

## 프로그램의 효율성 대두

커지는 데이터 input에 맞는 효율성 측정 필요  
 프로그램 간의 성능 측정의 비교 기준이 절실히  
 이러한 비교 기준은 시간 효율성과 공간 효율성으로 나뉨

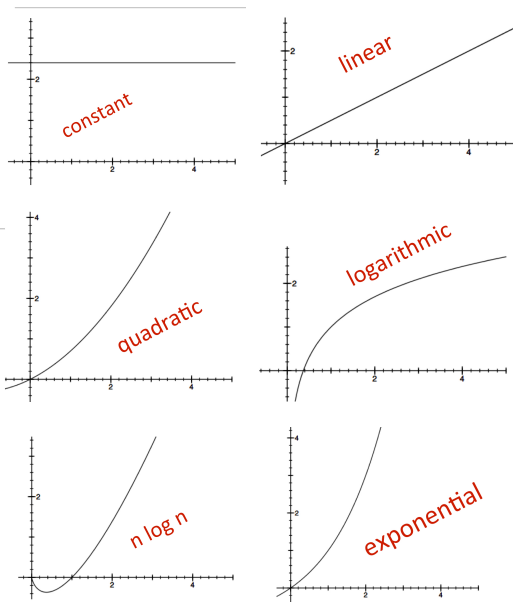
## 프로그램의 측정 방법 (Big-0)

타이머 (import time)나 직접 세는 거 의외에,  
 "최악의 케이스" 를 고려하는 Big-0 Notation을 사용  
 Big-0는 프로그램의 upper bound를 측정하는 것

## 측정의 세부사항

Big-0는 기본적으로 dominant terms를 사용  
 즉, 가장 변화가 큰 차수를 우선적으로 고려  
 Big-0는 그 효율성을 순서로,

↑  
 $O(1)$  = constant  
 $O(n)$  = linear  
 $O(n^2)$  = Quadratic  
 $O(\log n)$  = Logarithmic  
 $O(n \log n)$   
 $O(2^n)$  = exponential



## Law of addition

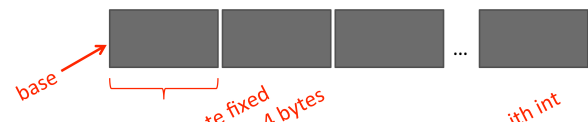
sequential statement인 알고리즘인 경우,  
 같은 indent level 상에 있으면 두 알고리즘을 합한다.

## Law of multiplication

nested statement/loop인 알고리즘인 경우,  
 다른 indent level 상에 있으면 두 big-0를 곱한다.

## 참조 테크닉 (파이썬)

파이썬에서는 접근(access)은 상수항의 Big-0를 갖는다.  
 파이썬의 데이터는 starting point를 기준,  
 하나씩 건너갈 때마다 4bytes를 곱하는 형식으로 참조



만약 list가 다차원 리스트일 경우에는  
 실제 data가 include되지 않고 이를 참조한 pointer만!

