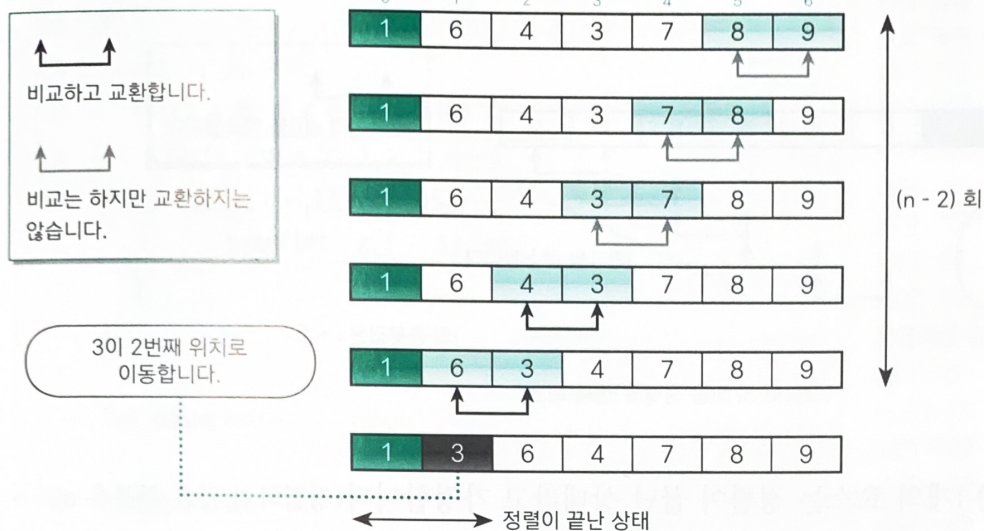


이어서 배열의 2번째 이후 요소에 대해 비교, 교환을 하는 패스(pass)를 수행합니다.



[그림 6-4] 버블 정렬의 두 번째 패스

이 패스를 수행하고 나면 3은 배열의 2번째 자리로 이동하고 그 결과 두 요소의 정렬이 끝납니다. 두 번째 패스의 비교 횟수는 첫 번째 패스보다 1회 적은 $n - 2$ 회입니다. 왜냐하면 패스를 수행할 때마다 정렬할 요소가 하나씩 줄어들기 때문입니다. 패스를 k 회 수행하면 앞쪽의 요소 k 개가 정렬된다는 것을 알 수 있습니다. 모든 정렬이 끝나려면 $n - 1$ 회의 패스가 수행되어야 합니다.

☞ 수행하는 패스의 횟수가 n 회가 아니라 $n - 1$ 회인 것은 $n - 1$ 개 요소의 정렬이 끝나면 마지막 요소는 이미 끝에 놓이기 때문입니다.

☞ '버블 정렬(bubble sort)'이라는 말은 액체 안의 공기 방울이(액체보다 가벼운 공기 방울이) 보글보글 위로 올라오는 모습에서 착안한 것입니다.

버블 정렬 프로그램

버블 정렬 알고리즘을 프로그램으로 구현해 보겠습니다. 변수 i 의 값을 0부터 $n - 2$ 까지 1씩 증가하며 $n - 1$ 회의 패스를 수행하는 프로그램은 아래와 같습니다.

```
for(i = 0; i < n - 1; i++) {
    //a[i], a[i + 1], ..., a[n - 1]에 대해
    //끝에서부터 앞으로 스캔하면서 이웃하는 두 요소를 비교하고 교환합니다.
}
```

여기서 비교하는 두 요소의 인덱스를 $j - 1, j$ 라 하고, 변수 j 의 값을 어떻게 변화하면 좋을지 그림 6-5를 통해 살펴보겠습니다. 배열의 끝(오른쪽)부터 스캔하기 때문에 j 의 시작값은 $n - 1$ 입니다.