

## 2.4

## 버블 정렬

## 개요

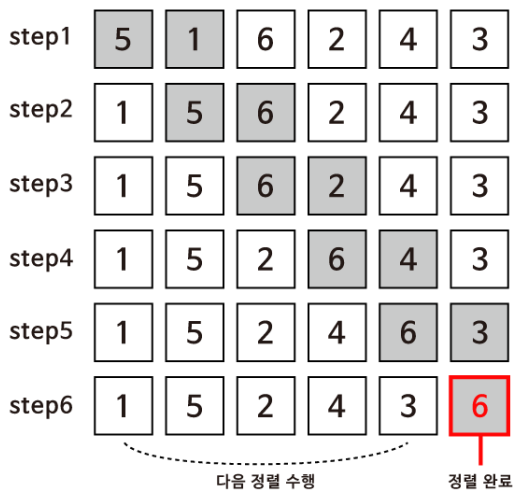
정렬되지 않은 리스트를 탐색하는 보다 정렬한 뒤 탐색하는 것이 더 효율적입니다. 정렬 알고리즘 중 하나는 **버블 정렬**입니다. 버블 정렬은 **두 개의 인접한 자료 값을 비교하면서 위치를 교환하는 방식으로 정렬하는 방법**을 말합니다. 버블 정렬은 단 두 개의 요소만 정렬해주는 좁은 범위의 정렬에 집중합니다. 이 접근법은 간단하지만 단 하나의 요소를 정렬하기 위해 너무 많이 교환하는 낭비가 발생할 수도 있습니다.

## 핵심개념

- \* 버블 정렬
- \* 배열

## 실행

```
for every element in the array
    check if element to the right is smaller
        if so swap the two elements
    else move on to the next element in the list
```



▲ &lt;코드 1&gt;

버블 정렬은 배열 안에 들어있는 **두 개의 인접한 수를 비교하고 만약 순서에 맞지 않는다면 교환해주는** 방식으로 작동합니다. <코드 1>을 보면 5, 1, 6, 2, 4, 3의 순서로 들어있는 배열이 있습니다. 버블 정렬을 사용하여 정렬해주고 싶다면 다음과 같은 의사 코드로 만들어볼 수 있습니다.

- ① 제일 먼저 배열 안에서 5와 1을 비교하기 시작할 것입니다. 1이 5보다 작기 때문에 두 수는 교환될 것입니다.
- ② 다음에는 5와 6을 비교하는데 올바른 순서로 되어있기 때문에 다음 요소로 넘어갈 것입니다.
- ③ 다음은 6과 2를 비교하고 계속 같은 방식으로 비교하여 교환합니다.

한번 정렬을 하면 1, 5, 2, 4, 3, 6의 순서로 정렬된 것을 확인할 수 있습니다. 완전히 정렬되지 않은 배열이지만 6은 이미 제 자리에 와 있습니다.  **$n$ 개의 원소에 대해서 버블 정렬을 한번 수행할 때마다  $n$ 번째의 원소가 제 자리를 찾게 됩니다.** 그렇기 때문에 다음 정렬에서는  $n-1$ 개의 원소를 정렬해 주면 됩니다. 이 예시의 경우 다음 정렬은 1, 5, 2, 4, 3 다섯 개의 숫자만 가지고 수행될 것입니다.

## 정렬된 배열

버블 정렬은 수행한 번 만에 모든 원소가 정렬되는 것을 보장하지 않습니다. 그러면 몇 번의 시도를 해줘야 할까요? 예를 들어 6, 5, 4, 3, 2, 1과 같이 거꾸로 정렬된 경우 다섯 번 시도해야 합니다. 즉  $n$ 개의 요소를 정렬해 주기 위해서는  $n-1$ 번 실행해주어야 합니다. 최악의 상황인 경우 최대한의 횟수를 실행해줘야 하므로 경제적이지 않습니다.