자료구조와 알고리즘

6강 – 기본 정렬 알고리즘

LECTURED BY SOONGU HONG

* 기본 정렬 알고리즘

*** 버블 정렬 (bubble sort)**

- 데이터의 인접 요소끼리 비교하고, swap연산을 수행하며 정렬하는 방식
- 시간 복잡도는 O(N^2)으로 다른 정렬보다 느린 편입니다.

*** 선택 정렬 (selection sort)**

- 대상에서 가장 크거나 작은 데이터를 찾아가 선택을 반복하면서 정렬
- 구현이 복잡하고 시간 복잡도도 버블 정렬과 같아 잘 사용하지 않습니다.

*** 삽입 정렬 (insertion sort)**

- 대상을 선택해 정렬된 영역에서 선택 데이터의 적절한 위치를 찾아 삽입하면서 정렬하는 방식
- 구현이 쉬우나 시간 복잡도는 버블 정렬과 같습니다.

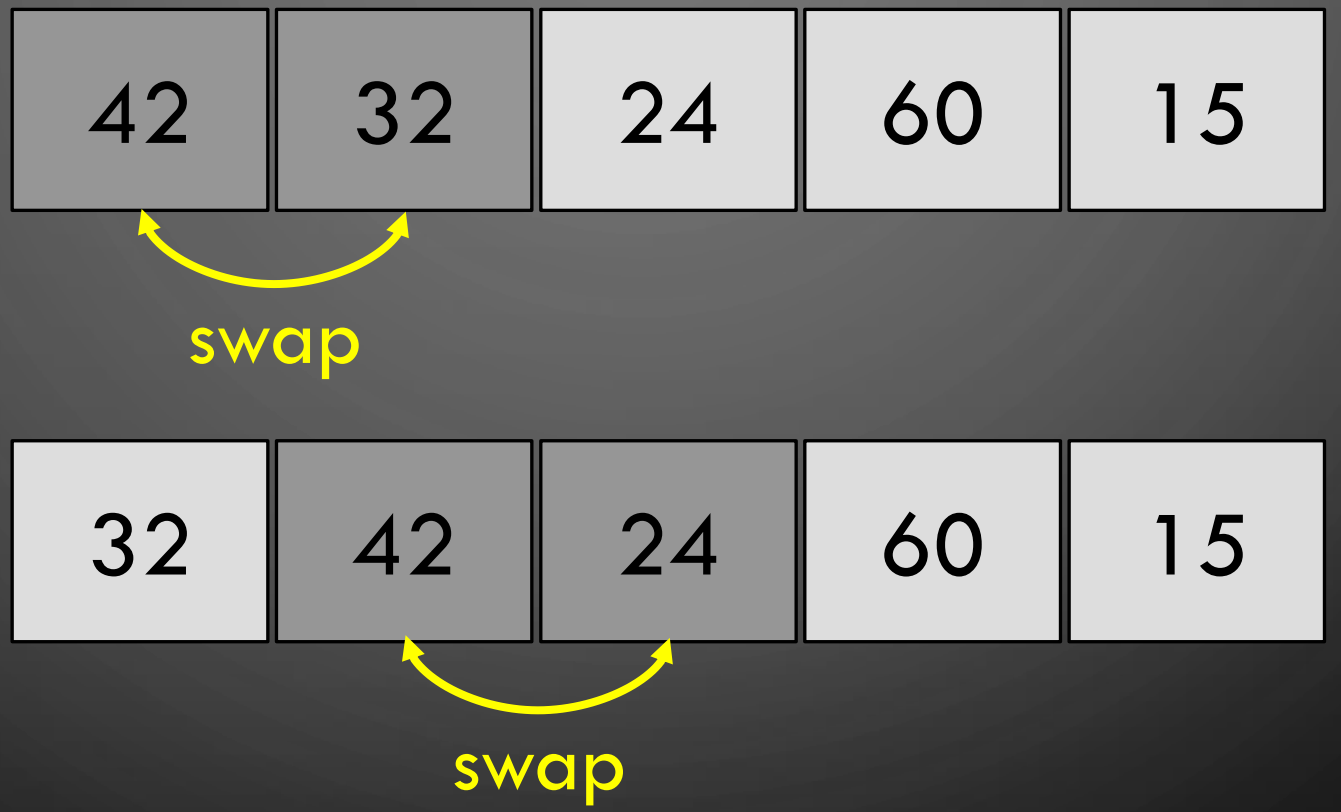
1. 버블 정렬

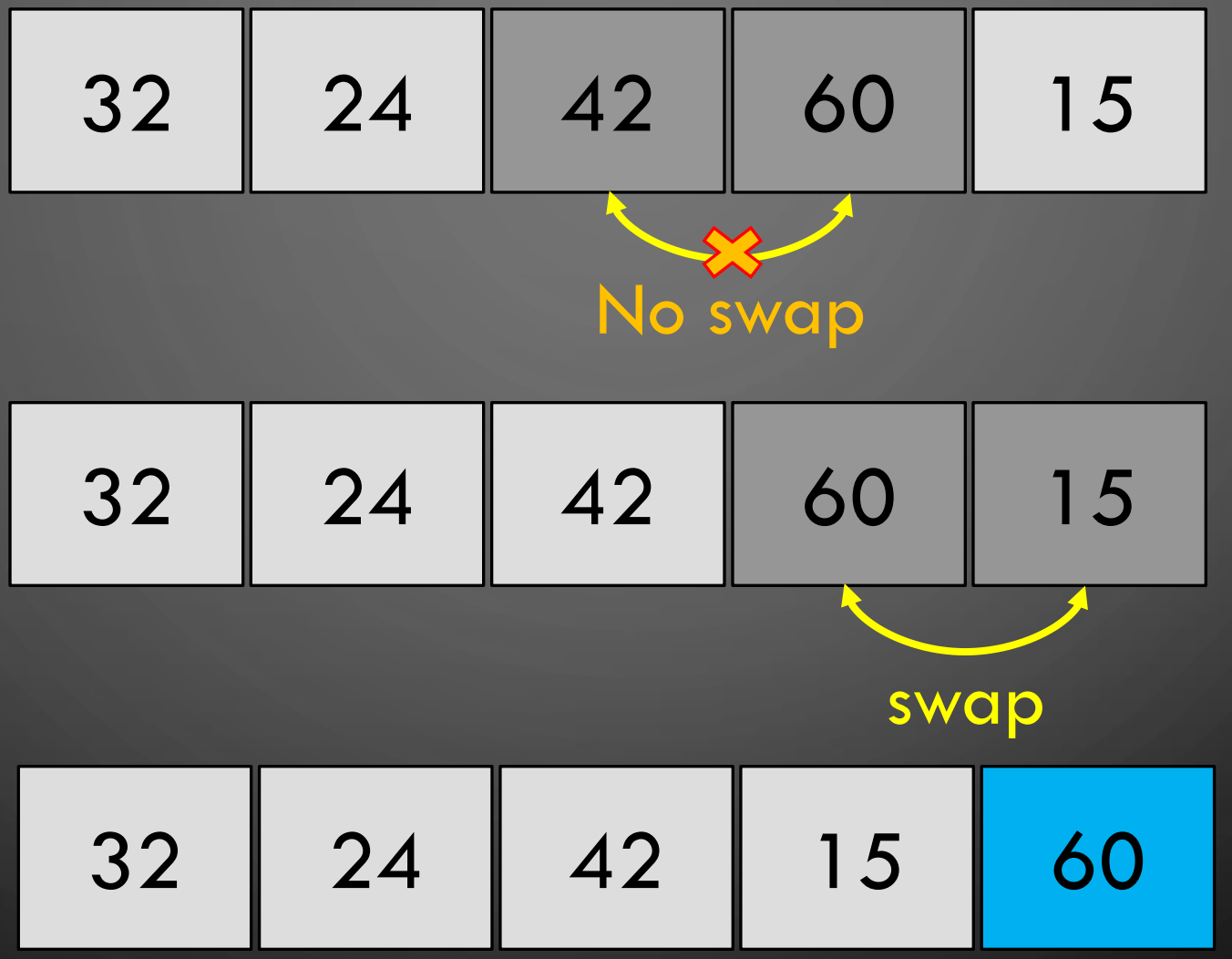
* 버블 정렬 과정

- 버블 정렬은 가장 기본적인 정렬방법으로서 서로 인접한 자료들을 서로 자리바꿈 하면서 뒤에서부터 정렬되는 방식입니다.

1. 비교 연산이 필요한 루프 범위를 설정한다.
2. 인접한 데이터 값을 비교한다.
3. Swap조건에 부합하면 swap연산을 수행한다.
4. 루프 범위가 끝날 때까지 2~3을 반복한다.
5. 정렬 영역을 설정합니다. 다음 루프를 실행할 때는 이 영역을 제외한다.
6. 비교 대상이 없을 때까지 1 ~ 5를 반복한다.

< 1번째 루프 >





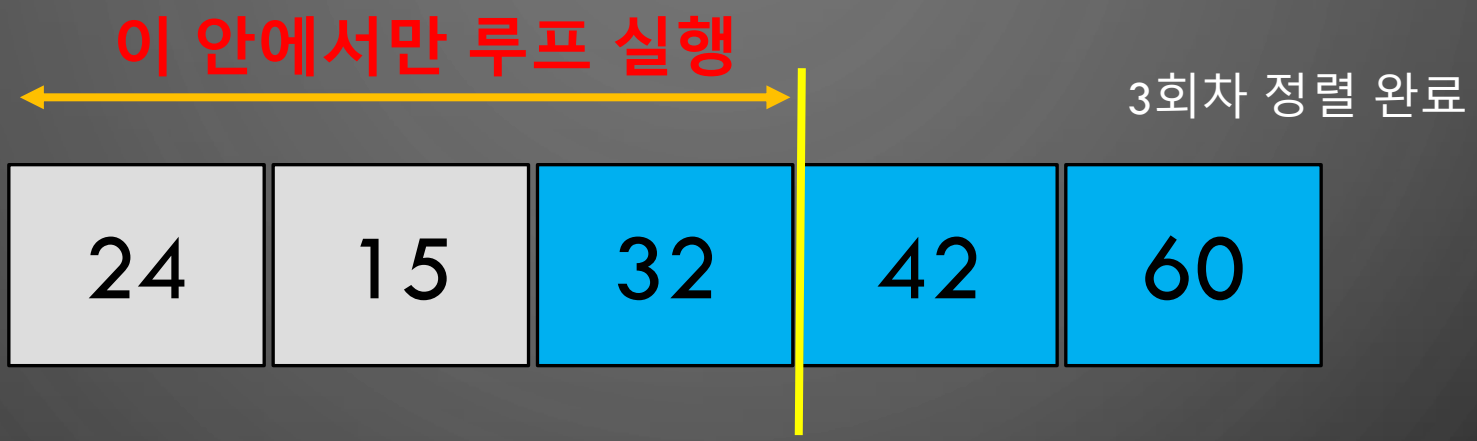
1회차 정렬 완료

< 2번째 루프 >

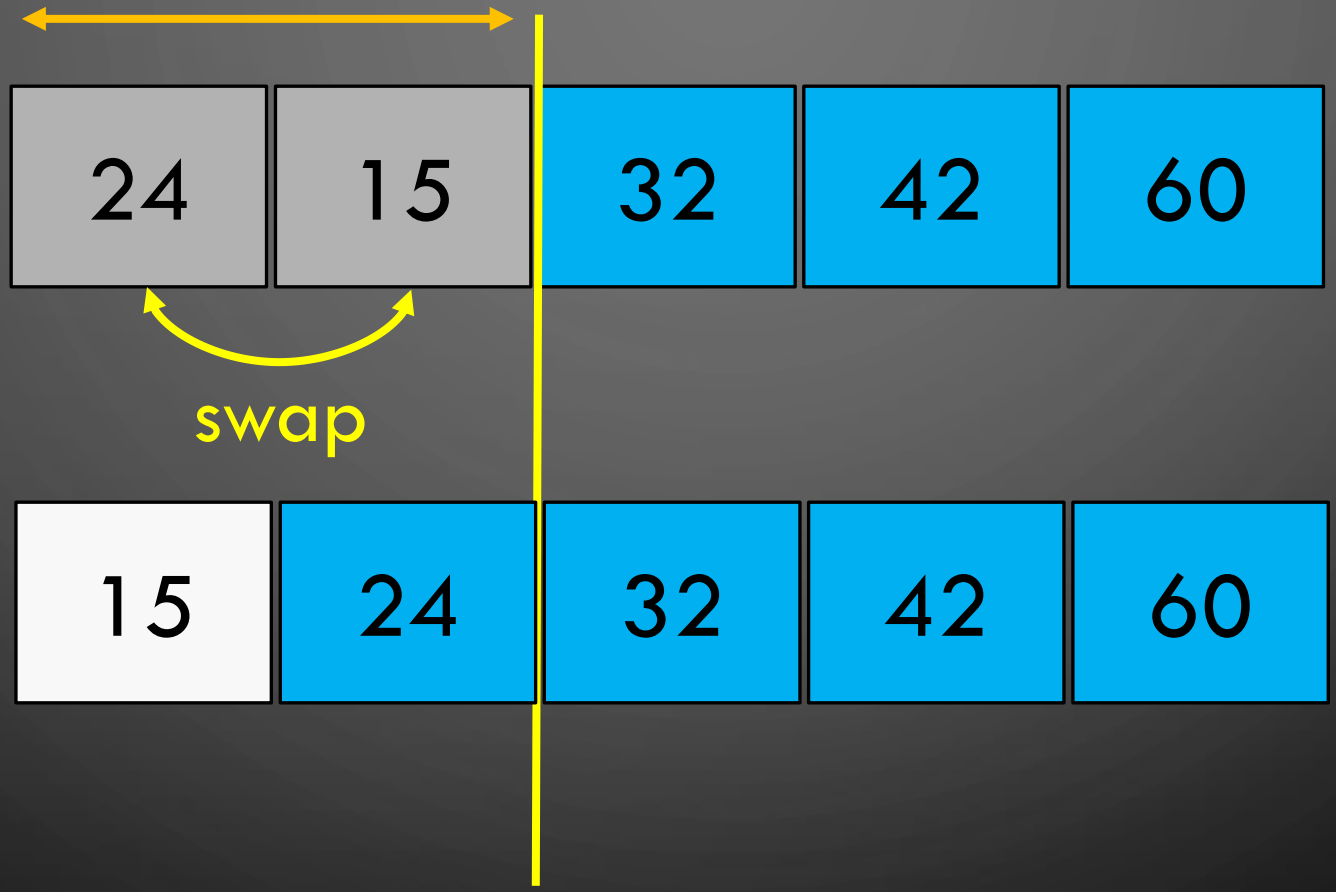

이 안에서만 루프 실행
24
32
42
15
60
swap
24
32
15
42
60
2회차 정렬 완료

< 3번째 루프 >

< 3번째 루프 >



< 4번째 루프 >



4회차 정렬 완료

모든 루프 종료

2. 선택 정렬

* 선택 정렬 과정

- 선택 정렬은 버블 정렬의 자리바꿈 횟수를 줄임으로써 성능을 개선한 알고리즘입니다.
- 그러나 선택 정렬의 비교횟수는 버블 정렬과 동일하여 성능 개선효과가 크지 않습니다.

1. 남은 정렬 부분에서 최소값을 찾는다.
2. 남은 정렬 부분에서 가장 앞에 있는 데이터와 최소값을 swap한다.
3. 가장 앞에 있는 데이터를 다음 위치로 이동하여 범위를 축소한다.
4. 남은 정렬 부분이 없을 때까지 반복한다.

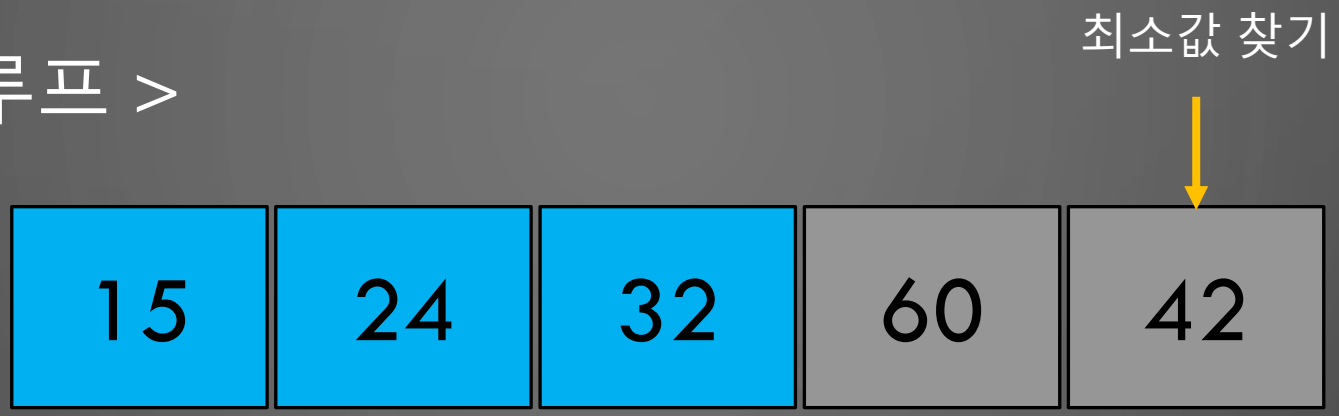
< 1번째 루프 >

< 2번째 루프 >

< 3번째 루프 >

세번째 데이터와 최소값이 같으므로 넘어감

< 4번째 루프 >



4번째 데이터와 최소값 교환

정렬 완료

3. 삽입 정렬

* 삽입 정렬 과정

- 삽입 정렬은 기본 정렬 중 가장 큰 효율성을 가집니다.
- 일반적으로 버블 정렬보다 거의 2배 빠르며 선택 정렬보다도 약간 빠릅니다.
- 일반적으로 퀵 정렬에서는 마지막 단계로 삽입 정렬을 쓰는데 그 이유는
  삽입 정렬이 이미 어느 정도 정렬된 배열에서 아주 큰 효과를 내기 때문입니다.

1. 현재 index에 있는 데이터를 선택한다.
2. 현재 선택한 데이터가 정렬 범위 내에 삽입될 위치를 탐색한다.
3. 삽입 위치부터 index위치까지 shift연산을 수행한다.
4. 삽입 위치에 현재 데이터를 삽입하고 index를 증가시킨다.
5. 선택할 데이터가 없을 때까지 반복한다.

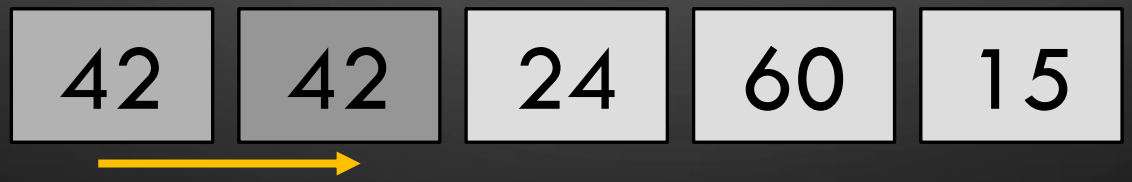
< 1번째 루프 >

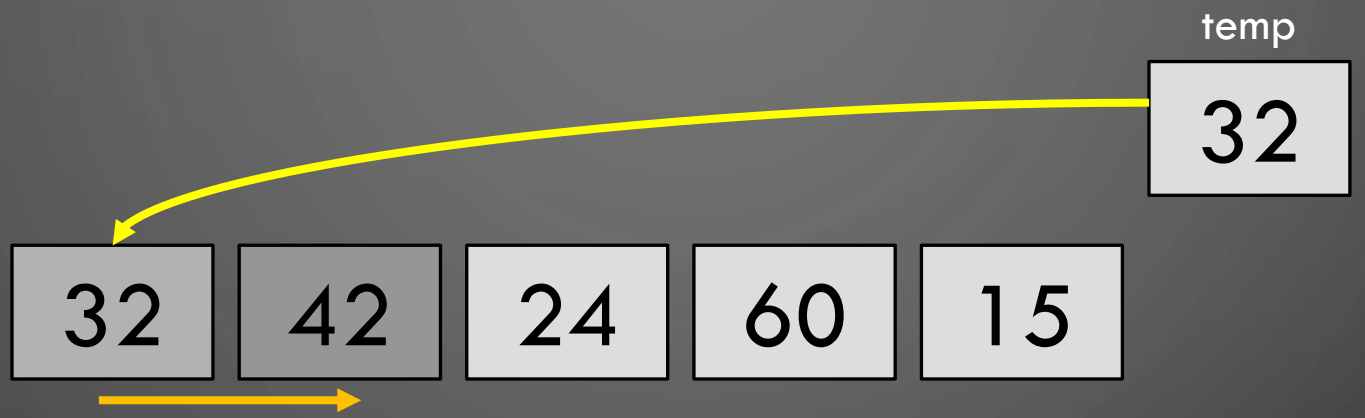
temp

32

타겟 데이터는 2번째 위치 데이터에서 시작하며 타겟 데이터를 백업 후
타겟 기준으로 왼쪽에 있는 값들을 비교하면서 타겟보다
크다면 왼쪽 데이터를 한 칸씩 오른쪽으로 이동(shift)한다.

비교 – 왼쪽이 크므로 왼쪽값 이동



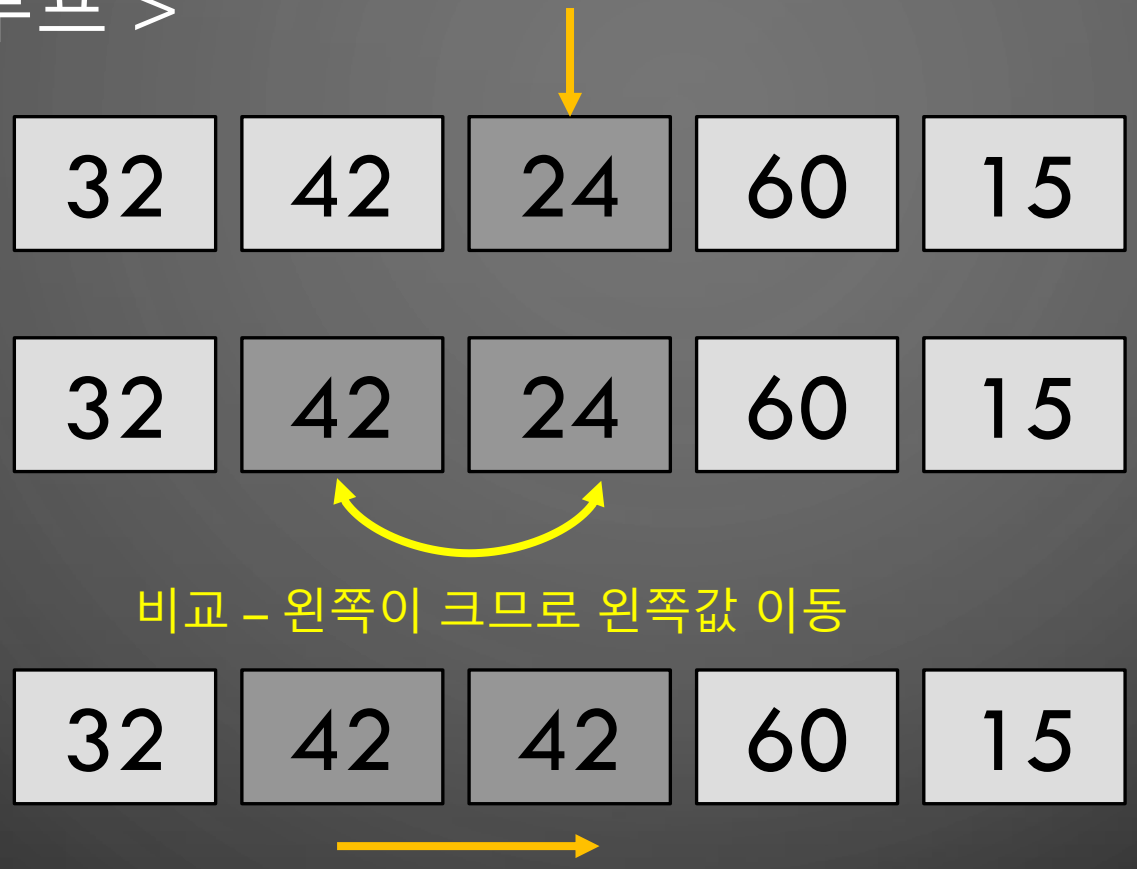


계속 왼쪽을 확인하며 타겟(temp)보다 크면 지속적으로 우측이동을 하는데 크지 않으면 중단하고 그 위치에 타겟을 삽입한다.

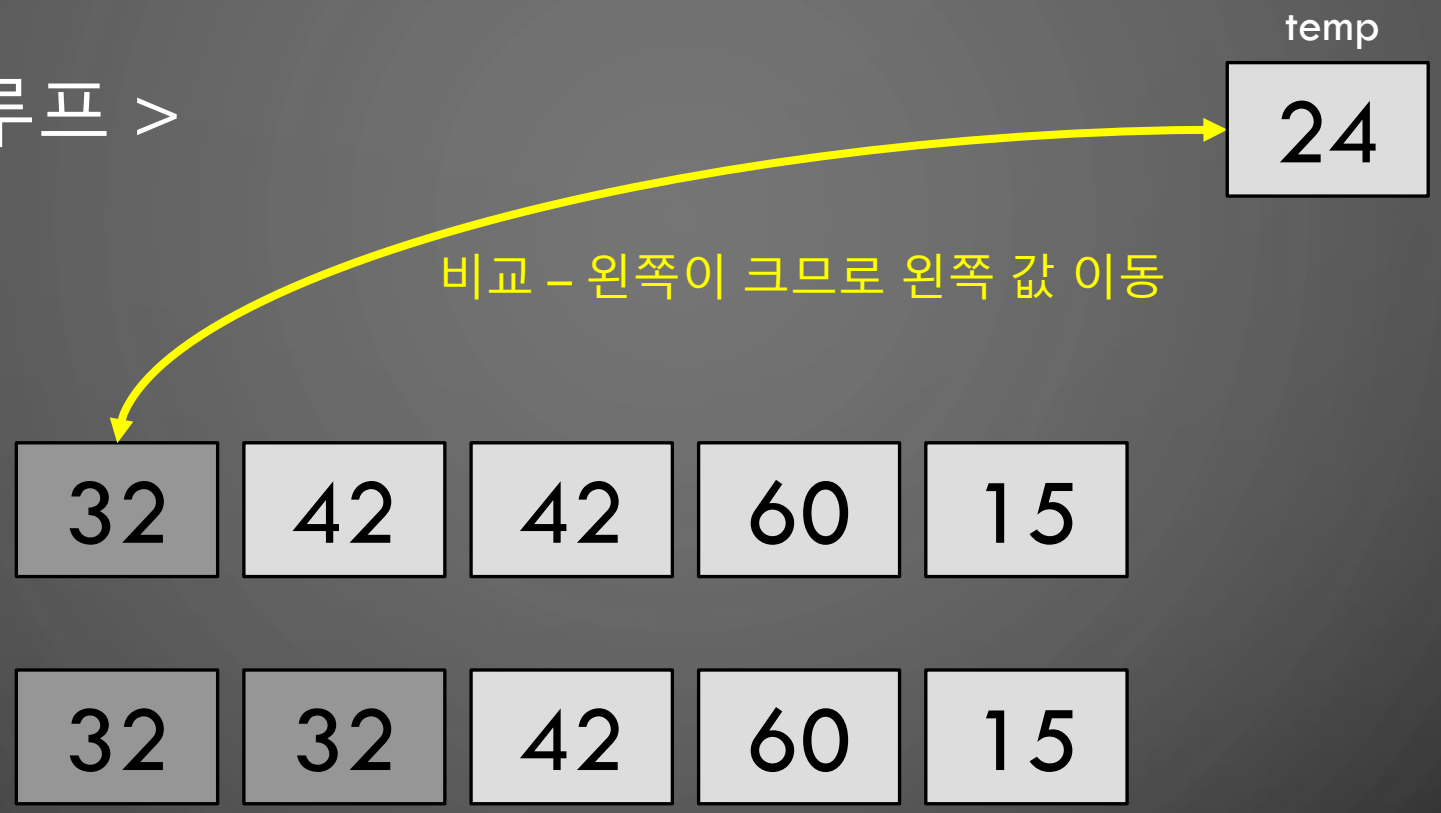
< 2번째 루프 >

temp

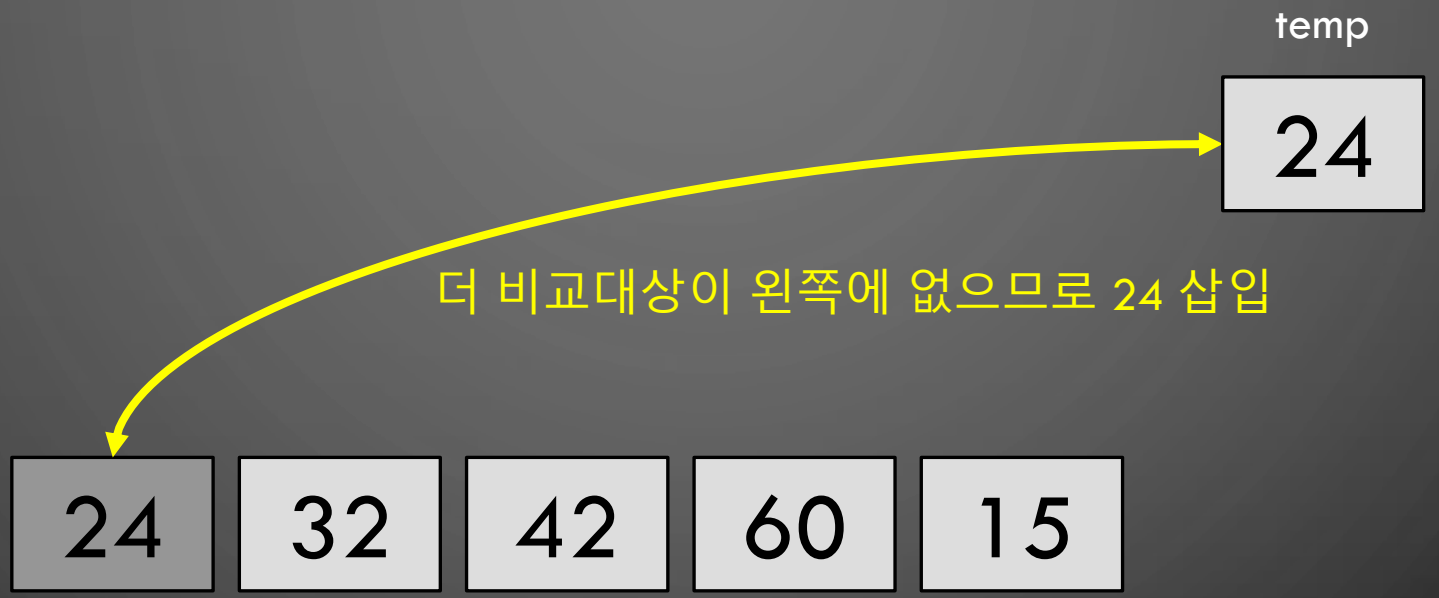
24



< 2번째 루프 >



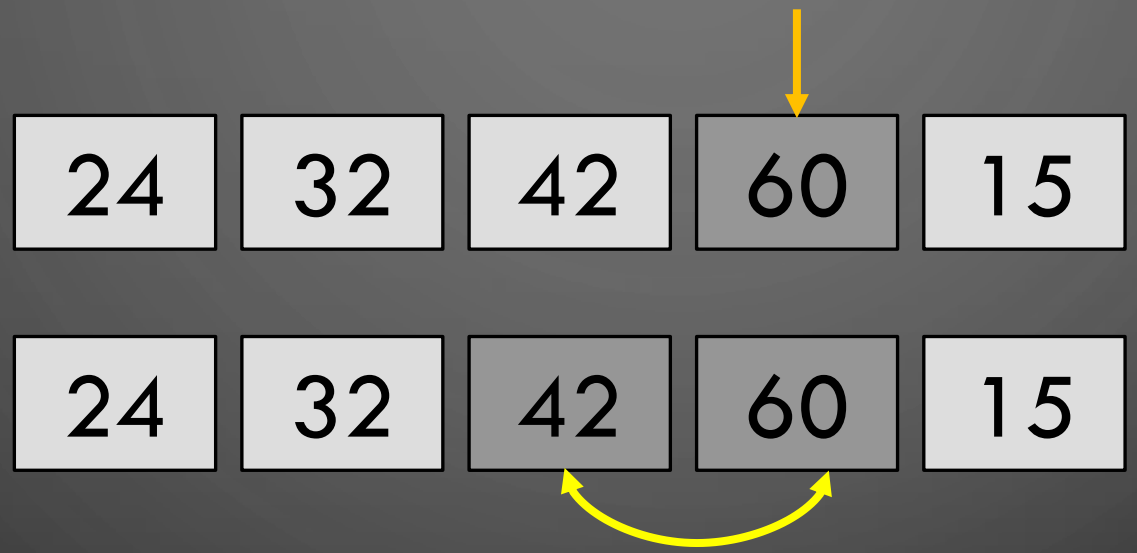
< 2번째 루프 >



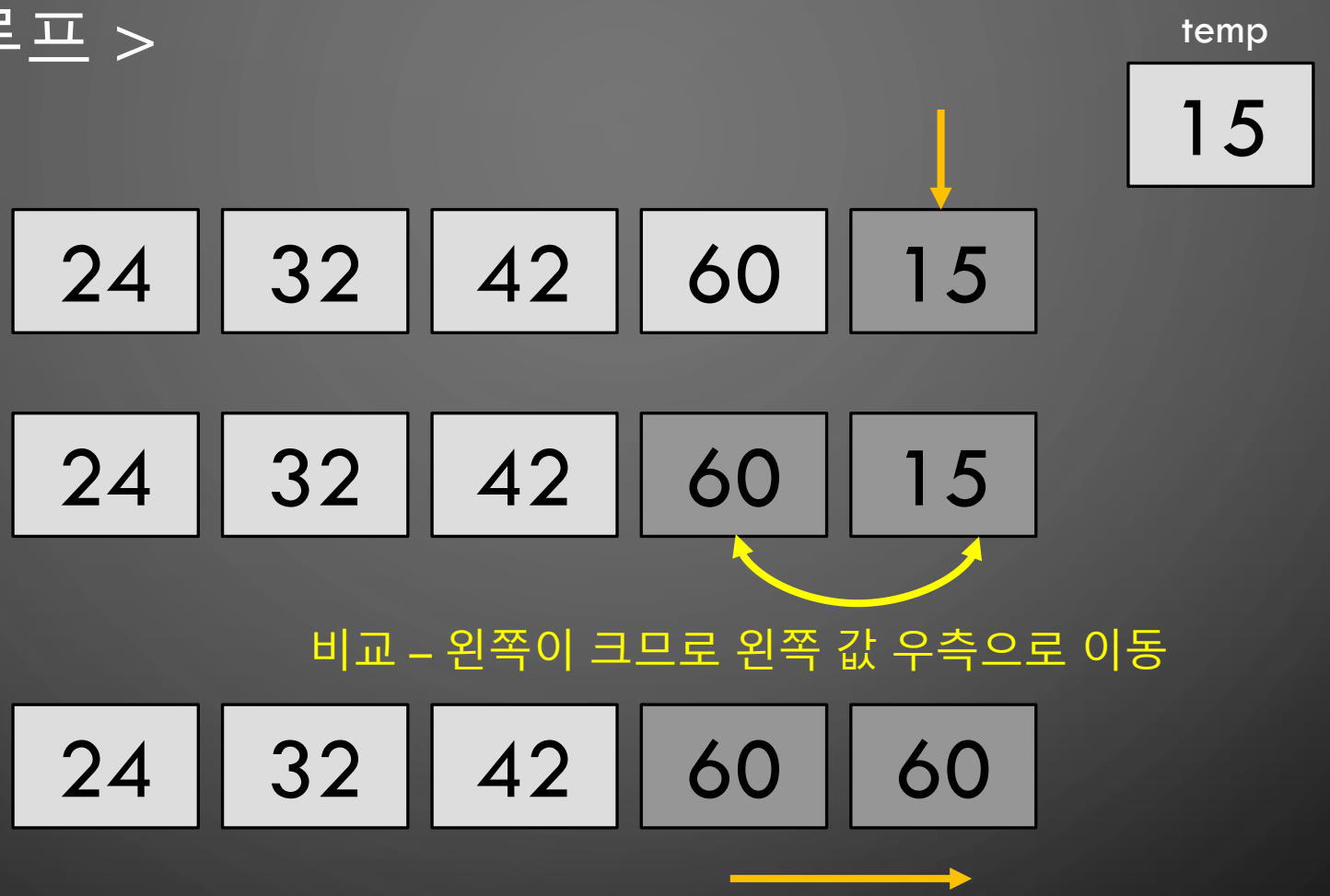
< 3번째 루프 >

temp

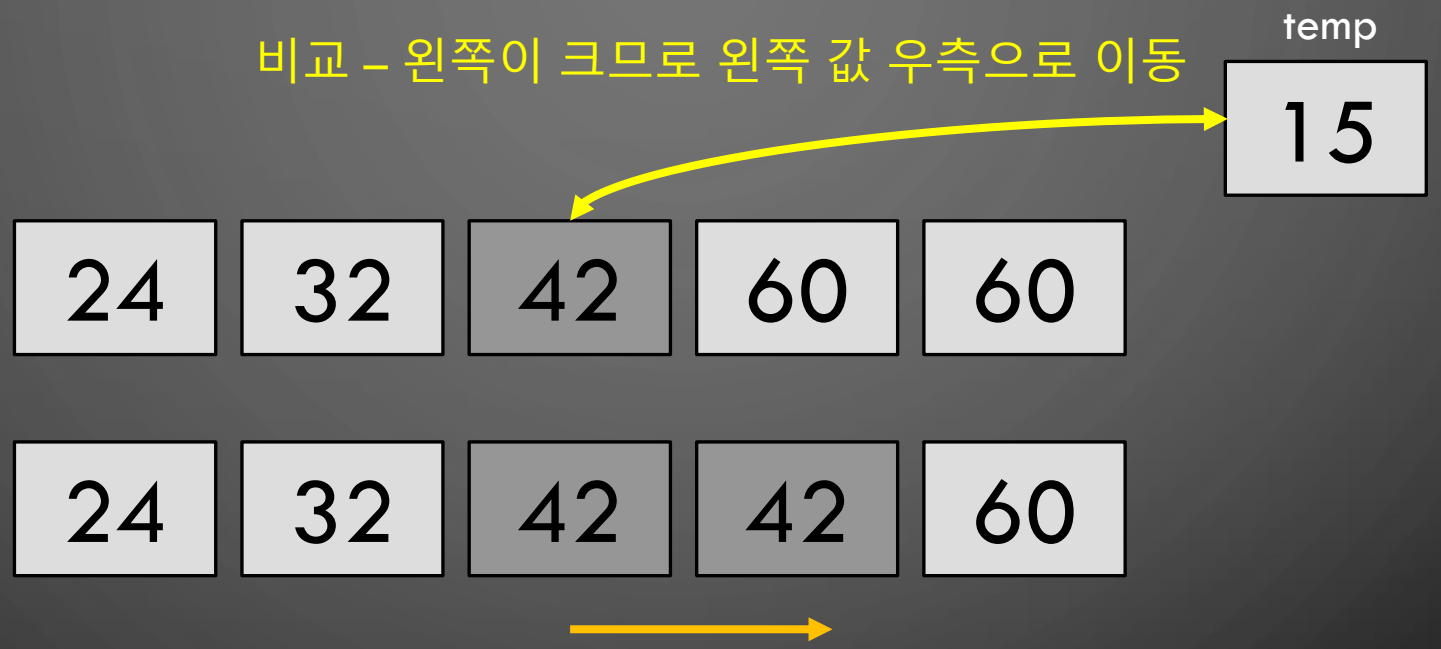
60



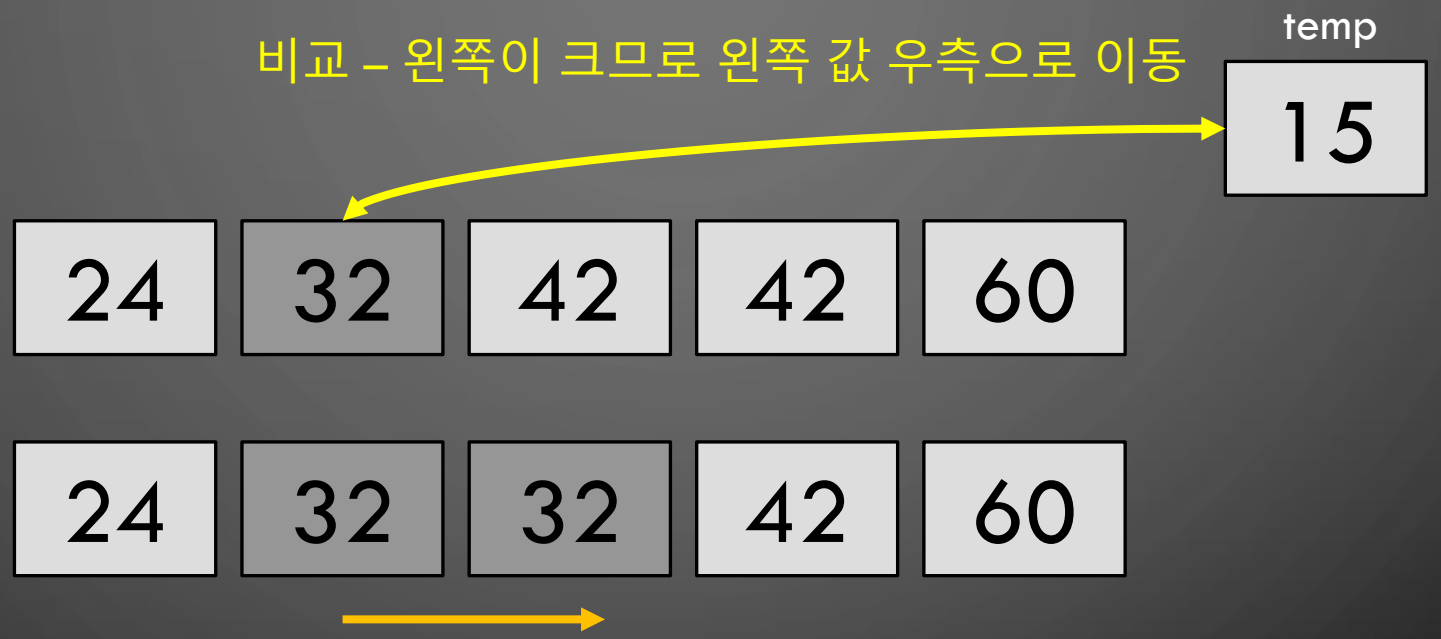
< 4번째 루프 >


temp
15
24 32 42 60 15
24 32 42 60 15
비교 – 왼쪽이 크므로 왼쪽 값 우측으로 이동
24 32 42 60 60

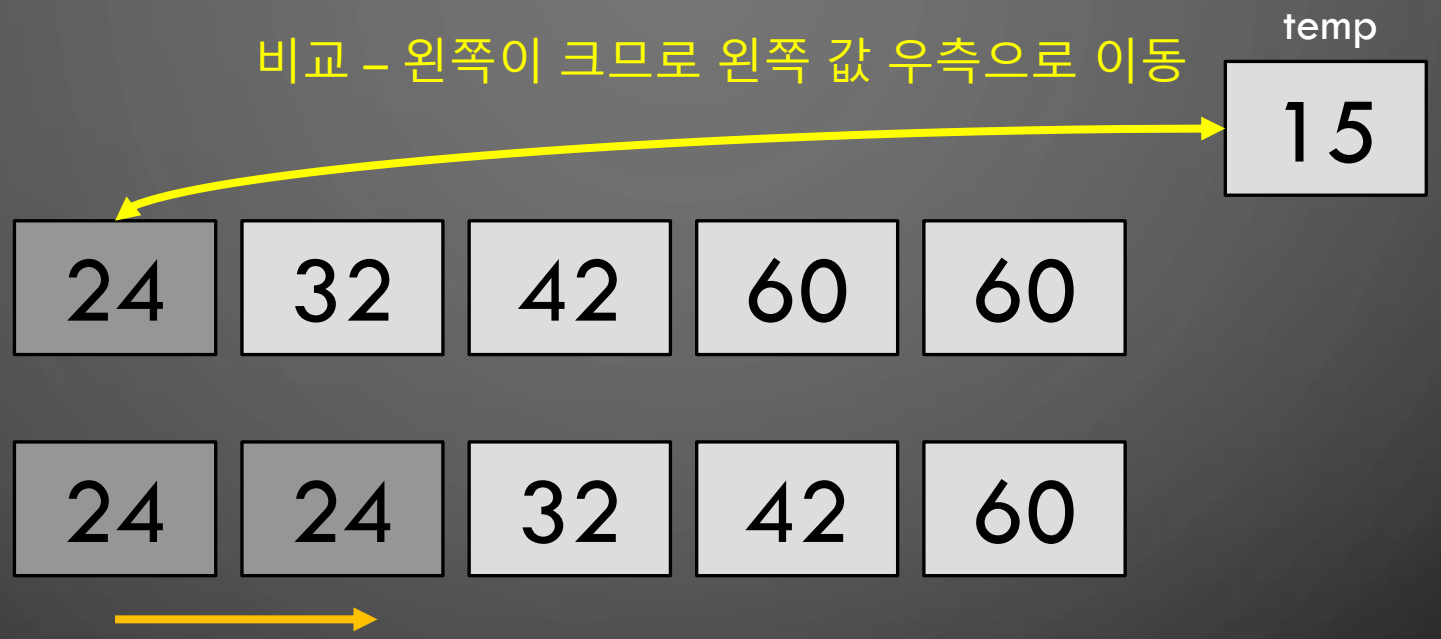
< 4번째 루프 >



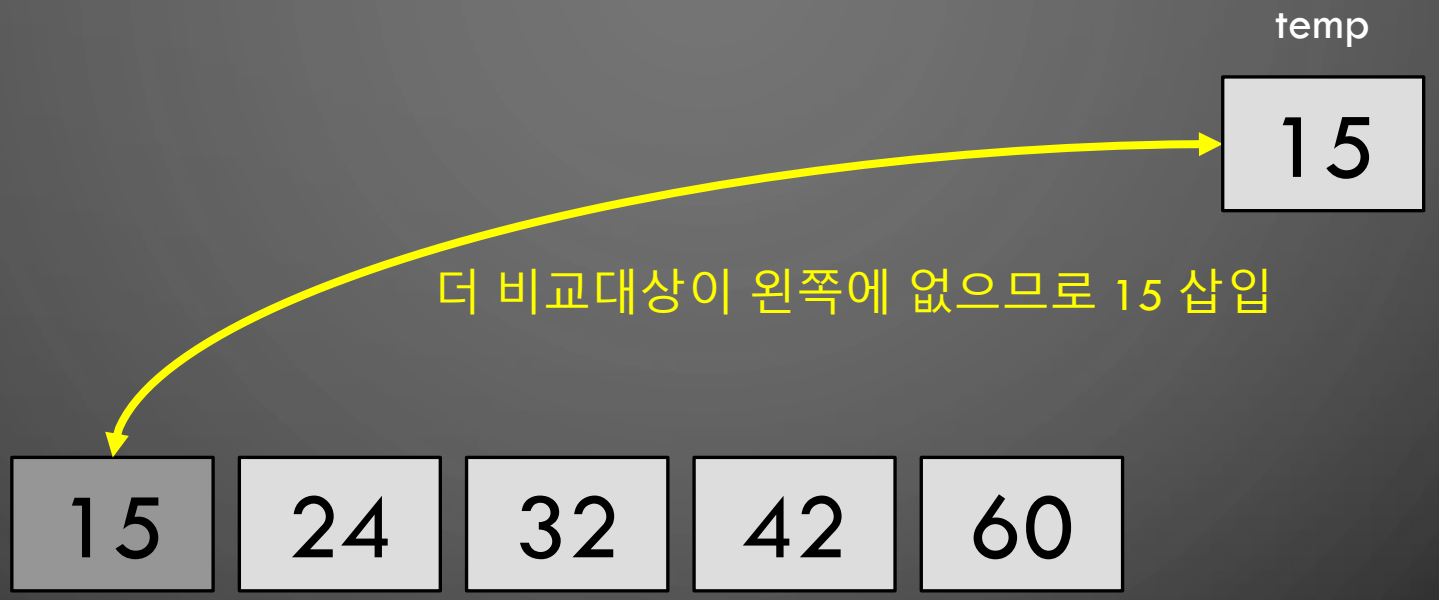
< 4번째 루프 >



< 4번째 루프 >



< 4번째 루프 >



< 5번째 루프 >

삽입 타겟이 없으므로 정렬 종료

15	24	32	42	60