

JavaScript 정렬 알고리즘

2) 버블 정렬

버블 정렬 | 알고리즘의 기본이 되는 정렬 알고리즘 이해하기

강사 나동빈

JavaScript

핵심 자료구조 알아보기

2) 버블 정렬

버블 정렬(Bubble Sort)

- 단순히 인접한 두 원소를 확인하여, 정렬이 안 되어 있다면 위치를 서로 변경한다.
- 서로 인접한 두 원소를 비교하는 형태가 **거품**과 같다고 하여 붙여진 이름이다.
- 시간 복잡도 $O(N^2)$ 로 비효율적인 정렬 알고리즘 중 하나다.

버블 정렬(Bubble Sort) 동작 방식

- 각 단계에서는 **인접한 두 개의 원소를 비교**하여, 필요시 위치를 변경한다.
- 첫째와 둘째를 비교, 둘째와 셋째를 비교, 셋째와 넷째를 비교하는 방식이다.

버블 정렬(Bubble Sort) 동작 방식

- 한 번의 단계가 수행되면, 가장 큰 원소가 맨 뒤로 이동한다.
- 따라서, 그 다음 단계에서는 맨 뒤로 이동한 데이터는 정렬에서 제외한다.

[핵심] 각 단계를 거칠 때마다 가장 큰 값을 하나씩 확실하게 결정하는 것으로 이해할 수 있다.

JavaScript 정렬 버블 정렬

버블 정렬(Bubble Sort) 예시

JavaScript 정렬 버블 정렬

- 정렬할 배열:

9	8	2	4	3
---	---	---	---	---

JavaScript 정렬 버블 정렬

버블 정렬(Bubble Sort) 예시

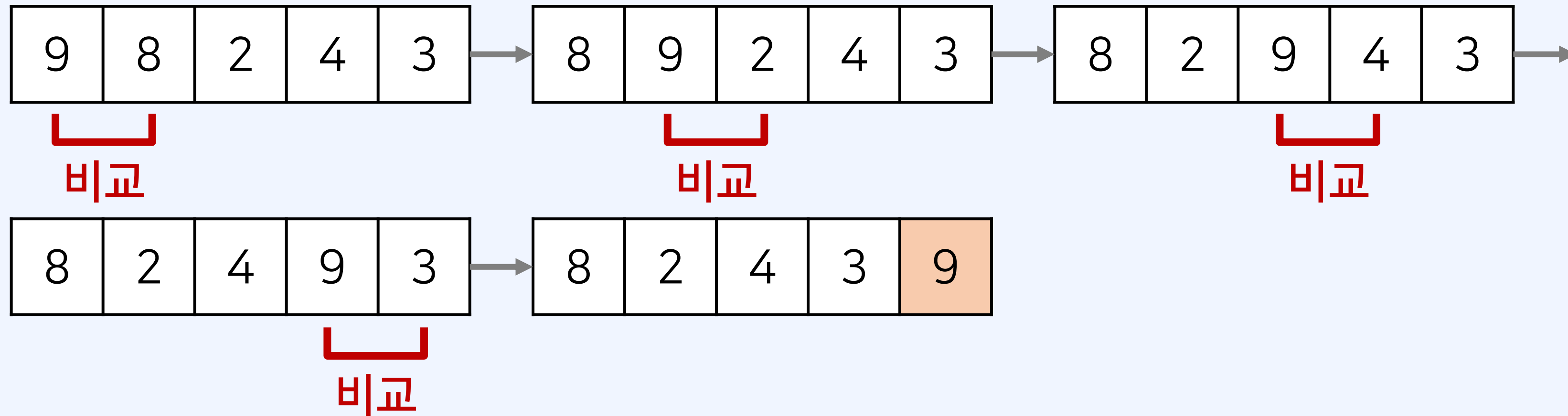
JavaScript 정렬 버블 정렬

- 정렬할 배열:

9	8	2	4	3
---	---	---	---	---

 : 정렬 완료

[1단계]



JavaScript 정렬 버블 정렬

버블 정렬(Bubble Sort) 예시

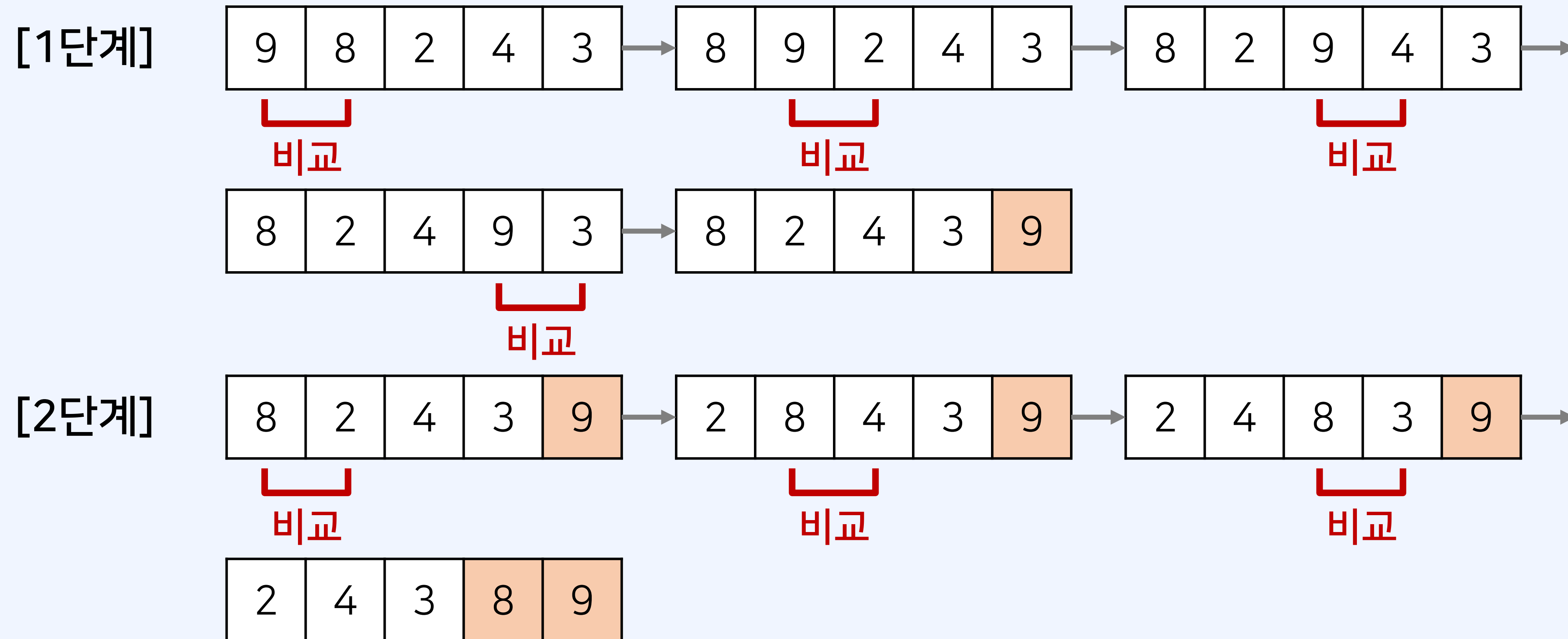
JavaScript 정렬 버블 정렬

- 정렬할 배열:

9	8	2	4	3
---	---	---	---	---

--

 : 정렬 완료



JavaScript 정렬 버블 정렬

버블 정렬(Bubble Sort) 예시

JavaScript 정렬 버블 정렬

- 정렬할 배열:

9	8	2	4	3
---	---	---	---	---

--

 : 정렬 완료



JavaScript 정렬 버블 정렬

버블 정렬(Bubble Sort) 예시

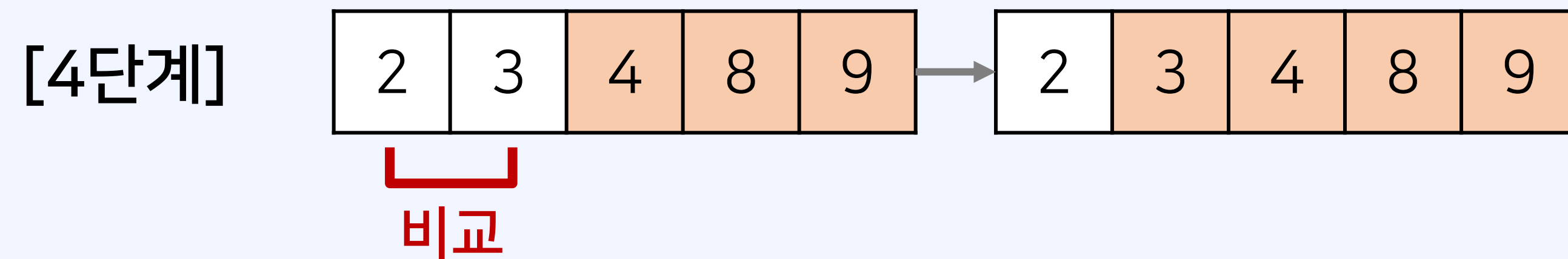
JavaScript 정렬 버블 정렬

- 정렬할 배열:

9	8	2	4	3
---	---	---	---	---

--

 : 정렬 완료



JavaScript 정렬 버블 정렬

버블 정렬(Bubble Sort) 예시

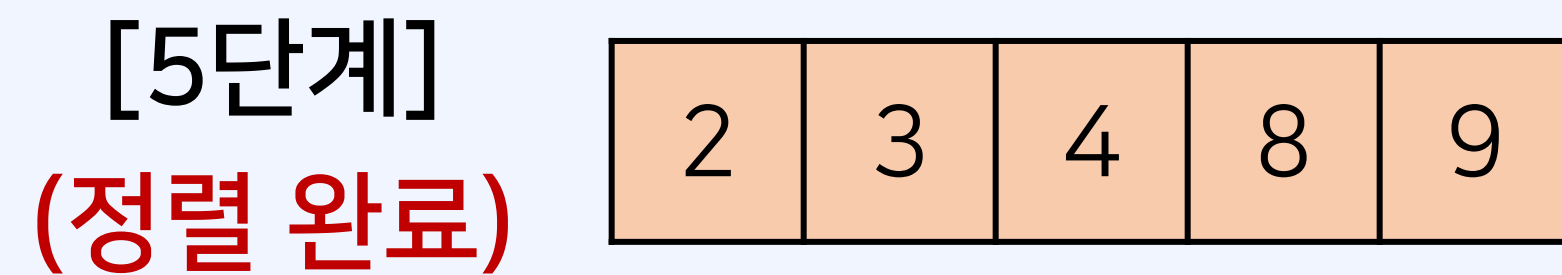
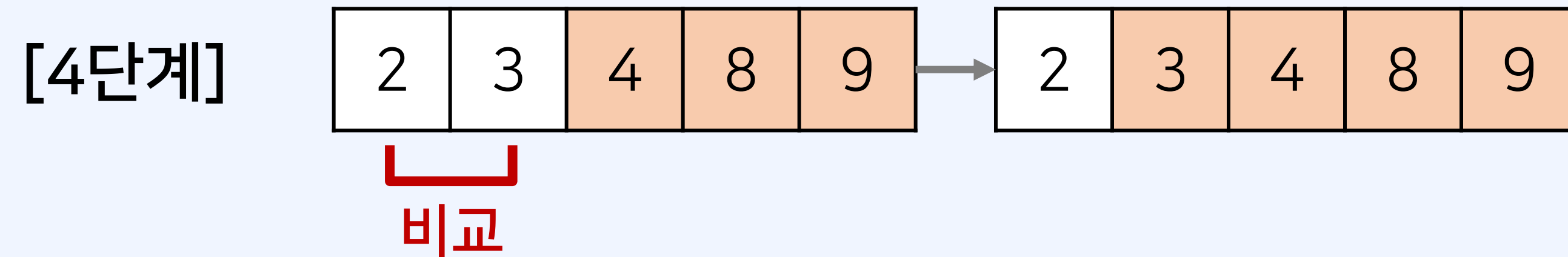
JavaScript 정렬 버블 정렬

- 정렬할 배열:

9	8	2	4	3
---	---	---	---	---

--

 : 정렬 완료



JavaScript 정렬 버블 정렬

버블 정렬(Bubble Sort) 소스 코드 예시

JavaScript 정렬 버블 정렬

```
// 버블 정렬 함수
function bubbleSort(arr) {
  for (let i = arr.length - 1; i > 0; i--) {
    for (let j = 0; j < i; j++) {
      if (arr[j] < arr[j + 1]) { // 내림차순 예시
        let temp = arr[j];
        arr[j] = arr[j + 1];
        arr[j + 1] = temp;
      }
    }
  }
}
```

JavaScript 정렬 버블 정렬

버블 정렬(Bubble Sort) 소스 코드 예시

JavaScript 정렬 버블 정렬

```
/* 1) 버블 정렬의 수행 시간 측정 */  
// 0부터 999까지의 정수 30000개를 담은 배열 생성  
let arr = Array.from({ length: 30000 }, () => Math.floor(Math.random() * 1000));
```

```
// getTime(): 1970-01-01부터의 시간차를 ms 단위로 계산  
startTime = new Date().getTime();  
bubbleSort(arr);  
endTime = new Date().getTime();
```

```
// 시간차 출력  
console.log("버블 정렬 소요 시간:", endTime - startTime, "ms.");
```

```
/* 2) 이미 정렬된 배열에 대한 버블 정렬의 수행 시간 측정 */  
// 값이 7인 정수 30000개를 담은 배열 생성  
arr = Array.from({ length: 30000 }, () => 7);
```

```
// getTime(): 1970-01-01부터의 시간차를 ms 단위로 계산  
startTime = new Date().getTime();  
bubbleSort(arr);  
endTime = new Date().getTime();
```

```
// 시간차 출력  
console.log("정렬된 배열에 대한 버블 정렬 소요 시간:", endTime - startTime, "ms.");
```

[실행 결과]

버블 정렬 소요 시간: 2912 ms.
정렬된 배열에 대한 버블 정렬 소요 시간: 2402 ms.

버블 정렬(Bubble Sort)의 시간 복잡도

- 최악의 경우 시간 복잡도 $O(N^2)$ 을 보장한다.
- 이미 정렬된 배열에 대해서 모든 비교가 필요하므로, 굉장히 비효율적인 정렬 알고리즘 중 하나에 속한다.