



# 강좌 소개

Introduction lecture

About lecturer, plans lecture

# 알고리즘

2022-2

강의 시간 (8반, 0반) – 블렌디드 수업

온라인 강좌 1.5시간 (e-class 동영상)

오프라인 강좌 1.5시간 (공학관 202호)

8반: 목 8M - 9M (16:30 - 17:45)

0반: 목 11M - 12M (19:25 - 20:40)



담당 교수 : 김 광섭 (공학관 A동 300호)



연구실TEL : 02)760 - 5890



공지 / 강의자료 : e-class 업로드



이메일 : engkwangseob@gmail.com

# 알고리즘

주차	내용	주차	내용
1	• 강의 소개 (오리엔테이션)	9	• 그래프 III
2	• 자료구조 복습 (배열, 구조체)	10	• 정렬 I
3	• 자료구조 복습 (스택, 큐, 연결리스트)	11	• 정렬 II
4	• 자료구조 복습 (연결리스트, 트리)	12	• 탐색 I
5	• 우선순위 큐 개념 및 응용	13	• 탐색 II
6	• 그래프 I	14	• 해싱
7	• 그래프 II	15	기말고사
8	중간고사	16	보강 주차(휴강시)

# 알고리즘

🔍 소프트웨어 개발 시 사용할 수 있는 고급 자료구조와 관련된 알고리즘에 대해 학습하고 이러한 이론을 C 프로그래밍 형태로 실습한다. 또한 다양한 정렬 및 탐색 알고리즘에 대해 학습한다.

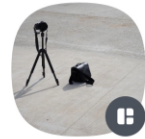
## 🔍 평가 방법

- 출석 + 중간고사 + 기말고사 + 과제 = 총점(100)
  - > 출석 = 10%
  - > 과제 = 20%
  - > 중간고사 = 35%
  - > 기말고사 = 35%
- 상대 평가: 학점 배분 비율에 맞게 학점 부여  
절대 평가: 총 결과에 따른 절대 평가를 통해 학점 부여  
(학교 상황에 따라 변경 가능성 있음)



## 이메일 외 소통 창구

- 공학관 A동 300호 면담
- E-class Q/A 게시판
- 카카오톡 오픈 채팅방
  - URL  
<https://open.kakao.com/me/engkwangseob>
  - QR 코드
  - 다른 수업도 모두 공통으로 사용하므로  
신분 반드시 얘기해줄 것



**kwangseobKim**

한성대학교 김광섭 교수 수업 수강생을 위한 오픈 채팅방



포스트

포스트

채팅

사진과 글로 관심사의 포스트를 작성해  
보세요.

추천 포스트 보기



## 1학기 수업인 자료구조와의 관계

- 아예 관계가 없지 않음
- 그렇다면 수업을 듣지 않은 학생의 경우??
- 기본적으로 C 언어를 수강을 했다면 듣는데 무리 없다고 생각
- 자료구조에 대한 내용 일부 학습 후 진행 예정



C언어로 쉽게 풀어쓴  
**자료구조**

권민석 · 송승혜 · 하현호 지음

▶ 제17판 (2020년 11월 15일)

본 책의 주요 소본권(제17판)은 2020년 11월 15일 현재 최신의 C 언어를 반영하여 개정되었습니다. 본 책의 주요 소본권(제17판)은 2020년 11월 15일 현재 최신의 C 언어를 반영하여 개정되었습니다. 본 책의 주요 소본권(제17판)은 2020년 11월 15일 현재 최신의 C 언어를 반영하여 개정되었습니다.



생능 출판사  
C언어로 쉽게 풀어쓴 자료구조

책을 구매할 필요는 없음  
별도 정리된 강의 자료 배포 예정

# 알고리즘

주차	내용
1	• 강의 소개 (오리엔테이션)
2	• 자료구조 복습 (배열, 구조체)
3	• 자료구조 복습 (스택, 큐, 연결리스트)
4	• 자료구조 복습 (연결리스트, 트리)
5	• 우선순위 큐 개념 및 응용 I
6	• 우선순위 큐 개념 및 응용 II • 그래프 I
7	• 그래프 II
8	중간고사

자료구조에 대한 내용  
(복습 내용이지만, 반드시 확인할  
필요가 있으므로 3주 정도 진행 예정)

자료구조 중 “큐”에서  
우선순위를 가진 항목들을 정하는 큐  
에 대해...

객체 간 관계를 표현하는 자료구조  
기초

# 알고리즘

주차	내용
9	• 그래프 III
10	• 정렬 I
11	• 정렬 II
12	• 탐색 I
13	• 탐색 II
14	• 해싱
15	기말고사
16	보강 주차(휴강시)

그래프를 활용한 알고리즘

자료 탐색의 기본인 정렬에 대한  
알고리즘 종류 학습

효율적인 탐색 알고리즘 종류 학습

탐색 시 키 값을 기본으로 하는 접근



# 알고리즘

---



## 알고리즘?

- 프로그램 = 자료구조 + 알고리즘
- 자료구조에 따라 알고리즘도 결정되므로, 프로그램에 가장 적합한 자료구조와 알고리즘 선택 중요
- 자료구조?
  - 데이터를 저장 시 효율적으로 저장하는 방법
- 알고리즘?
  - 문제를 풀기 위한 단계적인 절차



## 알고리즘의 조건

- 입력이 필요하다면 있어야 하며, 출력은 반드시 1개 이상 존재해야 함
- 각 명령어의 의미는 모호하지 않고 명확해야 함
- 한정된 수의 단계 후에는 반드시 종료되어야 함(유한성)
- 각 명령어들은 실행 가능한 연산이어야 함(유효성)



## 알고리즘의 기술(description) 방법

- 일상 언어를 통한 자연어
  - 우리가 가장 편하게 볼 수 있는 방법
  - 그러나 이를 실제 프로그래밍으로 옮기기에는 쉽지 않음 (모호하기 때문에...)

### 예) 배열에서 최대 값 찾기 알고리즘

#### 배열 안 최대 값 찾기

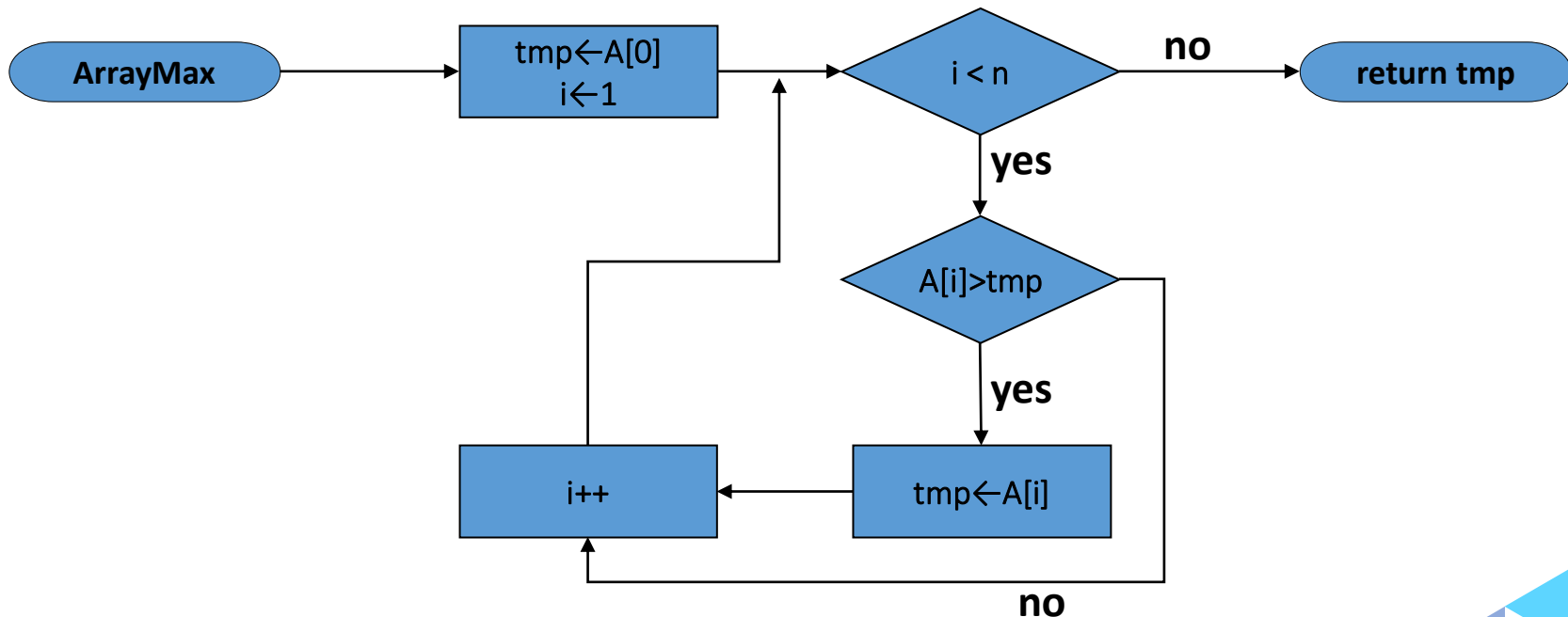
1. 배열 첫번째 요소를 변수 임시 변수에 복사
2. 배열 다음 요소들을 차례대로 임시 변수와 비교하여, 더 크면 임시 변수로 치환
3. 배열 모든 요소를 비교했으면 임시 변수를 최대 값으로 반환

# 알고리즘



## 알고리즘의 기술(description) 방법

- 흐름도 (flow chart)
  - 직관적이고 이해하기 쉬운 방법
  - 다이어그램에 대한 수식을 알고 있어야 하며, 복잡해질 경우 한번에 확인하기는 쉽지 않음



# 알고리즘



## 알고리즘의 기술(description) 방법

- 의사 코드(pseudo-code)
  - 알고리즘 기술에 가장 많이 사용
  - 자연어보다는 좀 더 구체적이면서 프로그래밍 언어보다는 덜 구체적인 표현 방법

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



## 알고리즘의 기술(description) 방법

- 프로그래밍 언어
  - 가장 정확한 기술 가능
  - 언어적인 특성이 따라가기 때문에 알고리즘의 핵심적인 내용에 대한 이해를 방해할 수 있음

```
#define MAX_ELEMENTS 100
int score[MAX_ELEMENTS];
int find_max_score(int n)
{
    int i, tmp;
    tmp=score[0];
    for(i = 1; i < n; i++){
        if( score[i] > tmp ){
            tmp = score[i];
        }
    }
    return tmp;
}
```



## 알고리즘 성능 분석

- 프로그램에 대한 시간 및 공간 효율성을 제고하기 위함
- 수행 시간 측정
- 복잡도 분석
  - 시간 복잡도 분석
  - 공간 복잡도 분석