

프로그래밍언어

교과목명 : 프로그래밍언어

담당교수 : 이 수 형

E-mail : soohyong@uu.ac.kr

교재명 : C언어의 완성

정렬 알고리즘

- 정렬
 - 주어진 데이터들을 크기 순서대로 재배치하는 것
- 오름차순, 내림차순
 - 오름차순 : 크기가 증가하는 순서 (예: 석차순, 학번순, ...)
 - 내림차순 : 크기가 감소하는 순서 (예: 성적순, ...)
- 내부정렬, 외부정렬
 - 내부정렬
 - 정렬대상을 한꺼번에 메인 메모리에 올릴 수 있을 때
 - 외부정렬
 - 정렬대상을 한꺼번에 메인 메모리로 올릴 수 없을 때
 - 메인 메모리와 보조 메모리 사이를 들락날락 하면서 정렬

정렬 알고리즘 : 선택 정렬

- 선택정렬 (Selection Sorting)
 - 가장 기본적인 정렬 알고리즘
 - 첫번째 단계에서 가장 큰 것을 선택하여 마지막과 교환
 - 두번째 단계에서 마지막(하늘색)을 제외한 나머지 중에서 가장 큰 것을 선택하여 마지막의 앞의 데이터와 교환
 - 동일한 방법으로 각 단계별로 가장 큰 것을 제일 뒤쪽의 데이터로 이동 (하늘색 제외)하면서 이 과정을 반복함 → for문 사용 (N개의 데이터인 경우 N-1번 반복)

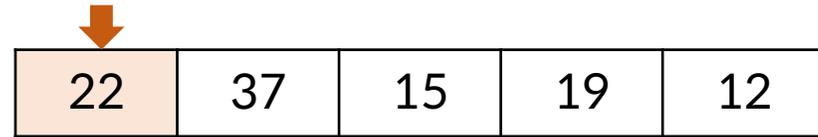
22	37	15	19	12
22	12	15	19	37
19	12	15	22	37
15	12	19	22	37
12	15	19	22	37

정렬 알고리즘 : 선택 정렬

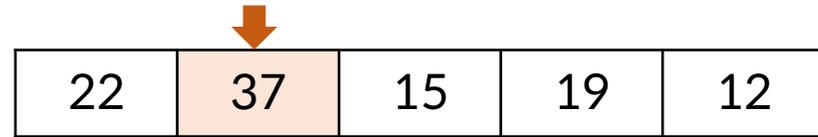
- 선택 정렬 : 첫번째 반복

- 목적 : 전체 데이터 중에서 가장 큰 것을 찾아서 마지막 데이터와 교환

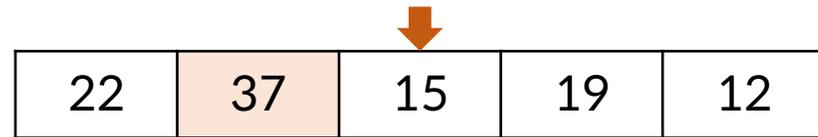
- 첫번째 데이터가 가장 큰 값으로 설정함



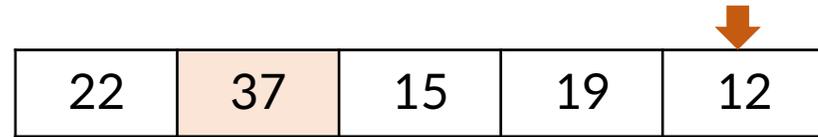
- 두번째 데이터를 가장 큰 값과 비교해서 두번째가 더 크다면 더 큰 값으로 설정



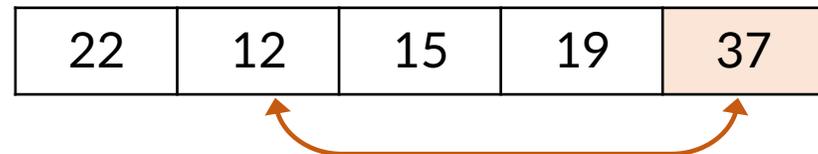
- 세번째 데이터를 가장 큰 값과 비교해서 세번째가 더 크다면 더 큰 값으로 설정



- 위의 과정을 마지막 데이터까지 비교하면서 반복 → for문으로 구현



- 큰 값과 마지막 값 교환

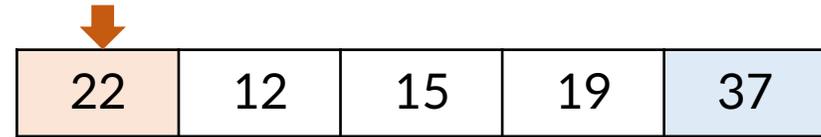


정렬 알고리즘 : 선택 정렬

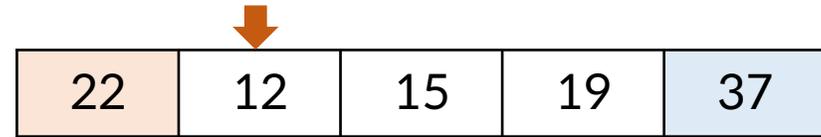
- 선택 정렬 : 두번째 반복

- 목적 : 전체 데이터 중에서 가장 큰 것을 찾아서 마지막 데이터와 교환

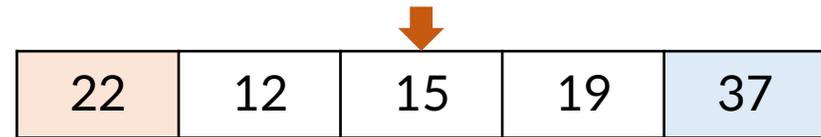
- 첫번째 데이터가 가장 큰 값으로 설정함



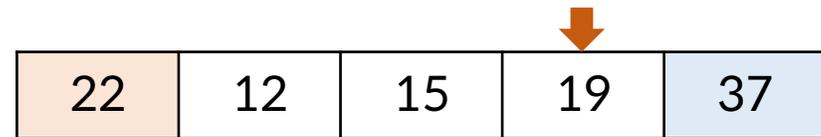
- 두번째 데이터를 가장 큰 값과 비교해서 두번째가 더 크다면 더 큰 값으로 설정



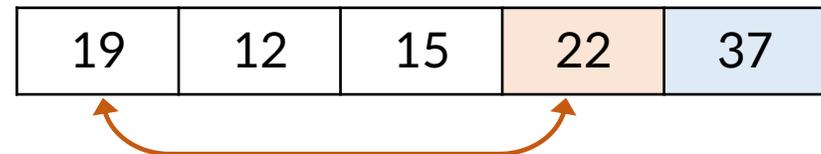
- 세번째 데이터를 가장 큰 값과 비교해서 세번째가 더 크다면 더 큰 값으로 설정



- 위의 과정을 마지막 앞 데이터까지 비교하면서 반복 → for문으로 구현 (데이터 수 N - 1번)



- 큰 값과 마지막 값 교환



최대값 탐색 알고리즘

- 배열의 요소 중에서 최대값을 탐색
 - 첫번째 요소를 최대값이라고 가정 : `maxvalue = data[0]`;
 - 두번째 값부터 마지막까지 반복하면서 최대값과 비교하여 큰 값이면 최대값을 수정

```
#include <stdio.h>
#define NO 5

int main()
{
    int i;
    int data[NO] = {22, 37, 15, 19, 12};
    int maxvalue;

    maxvalue = data[0];
    for (i = 1; i < NO; i++) {
        if (data[i] > maxvalue)
            maxvalue = data[i];
    }
    printf("Max. value : %d\n", maxvalue);
}
```

최대값 탐색 알고리즘

- 배열의 요소 중에서 최대값의 위치를 탐색
 - 최대값 위치를 처음이라고 가정 : `maxidx = 0`;
 - 두번째 값부터 마지막까지 반복하면서 최대값과 비교하여 큰 값이면 최대값 위치를 수정

```
#include <stdio.h>
#define NO 5

int main()
{
    int i;
    int data[NO] = {22, 37, 15, 19, 12};
    int maxidx;

    maxidx = 0;
    for (i = 1; i < NO; i++) {
        if (data[i] > data[maxidx])
            maxidx = i;
    }
    printf("Max. position : %d, value : %d\n", maxidx, data[maxidx]);
}
```

정렬 알고리즘 : 선택 정렬

- 선택정렬 (Selection Sorting)

- N개의 데이터의 경우 N-1번 반복하면서 차례대로 마지막부터 큰 값을 채움 : N-1번의 단계

```
for (i = 0; i < NO - 1; i++)
```

- i번째 단계에서 별로 가장 큰 것을 탐색하여
마지막과 교환

- 단계에서 가장 큰 값 ■의 위치 탐색 : maxidx

```
for(j = 1; j < NO - i; j++)
```

- 가장 큰 값과 마지막 데이터 교환 : ■ 앞의 값과 교환
data[maxidx] ↔ data[NO - i - 1]

22	37	15	19	12	i == 0
22	12	15	19	37	i == 1
19	12	15	22	37	i == 2
15	12	19	22	37	
12	15	19	22	37	

정렬 알고리즘 : 선택 정렬

- 프로그램 코드

```
#include <stdio.h>
#define NO 5

void print_data(int data[], int no);

int main()
{
    int i, j, tmp;
    int data[NO] = {22, 37, 15, 19, 12};
    int maxidx;

    // 데이터의 개수 - 1번 반복
    for (i = 0; i < NO - 1; i++) {
        // 0 ~ NO - 1 까지의 배열 요소 중에서 최대값의 위치 찾기
        // 최대값 data[maxidx]와 마지막 값 data[NO - 1 - i]의 값을 교환
        print_data(data, NO); // 반복할때마다 데이터 출력
    }
}
```

최대값 위치 찾는
프로그램 추가

두 값을 변경하는
프로그램 추가

정렬 알고리즘 : 선택 정렬

- 프로그램 코드

```
void print_data(int data[], int no)
{
    int i;
    for(i = 0; i < no; i++)
        printf("%d ", data[i]);
    printf("\n");
}
```

- 실행결과

```
22 12 15 19 37
19 12 15 22 37
15 12 19 22 37
12 15 19 22 37
```

정렬 알고리즘 : 선택 정렬

- 앞의 프로그램을 완성하기
 - 빈 부분의 내용을 추가하여 프로그램을 완성하기
- 큰 숫자부터 나오도록 (내림차순) 정렬하는 프로그램을 작성하시오.
 - 변수 이름 변경 : 최대값의 위치 (maxidx) → 최소값의 위치 (minidx)
 - 최대값의 변경을 위한 비교문을 반대로 수정

정렬 알고리즘 : 선택 정렬

- 선택 정렬 : 문자열 정렬

- 문자열의 배열을 정렬하는 경우 : 문자열 비교, 문자열 복사함수를 대신 사용

- 정수형 비교의 예)

```
int i, j;
```

```
i < j;
```

- 문자열 비교 : strcmp()함수 사용

```
char sa[] = "이순신";
```

```
char sb[] = "김유신";
```

```
strcmp(sa, sb) < 0; → sa문자열이 sb 문자열보다 작음 (사전에서 앞에 오는 경우)
```

- 문자열 복사

```
strcpy(sa, sb); // sb문자열을 sa문자열 변수에 대입
```

정렬 알고리즘 : 선택 정렬

- 실습 #1

- 다음의 문자열들을 이름순으로 정렬하는 프로그램을 작성하시오.

```
char strs[][10] = { "이순신", "강감찬", "홍길동", "신윤복", "김유신"};
```

- 실습 #2

- 다음의 구조체의 데이터를 정렬하는 프로그램을 작성하시오.

```
struct student  
{  
    int number;  
    char name[20];  
    double height;  
};
```

- 초기값은 다음의 초기화 문을 사용하여, 키가 큰 순서대로 정렬한다.

```
struct student list[3] = {  
    { 1, "이순신", 172.8 },  
    { 2, "강감찬", 169.9 },  
    { 3, "홍길동", 174.2 },  
    { 4, "신윤복", 182.1 },  
    { 5, "김유신", 180.3 }  
};
```

정렬 알고리즘 : 선택 정렬

- 실습 #3

- 다음의 구조체의 데이터를 입력파일(input.txt)에서 읽어와서 키가 작은 순서대로 정렬한 후 출력파일(output.txt)로 저장하는 프로그램을 작성하시오.

```
struct student
{
    int number;
    char name[20];
    double height;
};
```

- 입력 파일은 다음과 같이 작성한다.

1	이순신	172.8
2	강감찬	169.9
3	홍길동	174.2
4	신윤복	182.1
5	김유신	180.3