

컴퓨터 알고리즘 (Computer Algorithms)

우진운

제 1 장 알고리즘

- 학습목표
- 알고리즘이란?
- 알고리즘 표기
- 성능 분석과 측정

이 과목의 학습목표

- Design(설계)
 - 알고리즘을 설계하는 기법을 배운다.
- Analysis(분석)
 - 알고리즘을 분석하여 시간/공간 복잡도를 구하는 방법을 배운다.
- Computational Complexity(계산복잡도)
 - 문제를 분석하여 계산적 복잡도를 구하는 방법을 배운다.

알고리즘(Algorithm)이란?

- 문제에 대한 답을 찾기 위하여 계산하는 절차
- 단계별로 주의 깊게 설계된 계산과정
- 입력을 받아서 출력으로 전환시켜주는 일련의 계산절차

알고리즘의 연구영역

- 어떻게 알고리즘을 고안하는가?
 - 설계 기법, 설계 전략
- 어떻게 알고리즘을 확인하는가?
 - 입력에 대해 정확한 출력을 입증(validation)
- 어떻게 알고리즘을 분석하는가?
 - 알고리즘 분석, 성능 분석
- 어떻게 프로그램을 검사하는가?
 - 디버깅, 프로파일링(성능 측정)

알고리즘의 표기

- 자연어: 한글 또는 영어
- 프로그래밍언어: C, C++, Java 등
- 의사코드(Pseudo-code)
 - 직접 실행할 수 있는 프로그래밍언어는 아니지만, 거의 실제 프로그램에 가깝게 계산과정을 표현할 수 있는 언어
- 알고리즘은 보통 의사코드로 표현한다.
- 이 강의에서는 Java와 의사코드(pseudo-code)를 사용한다.

- 예: Java

```
void SelectionSort(int a[], int n)    // 강의자료 program(lec1-1)
// Sort the array a[0:n-1] into nondecreasing order.
{
    for (int i=0; i<n; i++) {
        int j = i;
        for (int k=i+1; k<n; k++)
            if (a[k]<a[j]) j=k;
        int t = a[i]; a[i] = a[j]; a[j] = t;
    }
}
```

- 자료구조: 일련의 자료들을 조직하고 구조화하는 것
예: 배열, 연결리스트, 트리 등
- 알고리즘과 자료구조는 깊은 연관성을 갖는다.
예: 순차탐색과 이진탐색

성능 분석과 측정

- 복잡도(complexity): 프로그램은 실행하는 하드웨어에 따라 실행 시간이 다르므로 컴퓨터에 관계없이 시간과 공간을 추산하는 것(시간 복잡도, 공간 복잡도)
- 성능평가: 성능 분석(사전 예측), 성능 측정(사후 검사)

(1) 성능 분석

1) 공간 복잡도

- 필요한 공간의 크기, 고정공간과 가변공간

2) 시간 복잡도

- 프로그램이 실행될 때 필요한 실행문의 실행 횟수
- 방법: 실행문의 빈도수를 계산. 예: 표 1.1

실행문의 빈도수를 계산

statement	s/e	frequency	total steps
float Sum(float a[], int n)	0	–	0
{			
float s = 0.0;	1	1	1
for (int i=0; i<n; i++)	1	n+1	n+1
s += a[i];	1	n	n
return s;	1	1	1
}			
Total			2n+3

3) 점근 표기법(O , Ω , Θ)

- 정확한 단계수를 계산하는 것이 힘들며, 그 의미도 명확하게 비교할 수 있는 것이 아니므로, 대략적으로 수행시간이나 공간을 예측하기 위해 사용한다.

- 정의: “빅오”

모든 n , $n \geq n_0$ 에 대해 $f(n) \leq cg(n)$ 인 두 개의 상수 c 와 n_0 가 존재한다면 $f(n) = O(g(n))$ 이다.

예: $f(n) = 2n + 3$ 일 때, $g(n) = n$ 으로 두면

$$f(n) = 2n + 3 \leq 6n, c = 6, n_0 = 1$$

따라서 $f(n) = O(g(n)) = O(n)$ 이다.

정리 1.2: 만약 $f(n) = a_m n^m + \dots + a_1 n + a_0$ 이면,
 $f(n) = O(n^m)$ 이다.

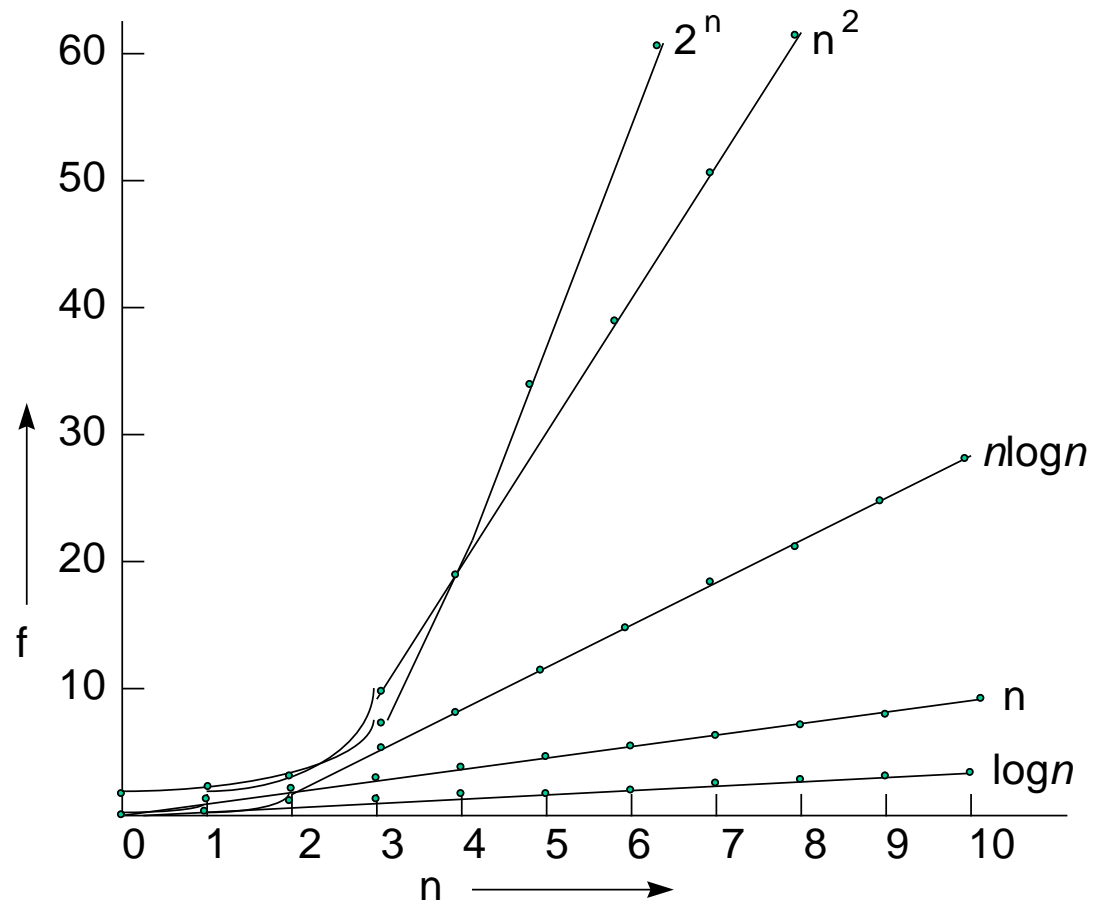
- 정의: “오메가”

모든 n , $n \geq n_0$ 에 대해 $f(n) \geq cg(n)$ 인 두 개의 상수 c 와 n_0 가 존재한다면 $f(n) = \Omega(g(n))$ 이다.

- 정의: “세타”

모든 n , $n \geq n_0$ 에 대해 $c_1 g(n) \leq f(n) \leq c_2 g(n)$ 인 세 개의 상수 c_1 , c_2 와 n_0 가 존재한다면 $f(n) = \Theta(g(n))$ 이다.

- $O(1)$: 상수형 (constant)
- $O(\log n)$: 로그형
- $O(n)$: 선형 (linear)
- $O(n^2)$: 제곱형 (quadratic)
- $O(n^3)$: 입방형 (cubic)
- $O(2^n)$: 지수형 (exponential)



(2) 성능 측정

- `System.currentTimeMillis()` 함수를 사용

예:

```
long start = System.currentTimeMillis();  
SelectionSort(arr, n);    // 선택 정렬 수행  
long stop = System.currentTimeMillis();  
runTime = stop - start;    // 걸린 시간 측정
```

* 시간이 0인 경우, 여러 번 반복한 후 전체 시간을 반복 횟수로 나눈다.

(3) 테스트 데이터의 생성

- 일반적으로 쉽게 테스트 데이터를 얻을 수 없다.
- 최악의 경우: 대규모의 무작위 데이터에 대한 결과값들 중 최대값
평균의 경우: “ ” 평균값