

알고리즘 설계와 구현(1)

Top-down approach에 의한 설계 - Stepwise Refinement

1. 문제정의 단계에서 얻은 문제해결방안을 바탕으로 Tree 형태로 문제를 분할한다.
2. 하위레벨로 내려 올 수록 구체화 된다.
3. 설계된 작업을 알고리즘으로 작성한다.
4. 최종적으로 필요한 함수의 종류를 마련하고 프로그램코드로 표현될 때까지 세분화한다.

예) 회사의 조직기법 - 일의 분할

식물이나 동물의 분류도

일상생활의 문제해결 방안

선택정렬(Selection Sort)-문제정의

가. 문제 정의

n개의 데이터로 구성되어 있는 정수 데이터집합의 데이터를 오름차순으로 정렬한다.

“정렬되지 않은 정수들 중에서 가장 작은 값을 찾아서

이미 정렬된 리스트의 다음자리에 놓는다.”

입력 : 데이터의 수(n), 정수 배열(list)

출력 : 정렬된 정수배열(list)

예)

0 1 2 3 4 5 6 7 8 9

int list[10] = { 57, 77, 100, 10, 35, 65, 30, 90, 20, 45 }

배열의 각 데이터는 인덱스에 의하여 참조 된다. list[6]=?

배열의 각 데이터를 처리하기 위하여 for문을 사용한다.

선택정렬-알고리즘 구상

나. 알고리즘 구상

문제분할

1. 처리할 데이터를 배열 list에 저장한다.
2. 정렬되지 않은 데이터 집합으로부터 최소값을 찾는다.
3. 단계2에서 찾은 데이터를 정렬된 list의 다음 자리에 놓는다.
4. 모두 정렬될 때까지 단계2와 단계3을 반복한다.
5. 결과를 출력한다.



구체화시키기 위한 준비

- (1) 단계2에서 최소값을 찾는 데이터의 구간을 어떻게 정할 것인가?

입력 데이터는 list[0]에서 list[n-1]에 저장되어있음

배열의 구간은 [0:n-1]에서 시작하고 단계2와 단계3의 과정이 진행되면서 구간이 줄어든다.

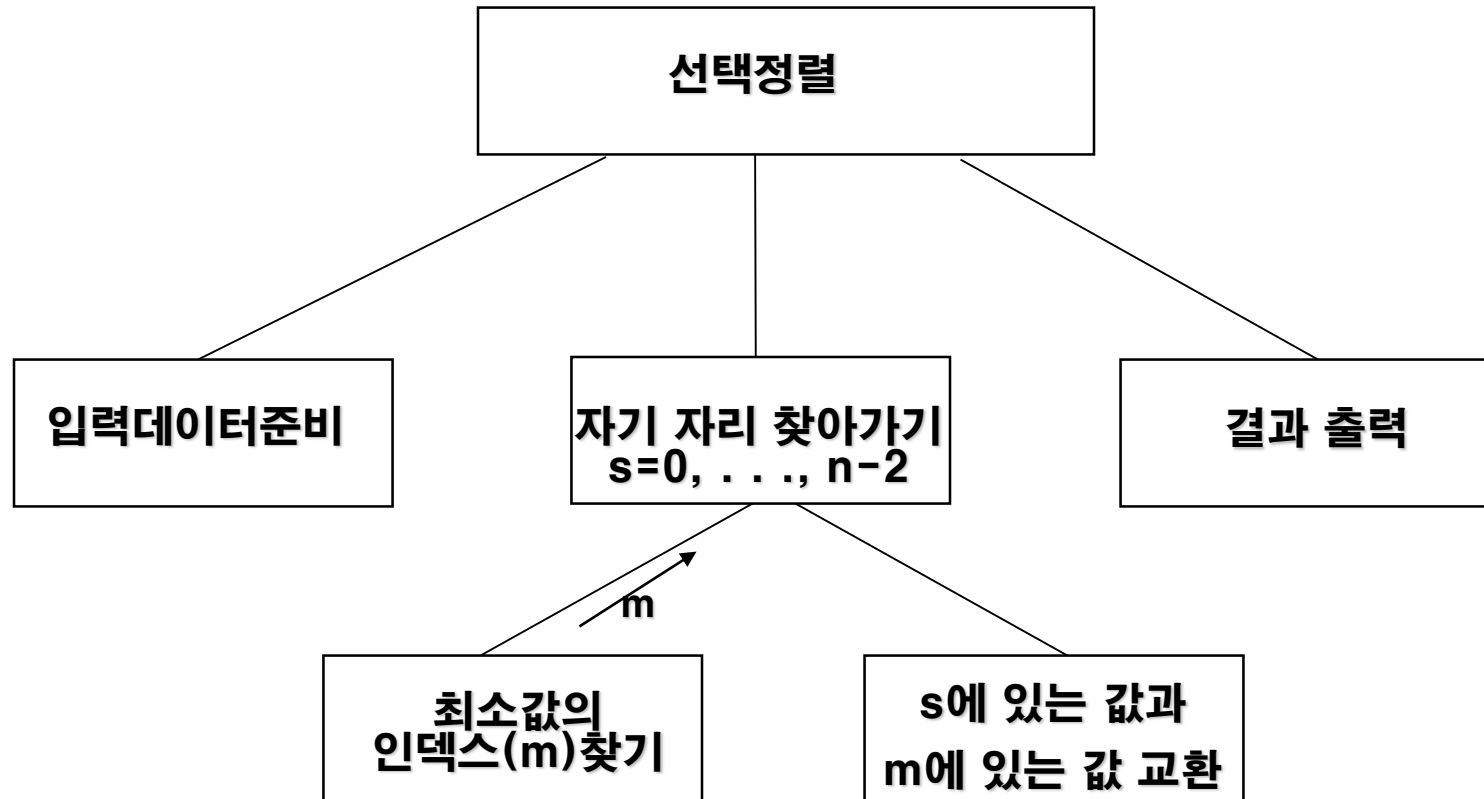
구간의 시작을 표현하는 변수 필요하다(s)

- (2) 단계2에서 최소값을 가지는 데이터의 위치정보(배열의 인덱스)가 필요하다.(m)

- (3) 단계3에서 list[s]의 위치에 list[m]을 저장해야한다.(list[s]와 list[m]의 값의 교환)

- (4) 단계4에서 “모두 정렬될 때까지” 더욱 명확하게하다. (s=0, 1, . . . ,n-2)

선택정렬-알고리즘 구상



선택정렬-Simulation

다. Simulation

가능한 입력데이터를 준비한다.

예) $n=7$

`list = {45, 100, 38, 90, 17, 75, 56}`

지금까지 구상한 알고리즘에 따라 데이터를 실행시킨 각 단계의 결과를 확인한다.

- 구상한 알고리즘 검토
- 좀 더 코딩레벨로 알고리즘을 기술할 수 있도록 구체화시킨다.

선택정렬-Simulation

다. Simulation

n=7	0	1	2	3	4	5	6	m
list	45	100	38	90	17	75	56	
S=0	17	100	38	90	45	75	56	4
S=1	17	38	100	90	45	75	56	2
S=2	17	38	45	90	100	75	56	4
S=3	17	38	45	56	100	75	90	6
S=4	17	38	45	56	75	100	90	5
S=5	17	38	45	56	75	90	100	6

선택정렬-알고리즘작성

라. 알고리즘 작성

단계 1 : 배열 list 에 입력 데이터 n개를 저장한다.

list[0], list[1], list[2] . . . list[n-1]

단계 2 : s=0로 한다. 즉 정렬된 데이터가 처음에는 없다고 가정한다.

단계 3 : s = n-1 이면 모든 데이터가 정렬되어 있으므로 단계 6으로 간다.

아니면 s부터 n-1의 위치에 있는 데이터 중 최소값을 찾는다.

(m의 위치라 가정한다.)

단계 4 : list[m]은 list[s]와 교환한다.

단계 5 : s를 하나 증가 시키고 단계 3으로 간다.

단계 6 : 정렬된 list를 출력하고 작업을 끝낸다.

```
prepare input data into list
for (s=0; s < n-1; s++) {
    m = min_index(list[s] . . . list[n-1])
    swap(list[m], list[s])
}
print the result
```

선택정렬-C함수 구현

마. C함수 구현

```
void selection_sort(int list[], int n)
{
    int s, m, j, temp;

    for (s = 0; s < n-1; s++) {
        m = s;
        for (j = s+1; j < n; j++)
            if (list[j] < list[m]) m = j;
        temp = list[s];
        list[s] = list[m];
        list[m] = temp;
    } /* for */
} /* sort */
```

```
prepare input data into list
for (s=0; s < n-1; s++) {
    m = min_index(list[s] . . . list[n-1])
    swap(list[m], list[s])
}
print the result
```


정리 및 실습

1. 선택정렬 알고리즘을 생각하며 다음의 예에 대하여 시뮬레이션 하면서 m 의 값과 변경되는 $list$ 의 값을 쓰시오.

$n=6$, $list=\{35, 95, 25, 70, 15, 58\}$

$n=6$	0	1	2	3	4	5	m
list	35	95	25	70	15	58	
$S=0$							
$S=1$							
$S=2$							
$S=3$							
$S=4$							

정리 및 실습

2. 배열 list에 입력데이터를 받고, selrection_sort(int list[], int n)에 의해 정렬하고 그 결과를 출력하는 함수 print_data(int list[], int n)을 작성하여 main()에서 사용하도록 프로그램을 작성하여 실행시켜 보시오

3. 내림차순으로 정렬한다면 어떤 부분을 바꾸어야 하는지 고려하여 selrection_sort_d(int list[], int n)를 작성하여 실행시켜 보시오.

```
m = s;  
for (j = s+1; j < n; j++)  
    if (list[j] < list[m]) m = j;
```

F:\waprogram\class5\w

```
0 [40]  
1 [30]  
2 [80]  
3 [70]  
4 [100]  
5 [10]  
6 [90]  
7 [20]  
8 [170]  
9 [60]  
10 [80]
```

```
0 [10]  
1 [20]  
2 [30]  
3 [40]  
4 [60]  
5 [70]  
6 [80]  
7 [80]  
8 [90]  
9 [100]  
10 [170]
```

자료구조복습(구조체)

◆ 구조체의 개념 및 활용

구조체 : 여러 형의 데이터를 하나의 객체로 선언하여 사용함

```
typedef struct member{  
    int id;  
    char name[20];  
    float score;  
} example;
```

struct member 라는 구조체 형이 선언된 것이고 동시에
typedef 에 의하여 example이라는 데이터 형을 정의함.

//구조체를 프로그램에서 변수로 사용할 수 있게 됨.

```
example onep, exmember[20];
```

◆ 구조체의 활용

```
example onep, exmember[20];
```

```
onep.id = 2102;
```

```
strcpy(onep.name, "Hong Gildong" );
```

```
onep.score = 4.23;
```

```
// exmember중에 성적이 4.00이상인 학생의 id를 출력하시오.
```

```
for (k=0; k < 20; k++)
```

```
    if (exmember[k].score >= 4.0)
```

```
        printf( "%d\n" , exmember[k].id);
```

구조체 활용 실습

```
typedef struct member {  
    int id;  
    char name[20];  
    float score;  
} example;
```

```
void over40(example *a, int n);  
example find_max(example *a, int n);
```

```
main()  
{  
    example onep, exmember[20];  
    int i, n;  
    printf( "Input the number of data : ");  
    scanf( "%d" , &n);  
  
    for (i=0; i < n; i++) {  
        printf("데이터 입력 : ");  
        scanf("%d %s %f",  
            &exmember[i].id, exmember[i].name,  
            &exmember[i].score);  
    }  
    over40(exmember, n);  
    onep = find_max(exmember, n);  
    printf("Information of the highest score student  
    = %d : %s : %.2f\n",  
        onep.id, onep.name, onep.score);  
}
```

구조체 활용 실습

```
void over40(example a[], int n)
{
    int k;

    printf("List of high score
           students(over 4.0)\n");
    for (k=0; k < n; k++)
        if (a[k].score > 4.0)
            printf("%d -- %s\n",
                   a[k].id, a[k].name);
}
```

k	id	name	score
0	121	Park	4.01
1	170	Lee1	3.98
2	200	Kim1	3.33
3	100	Lee2	4.32
4	300	Kim2	4.11
5	210	Lim	3.88

```
example find_max(example a[], int n)
{
    int k, max_index;

    max_index = 0;
    for (k=1; k < n; k++)
        if (a[k].score > a[max_index].score)
            max_index = k;

    return a[max_index];
}
```

F:\wprog\class5\structex.exe

```
Input the number of data : 6
데이터 입력 : 121 Park 4.01
데이터 입력 : 170 Lee1 3.98
데이터 입력 : 200 Kim1 3.33
데이터 입력 : 100 Lee2 4.32
데이터 입력 : 300 Kim2 4.11
데이터 입력 : 210 Lim 3.88
List of high score students(over 4.0)
121 -- Park
100 -- Lee2
300 -- Kim2
Information of the highest score student = 100 : Lee2 : 4.32
```

파일과 구조체 활용 실습

다음의 예제 입력파일과 같이 준비된 수강신청 데이터를 활용한 예제 프로그램을 작성해 보면서
파일의 데이터를 구조체로 읽어 활용하는 예제를 살펴봅니다

[입력파일 예]

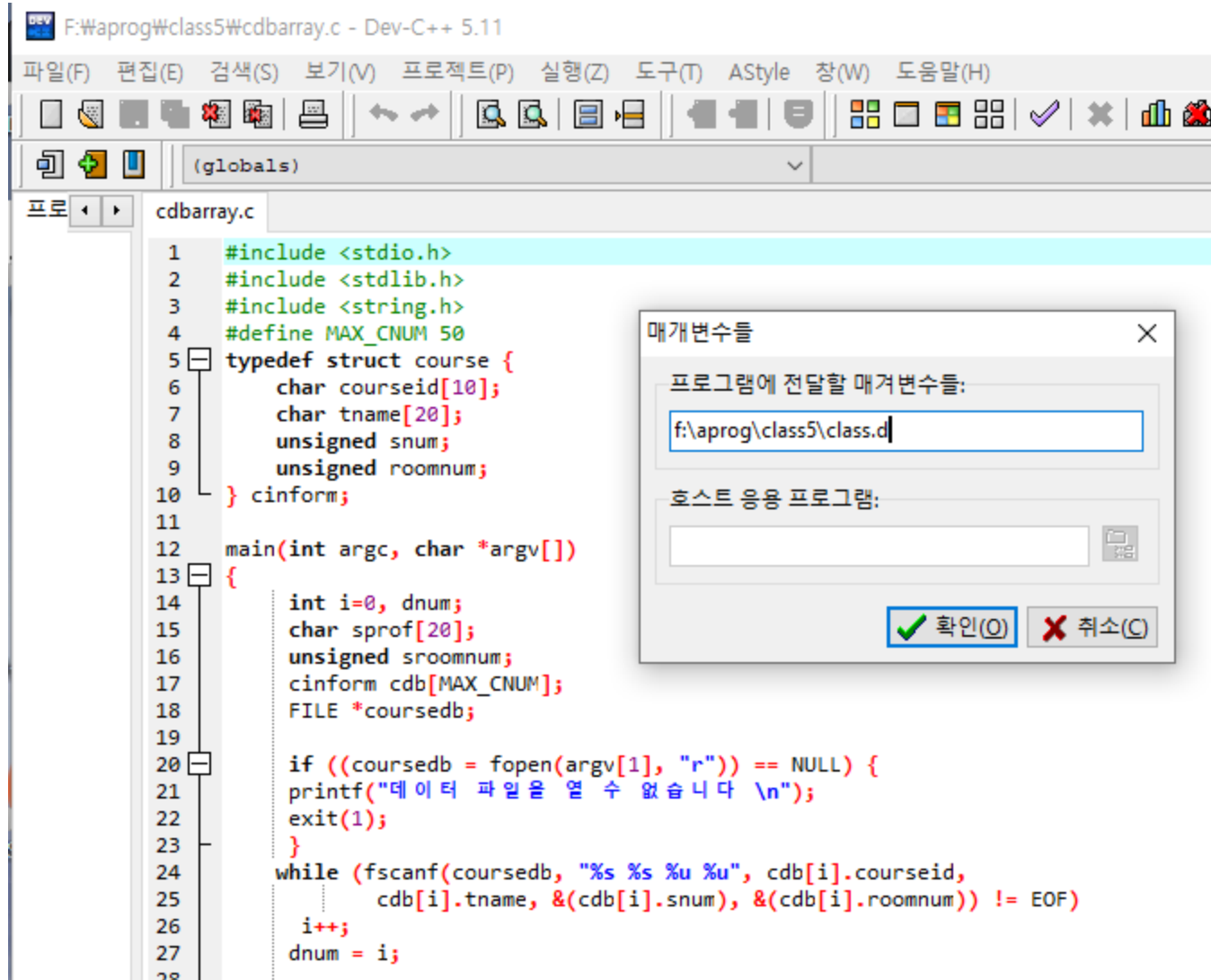
강좌코드 담당교수이름 수강생수 강의실번호

db01	YSKIM	37	1217
dmath	DHLEE	42	2301
arch01	HJCHOI	33	1108
ds01	HSRHEE	45	1307
db02	HJCHO	28	2311
algo01	HSRHEE	35	1305
ds02	HSRHEE	37	1307
arch02	HJCHOI	36	1108
algo02	HSRHEE	43	1305
network	HJYOON	22	2200
web	YSKIM	43	2100

```
typedef struct course {  
    char courseid[10];  
    char tname[20];  
    unsigned snum;  
    unsigned roomnum;  
} cinform;
```

1. 수강인원이 40명이상인 강좌코드를 출력하시오
2. 교수이름이 **HSRHEE**인 교수가 강의하는 강좌코드와 강의실번호를 출력하시오
3. 강의실 번호 **1108** 에서 강의하는 강좌코드와 담당교수를 출력하시오

파일과 구조체 활용 실습



파일과 구조체 활용 실습

```
printf("수강인원이 40명 이상인 강좌 코드 번호를 출력하시오.\n");  
for (i=0; i < dnum; i++)  
    if (cdb[i].snum >= 40) printf("%s\n", cdb[i].courseid);
```

```
printf("검색하고자 하는 교수이름 입력 : ");  
scanf("%s", sprof);  
printf("%s 교수가 강의하는 강좌코드와 강의실 번호를 출력하시오.\n", sprof);  
for (i=0; i < dnum; i++)  
    if (!strcmp(cdb[i].tname, sprof))  
        printf("%s\t%u\n", cdb[i].courseid, cdb[i].roomnum);
```

```
printf("검색하고자 하는 강의실 번호 입력 : ");  
scanf("%u", &sroomnum);  
printf("%u 강의실에서 수업하는 강좌코드와 담당교수를 출력하시오.\n", sroomnum);  
for (i=0; i < dnum; i++)  
    if (cdb[i].roomnum == sroomnum)  
        printf("%s\t%s\n", cdb[i].courseid, cdb[i].tname);
```

파일과 구조체 활용 실습

db01	YSKIM	37	1217
dmath	DHLEE	42	2301
arch01	HJCHOI	33	1108
ds01	HSRHEE	45	1307
db02	HJCHO	28	2311
algo01	HSRHEE	35	1305
ds02	HSRHEE	37	1307
arch02	HJCHOI	36	1108
algo02	HSRHEE	43	1305
network	HJYOON	22	2200
web	YSKIM	43	2100
mobile	EYKANG	39	3215
db03	YSKIM	28	2311

F:\wprog\class5\cdbarray.exe

수강인원이 40명 이상인 강좌 코드를 출력하시오.

dmath

ds01

algo02

web

검색하고자 하는 교수이름 입력 : HSRHEE

HSRHEE 교수가 강의하는 강좌코드와 강의실 번호를 출력하시오.

ds01 1307

algo01 1305

ds02 1307

algo02 1305

검색하고자 하는 강의실 번호 입력 : 1307

1307 강의실에서 수업하는 강좌코드와 담당교수를 출력하시오.

ds01 HSRHEE

ds02 HSRHEE