주요 내용 요약

선형 자료구조: 일차원적 자료구조인 리스트, 스택, 큐에 대해 공부합니다. 단순하지만 아주 많이 사용되는 유용한 자료구조입니다.

색인: 저장한 데이터를 효율적으로 찾기 위한 색인 기능을 지원하는 자료구조를 다룹니다. 이진 검색 트리로 시작해서 균형 잡힌 이진 검색 트리인 AVL 트리와 레드-블랙 트리, 용량이 큰 색인을 위한 B-트리까지 공부합니다. 또한 이런 색인들과 성격이 조금 다른, 내용에 의해 자리가 정해지는 해시 테이블을 공부합니다.

우선순위 큐 힙: 우선순위가 있는 데이터를 처리하기 위한 효과적인 자료구조인 힙을 공부합니다. 힙도 생각하는 방법을 훈련하기 좋은 도구이고, 정렬에도 사용되는 유용한 자료구조입니다.

정렬: 그 자체로도 중요하지만 체계적인 사고법을 훈련할 수 있는 좋은 도구입니다10가지 정렬 알고리즘을 소개하면서 정렬 알고리즘의 상대적 속도로 재어본 결과도 함께 제시합니다. 데이터 간 관계를 반영하는 그래프를 사용하여 최소 신장 트리 찾기, 최단 경로 찾기 등의 알고리즘을 공부합니다

1장. 자료구조 소개

Contents

주요 내용

- 01 자료구조란
- 02 자료구조와 알고리즘
- 03 자료구조의 추상 데이터 타입

학습목표

- 자료구조의 필요성을 직관적으로 이해하고, 알고리즘과의 관계에 대한 느낌을 갖도록 한다.
- 추상 데이터 타입에 대해 직관적으로 이해한다.

01 자료구조란

자료구조란

- 데이터를 저장, 조직, 관리하는 방법

그림 1-1 자료구조 적용 전(왼쪽)과 자료구조 적용 후(오른쪽)

그림 1-2 자료구조의 일상 예

자료구조란

- 문제 해결에 사용할 부품

- 건축물을 만들려면
  - 건축 재료와 구조 모듈에 대한 이해가 필요하다
  - 철근, 시멘트, 강화 유리, 벽돌, ...
  - 샷시, 철골, 거푸집, 배수 구조, 전기/인터넷 연결 구조, ...

- 프로그래밍과 문제 해결도
  - 데이터와 구조 모듈에 대한 이해가 필요하다
  - 프로그래밍 언어, 정수, 문자열, ...
  - 리스트, 스택, 큐, 우선순위 큐, 검색 트리, 해시 테이블, 그래프, ...

그림 1-3 다양한 형태의 건축물과 건축 재료

그림 1-4 부품(자료구조) 선택의 중요성

자료구조란

- 생각하는 방법을 훈련하는 도구

- 자료구조는 그 자체로 중요하다
  못지 않게, 생각하는 방법 훈련도 중요하다
  - 자료구조를 이용해서 문제를 해결하는 과정
  - 문제 해결 과정에서 논리의 골격이 구성되는 방법/스타일
  - 의미의 단위(의미의 매듭)를 설정하는 방법

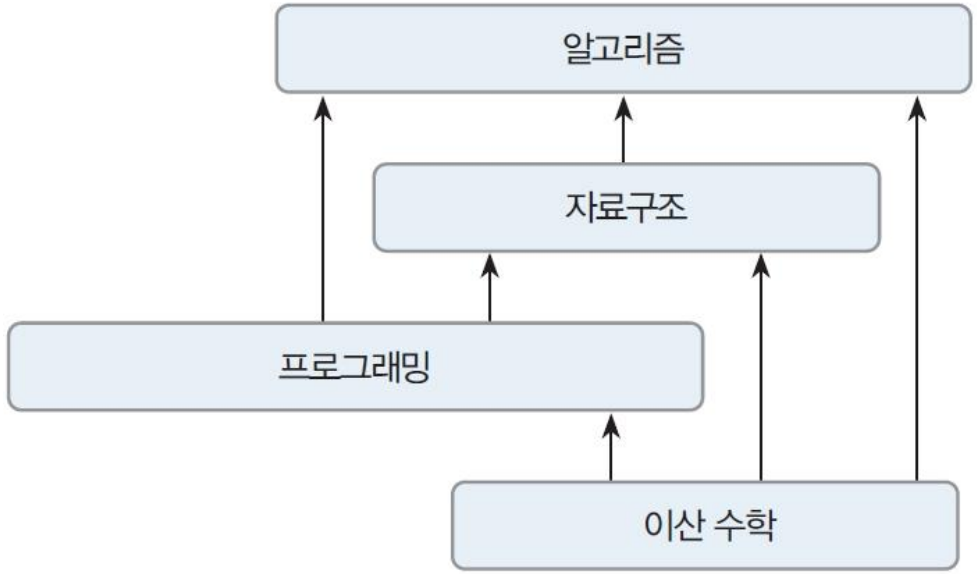

그림 1-5 알고리즘, 자료구조, 프로그래밍, 이산 수학의 관계

- 이 책에서 다루는 내용

자료구조
- 리스트
- 스택
- 큐
- 힙
- 검색 트리
- 해시 테이블
- 그래프

생각하는 방법
- 재귀
- 추상화
- 정렬
- 그래프

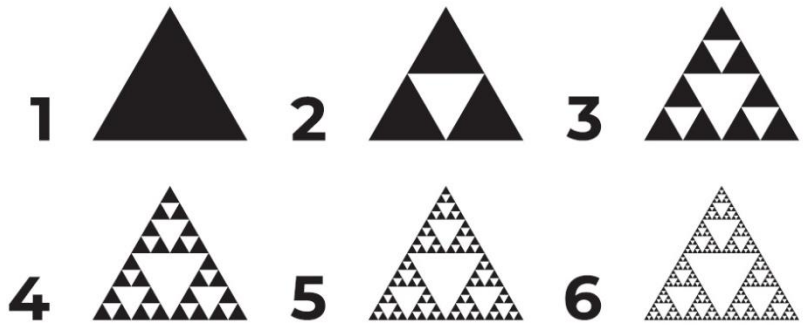

그림 1-6 재귀적 구조의 시에르핀스키 삼각형

자료구조 종류

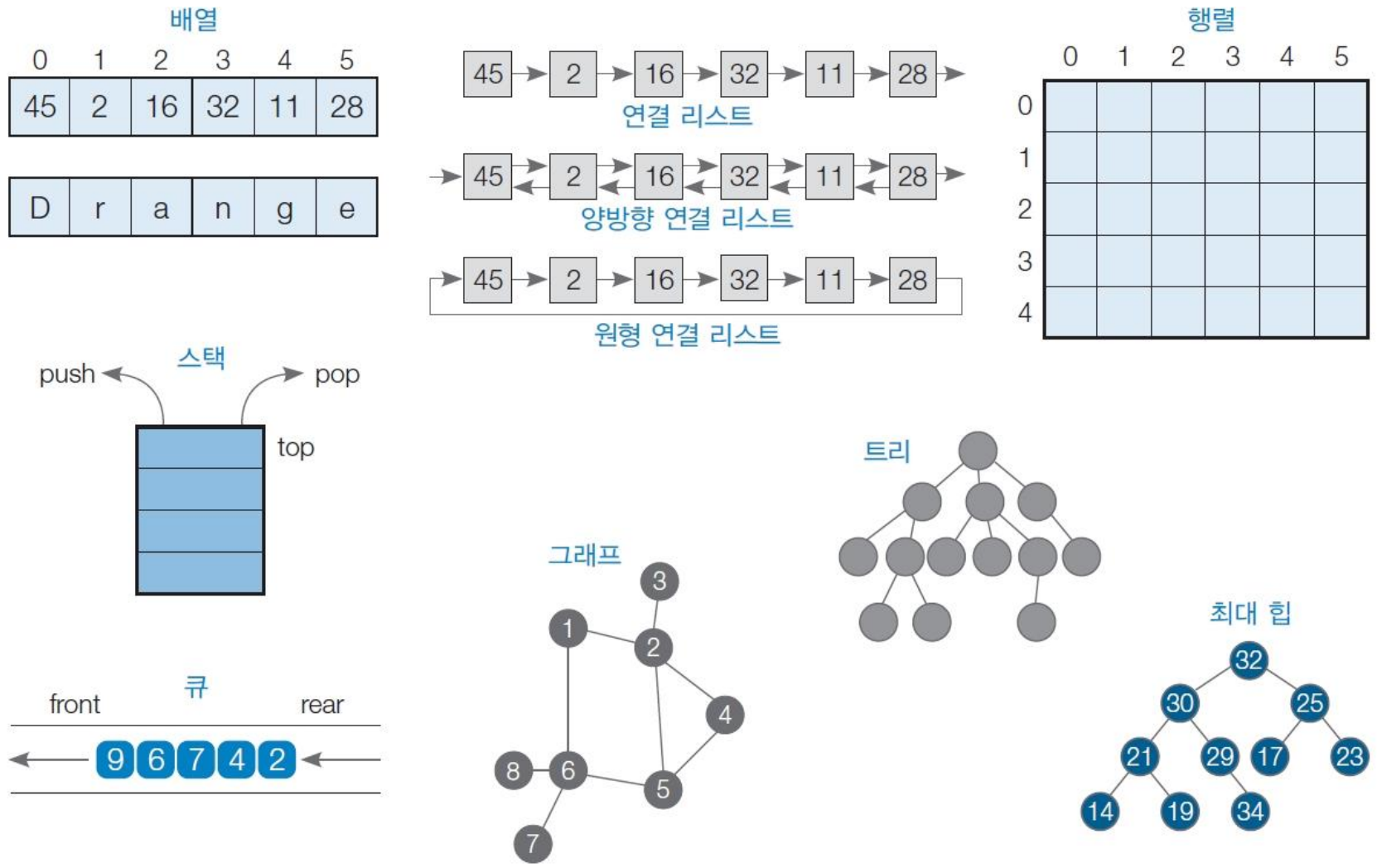

그림 1-7 자료구조의 종류

자료구조 종류

그림 1-8 이 책에서 다루는 자료구조

02 자료구조와 알고리즘

자료구조와 알고리즘의 관계

- 자료구조는 알고리즘 과목의 직전 단계이기도 하고, 그 자체로 여러 알고리즘을 포함

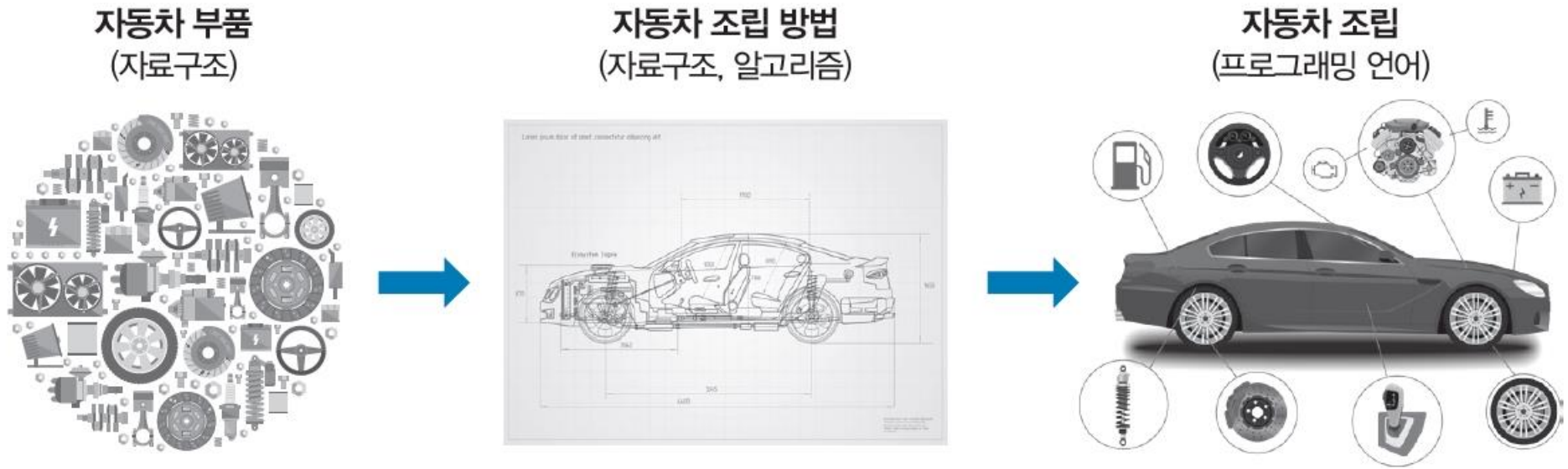

그림 1-9 자료구조와 알고리즘의 관계

알고리즘 표기법

- 자연어를 이용한 서술적 표현
  - 서술적일 뿐만 아니라 쓰는 사람에 따라 일관성이나 명확성을 유지하기 어려움
  - 누구라도 쉽게 이해하고 쓸 수 있어야 하는 알고리즘을 표현하는 데는 한계가 있음
- 순서도를 이용한 도식화
  - 명령의 흐름을 쉽게 파악할 수 있지만 복잡한 알고리즘을 표현하는 데는 한계가 있음
- 프로그래밍 언어를 이용한 구체화
  - 추가로 구체화할 필요가 없으나 해당 언어를 모르면 이해하기 어려움
  - 다른 프로그래밍 언어로 프로그램을 개발하는 경우에는 다른 프로그래밍 언어로 변환해야 하므로 범용성이 떨어짐
- 가상코드를 이용한 추상화
  - 가상코드Pseudo-Code는 직접 실행할 수는 없지만 일반적인 프로그래밍 언어와 형태가 유사해 프로그래밍 언어로 구체화하기가 쉬움

일반적인 알고리즘 표기법

```
move(n, a, b, c) {
    if (n > 0) then {
        move(n-1, a, c, b);
        a에 있는 원반을 b로 옮긴다;
        move(n-1, c, b, a);
    }
}
```

그림 1-10 일반적인 알고리즘 표기 예: 하노이 탑 알고리즘

이 책에서 사용하는 알고리즘 표기법

(a)

```
move(n, a, b, c): ❶
    if (n > 0)  ❷
        move(n-1, a, c, b)
        a에 있는 원반을 b로 옮긴다
        move(n-1, c, b, a)
```

(b)

```
move(n, a, b, c):
❸ if (n > 0)
        move(n-1, a, c, b)
        a에 있는 원반을 b로 옮긴다
        move(n-1, c, b, a)
    else
        에러 처리
```

(c)

```
sample(A[], n):
    i ← 0
    while i <= n
❹       sum ← sum + A[i]
        i++    ❺◀ i를 1 증가시킨다
    return sum
```

그림 1-11 이 책에서 사용하는 알고리즘 표현

❶ 함수 시작

❷ then 생략

❸ then 없이 if-else

❹ while 루프에 속하는 문장 들여쓰기

❺ 주석

03 자료구조의 추상 데이터 타입

추상

- 세세한 부분을 표현하지 않고 전체적인 이미지를 표현한 것

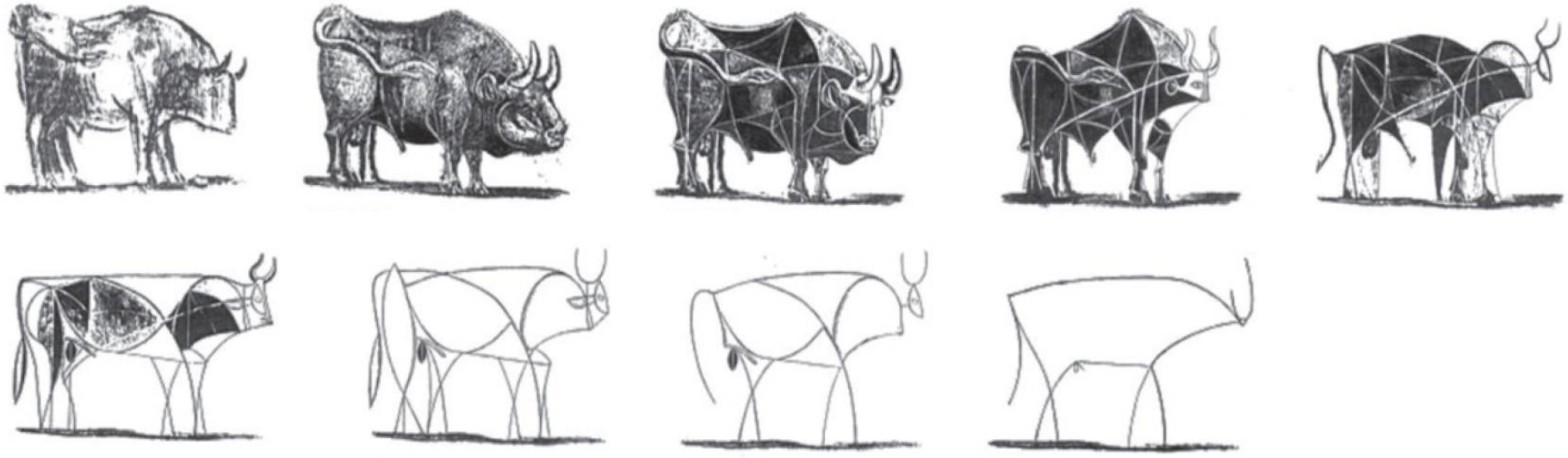

그림 1-12 추상적 묘사

Genetic Algorithm and Graph Partitioning

T. N. Bui, B. Moon • Published 1996 • Computer Science • IEEE Trans. Computers

Hybrid genetic algorithms (GAs) for the graph partitioning problem are described. The algorithms include a fast local improvement heuristic. One of the novel features of these algorithms is the schema preprocessing phase that improves GAs' space searching capability, which in turn improves the performance of GAs. Experimental tests on graph problems with published solutions showed that the new genetic algorithms performed comparable to or better than the multistart Kernighan-Lin algorithm and the simulated annealing algorithm. Analyses of some special classes of graphs are also provided showing the usefulness of schema preprocessing and supporting the experimental results. Collapse

그림 1-13 논문 초록

추상 데이터 타입: ADT

- 세부 사항에서 벗어나 추상적으로 정의한 데이터 타입
- 즉, 어떤 데이터 타입이 어떤 작업으로 이루어지는지만 표현한 것

(a)

```
원소를 첫 번째, 두 번째, ..., i번째 원소로 가리킬 수 있는 자료구조
```

(b)

```
i번째 자리에 새 원소 넣기
원소 x 찾기
i번째 자리의 원소 삭제하기
```

그림 1-14 리스트의 ADT 표현 예

추상 데이터 타입: ADT

(a)

```
ADT '리스트'
    작업:
        i번째 자리에 새 원소 넣기
        원소 x 찾기
        i번째 자리의 원소 삭제하기
```

(b)

```
ADT '리스트'
    작업:
        i번째 자리에 새 원소 넣기
        원소 x 찾기
        i번째 자리의 원소 삭제하기
        비어 있는지 확인하기
        깨끗이 청소하기
```

그림 1-15 ADT 리스트