

자료구조

L02 Data Structures and Algorithms

2022년 1학기

국민대학교 소프트웨어학부

In this lecture

- 올바른 자료구조(data structure)를 선택할 때 고려할 점
- ADT(Abstract Data Type)가 필요한 이유, 자료구조와의 차이점
- Problem, Algorithm, Program의 구분

Outline

- ❖ 자료구조의 철학
- ❖ 추상자료형(ADT), 자료구조, 자료형
- ❖ Problems, Algorithms, Programs

자료구조 학습목표

- 자료구조마다 **비용(cost)**과 **이득(benefit)**이 서로 다름을 이해한다.
- 자주 사용되는 자료구조를 배운다.
- 자료구조나 프로그램의 비용을 측정하는 법을 이해한다.
 - 새로운 자료구조나 프로그램의 이점을 판단할 수 있다.

좋은 프로그래밍이란?

컴퓨터 프로그램 디자인의 두 가지 핵심 목표:

- 이해하기 **쉽고**, 구현하기 **쉽고**, 디버깅하기 **쉬운** 알고리즘을 설계한다.
- 컴퓨터의 자원(CPU, 메모리 등)을 **효율적으로** 사용하는 알고리즘을 설계한다.

두 목표를 동시에 이룰 수 있다면... 최고!

하지만, 세상은 언제나 내 마음처럼 흘러가진 않는다...

자료구조는 효율성에 더 집중한다.

하지만, 좋은 자료구조는 효율적이면서도 이해하기도 쉽다.

자료구조가 필요한 이유

- 자료구조는 자료를 구조화함
⇒ 프로그램 효율성 UP!
- 더 비싸고 강력한 컴퓨터
⇒ 더 복잡한 문제(application) 해결 가능
- 컴퓨터 하드웨어가 계속 발전하는데, 효율적인 프로그램이 필요할까?
⇒ 더 강력한 컴퓨터 + 더 효율적인 프로그램 =
더 강력한 문제해결력

효율성 (Efficiency)

- 어떤 자료구조/알고리즘을 선택하느냐에 따라 프로그램의 실행시간이 달라짐
⇒ 몇 초 vs 며칠
- **효율적이다(Efficient)**: 자원 제약사항 (Resource constraints) 내에 문제를 해결한다. (a.k.a. 자원절약)
- **자원(Resource)**: Time, Space
- **비용(Cost)**: 자료구조나 프로그램 등이 소비하는 자원의 양

자료구조 선택하기

자료구조 선택하는 방법

1. 문제(Problem)를 푸는데 꼭 필요한 연산(Operation)에는 무엇이 있는지 분석한다.
2. 각 연산마다 필요한 자원(Resource constraints)을 정량화 한다.
3. 이러한 요구사항을 가장 잘 만족하는 자료구조를 선택한다.

비용과 이득 (Costs and Benefits)

- 자료구조마다 **비용**과 **이득**이 있음.
- 한 자료구조가 다른 자료구조보다 항상 좋은 경우는 잘 없음.
- 모든 자료구조는 **공간, 시간, 노력**을 필요로 한다.
 - 공간(Space): 자료를 저장
 - 시간(Time): 각 기본 연산(basic operation)을 실행
 - 프로그래머의 노력 (Programming effort)

비용과 이득 (Costs and Benefits)...

- 문제(Problem)를 해결할때, **공간과 시간에 제약**이 있음.
- 문제의 특성을 잘 파악해야, 가장 적절한 자료구조를 선택할 수 있음.
- 은행 시스템 예시:
 - 계좌 생성: 몇 분 내에 완료해야함
 - 입출금: 몇 초 내에 완료해야함
 - 계좌 폐기: 하루~정도 걸려도 괜찮음

은행 시스템을 위해서 어떤 자료구조를 사용해야할까?

자료구조 선택을 위한 질문

- 자료구조에 자료가 처음에 **한꺼번에** 입력되는가? 혹은 **띠엄띠엄** 입력되는가? (예시?)
- 자료는 **지워질 수 있는가?**
- 자료가 **미리 정의된 순서**로 처리되는가? 혹은 **임의 접근 (random access)**이 필요한가?
 - 예) word 문서에서 성과 이름의 순서 바꾸기
 - 예) 도서관리시스템에서 원하는 책의 저자 알아보기

자료구조 선택하기 예제

학생의 학점(GPA) 정보가 학교의 데이터베이스에 저장되어있다.

Task 1) 2022년 봄 자료구조 수업에서 B0받은 학생 모두 찾기

- 이러한 종류의 연산을 “**exact query**”라 함
- Hash table 자료구조가 적절하다.

Task 2) 2022년 봄 학점이 1.0~2.5 사이인 학생 모두 찾기

- 이러한 종류의 연산을 “**range query**”라 함
- binary-tree, B-tree 등의 자료구조가 적절하다.

Outline

- ❖ 자료구조의 철학
- ❖ 추상자료형(ADT), 자료구조, 자료형
- ❖ Problems, Algorithms, Programs

추상자료형 (Abstract Data Types)

- 추상자료형 (ADT; Abstract Data Type):
 - 자료형(data type)을 값과 연산으로 명세(specification)한 것
- ADT 연산은 입력과 출력으로 정의됨
- Encapsulation: 연산의 구현 방법은 숨긴다

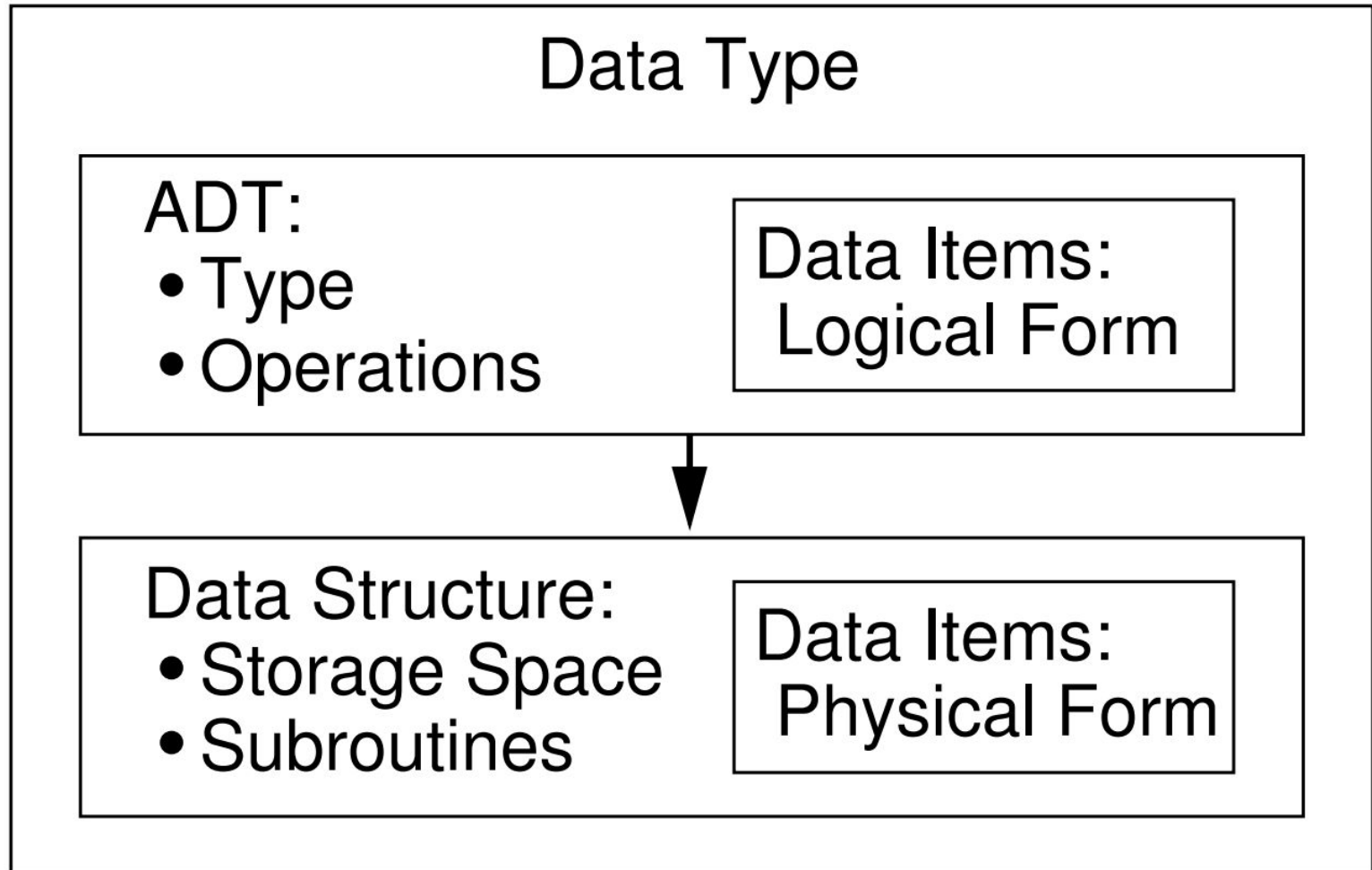
자료구조 (Data Structure)

- 자료구조 (Data Structure): ADT를 실제적으로 구현 (physical implementation)한 것.
 - main memory에 자료가 어떻게 저장되나?
 - 연산은 어떤 구체적인 방법으로 동작하나?

참고)

- 파일구조(File Structure): 하드디스크와 같은 보조저장소에 자료가 저장되는 구조

자료형 (Data Type)



ADT와 Interface, 자료구조와 Class

- Java에서는 ADT를 Interface로 표현 (Python의 경우 추상클래스)
- 자료구조는 Class로 표현

```
interface Dog{  
    public void bark();  
}  
  
class KoreanDog implements Dog{  
    public void bark(){  
        System.out.println("멍멍!");  
    }  
}
```

Java

```
from abc import *  
class Dog(metaclass=ABCMeta):  
    @abstractmethod  
    def bark(self):  
        pass  
  
class KoreanDog(Dog):  
    def bark(self):  
        print("멍멍")
```

Python

왜 ADT를 사용할까?

- 자료구조를 몰라도, **ADT만으로도 자료형을 사용할 수 있다.**
- 더 복잡하고 더 규모가 큰 프로젝트를 할 수 있다!
 - 우리의 시간은 소중하니까.

Outline

- ❖ 자료구조의 철학
- ❖ 추상자료형(ADT), 자료구조, 자료형
- ❖ **Problems, Algorithms, Programs**

Problem, Algorithm, Program

- **문제 (Problem):** 풀어야 할 작업
 - 특정 입력에 특정 출력이 나와야 함
 - 예) 수많은 웹 문서 데이터에서 “Data Mining” 이라는 단어를 포함하는 문서 모두 찾기
 - 문제를 해결하는 방법은 문제에 포함되지 않음

Problem, Algorithm, Program

- **알고리즘 (Algorithm):** 문제를 풀기위한 방법(method)이나 과정(process)
 - 한 문제를 푸는 알고리즘은 여러개 존재 가능
 - 속성
 - 알고리즘은 정확해야한다
 - 알고리즘은 구체적인 절차(step)로 구성된다
 - 알고리즘은 유한한(finite) 절차로 구성된다
 - 알고리즘은 언젠가 끝나야 한다

Problem, Algorithm, Program

- **프로그램 (Program):** 프로그래밍 언어로 알고리즘을 구현한 것

In this lecture

- 올바른 자료구조(data structure)를 선택할 때 고려할 점
- ADT(Abstract Data Type)가 필요한 이유, 자료구조와의 차이점
- Problem, Algorithm, Program의 구분

Questions?