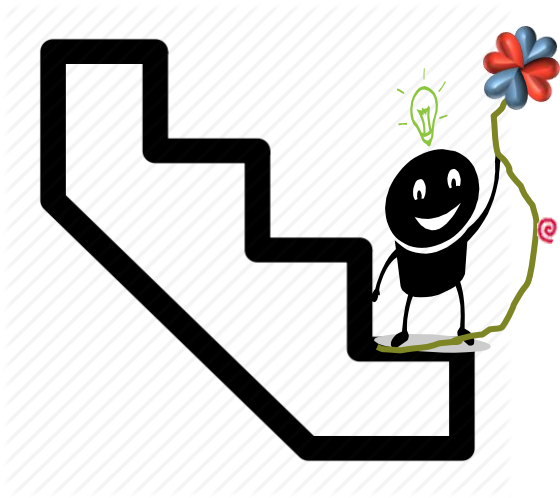


# 제 1과목 - 알고리즘

## <part2. 알고리즘 순서도작성>

### 14. 자료처리 알고리즘 - 삽입정렬

오늘의 핵심내용



데이터 정렬 알고리즘 중 삽입정렬 알고리즘 학습

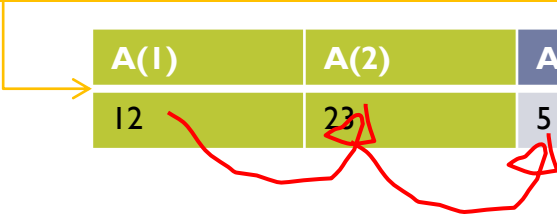


삽입정렬 – 두 번째 자료부터 시작하여 그 앞의 자료를 비교하여 순서대로 정렬하는 방식

예) 입력된 12,23,5,43,7을 차례대로 오름차순 정렬하는 과정


A(1)	A(2)	A(3)	A(4)	A(5)
12	23	5	43	7

1. 두 번째 자료 23을 첫번째 자료 12와 비교→첫번째 값 12가 작으므로 변화없음



A(1)	A(2)	A(3)	A(4)	A(5)
12	23	5	43	7

2. 세번째 값5를 두번째, 첫 번째 값과 비교하여 알맞은 곳에 삽입하고 나머지는 뒤로 이동



A(1)	A(2)	A(3)	A(4)	A(5)
5	12	23	43	7

3. 네번째 값 43을 세번째,두번째,첫번째 값과 비교하여 자리배치 =>네번째 값이 크므로 자리변동 없음

A(1)	A(2)	A(3)	A(4)	A(5)
5	12	23	43	7

4.다섯 번째 값 7을 네 번째, 세번째, 두 번째, 첫번째 순으로 값을 비교

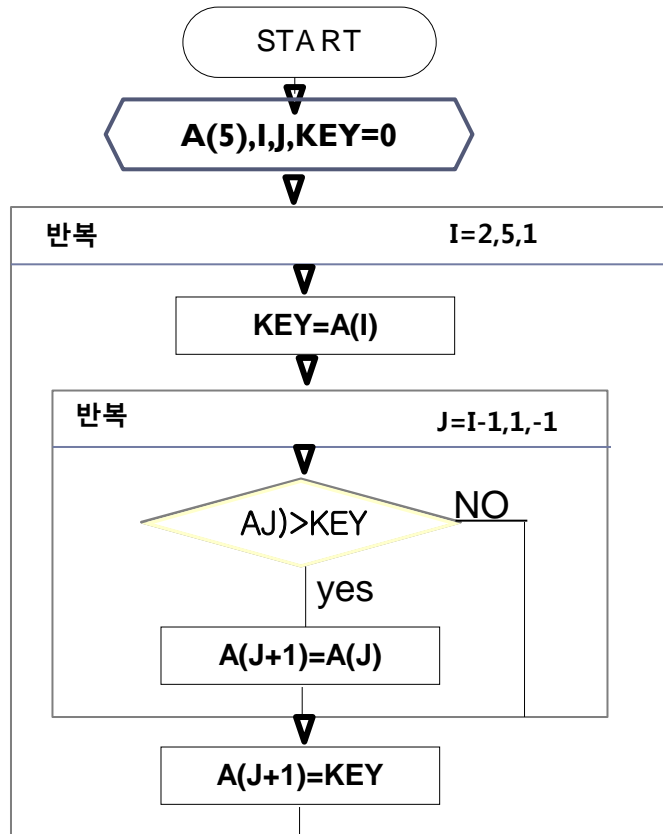
A(1)	A(2)	A(3)	A(4)	A(5)
5	12	23	<del>43</del> 7	<del>7</del> 43

A(1)	A(2)	A(3)	A(4)	A(5)
5	12	<del>23</del> 7	<del>43</del> 23	43

A(1)	A(2)	A(3)	A(4)	A(5)
5	<del>12</del> 7	<del>7</del> 12	23	43

A(1)	A(2)	A(3)	A(4)	A(5)
5	7	12	23	43



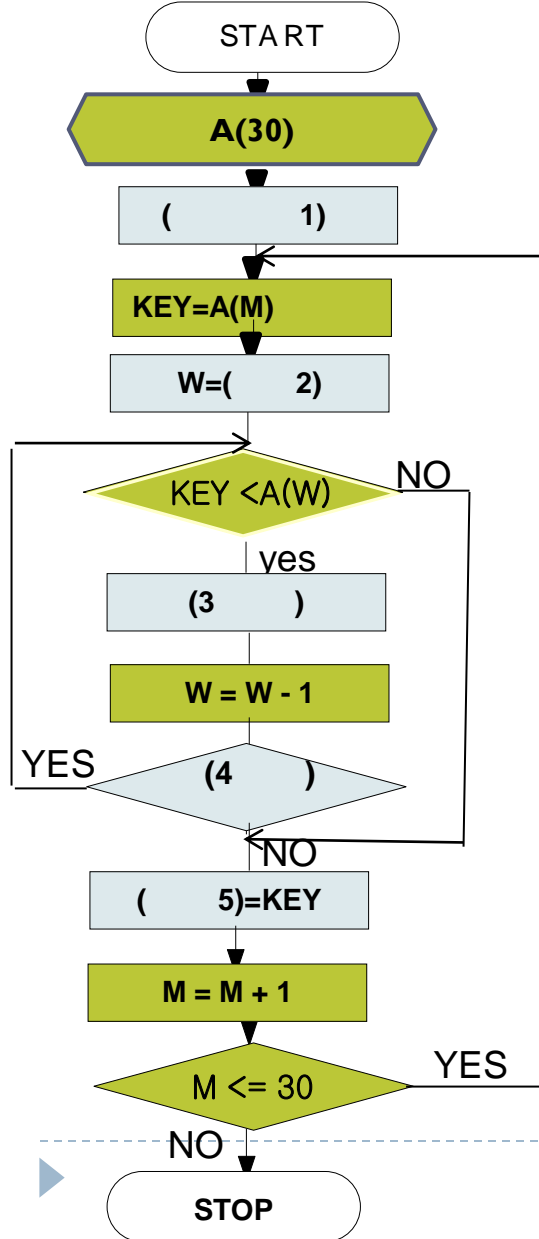


```
#include<stdio.h>
void main()
{
    int a[5], i, j, key=0;
    for (i = 1; i <= 5; i++)
        scanf_s("%d", &a[i]);
    for (i = 2; i <= 5; i++)
    {
        key = a[i];
        for (j = i - 1; j >= 1; j--) {
            if (a[j] > key) a[j + 1] = a[j];
            else break;
        }
        a[j + 1] = key;
    }

    for (i = 1; i <= 5; i++)
        printf("%d\t", a[i]);
}
```

The screenshot shows a Windows command prompt window titled 'C:\Windows\system32\cmd.exe'. The program has been executed, and the output displays the sorted array: 5, 12, 17, 33. The prompt is currently showing '계속하려면 아무 키나 누르십시오 . . .', which means 'Press any key to continue...'.

1. 100보다 작은 30개의 자연수가 배열 A(30)에 정렬되지 않은 상태로 저장되어 있을 때, 이 데이터를 오름 차순으로 정렬하는 삽입 정렬 알고리즘



- (1) 답 -
- (2) 답 -
- (3) 답 -
- (4) 답 -
- (5) 답 -

답 : \_\_\_\_\_

(1)	← M = 2
(2)	← M - 1
(3)	← A(W+1) = A(W)
(4)	← W = 1
(5)	← A(W+1)