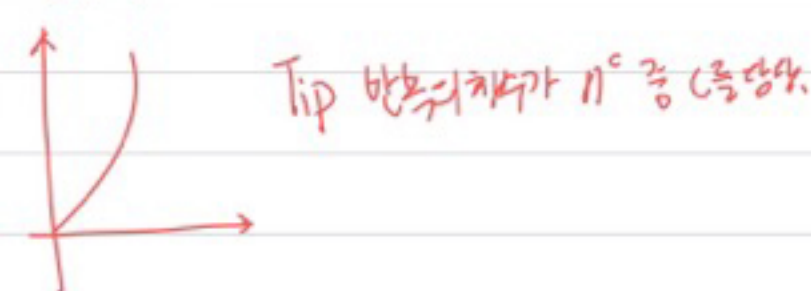
$O(1)$ : 상수 시간



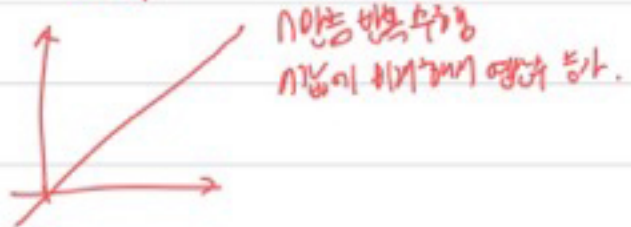
$O(\log n)$



$O(n^3)$



$O(n)$



$O(n^4)$

