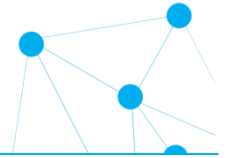


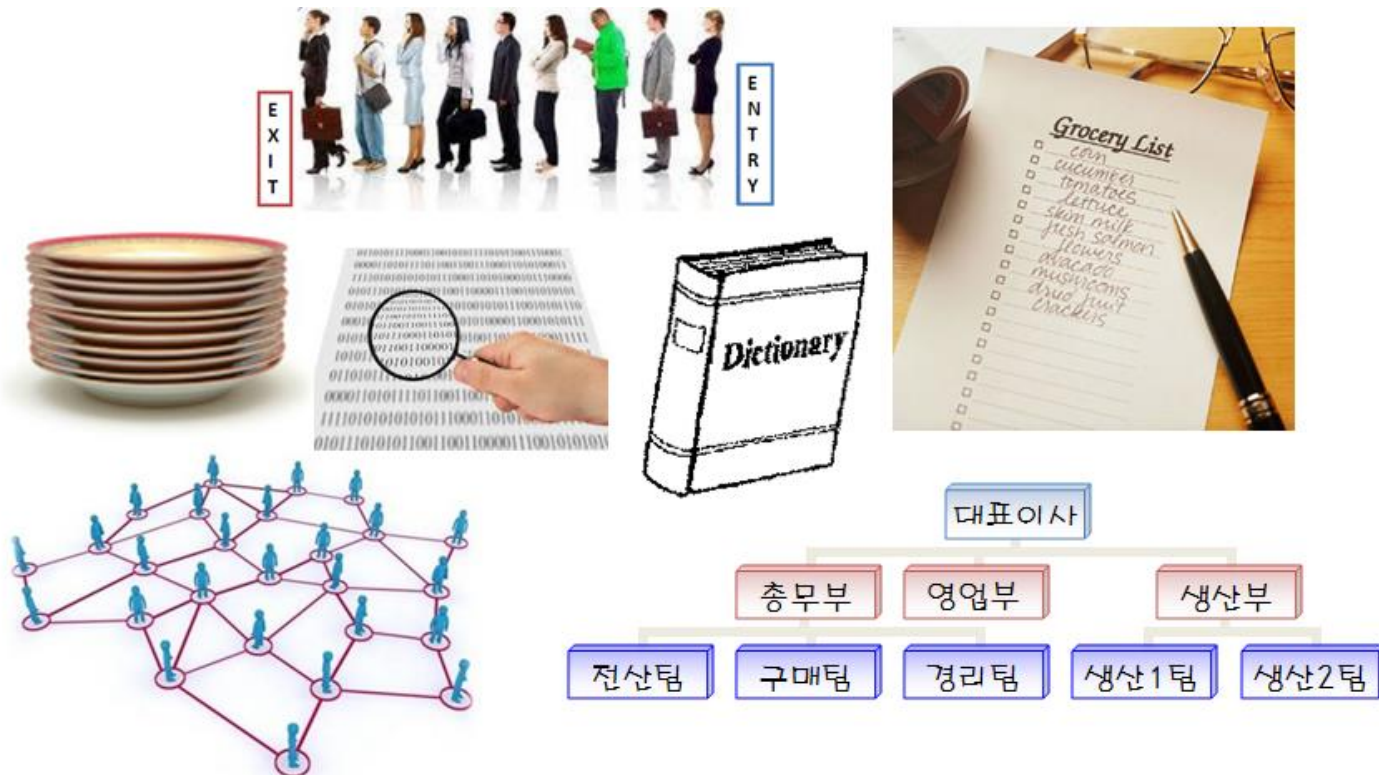
# 데이터구조 1장

## 01-1. 자료구조의 개념

# 자료구조



- 일상 생활에서 자료를 정리하고 조직화하는 이유는?
  - 사물을 편리하고 효율적으로 사용하기 위함
  - 다양한 자료를 효율적인 규칙에 따라 정리한 예



# 컴퓨터에서의 자료구조



- 자료구조(Data Structure)
  - 프로그램에서 자료들을 정리하여 보관하는 여러 가지 구조
  - 컴퓨터 과학에서 자료의 효율적인 접근 및 수정을 가능케 하는 자료의 조직, 관리, 저장의 의미
  - 컴퓨터에서 원하는 것을 효율적으로 표현하기 위해 필요
    - 시간효율/공간효율

메모리를 어떻게 구조화해서 사용할 것인가?

기억장소의 활용이나 처리 시간의 단축 등

많은 자료구조를 알수록 개발에 유리

# 자료구조의 분류

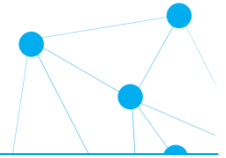


자료들 사이 앞뒤 관계가 1:1인 경우

선형 구조	선형 리스트(Linear List, 배열), 연결 리스트(Linked List), 스택(Stack), 큐(Queue), 데크(Deque)
비선형 구조	트리(Tree), 그래프(Graph)

자료들 사이 앞뒤 관계가 1:n(계층구조),  
n:m(망구조)인 경우

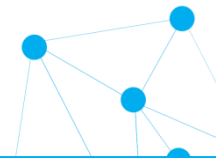
# 일상생활과 자료구조



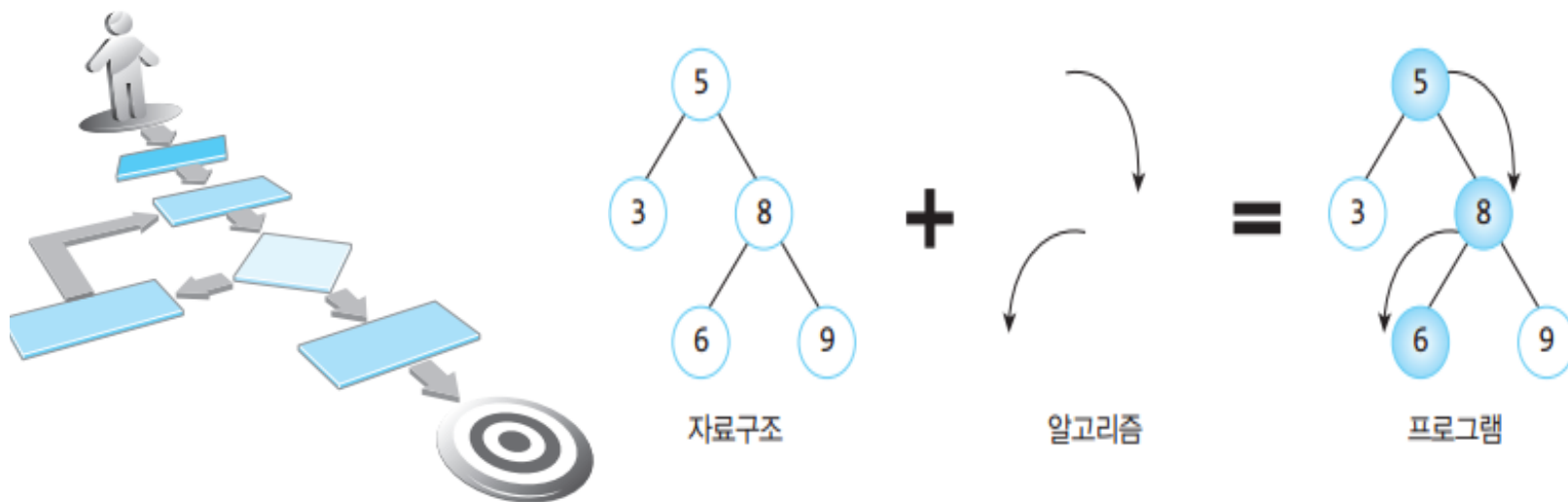
일상생활에서의 예	해당하는 자료구조
그릇을 쌓아서 보관하는 것	스택
마트 계산대의 줄	큐
버킷 리스트	리스트
영어사전	사전
지도	그래프
컴퓨터의 디렉토리 구조	트리

현실에서 행했던 이런 행동들을 컴퓨터상에서 행해야 하는 상황에 직면하게 됨. 데이터가 늘어나면 이를 효율적으로 관리하기 위한 다양한 방법이 필요함.

# 컴퓨터 프로그램



- 컴퓨터 프로그램은 무엇으로 이루어져 있나?
  - 프로그램 = 자료구조 + 알고리즘



## 자료구조

“프로그램이란 데이터를 표현 하고,

표현에는 저장의 의미가 포함된다!

## 알고리즘

그렇게 표현된 데이터를 처리 하는 것이다.”

# 알고리즘



- 주어진 입력을 이용하여 원하는 결과를 만드는 과정
- 자료구조를 어떻게 처리할 것인가를 나타내는 처리 방법
- 컴퓨터로 문제를 풀기 위한 단계적인 절차
- **알고리즘의 조건**
  - 입력 : 0개 이상의 입력이 존재하여야 한다.
  - 출력 : 1개 이상의 출력이 존재하여야 한다.
  - 명백성 : 각 명령어의 의미는 모호하지 않고 명확해야 한다.
  - 유한성 : 한정된 수의 단계 후에는 반드시 종료되어야 한다.
  - 유효성 : 각 명령어들은 실행 가능한 연산이어야 한다.



# 알고리즘의 기술 방법



- (1) 자연어로 표기된 알고리즘
- (2) 흐름도(순서도)로 표기된 알고리즘
- (3) 의사코드로 표현된 알고리즘(=pseudo, 슈도코드)
  - 알고리즘 기술에 가장 많이 사용
  - 알고리즘의 핵심적인 내용에만 집중

```
ArrayMax(list, N):  
    largest←list[0]  
    for i←1 to N-1 do  
        if list[i]>largest  
            then largest←list[i]  
    return largest
```

- (4) 특정 프로그래밍 언어로 표현된 알고리즘

## 01-2. 자료의 표현(수표현)

# 진법변환(2진수→10진수)

- 컴퓨터는 2진 시스템
  - 복잡한 2진수 표현을 간단하게 8진수, 16진수로 표현하여 사용
- 10진수의 표현
  - 0~9 열개의 기호로 표현

$$(1234.56)_{10} = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0 + 5 \times 10^{-1} + 6 \times 10^{-2}$$

- 2진수의 표현
  - 0, 1 두개의 기호를 사용하여 표현

$$(110.11)_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$$

# 진법변환(10진수→2진수)



- 컴퓨터는 2진 시스템 - 10진수의 2진수 변환필요
- 10진수의 변환
  - 정수부분 : 해당 진수로 나누어서 나머지를 역순표현

〈정수 부분〉

2진수
$\begin{array}{r} 2 \overline{) 47} \\ 2 \overline{) 23} \dots 1 \\ 2 \overline{) 11} \dots 1 \\ 2 \overline{) 5} \dots 1 \\ 2 \overline{) 2} \dots 1 \\ 1 \dots 0 \end{array}$
$(47)_{10} = (101111)_2$

- 소수부분 : 해당 진수를 곱해서  
결과의 정수부분을 차례대로 표  
현(0 또는 반복되는 수가 나올  
때까지 진행)

〈소수 부분〉

2진수		
$\begin{array}{r} 0.625 \\ \times 2 \\ \hline 1.250 \end{array}$	$\begin{array}{r} 0.250 \\ \times 2 \\ \hline 0.5 \end{array}$	$\begin{array}{r} 0.5 \\ \times 2 \\ \hline 1.0 \end{array}$
$(0.625)_{10} = (0.101)_2$		

10진수 2진수  
변환

실수의 고정소수점 방식 표현

$$(47.625)_{10} \rightarrow (101111.101)_2$$

- $2^K$  값
  - 2의 배수 값을 이용
  - 1, 2, 4, 8, 16, 32, 64....

연습문제4

$$123 = 64 + 32 + 16 + 8 + 2 + 1$$

의 2의 배수승들로 표현가능

$$\text{즉, } 2^6 + 2^5 + 2^4 + 2^3 + 2^1 + 2^0$$

해당 자리만 1로 표현하면, 123은

1	1	1	1	0	1	1
---	---	---	---	---	---	---

실제 컴퓨터에 표현할 땐 바이트 크기가 중요! 크기에 따라 0 추가

- 컴퓨터의 수는 2진수 – 사람이 다루기 어려움
  - 복잡한 2진수 표현을 간단하게 8진수, 16진수로 표현하여 사용
- 8진수( $2^3 = 8$ ) : 2진수 세 자리가 8진수 한 자리에 해당
- 16진수( $2^4=16$ ) : 2진수 네 자리가 16진수 한 자리에 해당
  - 16진수 기호 : 0~9, A(10), B(11), C(12), D(13), E(14), F(15)

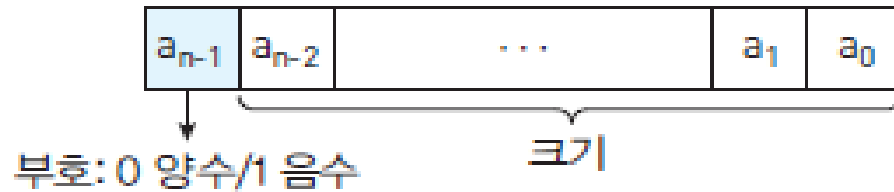
# 정수의 표현

- 처음 1 Bit에 부호 나머지 Bit에 정수 표현
- 부호 없는 수(unsigned)
  - 0을 포함하여 양정수만 이진수 변환 그대로 표현
- 부호 있는 수(signed)
  - 처음 1비트를 부호비트로 사용(0 양수, 1 음수)
  - 부호화 크기(signed magnitude) = 부호화 절대치
  - 부호화 1의 보수(1's complement)
  - 부호화 2의 보수(2's complement)



- 부호화 절대치

- 부호비트에 부호를 채우고 나머지 비트에 2진수를 채운다.



- 문제점

- 두 개의 0이 존재함(음수 0, 양수 0)
    - 연산시 부호에 따라 계산 알고리즘이 달라짐

- 보수

- 보충하는 수(보호하는 수)
  - R의 보수 : R이 되기 위해 필요한 수
- R진법 수의 보수는 (R의 보수)와 (R-1의 보수) 존재
  - 10진수는 10의 보수와 9의 보수
  - 2진수는 2의 보수와 1의 보수
- R의 보수 = (R-1의 보수) + 1

- 2의 보수

- 컴퓨터 정수 표현의 표준 방법

- 표현범위

종류	범위	n=8	n=16	n=32
부호화 절대치법	$-2^{n-1}+1 \sim +2^{n-1}-1$	$-127 \sim +127$	$-32767 \sim +32767$	$-2^{31}+1 \sim +2^{31}-1$
부호화 1의 보수법				
부호화 2의 보수법	$-2^{n-1} \sim +2^{n-1}-1$	$-128 \sim +127$	$-32768 \sim +32767$	$-2^{31} \sim +2^{31}-1$

# 실수의 표현



- **부동소수점 표현(floating-point)**

- 소수점의 위치를 변경하여 지수를 조정하는 표현법
- 제한된 비트로 큰 숫자 또는 정밀한 숫자를 표현하기 위해 사용 (즉, 유효한 숫자의 범위를 늘리기 위해)
- 연산절차가 복잡하고 많은 시간이 걸리며 하드웨어적으로 복잡(컴퓨터에 부담이 되는 표현법임)

> 10진 부동소수점 수

$$976,000,000,000 = 9.76 \times 10^{11}$$

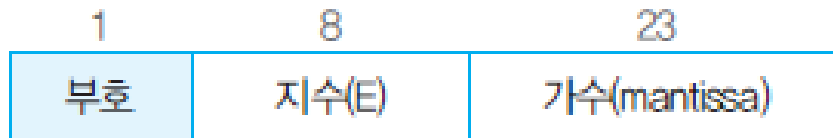
부호: \_\_, 가수: \_\_ 기수: \_\_ 지수: \_\_

$$-0.000000000000976 = -9.76 \times 10^{-12}$$

부호: \_\_, 가수: \_\_ 기수: \_\_ 지수: \_\_

- 2진 부동소수점수(IEEE 754 형식)

- 기수는 2 (표현 생략)
- 단정도 형식(32bits)
- 배정도 형식(64bits)



(a) 단정도 형식



(b) 배정도 형식

- 정규화

- 같은 수에 대한 부동소수점 표현이 여러가지 존재
- 여러 가지 표현법 중 표준 필요
- 정규화 과정 진행(가수와 지수부에 모두 정규화 과정 필요)

- 가수의 정규화

- 가수를  $1.bbb...b$  형식으로 표현할 수 있도록 지수를 조정
- 가수를 표현할 때 소수의  $bbb...b$ 만 표현(1생략 : 표현범위 넓어짐)
- 숫자 0의 경우 모두 0

- 지수의 정규화

- 실수의 연산시 지수에 음수가 있는 경우 표현 및 연산 복잡
- 지수를 부호없는 수로 표현
- 바이어스 지수 (biased exponent)
- 바이어스수를 실제 지수에 더하여 계산