Data - Structure - Study(DSS)

Winter Vacation Soo-lab Study

TEAM Kai.saekies

발표자 : 김영웅 양근제

2020.01.03 (FRI)

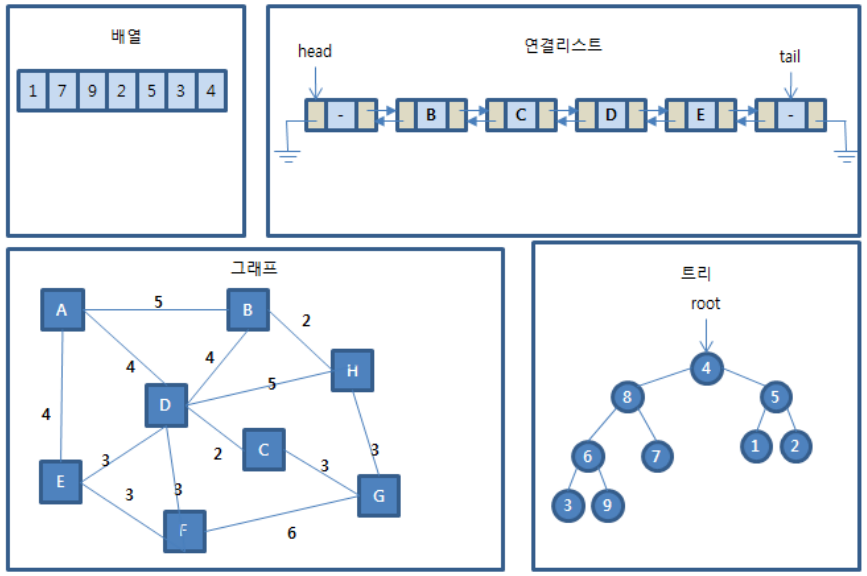
Index

Data Structure

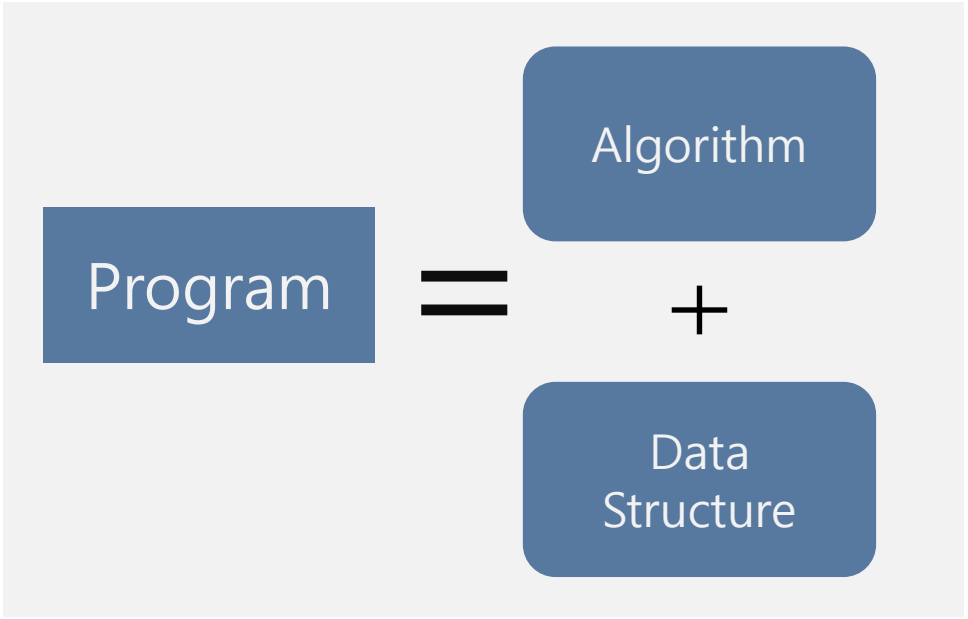
컴퓨터 과학에서 효율적인 접근 및 수정을 가능케 하는 자료의 조직, 관리, 저장을 의미.

↓

데이터 값의 모임, 데이터 간의 관계, 그리고 데이터에 적용할 수 있는 함수나 명령을 의미

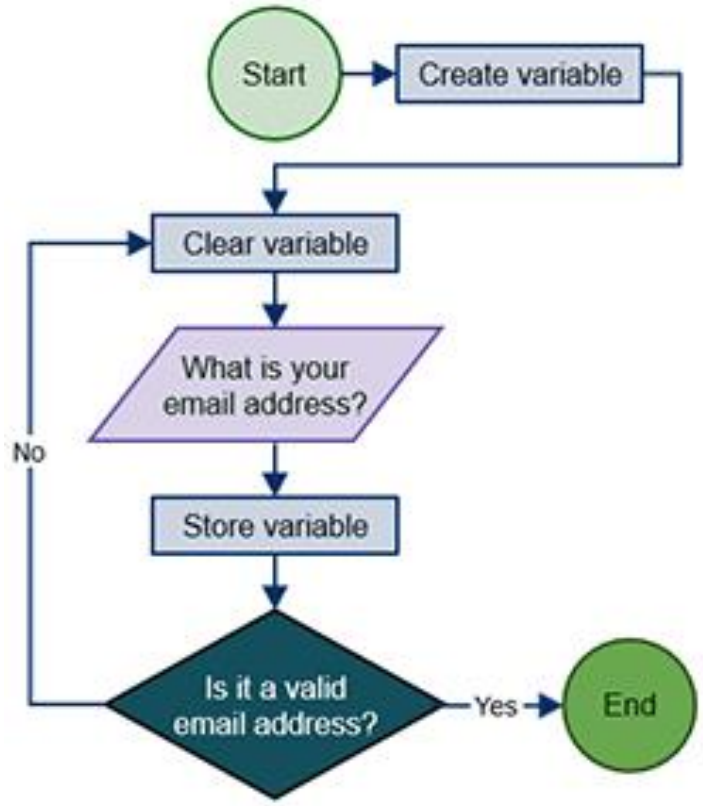


다양한 형태의 자료구조:array, linked list, graph, tree



Algorithm

- 어떤 문제가 주어졌을 때, 문제를 해결하기 위한 단계적 절차



Email 발송 알고리즘

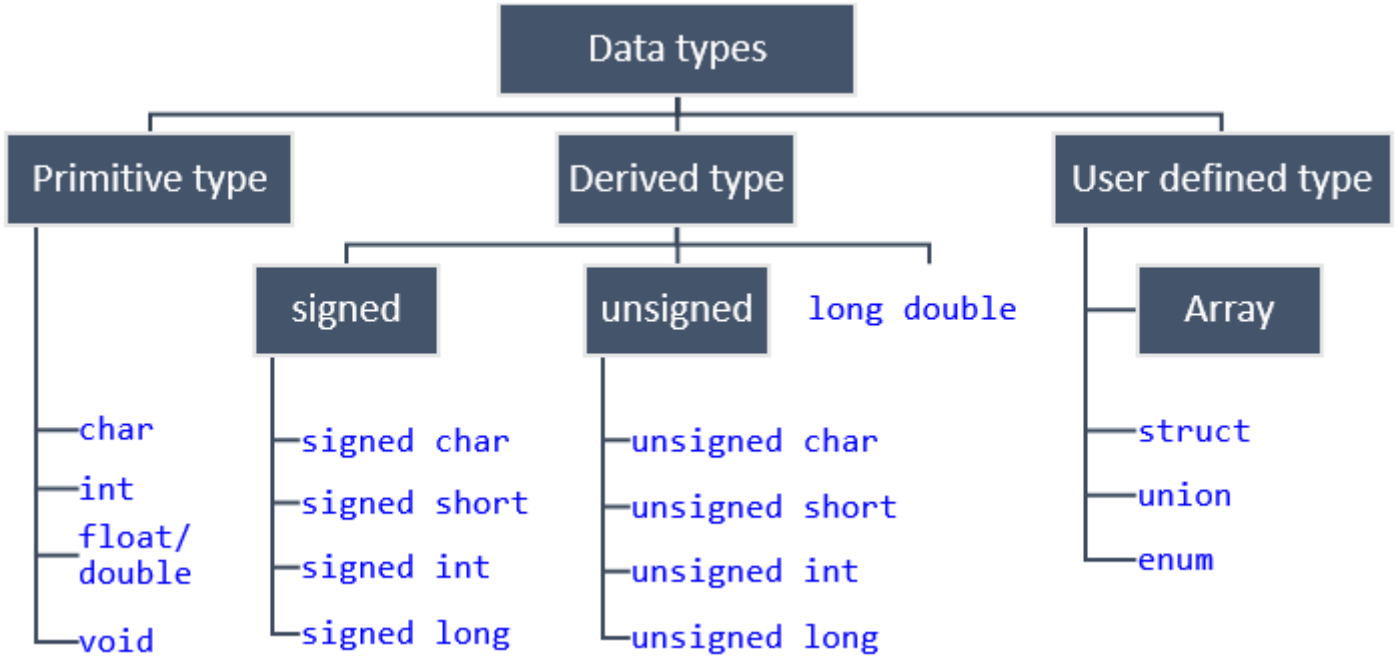
알고리즘을 기술하는 4가지 방법

- 자연어
- Flowchart
- Pseudo – code
- Programming language

Data Type(자료형)

Data Type

- 여러 종류의 데이터를 식별하는 분류
- 넓은 의미로 해당 자료형에 대한 가능한 값, 해당 자료형에서 수행을 마칠 수 있는 명령들, 데이터의 의미, 해당 자료형의 값을 저장하는 방식을 의미

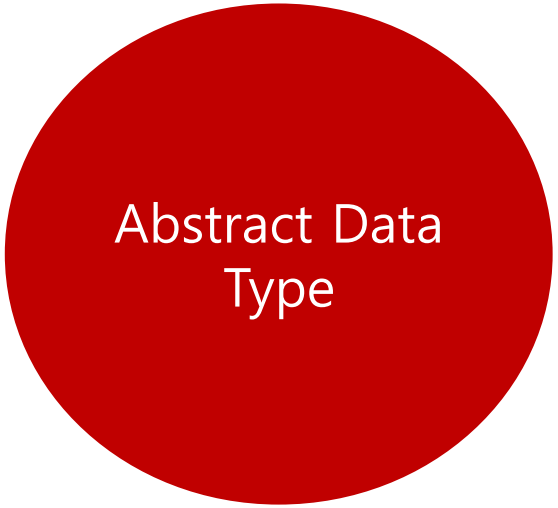


다양한 자료형(C)

Data Type(자료형)

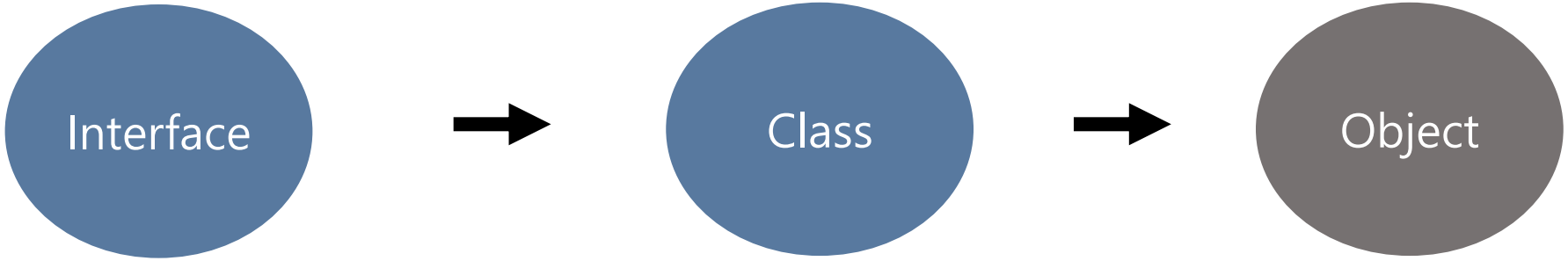
Abstract Data Type(ADT) – 추상 자료형

- 프로그램의 대상이 되는 사물 또는 대상을 추상화 하여 정의



ADT의 장점

- 설계와 구현의 분리
- Data를 사용할 때 모든 정보를 다 이해하지 않아도 된다.
  정보은닉 기법(information hiding)



Java의 추상 자료형

알고리즘 성능분석

- **Time Complexity(시간 복잡도)**
  알고리즘에 사용되는 연산 횟수 ≠ 실행 시간

```
for(int i = 0; i < n ; i++)
{
    printf("hello!");
}
```

n

```
for(int i = 0; i < n ; i++)
{
    for(int j = 0; j < n;j++)
    {
        printf("hello");
    }
}
```

n^2

- **Space Complexity(시간 복잡도)**
  알고리즘에 사용되는 메모리의 총량

```
int n[2] = {1,1};
```

n

```
int n[2][2] = {{1,1},{1,1}};
```

n^2

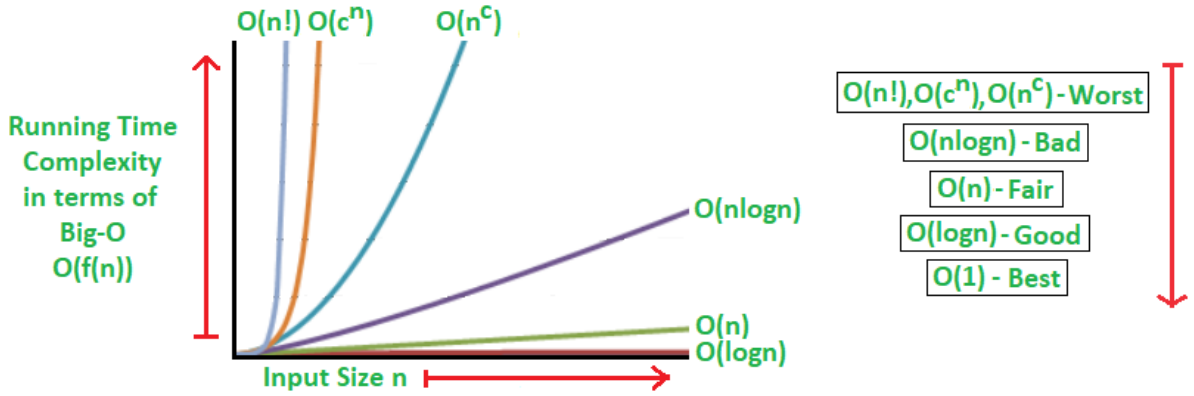
알고리즘 성능분석

시간 복잡도 표현 방식

- **Big – o notation** – 상한 표기
- Big – omega notation – 하한 표기
- Big – theta notation – 상한과 하한 표기

빅오 표기법 : 두 함수 f(n)과 g(n)이 주어졌을 때 모든 n>n0에 대하여 |f(n)| <= c*|g(n)|을 만족하는 2개의 상수 c와 n0가 존재하면 f(n)=O(g(n))이다.

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



알고리즘 성능분석

Big - O	Example
O(1)	단순 출력문 , push, pop
O(log n)	Tree
O(n log n)	병합 정렬
$O(n^2)$	Nested for, Bubble sort
$O(2^n)$	피보나치 수열
O(n!)	n순열