



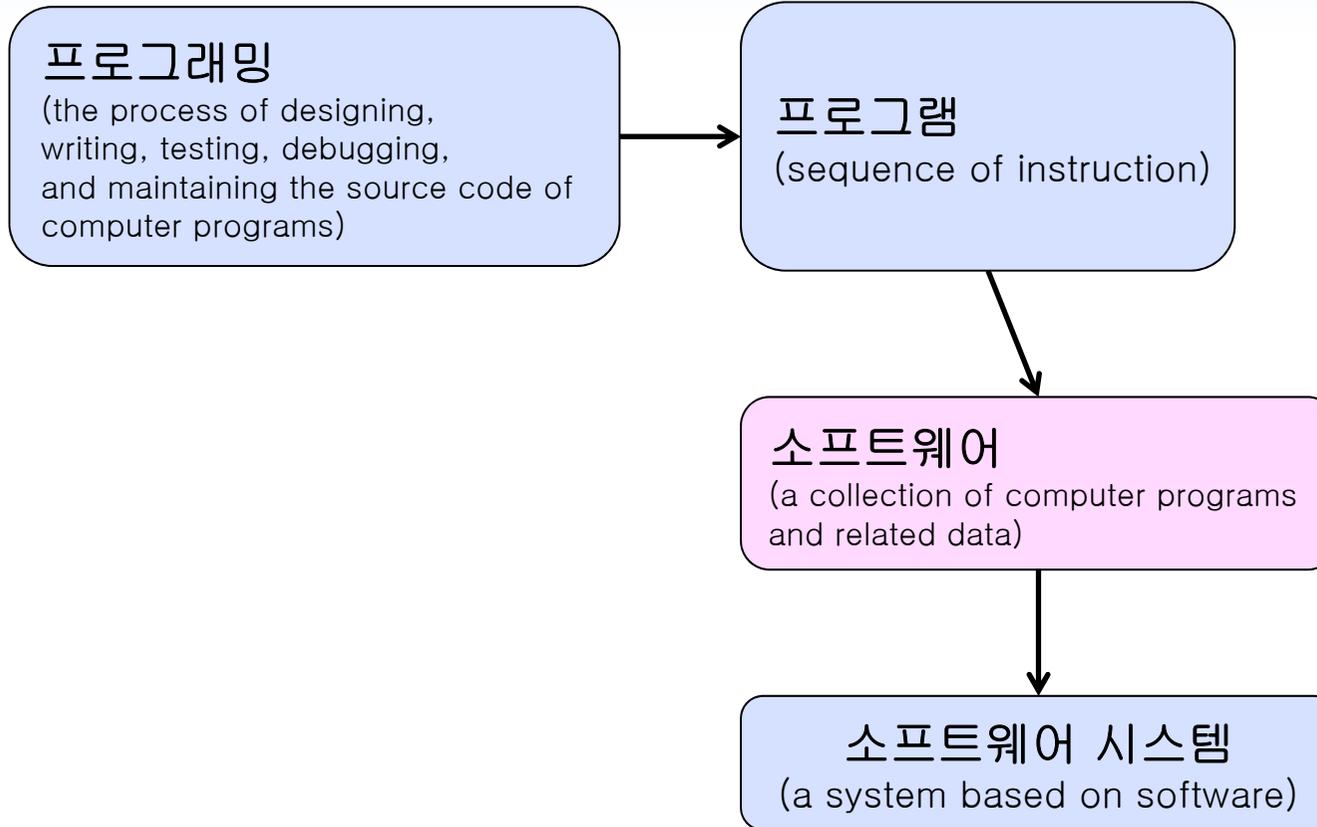
제 1 장 . 소프트웨어 개발과정

1. 소프트웨어 개발

소프트웨어 개발의 예
소프트웨어 시스템 개발 단계(life cycle)
소프트웨어 개발 각 단계에서의 문제점

2. 자료구조의 개념

자료구조의 개념
좋은 소프트웨어 란?
자료구조에서 배우는 내용



* 용어 :
System : a set of interacting or interdependent components forming an integrated whole
(예 : computer system? 정보시스템 ...)



1. 소프트웨어 개발

1.1 소프트웨어 개발 예

(소프트웨어 개발 문제 예) n 개의 데이터 중 가장 큰 수를 찾아 주시요?

What is
the
problem?



철수 -> 프로그래머(영희)



How to
solve the
problem...

(문제 개요)

철수는 연구를 하기 위하여 실험을 하는데 실험 결과 항상 n 개의 자료가 무작위로 만들어진다고 한다. 철수는 이 때마다 제일 큰 수를 찾기 위하여 고생을 한다. 그래서 실험 결과로 만들어진 수를 컴퓨터를 이용해 제일 큰 수를 찾을 수 없을까 고민하다 프로그래머에게 의뢰하기로 하였다.

철수가 컴퓨터프로그램을 직접 배워서 해볼까 생각을 해보았지만,

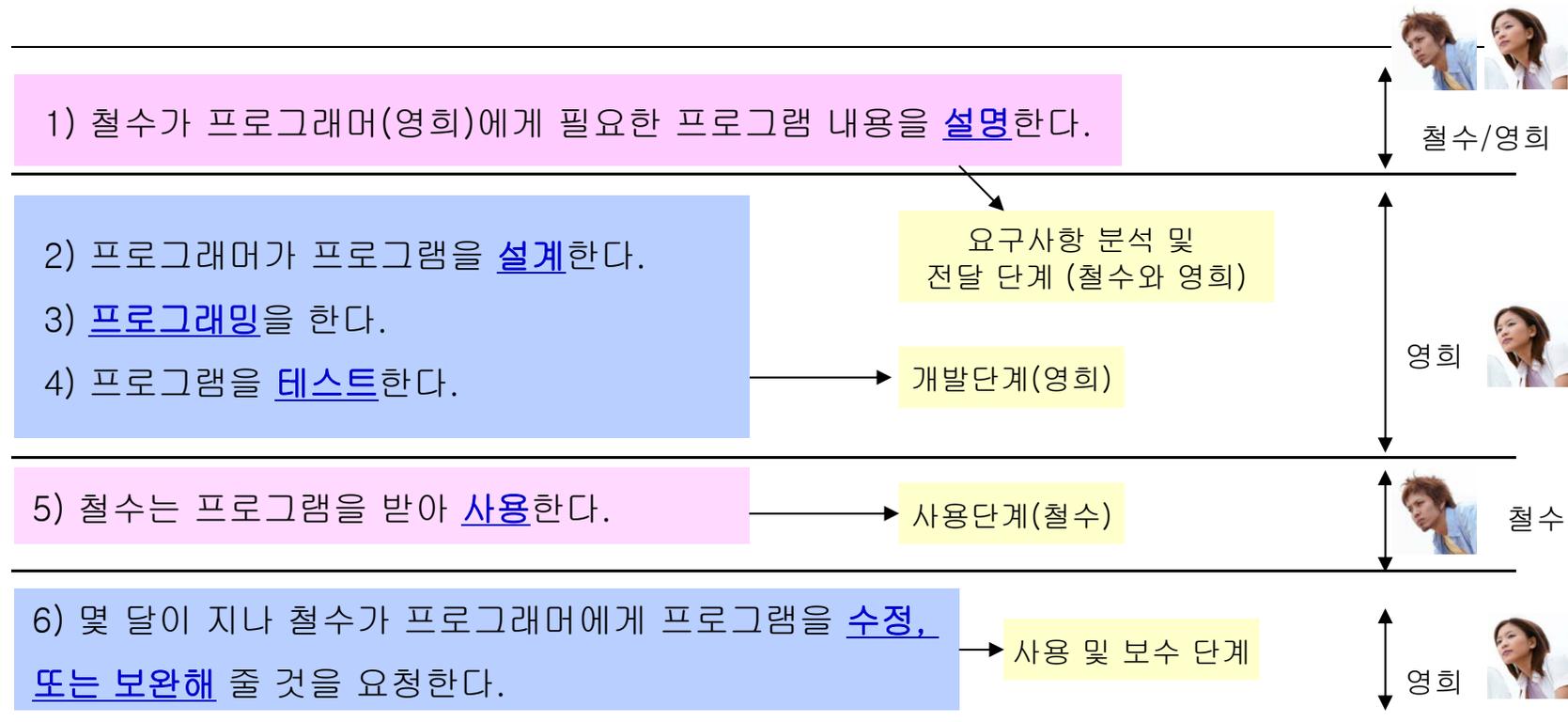
1. 프로그램이 쉽지 않고
2. 앞으로 프로그램을 자주 할 것 같지 않고
3. 컴퓨터 프로그램 언어가 기술이 발달하면 자주 바뀐다고

비용은 프로그래머가 개발하는 데 소요되는 시간에 대하여 비용을 지불하기로 한다.



1.1 소프트웨어 개발 예

(소프트웨어 개발 문제 예) n 개의 데이터 중 가장 큰 수를 찾아 주시오.





1.1 소프트웨어 개발 예

(프로그램 개발 단계)

1. **요구사항분석** - 프로그램 개발에 대한 사용자 및 기능/성능 요구사항
2. **설계** - 플로우차트 등 설계 도구 사용
3. **프로그래밍** : 플로우차트를 프로그래밍언어로 바꾼다.

위 내용을 정리하면 다음과 같은 C 언어 프로그램이 된다.(**실행해보자!!**)

```
#include <stdio.h>
int main()
{
    int big, x;
    big =0;
    while(!feof(stdin)) // 키보드를 stdin 이라고함, Ctrl-Z로 종료
    {
        scanf("%dWn", &x);
        if(x>big) big=x;
    }
    printf("the big is : %dWn ", big);
}
```



1.1 소프트웨어 개발 예

(프로그램 개발 단계) - 계속

4. **테스트** : 프로그램을 테스트한다. 철수에게 샘플 데이터를 받아서 실험을 한다.

5. **사용** : 철수는 프로그래머(영희)에게 받아서 사용을 한다. 철수는 프로그래머에게 컴퓨터에 설치하는 법, 프로그램을 수행하는 법, 문제가 있을 때 해결하는 법을 배우거나 문서로 설명된 자료를 받는다.

6. **사용 및 보수 단계** : 철수는 사용을 하다 실험의 내용이 변경되어 제일 작은 수도 함께 필요하게 되었다. 철수는 이것을 역시 혼자해결이 힘들어서 다시 프로그래머를 찾아서 수정을 의뢰하게 된다. 그 외 실수데이터 처리가 필요하다든지, 운영체제가 바뀌어 프로그램을 다시 설치한다든지 등의 문제가 발생할 수 있다.



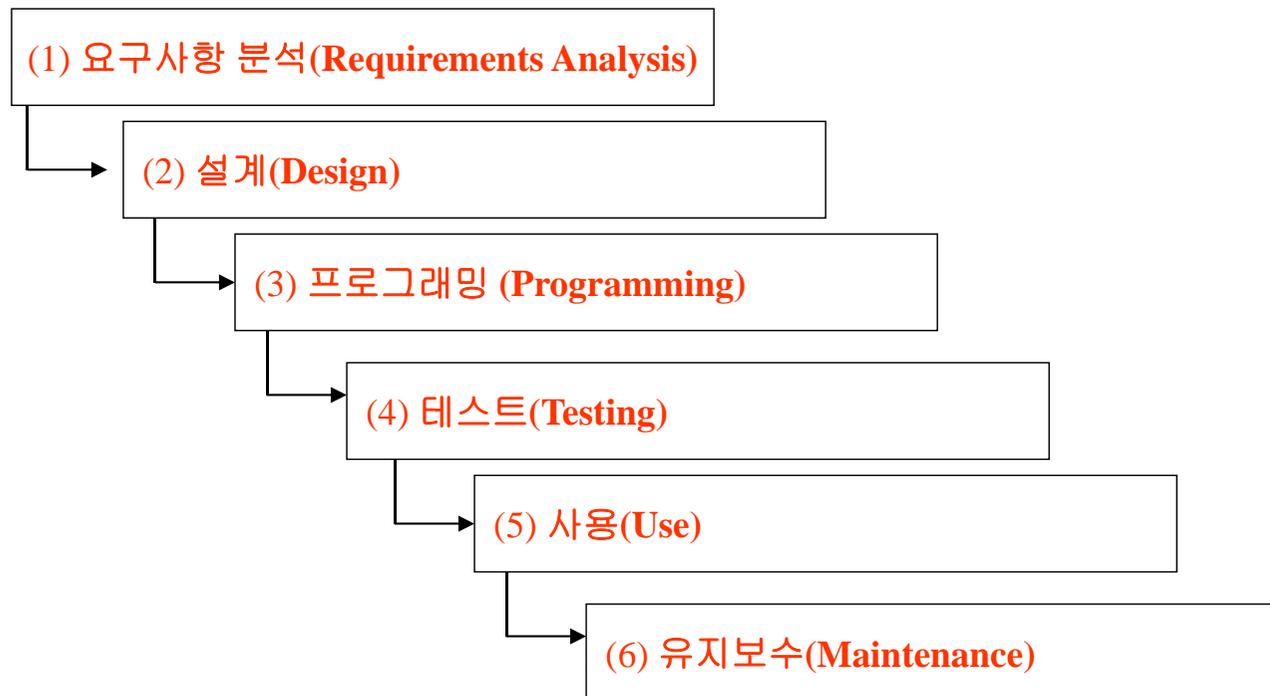
1.1 소프트웨어 개발 예

(프로그램 개발 시 생각할 점)

1. 철수가 영희에게 **음수** 데이터도 있다고 말했다면 영희의 프로그램은 오류이다 (영희가 이것을 무시했다면). 데이터가 모두 음수이면 결과가 틀리다. - 왜 ?
-> 소프트웨어 개발시 의뢰자가 개발자에게 정확히 설명할 필요가 있다.
2. 철수가 영희에게 프로그램 **max.exe** 를 받고 **max.c** 프로그램을 전달받지 않으면 유지 보수 단계에서 어떤 문제가 생길까? 유지보수 할 때 영희가 없다면?
-> 소스코드의 소유권을 의뢰자가 받아둘 필요가 있다
3. 철수가 프로그램을 배운다면 어떻게 될까? **프로그래머**의 역할이 축소되면?
-> 소프트웨어 개발의 난이도에 따라 프로그래머의 역할이 달라진다
4. 철수가 사용하다가 컴퓨터의 **운영체제**가 바뀌면? 혹은 실수 데이터의 처리가 필요하면?
-> 소프트웨어는 개발 후 소프트웨어 사용 환경, 혹은 새로운 요구에 따라 재 개발이 필요하다.



1.2 소프트웨어 개발 단계(**Software Life Cycle**, 소프트웨어 생명 주기)





1.2 소프트웨어 개발 단계(System Development Life Cycle)

(소프트웨어 개발 단계 설명)

(1) 요구사항 분석(Requirements Analysis)

- 소프트웨어 개발 전 개발자가 사용자의 정확한 요구 사항, 즉 정보(입력, 출력, 처리 논리 등)를 파악한다. 사용자와의 대화와 업무 범위를 파악하는 과정이다.

(2) 소프트웨어/시스템 설계(Software/System Design)

- 개발될 프로그램을 설계한다. 분석 단계보다는 좀 구체적인 과정으로 프로그램의 흐름, 데이터 내용에 관한 설계가 된다. 객체의 표현 방식을 택하고 각각의 오퍼레이션들을 위한 알고리즘을 작성한다

(3) 프로그래밍 (Programming, Coding)

- 설계된 논리를 직접 프로그램 언어로 바꾸는 작업이다.

(4) 테스트(Software Testing)

- 작성된 프로그램을 실제 데이터나 모의 데이터를 이용하여 검증한다.

(5) 사용(Use)

- 프로그램을 사용자가 받아서 사용한다.

(6) 유지보수(Maintenance)

- 프로그램을 사용자가 받아서 사용하는 중에 발생하는

1. 오류(bug)이나, 2. 새로 추가될 프로그램 기능, 3. 운영체제변경(upgrade)을 재 개발하는 과정이다.



1.3. 소프트웨어 개발 각 단계에서의 문제점

- (1) 요구사항분석(Requirements Analysis) 단계
 - 필요한 개발 내용을 정확하게 프로그래머에게 전달하는 문제
- (2) 설계 혹은 디자인 (Design)
 - 설계도구의 필요성
- (3) 프로그래밍 혹은 코딩 (Coding)
 - 좋은 프로그램 언어의 필요
 - 어떻게 하면 빨리 개발을 할 것인가
 - **정확한 프로그램을 만들 것인가**
 - **빨리 수행되도록 할 것인가**
 - **입력데이터나 프로그램내의 자료를 어떻게 구성할 것인가(자료구조)**
 - 프로그램을 다른 사람이 쉽게 알아볼 수 있도록 작성
- (4) 테스트(Test)
 - 여러 가지 데이터에 대하여 검증필요
 - 실험 데이터의 생성
- (5) 사용(Use)
- (6) 유지보수(Maintenance)
 - 프로그램의 변경이 필요할 때 효과적으로 재개발하는 문제
 - 프로그램 개발자는 보통 개발 업체를 떠난 경우가 많다.



Q/A

- * 소프트웨어 개발단계를 순서대로 적은 것 중 맞는 것은?
 - ① 요구사항분석 - 설계 - 유지보수 - 사용 - 테스트 - 프로그래밍
 - ② 요구사항분석 - 설계 - 프로그래밍 - 테스트 - 사용 - 유지보수
 - ③ 요구사항분석 - 프로그래밍 - 설계 - 유지보수 - 사용 - 테스트
 - ④ 설계 - 요구사항분석 - 유지보수 - 사용 - 테스트 - 프로그래밍

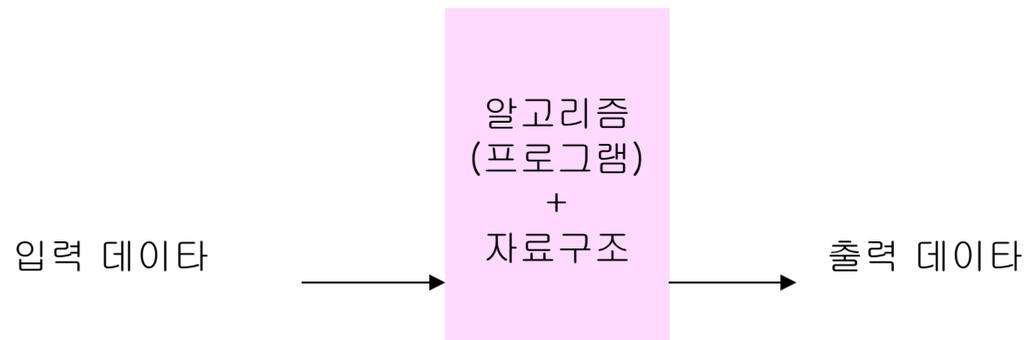


2. 자료 구조의 개념



2.1 자료구조의 개념

- * 자료구조는 소프트웨어 개발 단계에서 “좋은 소프트웨어”를 만들기 위한 “자료의 구조 + 자료의 구조에 대한 프로그래밍”에 관한 분야이다. 소프트웨어 개발의 단계 (3) 과정이다.
- * 소프트웨어는 다음과 같이 구성되어 있다.
소프트웨어 = 알고리즘 + 자료구조
- * 알고리즘은 C, Java 같은 프로그램 언어를 이용하여 프로그램으로 바뀐다.
- * 대표적인 자료구조는 리스트, 트리, 그래프가 있다.





2.2 좋은 소프트웨어(프로그램)란?

좋은 프로그램은 무엇인가?

좋은 프로그램과 나쁜 프로그램을 비교하는 척도는 무엇인가?

좋은 프로그램은?

1. 프로그래머가

개발하기 쉽고(빨리 개발),
검증이 쉬운(알기 쉬운) 프로그램



2. 사용하는 사람(운영자)

사용이 쉽고,
유지보수하기 편하고,
요구사항변경에 적응하기 쉬운 프로그램



3. 컴퓨터(시스템)

실행 시 수행 시간이 빠르고
기억장소를 적게 사용하는 프로그램





2.3 프로그램과 자료구조의 관계

(도서관 사례)

도서관에 책이 1만권 있다. 책을 찾으러 서가를 뒤진다고 해보자. 이 때 도서관의 책 1권 1권은 컴퓨터의 “자료”에 해당이 된다. 책을 찾는 과정은 자료에 대한 작업이 되며 컴퓨터로 말하면 프로그램이 된다.

(사례 1) 도서관의 책이 아무 순서 없이 1만 권이 놓여 있다. 이용자는 책을 찾기 위해 서 많은 도서를 살피고 운이 좋으면 300권 정도 살피다 찾을 수 있고 운이 나쁘면 8,000권 정도를 살펴야 한다. 그렇지만 도서관 운영자는 새로운 도서가 들어올 때 정리 하는 노력을 할 필요가 없다.

사례 1	도서의 상태	책을 찾는 결과
도서관	도서관의 책 1만 권이 아무 순서 없이 놓여있다	책을 찾는 사람의 시간이 많이 소요된다
컴퓨터 자료구조	자료를 컴퓨터에 순서 없이 저장한다.	프로그램의 응답 시간이 길어진다.



사서의 알고리즘
도서등록 프로그램
- 편리

도서 자료구조

이용자 알고리즘
도서검색 프로그램
- 불편





2.3 프로그램과 자료구조의 관계

(사례 2) 도서관의 책을 도서 이름에 따라 **가나다순**으로 14개 그룹으로 정리를 한다. 이용자는 책을 찾기 위해서 먼저 가나다순으로 찾은 후 그 안에서 책을 뒤진다. 운이 좋으면 300권 정도 살피다 찾을 수 있고 운이 나빠도 한 분류에 최대 1,000권이 있다면 1,000권 내에서 찾을 수 있다. 그렇지만 도서관 운영자는 새로운 도서가 들어올 때 가나다순으로 정리하는 노력을 할 필요가 있다.

사례 2	도서의 상태	책을 찾는 결과
도서관	가나다순으로 14개 그룹으로 책 1만 권을 분류하여 각 항목에서는 아무 순서 없이 놓여있다	책을 찾는 사람의 시간이 많이 줄어든다.
컴퓨터 자료구조	자료를 14개 큰 기억 장소에 그룹으로 정리한다.각 그룹에는 무순으로 저장한다.	프로그램의 응답 시간이 많이 줄어 든다.



사서의 알고리즘
도서등록 프로그램
 - 약간 복잡

도서 자료구조

이용자 알고리즘
도서검색 프로그램
 - 약간 불편





2.3 프로그램과 자료구조의 관계

(사례 3) 도서관의 책을 도서 이름에 따라 **가나다순**으로 14개 그룹으로 정리를 한다. 또 각 그룹 안에서는 **가나다순**으로 정리를 한다. 이용자는 책을 찾기 위해서 먼저 가나다순으로 찾은 후 그 안에서 책을 뒤진다. 운이 좋건 나쁘건 10번 이하의 빠른 시간 안에 찾을 수 있다. 그렇지만 도서관 운영자는 새로운 도서가 들어올 때 가나다순으로 정리하는 노력과 각 분류 안에서 가나다 순으로 책을 정리할 필요가 있다. 책을 꽂을 자리가 없으면 서가의 1줄을 이동해야 하는 노력이 필요하다.

사례 3	도서의 상태	책을 찾는 결과
도서관	가나다순의 14개 그룹으로 책 1만 권을 분류하여 각 항목에서는 가나다 순으로 놓여있다	책을 찾는 사람의 시간이 많이 줄어든다.
컴퓨터 자료구조	자료를 14개 큰 기억 장소에 그룹으로 정리한다. 각 그룹에는 데이터를 가나다 순으로 순서대로 저장한다. 혹은 빈 자리가 없으면 기억 장소를 많이 이동한다.	프로그램의 응답 시간이 많이 줄어 든다.



사서의 알고리즘
도서등록 프로그램
- 복잡

도서 자료구조

이용자 알고리즘
도서검색 프로그램
- 편리





2.3 프로그램과 자료구조의 관계

(도서관 사례를 통한 자료 구조의 정리)

앞에서 열거한 여러 경우를 살펴보면 몇 가지를 재미있는 사실을 알 수 있다.

평균 검색 시간(비교횟수)를 줄이기 위한 방법들은 많다(사례 3이 좋다).

자료구조를 신경을 쓰지 않으면 사례 1과 같이 도서검색 프로그램은 시간이 오래 걸린다.

자료구조에 신경을 쓰지 않으면 사례 1과 같이 도서등록 프로그램(운영자)은 간편하다.

자료구조가 바뀌면 프로그램도 같이 영향을 받는다.

알고리즘들 간에(예, 도서검색(search)과 도서등록(insert) 프로그램은 서로 충돌한다.

따라서 자료구조는 프로그램과 긴밀한 관계를 가지며 프로그램의 효율, 복잡도 등에 영향을 미친다. 사용자에게 좋은 프로그램을 만들려면 자료의 구조와 프로그램에 많은 관심을 가져야 한다.

자료구조

< C 자료구조 입문 >

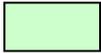


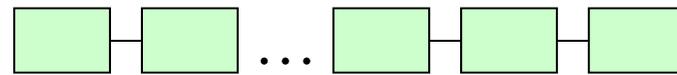
2.4 자료구조에서 배우는 내용

- 1) 자료의 구조
- 2) 자료구조에 대한 알고리즘과 구현
- 3) 정렬과 검색

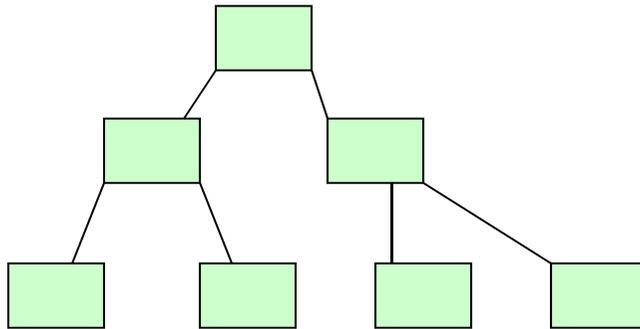


2.4 자료구조에서 배우는 내용

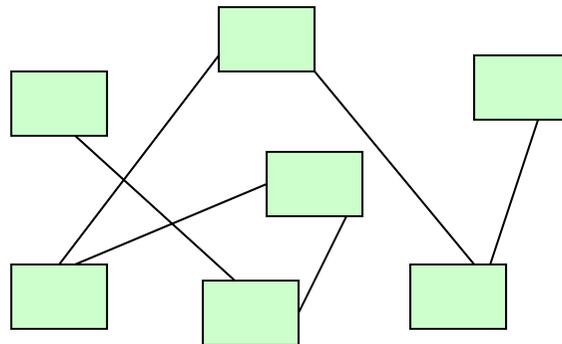
자료의 구조 (자료 : )



리스트
자료구조



트리
자료구조



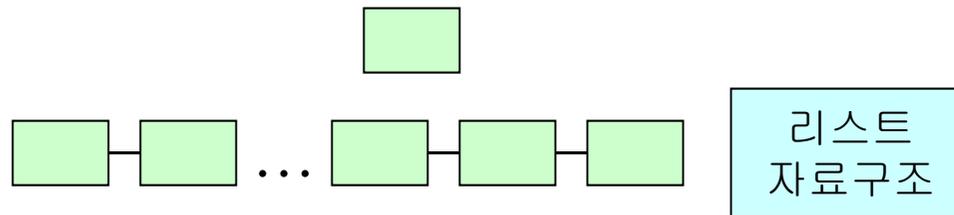
그래프
자료구조

< C 자료구조 입문 >



2.4 자료구조에서 배우는 내용

자료의 구조에 대한 알고리즘 – 1. 리스트 자료구조



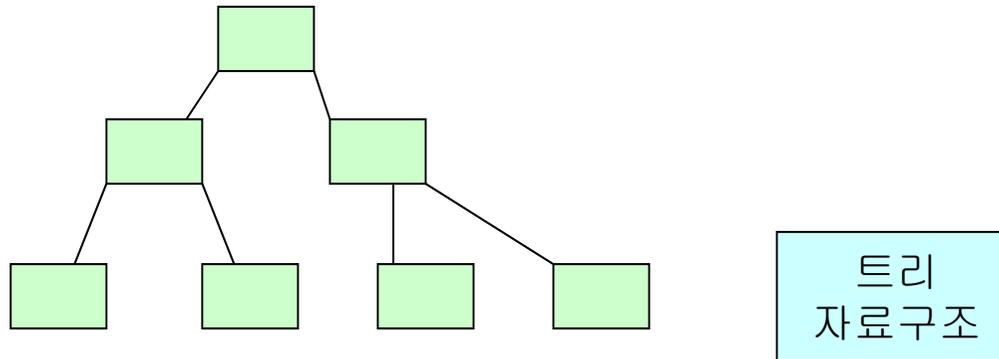
리스트 자료에 대한 알고리즘

1. 일반 리스트를 프로그램으로 구현하는 방법(배열에 저장)
2. 리스트에 관한 연산의 구현(검색과 삽입, 삭제)
3. 스택의 구현(리스트의 한쪽에서 삽입과 삭제가 일어나는 리스트 타입)
4. 큐의 구현(리스트의 한쪽에서 삽입, 반대쪽에서 삭제가 일어나는 리스트 타입)
5. 리스트를 연결리스트에 저장하여 관리하는 방법



2.4 자료구조에서 배우는 내용

자료의 구조에 관한 알고리즘 - 2. 트리 자료구조



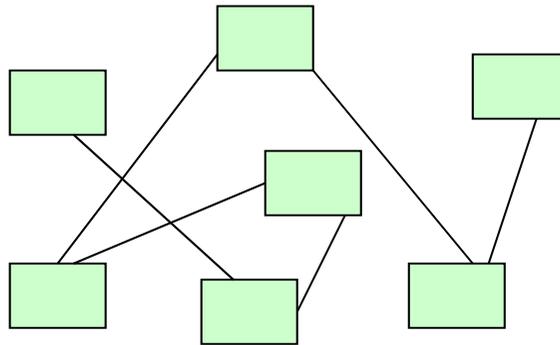
트리 구조 자료에 대한 알고리즘

1. 트리 구조를 생성하고 데이터를 트리 구조에 저장하는 방법
2. 이진 트리(자식노드가 2개 이하인 트리)의 성질
3. 트리의 탐색(Traversal)
4. 트리를 이용한 문제 해결 방법(검색트리, 힙 등)



2.4 자료구조에서 배우는 내용

자료의 구조에 관한 알고리즘 - 3. 그래프



그래프
자료구조

그래프 구조 자료에 대한 알고리즘

1. 그래프 구조를 생성하고 데이터를 그래프 구조에 저장하는 방법
2. 그래프의 성질
3. 그래프의 탐색(Traversal)
 깊이우선탐색, 너비우선탐색
4. 그래프를 이용한 문제 해결 알고리즘(최단거리문제, 스패닝 트리 등)



2.4 자료구조에서 배우는 내용

3. 정렬과 검색

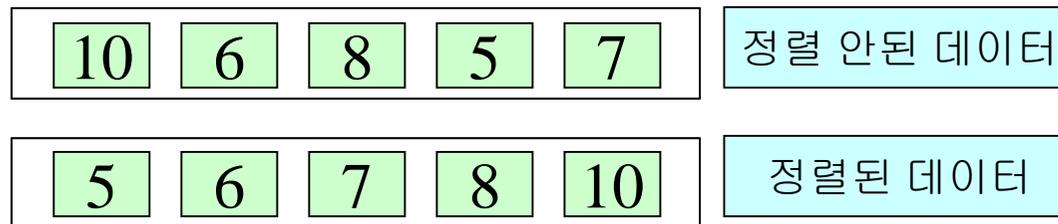
- 정렬(sorting)

정렬은 데이터를 값에 따라 순서대로 배열하는 것이다.

정렬은 검색을 효율적으로 할 수 있도록 돕는다.

정렬하는 방법은

버블정렬, 삽입정렬, 선택정렬, 퀵 정렬, 힙 정렬 등 여러 가지가 있다.





2.4 자료구조에서 배우는 내용

- 검색(searching)

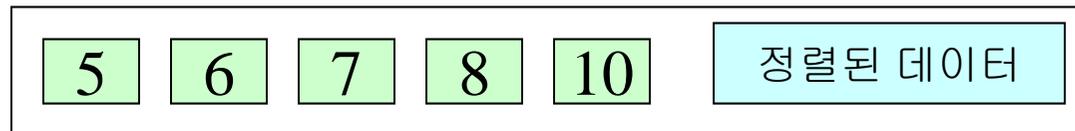
도서관의 많은 도서 자료를 컴퓨터에 저장하여 검색하는 작업의 경우처럼 검색은 데이터에서 원하는 값에 대한 자료를 찾는 것이다.

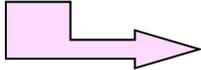
검색은 검색 시간이 가장 문제가 된다.

검색하는 방법은

선형검색 : 처음부터 하나씩 찾는 방법

이진검색 : 정렬된 데이터의 가운데 데이터를 검색하여
데이터의 검색 대상 개수를 $\frac{1}{2}$ 씩 줄여나가는 방법



 8에 관한 데이터 찾기



Q/A

- * 프로그램(소프트웨어) 기능 중 가장 중요한 것은?
 - ① 속도가 빨라야 한다
 - ② 프로그램 기능 수행이 정확해야 한다
 - ③ 기억장소를 적게 사용해야 한다.
 - ④ 프로그램이 알기 쉽게 작성되어야 한다.