

버블정렬

기초 알고리즘 공부하기

23.08.21 박송이

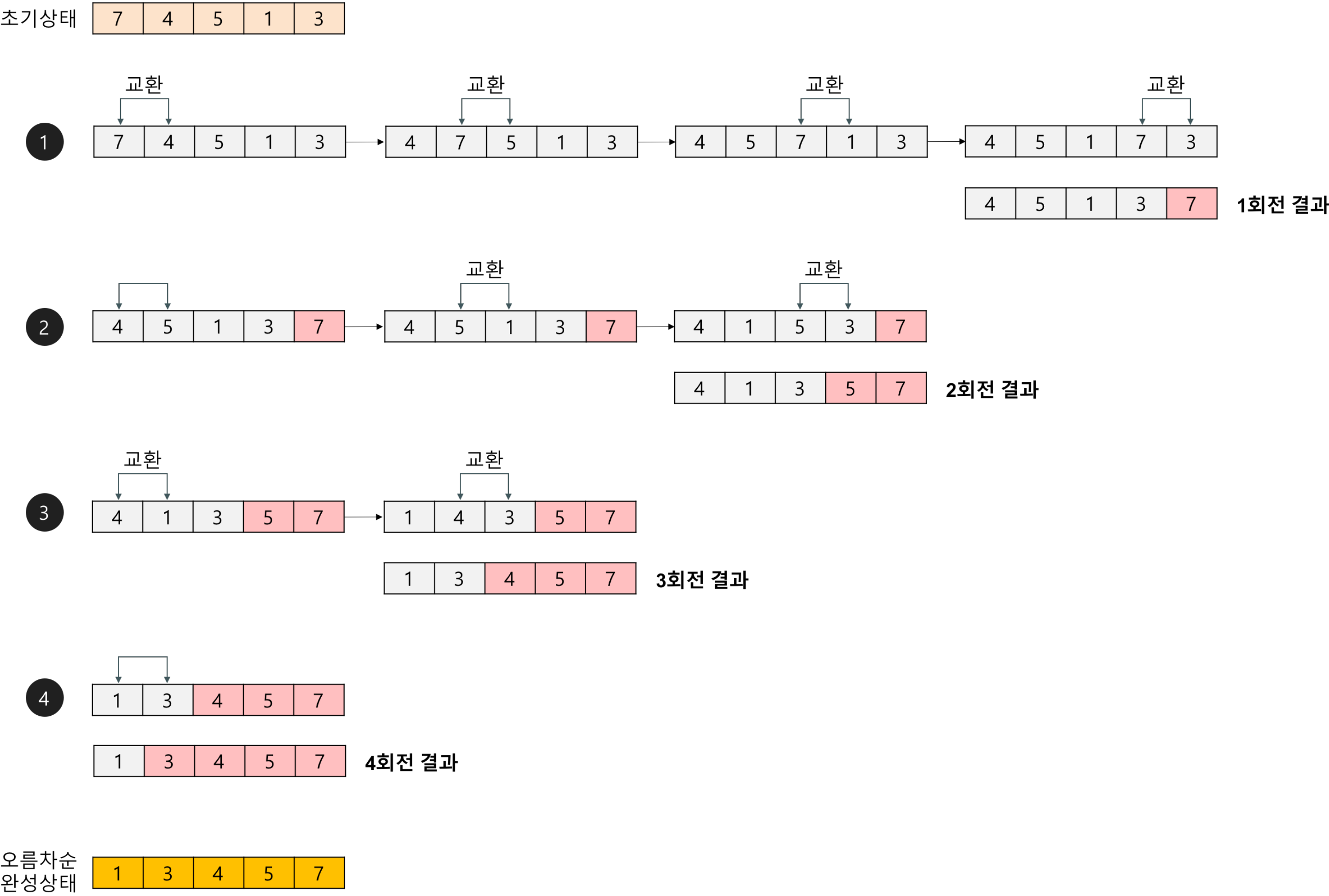
버블정렬이란?

데이터의 인접 요소끼리 비교하고 swap 연산을 수행하며 정렬하는 방식
핵심!

장점 : 구현하기 쉽다.

단점 : 시간복잡도 $O(n^2)$ 으로 다른 정렬 알고리즘보다 속도가 느리다.

버블정렬이란?



015 - 수 정렬하기 (1)

브론즈 1

- 수의 개수 ($1 \leq N \leq 1,000$)
- 두 번째 줄부터 N개의 줄에 숫자가 주어진다.
- 여기서는 N의 최대 범위가 1,000으로 매우 작아서 버블 정렬으로 풀어도 된다.

5 = N

N개의 정렬해야 할 수들

5

2

3

4

1

015 - 수 정렬하기 (1)

브론즈 1

for i in range(N-1) outer loop

for j
in
range(N-i-1)
inner loop

1번째 루프

5	2	3	4	1
2	5	3	4	1
2	3	5	4	1
2	3	4	5	1
2	3	4	1	5

N=5 i=0
N-i -1 = 4
j = 0 ~3

2번째 루프

2	3	4	1	5
2	3	4	1	5
2	3	4	1	5
2	3	1	4	5

N=5 i=1
N-i -1 = 3
j = 0 ~2

3번째 루프

2	3	1	4	5
2	3	1	4	5
2	1	3	4	5

N=5 i=2
N-i -1 = 2
j = 0 ~1

4번째 루프

2	1	3	4	5
1	2	3	4	5

N=5 i=3
N-i -1 = 1
j = 0

016 - 버블 정렬 프로그램 1

골드 2 - 문제에 나와있는 c++ 코드부터 이해하기

```
> C++ code.cpp
bool change = false;

for(int i=1; i<=n+1; i++){
    change = false;
    for(int j=1; j<=n-i; j++){
        if(a[j]>a[j+1]){
            change = true;
            swap(a[j],a[j+1])
        }
    }
    if(change==false){
        cout << i << '\n';
        break;
    }
}
```

outer loop

inner loop

뒤에 있는 숫자가 더 큰 숫자라면

bool change를 true로 바꾸고
두 값을 swap해라

inner loop가 끝난 후 만약
여전히 change가 false라면
outer loop의 i를 출력하고
끝내라.

change가 false라는 것은 조건이 성립하지 않아
한번도 swap이 일어나지 않음을 말한다.

몇번째 outer loop에서 한번도 swap이
일어나지 않았는지 찾는 문제.

016 - 버블 정렬 프로그램 1

골드 2 - 문제의 조건 이해하기

- 수의 개수 ($1 \leq N \leq 500,000$)
- 두 번째 줄부터 N개의 줄에 숫자가 주어진다. (각 수는 1,000,000보다 작거나 같은 자연 수 또는 0이다.)
- 버블정렬의 swap이 한번도 일어나지 않은 루프가 언제인지 알아내는 문제
- 다만 여기서는 N의 크기가 매우 커서 안쪽 for문이 몇번 수행되었는지 구하는 아이디어 필요

016 - 버블 정렬 프로그램 1

골드 2 - 안쪽 for문이 몇 번 수행됐는지 구하는 아이디어

1번째 루프

5	2	3	4	1
2	5	3	4	1
2	3	5	4	1
2	3	4	5	1
2	3	4	1	5

안쪽 루프는 왼쪽 -> 오른쪽으로 이동하면서 swap 진행

바로 다음 인덱스에 있는 값과 비교하기 때문에
swap을 해도 이동할 수 있는 거리는 최대 1이다.

즉 데이터의 정렬 전 인덱스와 정렬 후 인덱스를 비교해 왼쪽으로 가장 많이 이동한 값을 찾으면
문제를 해결 할 수 있다.

016 - 버블 정렬 프로그램 1

골드 2 - 손으로 풀어보기

원래 데이터

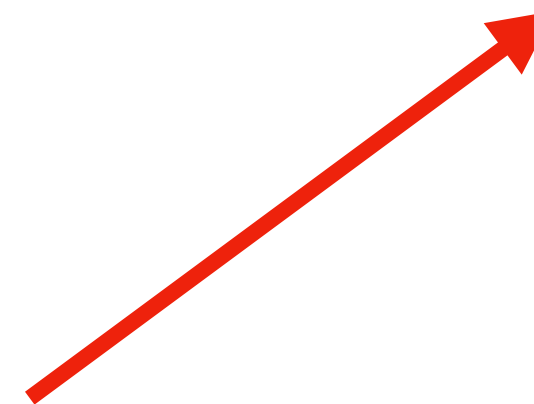
데이터	10	1	5	2	3
인덱스	0	1	2	3	4



data	1	2	3	5	10
정렬 전 index	1	3	4	2	0
정렬 후 index	0	1	2	3	4
결과값	1	2	2	-1	-4

정렬 후

데이터	1	2	3	5	10
인덱스	0	1	2	3	4



016 - 버블 정렬 프로그램 1

골드 2 - 손으로 풀어보기

원래 데이터

데이터	10	1	5	2	3
인덱스	0	1	2	3	4

```
for i in range(N):  
    A.append((int(input()),i))
```

[(10,0),(1,1),(5,2),(2,3),(3,4)]

위 과정을 통해 데이터와 원래의 인덱스를 함께 보존

016 - 버블 정렬 프로그램 1

골드 2 - 손으로 풀어보기

A

데이터	10	1	5	2	3
인덱스	0	1	2	3	4



sorted_A

데이터	1	2	3	5	10
인덱스	0	1	2	3	4

sorted_A = sorted(A)

sorted 함수를 통해 데이터가 오름차순으로 정렬된 새로운 배열 sorted_A를 만든다.

016 - 버블 정렬 프로그램 1

골드 2 - 손으로 풀어보기

data	1	2	3	5	10
정렬 전 index	1	3	4	2	0
정렬 후 index	0	1	2	3	4
결과값	1	2	2	-1	-4

정렬전 인덱스 - 정렬후 인덱스 연산
후 최대값 max 얻기

이렇게 얻어진 max값에 swap이 일어나지 않는 반복문이 한번 더
실행되는 것을 감안해 1을 더해주기