

알고리즘 수업을 종강하며

1. 교과목 목표 : 프로그래밍 원리의 이해와 스스로 프로그래밍하는 능력 향상

- 컴퓨터로 문제해결 하는 과정(알고리즘)을 예제를 들어 학습
- 이미 알려진 알고리즘의 논리 및 구현 과정의 이해
- 고급자료구조와 프로그래밍 연습

2. 수업 내용 및 평가 : 강의계획서 참조 (중간고사범위)

자료구조를 바탕으로 문제에 대한 해결책을 구상하고 C로 구현한다.

- 컴퓨팅사고와 기본개념(1주)
- 알고리즘의 정의, 표현, 제어(순차, 조건, 반복)
예제를 통한 알고리즘의 이해 및 작성연습(2주)
- 알고리즘을 프로그래밍 언어로 구현
C의 기본 자료처리와 함수 작성에 의한 모듈화(3주)
- Recursive Algorithm(gcd, Hanoi)
반복제어구조와 recursion의 이해(4주)
- 알고리즘 설계와 구현단계 : Top-down 방식의 문제해결절차 연습
Selection Sort(선택정렬)(5주)
Binary Search(이진검색)(6주)
- 규칙을 코드로 구현(Magic Square)(6주)
- 기타 : 배열, 파일처리, 구조체의 자료구조활용/컴퓨팅사고-분해/추상화/규칙찾기

기말고사준비 가이드라인

2. 수업 내용 및 평가 : 강의계획서 참조(중간고사 이후=아래 내용 기말고사 범위)

자료구조를 바탕으로 문제에 대한 해결책을 구상하고 C로 구현한다.

- (1) 난수(random number) 생성기 (9주) - 수업동영상9주차01(시험범위 제외)
상태도 분석을 통한 알고리즘의 이해
- (2) 스택을 이용한 알고리즘 (10주)
수식의 변경과 계산
- (3) 노드기반 자료구조 구축 연습(10주) - 수업동영상10주차02(시험범위 제외)
- (4) 자기참조구조체를 이용한 이진탐색트리 구축 및 활용 (11주)
- (5) 정렬알고리즘(버블, 삽입, 퀵, 합병) (12-13주)
- (6) 알고리즘의 성능분석 big-O 개념(12주)

3. 평가 : 중간(30), 기말(30), 과제물및 기타(20), 출석(20)

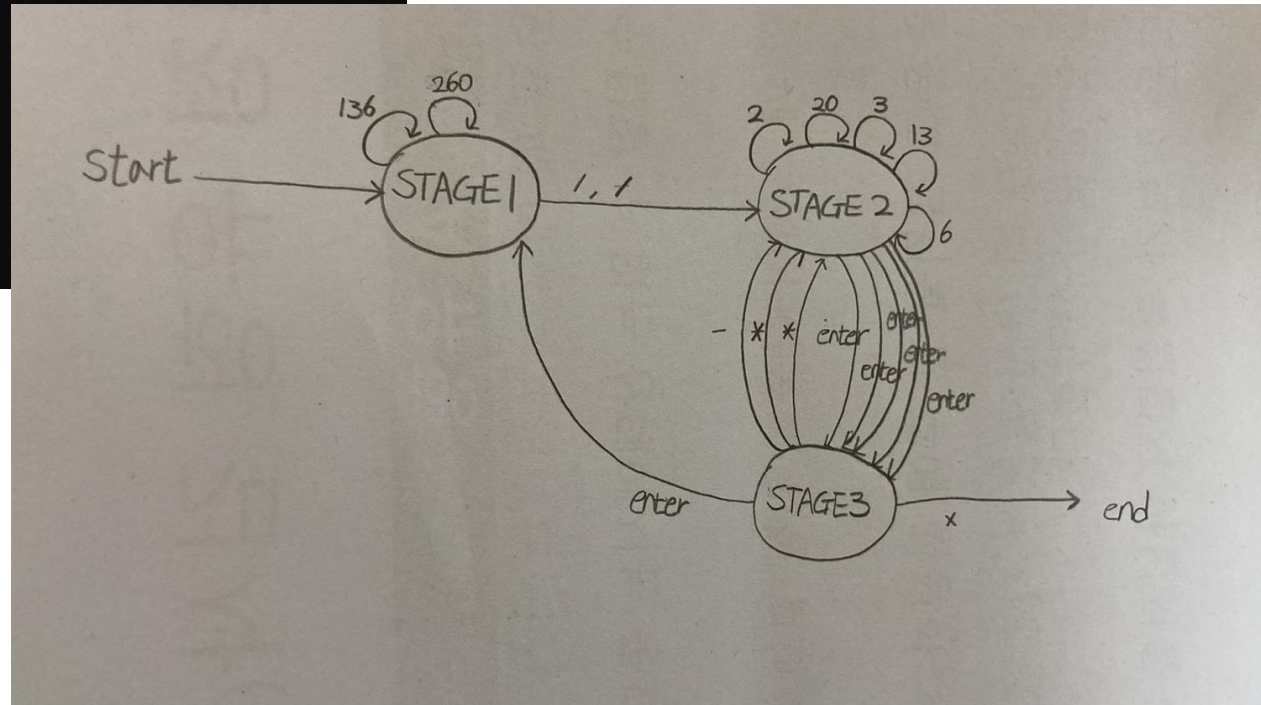
9주차부터 13주차

수업자료pdf(교재)+동영상수업+수업정리과제를
개념과 논리 이해하고 C코딩(적으면서 정리)으로 알고리즘 표현 연습

상태도 분석

1. 상태도 분석을 통한 알고리즘의 이해-STD 그리기

```
D:\Waprogram\wint_cal.exe
계산기 프로그램을 시작합니다.
정수형의 피연산자와 사칙연산자 중 하나를 입력하시오.
결과를 받은 후 끝내려면 문자 x를 입력하시오
136/2
= 68-20
= 48*3
= 144
260/13
= 20*6
= 120x
계산기 프로그램을 종료합니다.
```



스택을 이용한 알고리즘

다음의 중위 표기식을 가지고 물음에 답하시오.

$$2*((3+9)/3+4)/2$$

(1) 스택의 변화를 나타내며 후위표기식으로 변경하시오.

+			
(/	+	
(((
*	*	*	-

후위표기식 : 2 3 9 + 3 / 4 + * 2 /

스택을 이용한 알고리즘

```
int cal(void) {
    char symbol;
    int op1, op2, n = 0;
    top = -1;
    symbol = pexpr[n++];
    while (symbol != '\0') {
        if (is_operator(symbol)) {
            op2 = delete_stack();
            op1 = delete_stack();
            switch (symbol) {
                case '+':
                    add_stack(op1 + op2);
                    break;
                case '-':
                    add_stack(op1 - op2);
                    break;
                case '*':
                    add_stack(op1 * op2);
                    break;
                case '/':
                    add_stack(op1 / op2);
            }
        }
        else
            add_stack(symbol - '0');
        symbol = pexpr[n++];
    }
    return delete_stack();
}
```

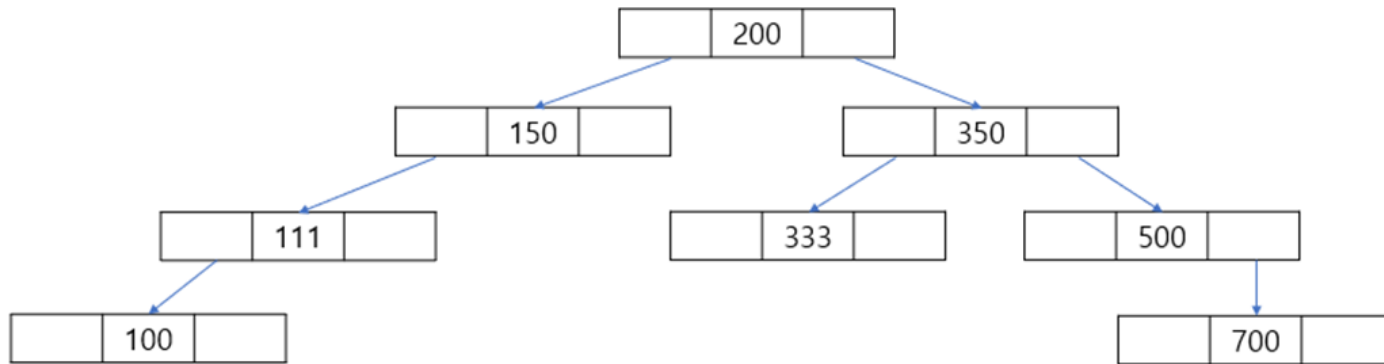
후위표기식 : 2 3 9 + 3 / 4 + * 2 /

n	symbol	cal()에서 수행한 함수	top
0, 1, 2	'2', '3', '9'	add_stack(2), add_stack(3), add_stack(9)	2
3	'+'	2delete_stack() => 9, 3, add_stack(12)	1
4	'3'	add_stack(3)	2
5	'/'	2delete_stack() => 3, 12, add_stack(4)	1
6	'4'	add_stack(4)	2
7	'+'	2delete_stack() => 4, 4, add_stack(8)	1
8	'*'	2delete_stack() => 8, 2, add_stack(16)	0
9	'2'	add_stack(2)	1
10	'/'	2delete_stack() => 2, 16, add_stack(8)	0

이진탐색트리의 구축 활용

정수형 key 값이 다음과 같이 차례로 주어지는 경우 만들어진
이진검색트리를 그리고 중위순회(inorder traversal)한 결과를 쓰시오.

key 값 : 200, 350, 150, 111, 500, 333, 100, 700



중위순회의 결과 값 : 200 150 111 100 350 333 500 700

이진탐색트리의 구축 활용

사원번호(정수형), 입사년도(정수형), 인사등급(문자형)을 자료로 가지는 이진탐색트리의 노드 구조(ENODE)를 정의하고 준비한 사원정보를 입력하고 입력의 끝에는 (0 0 0)을 입력한다.

다음은 이렇게 입력된 정보를 사원번호를 기준으로 이진 검색트리를 구축하고 그 정보를 출력하는 프로그램이다.

또한 이 프로그램에서는 다음의 두가지 정보를 출력한다.

- 1) 입사년도를 입력하면 그 해에 입사한 사원의 정보(사원번호, 인사등급)를 출력한다
 - 2) 사원번호를 입력하면 그 번호에 해당하는 사원정보(입사년도 인사등급)를 출력한다
- inorder()와 year_search() 의 코드를 작성하여 추가하시오.

이진탐색트리의 구축 활용

```
typedef struct node {  
    struct node *left;  
    int eid;  
    int syear;  
    char grade;  
    struct node *right;  
} ENODE;
```

```
void inorder(ENODE *root);
```

```
int insert_node(ENODE *root, int id, int year, char score) {  
    ENODE *tptr = root, *before;  
    while (tptr) {  
        if (id < tptr->eid) {  
            before = tptr;  
            tptr = tptr->left;  
        }  
        else if (id > tptr->eid) {  
            before = tptr;  
            tptr = tptr->right;  
        }  
        else // found  
            return 0;  
    }  
    tptr = (ENODE *)malloc(sizeof(ENODE));  
    tptr->eid = id;  
    tptr->syar = year;  
    tptr->grade = score;  
    tptr->left = tptr->right = NULL;  
    if (id < before->eid) before->left = tptr;  
    else before->right = tptr;  
    return 1;  
}
```

```
void year_search(ENODE *root, int year);
```



```

void year_search(ENODE *ptr, int year) {
    if (ptr) {
        year_search(ptr->left, year);
        if ((ptr->year) == year)
            printf("%d\t%c\n", ptr->eid, ptr->grade);
        year_search(ptr->right, year);
    }
}

```

```

ENODE *search(ENODE *root, int key) {
    ENODE *tptr = root;
    while (tptr) {
        if (key < tptr->eid)
            tptr = tptr->left;
        else if (key > tptr->eid)
            tptr = tptr->right;
        else // found
            return tptr;
    }
    return NULL; // not found
}

```

검색 트리에 저장할 직원정보를 입력하세요.
 직원번호, 입사년도, 인사등급을 입력하고
 입력의 끝에는 (0 0 0)를 입력하세요.

```

199 2015 B
230 2017 C
101 2014 B
707 2020 C
355 2019 A
717 2020 B
555 2019 B
150 2015 A
220 2016 A
111 2014 C
600 2019 C
299 2017 A
500 2019 B
414 2019 A
0 0 0

```

트리에 구축된 직원정보 :

```

101 2014 B
111 2014 C
150 2015 A
199 2015 B
220 2016 A
230 2017 C
299 2017 A
355 2019 A
414 2019 A
500 2019 B
555 2019 B
600 2019 C
707 2020 C
717 2020 B

```

입사년도 : 2019

```

355 A
414 A
500 B
555 B
600 C

```

직원 번호 : 555
 555번 직원의 정보 : 2019 B

정렬 알고리즘

1. 개요

정보처리에 있어 정렬의 중요성

기준 값(key)이 커지는 순으로(오름차순)

또는 작아지는 순으로(내림차순) 데이터를 나열하는 것

2. 버블 정렬(Bubble Sort)

3. 삽입 정렬(Insertion Sort)

4. 퀵 정렬(Quick Sort)

5. 합병 정렬(Merge Sort)

수업정리과제 이해! 확인

알고리즘 성능분석

▶ 시간 복잡도 분석

산술, 대입, 비교, 이동의 기본적인 연산 고려

- 알고리즘 수행에 필요한 연산의 개수를 계산
- 입력의 개수 n 에 대한 함수->**시간 복잡도 함수**

big-O표기법 :

알고리즘 수행 단계를 바탕으로

포괄적(점근적)으로 알고리즘성능의 범주를 정한 것

$O(1)$	상수시간
$O(N)$	선형시간
$O(N^2)$	이차시간
$O(\log N)$	로그시간

시간 복잡도 분석

▶ 시간 복잡도 분석

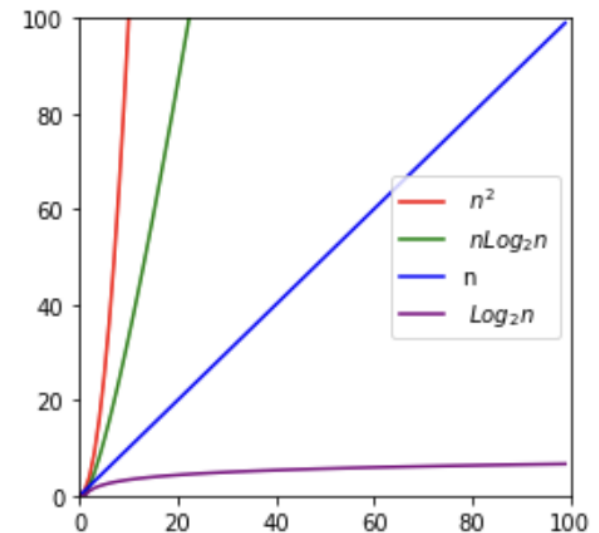
- 알고리즘 수행에 필요한 연산의 개수를 계산
- 입력의 개수 n 에 대한 함수 → **시간 복잡도 함수**

big-O표기법 :

알고리즘 수행 단계를 바탕으로

포괄적으로 알고리즘성능의 범주를 정한 것

순차검색	$O(n)$
이진검색	$O(\log n)$
버블정렬	$O(n^2)$
삽입정렬	$O(n^2)$
퀵정렬	$O(n \log n)$
합병정렬	$O(n \log n)$



기말고사 준비는 이렇게

9주차부터 13주차

**수업자료pdf(교재)+동영상수업+수업정리과제를
개념과 논리 이해하고 C코딩(적으면서 정리)으로 알고리즘 표현 연습**

시험공부 하면서 마지막 14주차 과제함에 스스로 문제 내고 답하기

한학기 동안 수고 많이 하셨습니다!!!