

1

이산수학 개념

1 이산수학

이산적인 수학구조에 대해 연구하는 학문

- ▶ 실수와 같이 연속적인 성질을 가진 값에 대한 것이 아니라 자연수, 정수와 같이 하나하나 구분되는 집합에 대해 공부하는 학문

* 이산수학의 특징

- 유한하거나 셀 수 있음

1 이산수학

- ◆ 명제, 논리의 참과 거짓, 집합의 포함, 관계의 유무, 함수의 입출력 등과 같이 확실하게 분리되는 개념을 다룸
- ◆ 컴퓨터에 활용되는 많은 개념들은 이산적인 개념이 포함됨

* 이산수학이란?

- 디지털 컴퓨터가 데이터를 처리하는 과정에 필요한 수학적인 개념과 원리

2 연속과 이산(Discrete)

연속

- 중단 또는 급격한 변화가 없는 상태
- 연속적인 성질을 가지는 수학 분야를 연속수학이라 함
- 미적분학, 위상수학, 복소수, 추상대수학, 해석학 등



이산(Discrete)

- 연속성이 전혀 없는 분리된 개념, 불연속적인 상태
- 디지털 컴퓨터가 다루는 데이터는 이산적인 값(0, 1)을 가짐
- 이산과 연속은 서로 반대의 의미

2 연속과 이산(Discrete)

◆ 처리하는 자료가

연속적인 형태  아날로그 컴퓨터

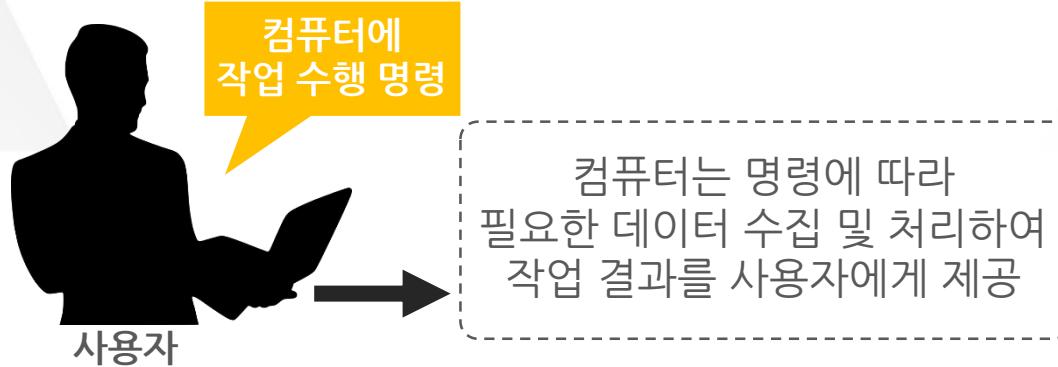
이산적인 형태  디지털 컴퓨터

3 이산수학 학습의 필요성

현대 사회에서 디지털 컴퓨터의 영향력이 증가되면서 이산수학의 중요성은 점점 증가되고 있음

- ▶ 현재의 디지털 컴퓨터들은 기본적으로 유한한 이산시스템이므로 디지털 컴퓨터의 상당수 내용은 이산수학을 통해 설명됨

3 이산수학 학습의 필요성



이산수학의 여러 개념들은 이러한
입력에서부터 출력까지 컴퓨터의
모든 과정을 이해하는 데 도움을 줌

2 도구, 기법, 방법론

도구, 기법, 방법론

1 축구를 하기 위한 도구, 기법, 방법론

도구

- ◆ 축구공, 축구장, 골대, 출구장, 보호대, 유니폼 등

기법(기술)

- ◆ 킥헤딩, 트래핑, 스토킹, 태클 등

1 축구를 하기 위한 도구, 기법, 방법론

방법론

- ▶ 피라미드 시스템, 4-2-4 시스템, 수비시스템 등
- ▶ 게임을 이기기 위해서는 상대방의 전략에 따라 적재적소에 알맞게 사용해야 함

* 방법론이란?

- 누가, 언제, 어디서, 왜, 어떠한 도구와 기법을 사용해야 되는지에 대한 것

2 수학의 도구, 기법, 방법론

- ◆ 수학은 너무나 많은 도구, 기법, 방법론이 있고 때로는 이들을 도구, 기법, 방법론에 맞추어 구분하기 모호한 경우가 많음

도구, 기법, 방법론

2 수학의 도구, 기법, 방법론

도구

- ◆ 수학의 정의, 정리

기법

- ◆ 가우스 소거법 (일차연립 방정식)
- ◆ 근의 공식 (2차, 3차 방정식)

방법론

- ◆ 상황에 따라 가장 효과적이고 효율적인 도구와 기법을 선택하는 것

2 수학의 도구, 기법, 방법론

수학적 모델링

- ◆ 수학의 도구, 기법, 방법론을 이용해 문제를 해결하려면 우선 실생활의 문제를 수학적 모델로 변화시켜야 함

2 수학의 도구, 기법, 방법론

정보 모델링

- ▶ 실생활의 문제를 컴퓨터로 해결하려고 한다면 마찬가지로 문제를 컴퓨터의 영역으로 옮기는 과정이 필요함



수학적 모델링과 정보 모델링은

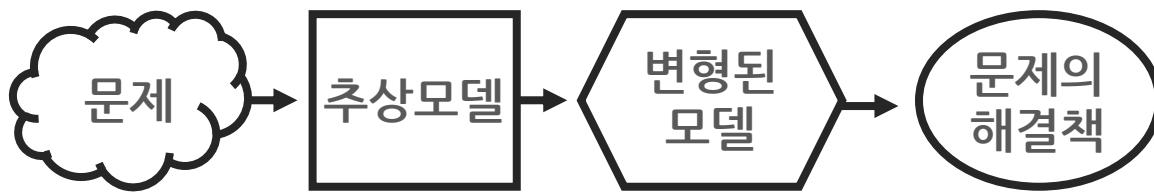
추상화를 통해서 진행됨

3 추상화(Abstraction)

- ▶ 문제와 관련된 핵심내용만 남기고
관련 없는 내용을 제거하여 **문제를 단순화 시키는 과정**
- ▶ 일정한 인식 목표를 추구하기 위하여 여러 가지
표상이나 개념에서 특정한 특성이나 속성을 빼냄

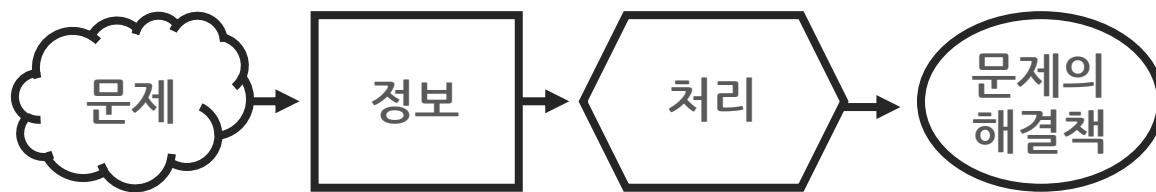
4 문제 해결 과정

수학적 모델링을 이용한 문제해결



4 문제 해결 과정

정보 모델링을 이용한 문제해결



5 추상화의 예시

▶ 문제 1

사과 1개를 600원 주고 사서 800원에 팔고, 배 1개를 1200원에 주고
사서 1500원에 팔았다고 할 때, 사과와 배를 합하여 총 10개를 팔고
이익이 2400원이 생겼다고 한다면 사과는 몇 개를 팔았습니까?

 추상화주어진 문제를
수학적 표현으로 변환함

<간단한 문제 표현>

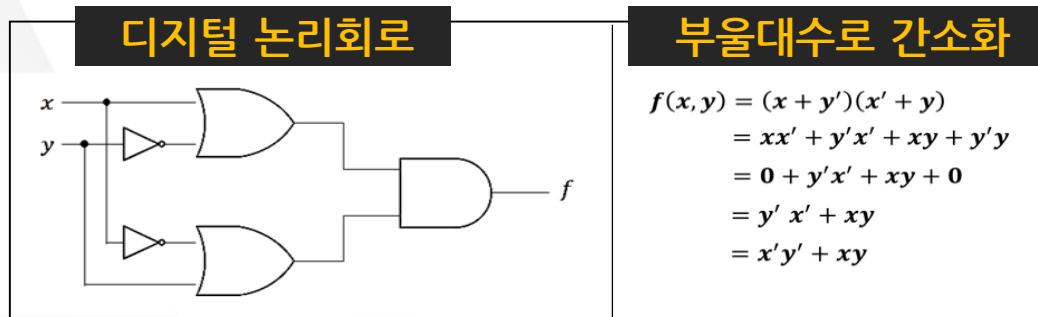
A : 사과의 판매 개수 B : 배의 판매 개수

$$A+B = 10$$

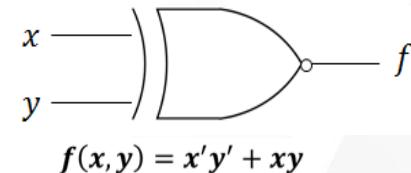
$$200A + 300B = 2,400$$

5 추상화의 예시

◆ 디지털 논리회로를 부울대수를 통해 단순화시키는 과정



실생활의 문제(복잡한 논리회로)가
수학적 도구(부울대수식)로 표현되고
수학적 기법(전개)을 통해서 해결됨
(간단한 논리회로)



3 알고리즘 언어

1

알고리즘

주어진 문제를 해결하기 위한 일련의 과정이나
절차를 순서대로 정확하게 나열한 것

- ◆ 어떠한 문제를 해결하기 위한 여러 동작들의 유한한 모임
- ◆ 알고리즘의 각 작업은 모호함이 없이 분명하게 표현되어야 함
- ◆ 프로그램 작성의 중요한 기초가 됨

1

알고리즘

주어진 문제를 해결하기 위한 일련의 과정이나
절차를 순서대로 정확하게 나열한 것

- ◆ 한 문제를 해결하기 위한 여러 알고리즘 중
가장 효율적이고 효과적인 알고리즘을 선택하는 것은
곧 좋은 프로그램의 작성의 의미
- ◆ 프로그램을 개발하기 위한 초기에 작성되며
전체 프로그램을 총괄하는 역할

2

알고리즘의 표현

- ① 컴퓨터 프로그래밍 언어
- ② 순서도(Flow Chart)
- ③ 의사코드(Pseudocode)

2 알고리즘의 표현

① 컴퓨터 프로그래밍 언어

◆ 컴퓨터 작동을 위한 동작을 세밀하게 지시

- [문제점]

- 알고리즘의 핵심요소가 잘 드러나지 않아 사람이 읽고 이해하기 쉽지 않음
- 중요하지 않은 부차적인 표현 (중괄호와 세미콜론 등)에 신경써야 함
- 통일된 언어가 존재하지 않음 (C, JAVA, Pascal 등)

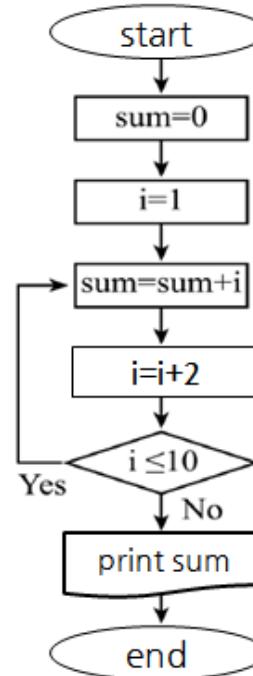
2 알고리즘의 표현

② 순서도(Flow Chart)

- ◆ 문제를 해결하는 방법과 절차에 따라 약속된 기호를 사용하여 명령들을 순서대로 표현하는 방식
- ◆ 알고리즘의 작동방식을 도식화하여 전달가능
 - [문제점]
 - 내용이 복잡하거나 큰 프로그램의 경우 표현 어려움
 - 반복과 조건 분기와 같은 구조는 많은 선과 도형들 속에서 어디부터 어디까지가 범위인지 찾기 쉽지 않음

2 알고리즘의 표현

② 순서도(Flow Chart)



2 알고리즘의 표현

③ 의사코드(Pseudocode)

- ◆ 프로그램 명령문 형식을 취하고 각 명령을 사람이 이해하기 쉽게 적당한 뜻을 가진 단어로 나타냄
- ◆ 프로그램 코드와 비슷한 형태로 알고리즘 작성
- ◆ 모호한 부분은 프로그래밍 언어의 문법을 채용하여 명확하게 기술
- ◆ 일반적으로 C언어 형식으로 작성

2 알고리즘의 표현

③ 의사코드(Pseudocode)

* 의사코드에 사용되는 명령어

할당문

제어문

- ━ 순차문
- ━ 조건문
- ━ 반복문

```
algorithm total()
{
    sum=0;
    i=1;
    while(i <=10){
        sum=sum + i;
        i=i+2;
    }
    print sum;
}
```

3 할당문

- ◆ 많이 사용하는 의사코드 중 하나이며 연산의 결과를 특정 변수에 할당해 주는 역할
- ◆ 값을 저장하기 위한 구문

변수명 = 계산식;

- 할당문의 예시

$x = a+2;$

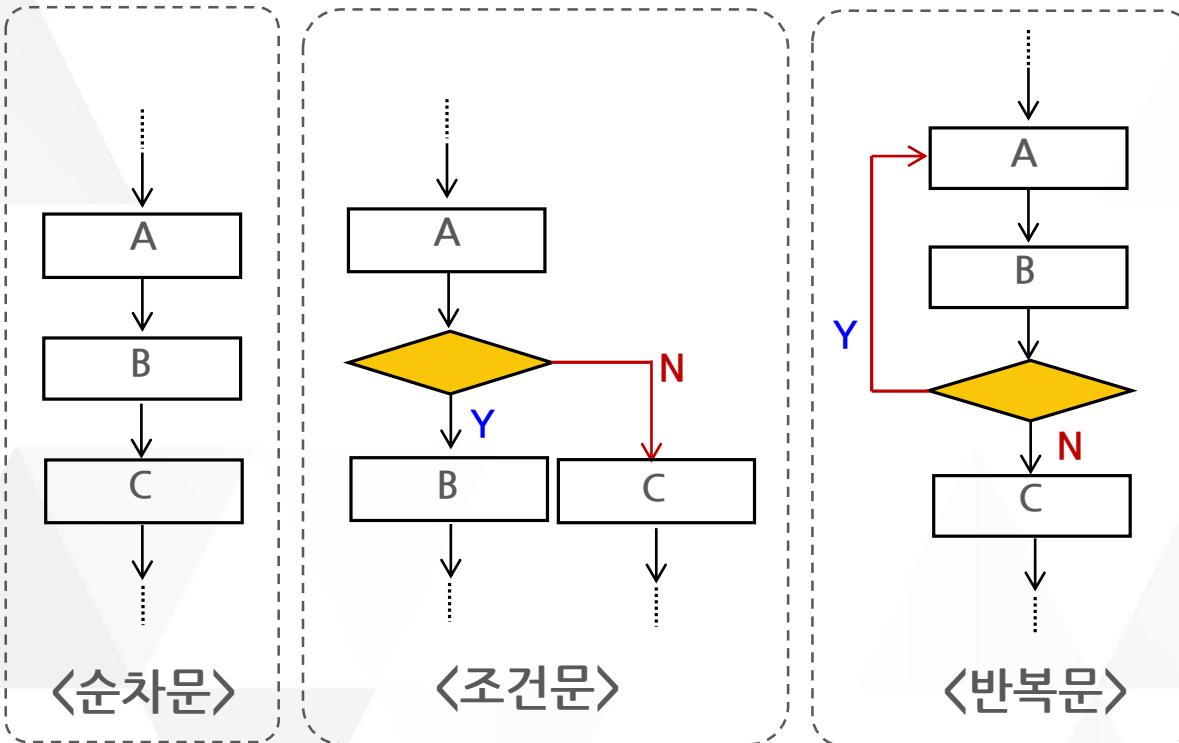


등호(=) 오른쪽을 먼저 계산하는데,
a의 값에 2를 더한 결과를 왼쪽의
변수 x에 저장하라는 의미

4 제어문

- ◆ 특정 조건에 의해 명령의 실행 순서가 변경되거나 명령을 반복적으로 수행해야 하는 경우에 해당하는 구문
- ◆ 순차문, 조건문, 반복문 등

4 제어문



4 제어문

① 순차문

- ◆ 순차적으로 실행되는 여러 개의 구문으로 구성
- ◆ 위에서 아래로 실행
 - 순차문의 예

```
x = 1;  
x = x+2;  
x = x+3;
```

② 조건문

- ◆ 조건에 따라 서로 다른 구문을 수행
- ◆ 특정 조건이 기준이 되어 해당 조건을 만족할 때 명령문이 실행하도록 하는 구문
- ◆ 조건이 하나인 경우도 있고 여러 개인 경우도 있음
- ◆ If문, Switch문 등이 있음

4 제어문

② 조건문 - if문

- ◆ 조건이 참이면 다음에 나오는 명령문1을 실행하고 거짓이면 명령문2를 실행함

```
if 조건문  
    명령문1;  
else  
    명령문2;
```

■ If문의 예

```
if ( x > 0 )  
    print “양수”;  
else if ( x < 0 )  
    print “음수”;  
else  
    print “제로”;
```

4 제어문

③ 반복문

- ◆ 명령문을 반복적으로 실행하게 하는 것(루프)
- ◆ 일정한 횟수만큼 반복하거나 특정 조건을 만족하는 동안 반복 수행
- ◆ for 문, while 문 등

4 제어문

③ 반복문 - for문

- ◆ 변수의 초기값을 주고 명령문을 한 번 반복 실행할 때마다 변수의 값을 하나씩 증가시키면서 최종값이 될 때까지 반복
- ◆ 변수가 최종값이 되면 해당 for문을 빠져나와 더 이상 반복 실행하지 않음

4 제어문

③ 반복문 - for문

for 변수=초깃값 to 최종값
명령문;

■ for문의 예

```
for i = 1 to 5
{
    print i;
}
print “출발”;
```

4 제어문

③ 반복문 - while문

- ▶ 조건을 만족하는 동안 명령문을 반복 실행
- ▶ 조건을 더 이상 만족하지 않으면 반복 종료

while 조건문
명령문:

```
i = 5;  
while i > 0  
{  
    print i;  
    i = i-1;  
}  
print “종료”;
```

4 제어문

예시) 다음 알고리즘의 출력은 무엇인가?

```
sum = 0;  
i=1;  
while i ≤ 10  
{  
    sum = sum + i;  
    i = i + 3;  
}  
print sum;
```

화면에 22를 출력함

4 이산수학의 응용 분야

이산수학의 응용 분야

1 이산수학의 응용 분야

◆ 이산수학은 컴퓨터 여러 분야에 광범위하게 응용되고 있음

이산수학 주제	컴퓨터 응용분야
논리	전문가 시스템
증명	프로그램의 입증
집합론	자료구조, 데이터베이스
행렬	그래픽, 기계학습
관계	관계 데이터베이스
함수	프로그램 언어

이산수학의 응용 분야

1 이산수학의 응용 분야

◆ 이산수학은 컴퓨터 여러 분야에 광범위하게 응용되고 있음

이산수학 주제	컴퓨터 응용분야
부울대수	디지털 회로 설계
그래프	자연어 처리, 컴퓨터 네트워크의 라우팅 알고리즘
트리	통신 네트워크, 데이터베이스 인덱스, 자료의 탐색, 전기회로망 설계
조합이론	알고리즘의 효율성 분석
알고리즘	컴퓨터 프로그램
오토마타	문제 해결 가능성

이산수학의 응용 분야

1 이산수학의 응용 분야

논리, 증명(전문가 시스템, 프로그램의 입증)

- ◆ 문제의 해결 과정은 애매하지 않고 객관적으로 정의되어야 하며 논리적이어야 함

집합론(자료구조, 데이터베이스)

- ◆ 명확한 기준으로 데이터를 수집해야 하고 그 기준에 따라 데이터를 분류/관리 해야 함

이산수학의 응용 분야

1 이산수학의 응용 분야

행렬, 관계(2차원/3차원 그래픽, 기계학습, 관계 DB)

- ◆ 데이터들 사이의 관련성을 찾아내는 것은 매우 중요하며 이러한 관련성을 기초로 데이터를 분석하고, 처리, 연산할 수 있음

함수(프로그램 언어)

- ◆ 컴퓨터에서 입력, 처리, 출력 관계의 정확한 정의는 매우 중요하며 이러한 관계를 이해하기 위한 수학적 개념

이산수학의 응용 분야

1 이산수학의 응용 분야

부울대수(디지털 회로 설계)

- ◆ 하드웨어는 복잡한 회로로 구성되며 각 회로의 연산을 정의하고 그에 맞는 회로도를 그리는데 부울대수가 필요

그래프, 트리(통신 네트워크, 라우팅 알고리즘, 탐색)

- ◆ 문제를 가시적인 형태인 그래프로 표현하면 효율성을 높일 수 있고, 트리를 활용하면 데이터의 검색이 쉬워짐

이산수학의 응용 분야

1 이산수학의 응용 분야

조합이론(알고리즘의 효율성 분석)

- ▶ 순열, 조합, 경우의 수 등은 컴퓨터의 연산이나 처리 등을 결과를 예측하거나 SW 프로그래밍 방식을 결정하는 기준으로 활용할 수 있음

알고리즘(컴퓨터 프로그램)

- ▶ 알고리즘은 프로그램을 작성하기 전에 명령 과정을 대략적으로 기술한 것으로 프로그램의 정확성과 효율성을 검증하는데 활용됨