

정렬

목차

1. 정렬의 종류
2. 선택 정렬
3. 버블 정렬
4. 퀵 정렬
5. 삽입 정렬
6. 셀 정렬
7. 병합 정렬
8. 기수 정렬
9. 히프 정렬

1. 정렬의 종류 (1/2)

- 정렬이란?
 - 자료를 순서대로 재배열
 - 키(Key)
 - 정렬에 사용되는 자료의 속성
 - 정렬 순서
 - 오름차순
 - 내림차순
- 정렬의 효율성
 - 시간복잡도: 얼마만큼 빨리 정렬
 - N , N^2 $N\log N$
 - 이동/비교 연산의 횟수
- 정렬의 안정성
 - 같은 키 값을 가지는 자료의 순서가 지켜지는지 여부
 - 다중 키 정렬

1. 정렬의 종류 (2/2)

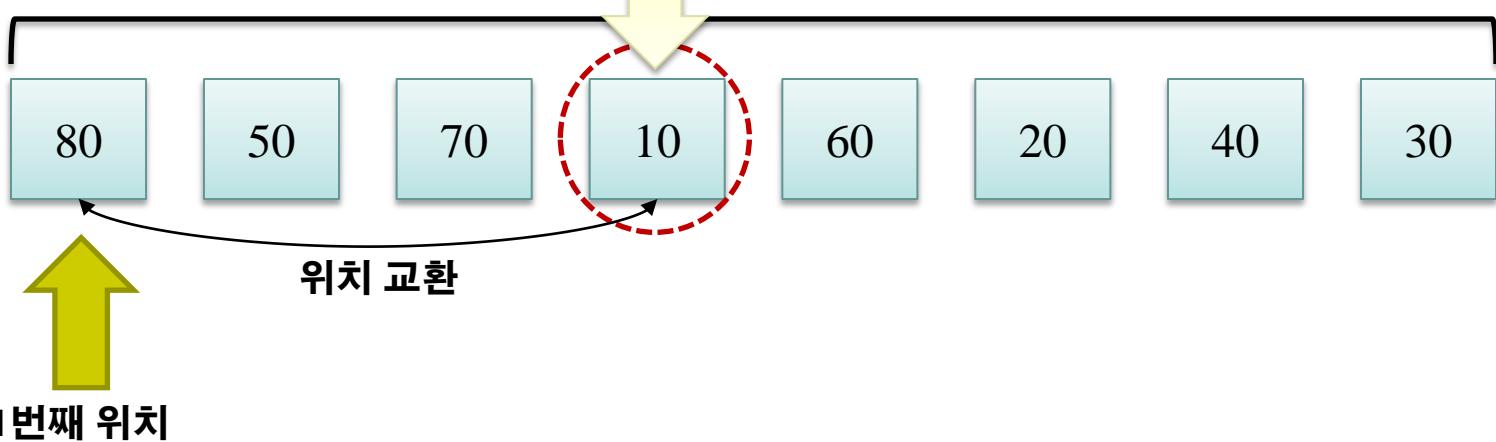
- 수행되는 장소에 따른 분류
 - 외부: 디스크
 - 내부: 메모리
- 실행 방법에 따른 분류
 - 교환
 - 버블 정렬, 퀵 정렬
 - 삽입
 - 삽입 정렬, 셀 정렬
 - 병합
 - 2-way 병합, n-way 병합
 - 분배
 - 기수 정렬
 - 선택: 특정 자료구조 이용
 - 선택 정렬, 히프 정렬

2. 선택 정렬 (1/2)

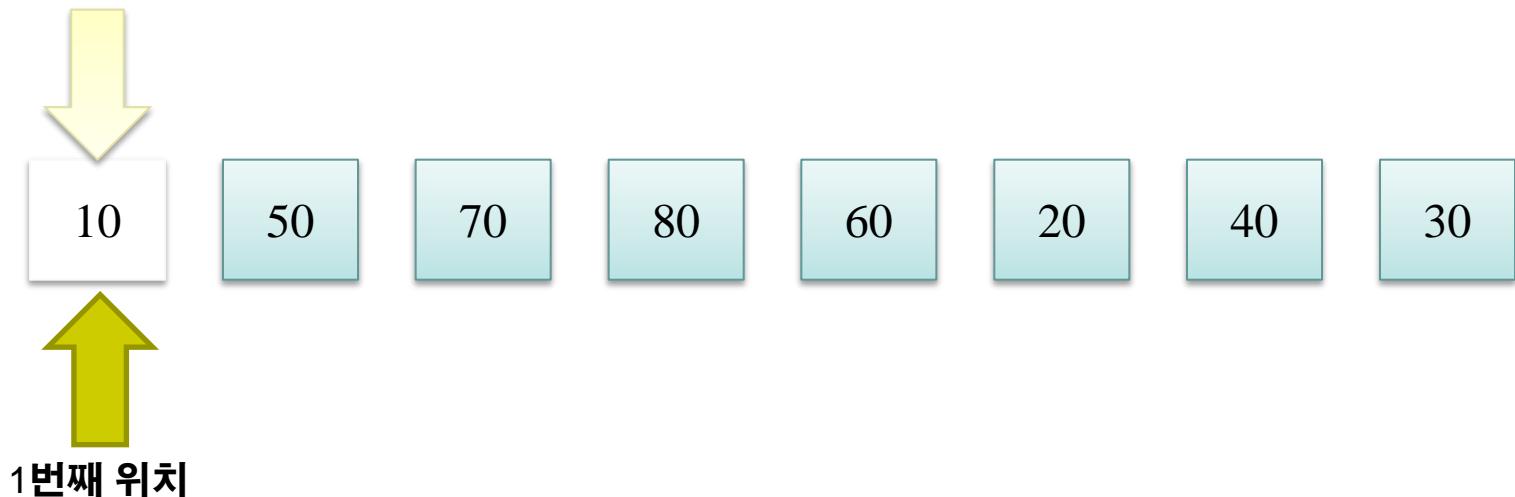
- 선택 정렬(Selection Sort)
 - 정렬되지 않은 자료 중에서 해당 위치에 맞는 자료를 선택하여 위치 교환

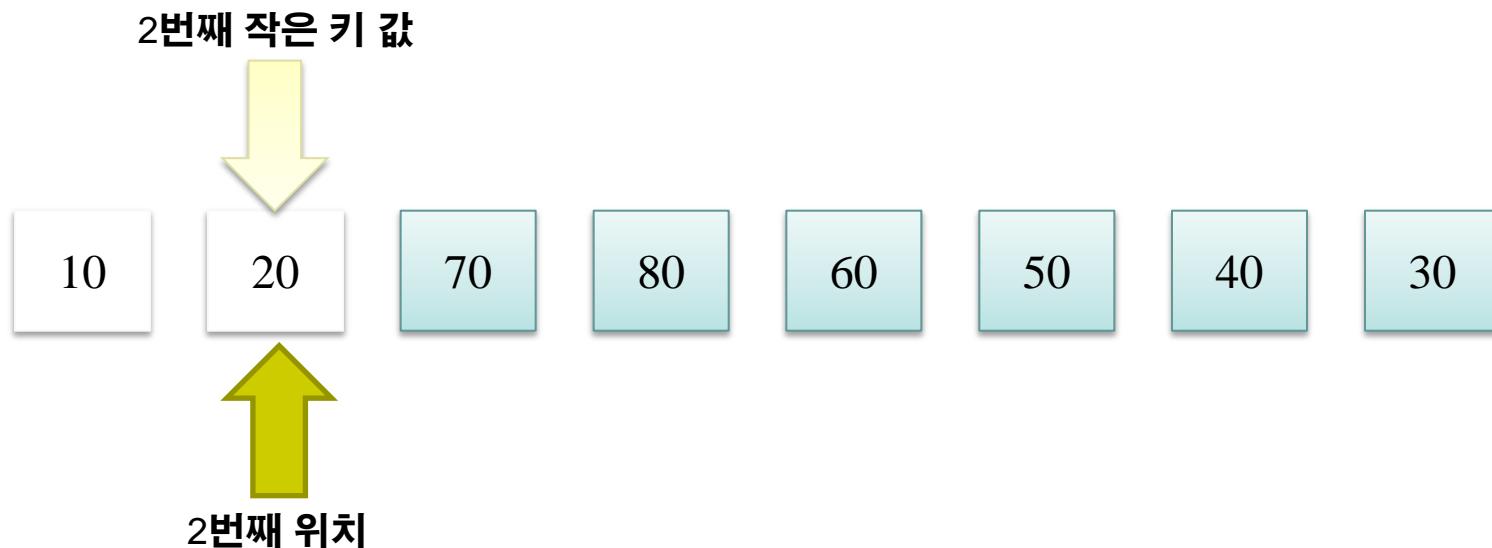
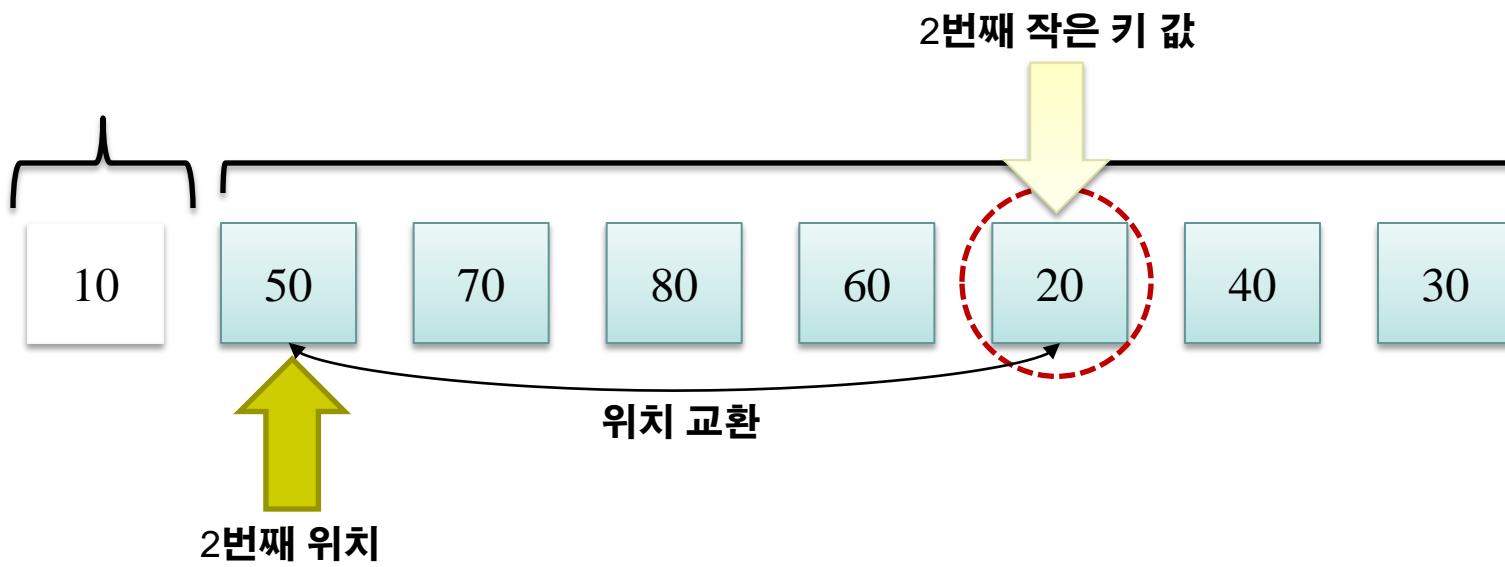


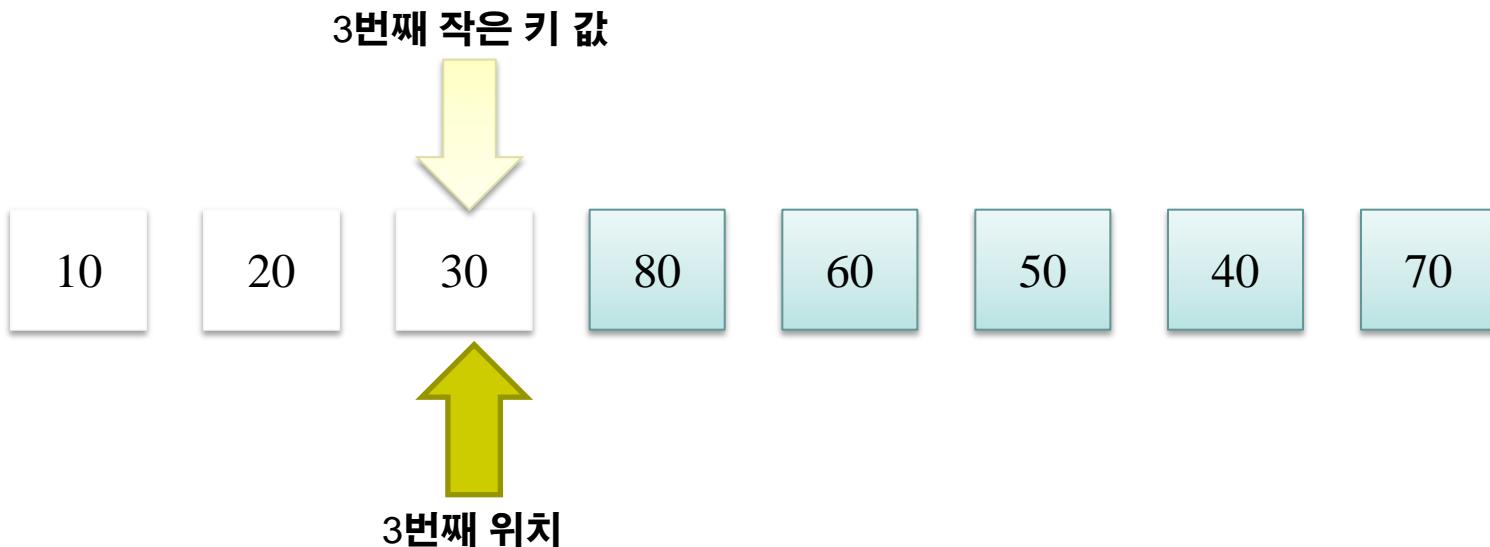
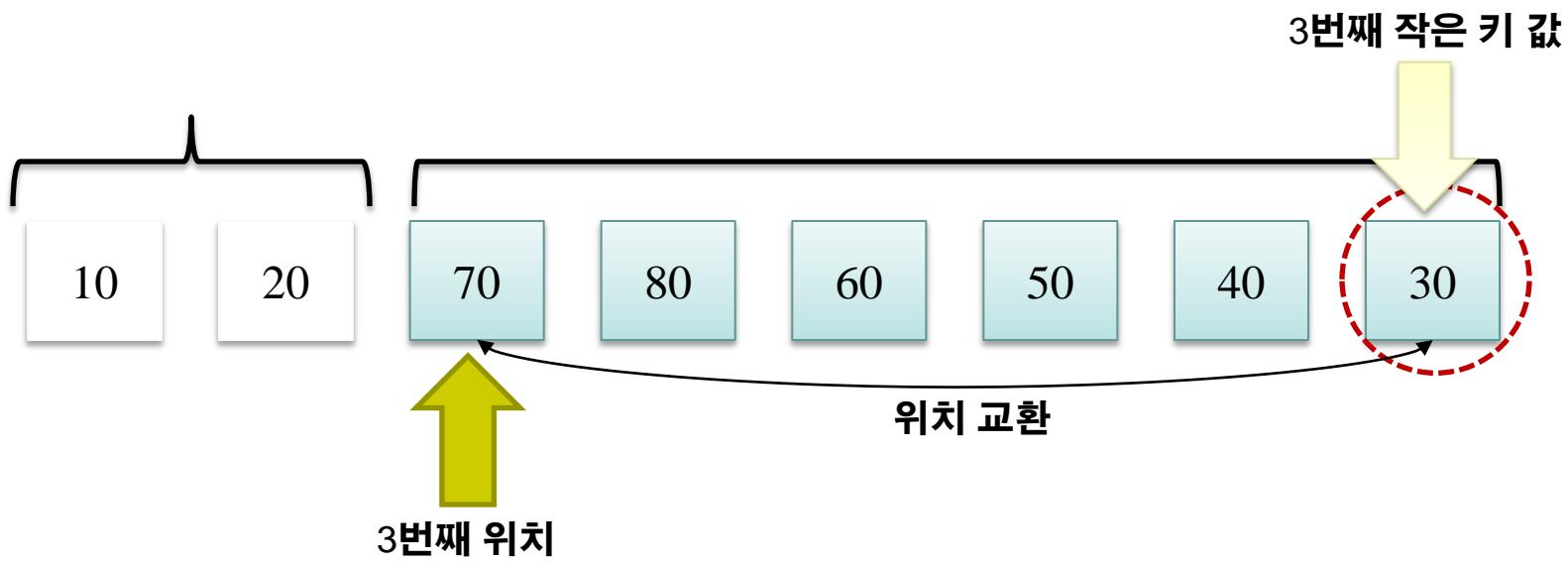
가장 작은 키 값

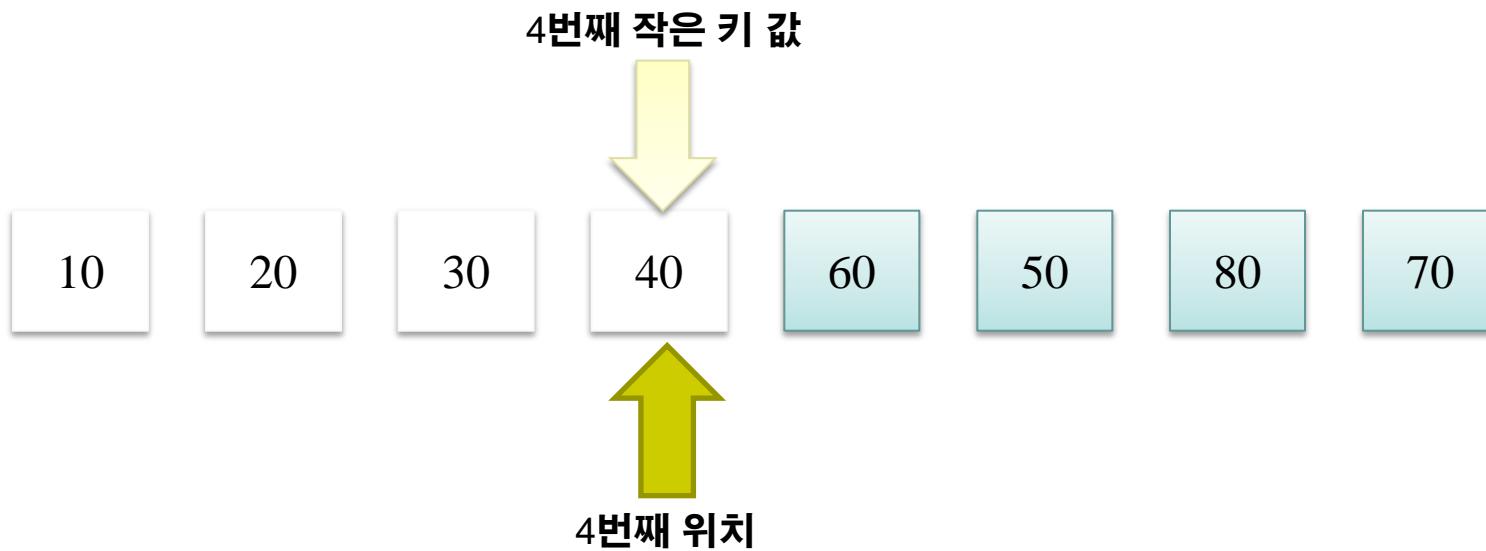
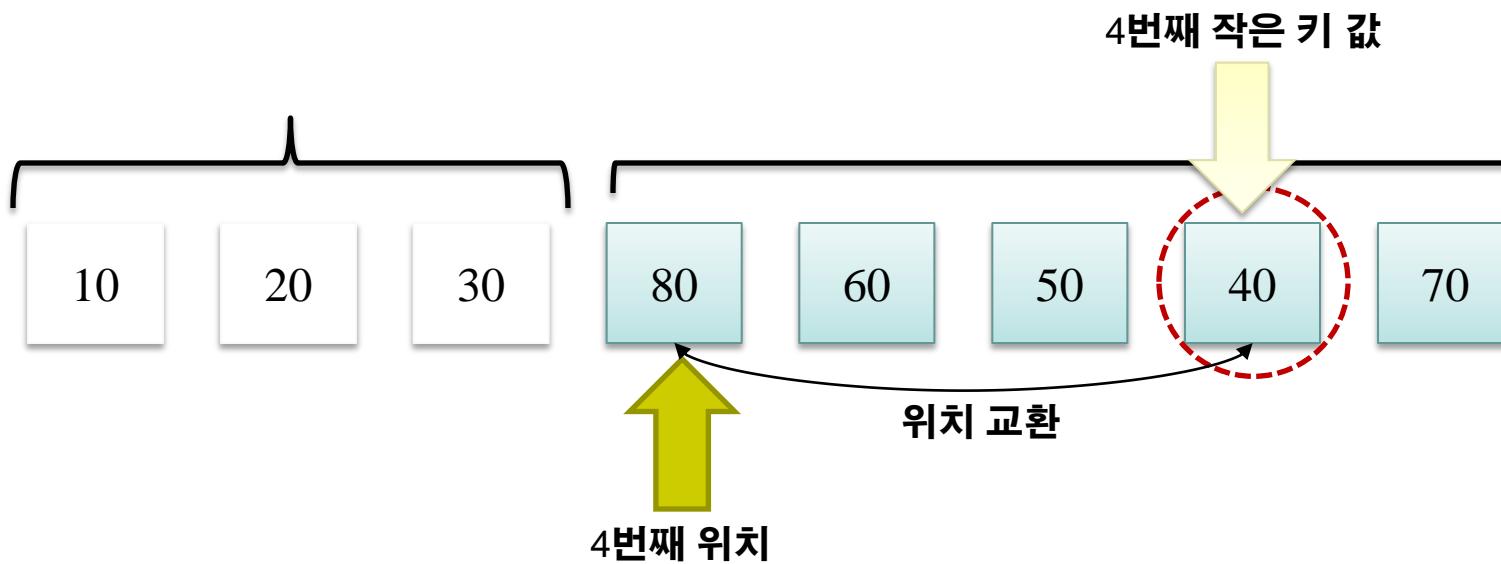


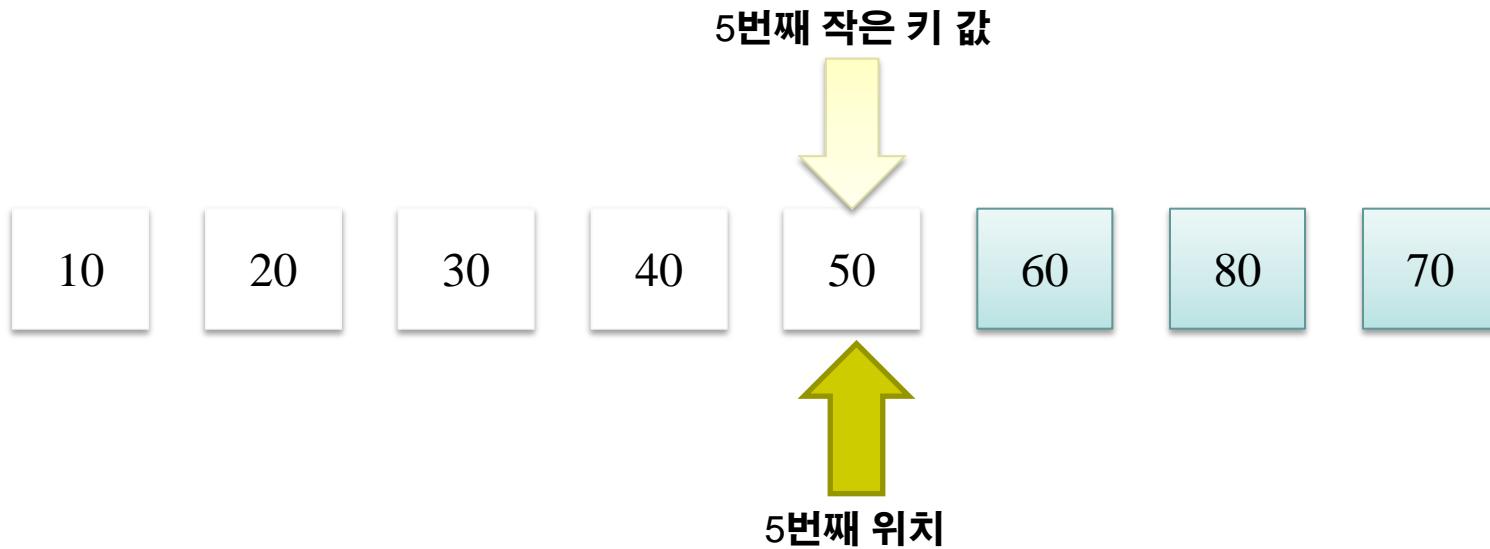
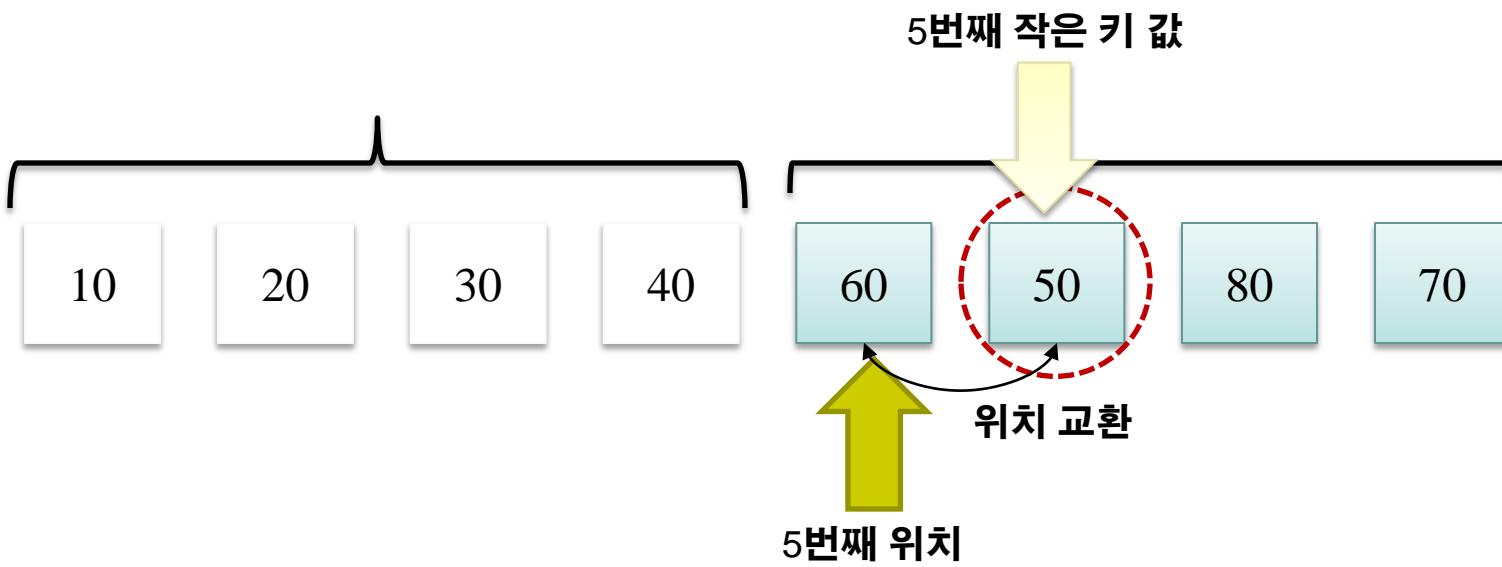
가장 작은 키 값



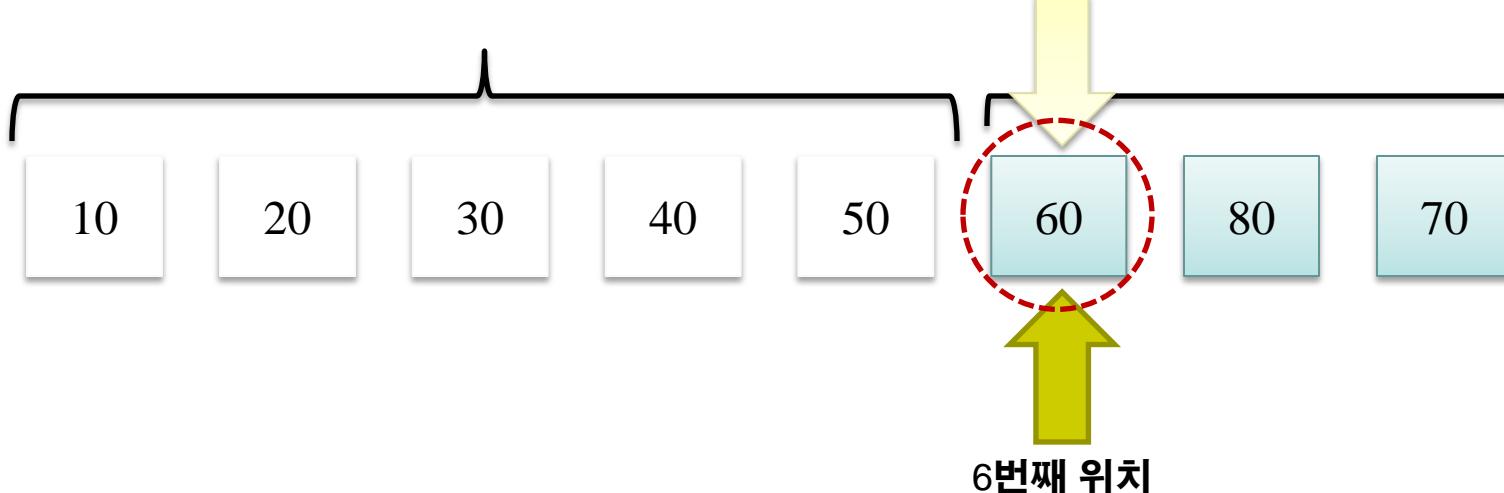




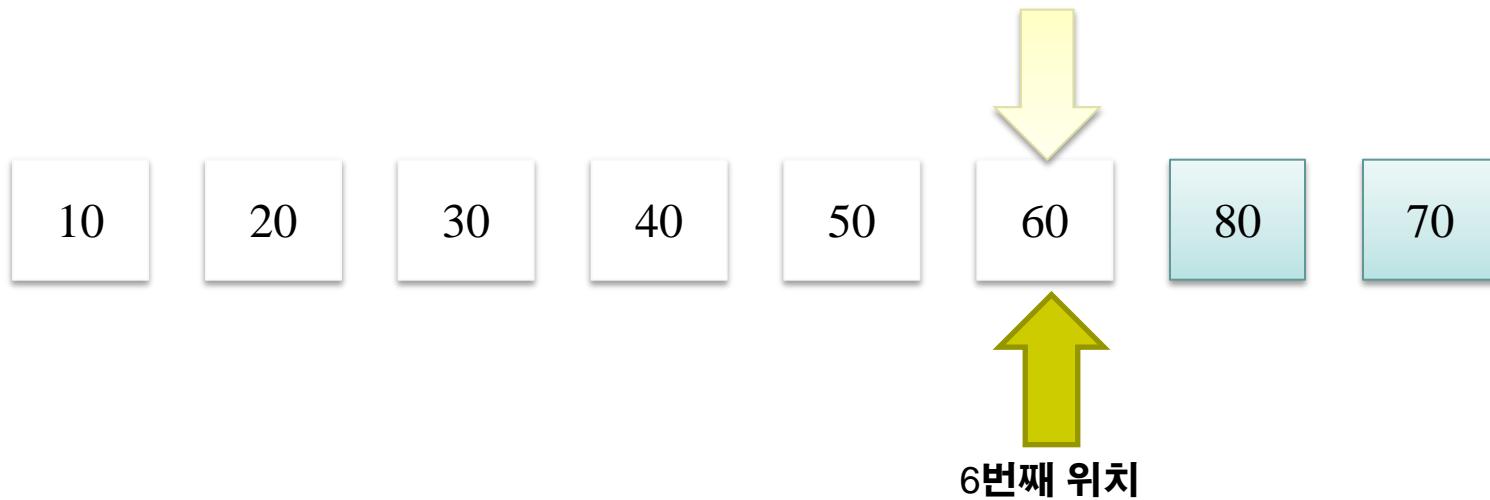


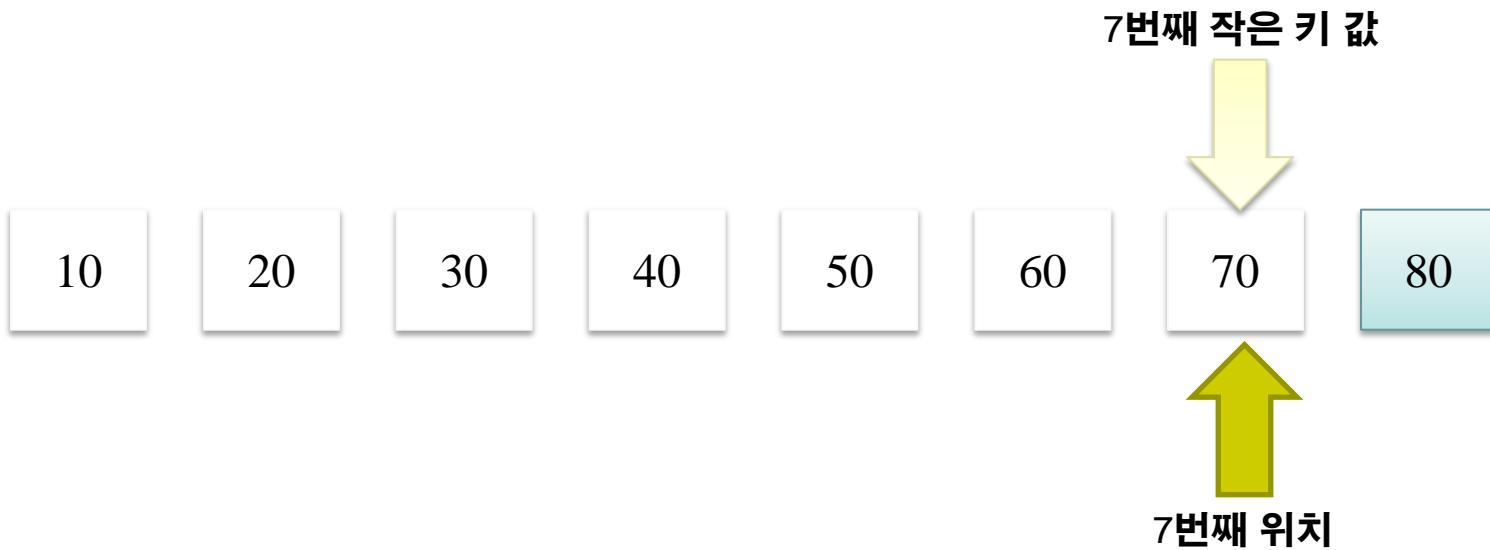
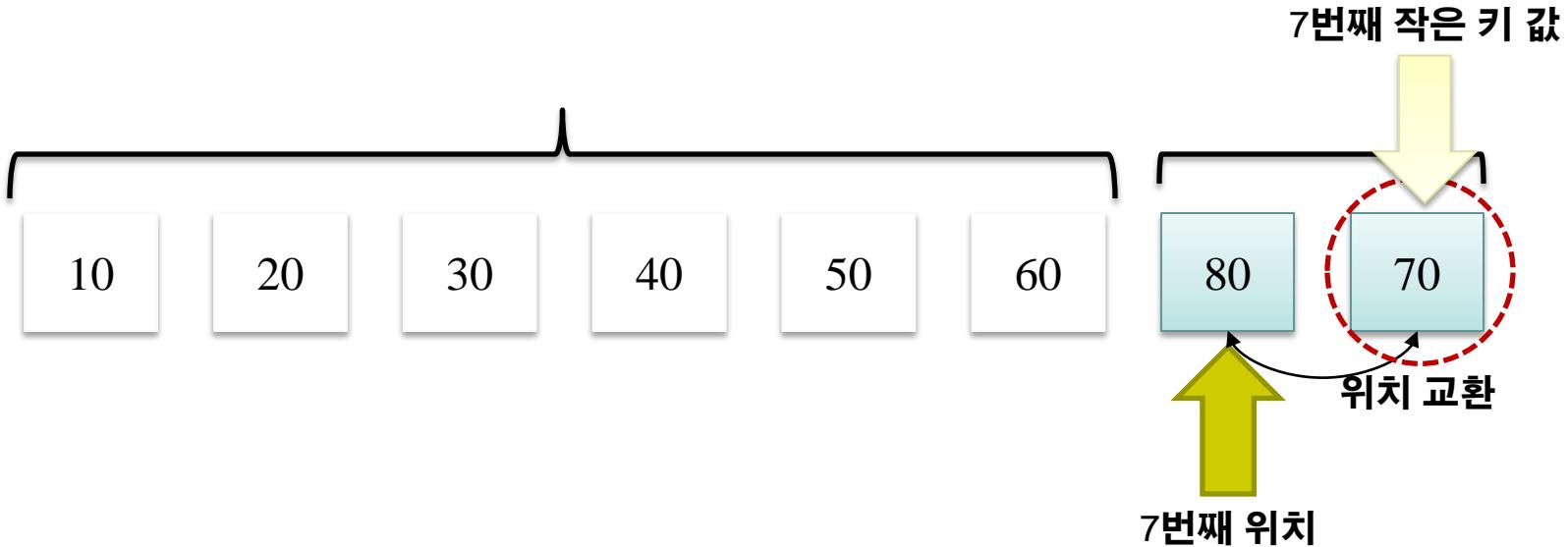


6번째 작은 키 값

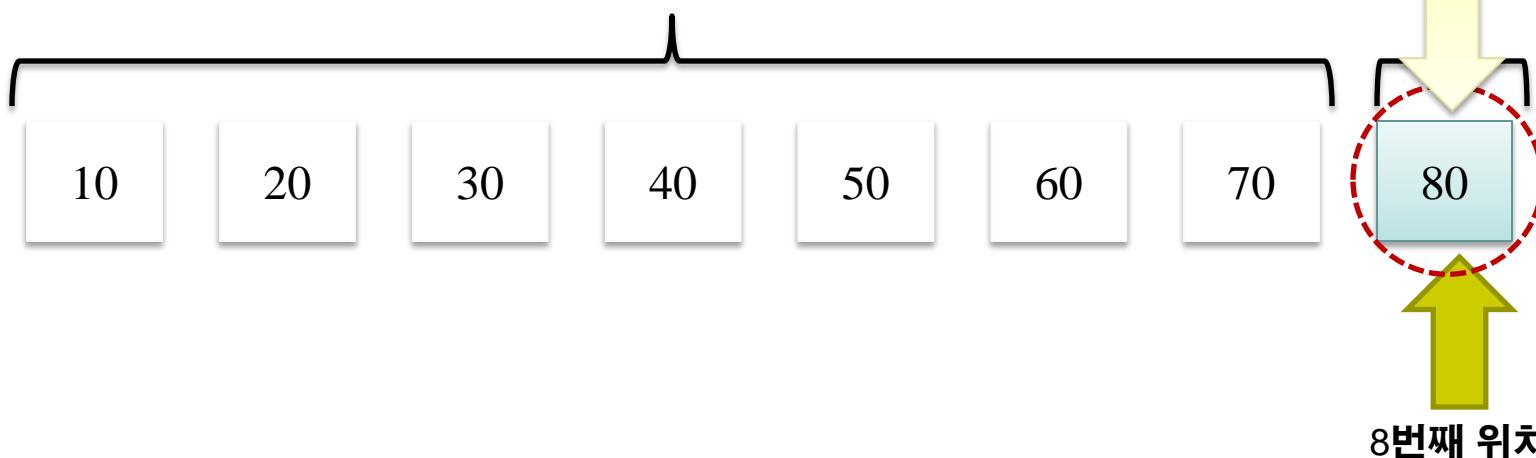


6번째 작은 키 값

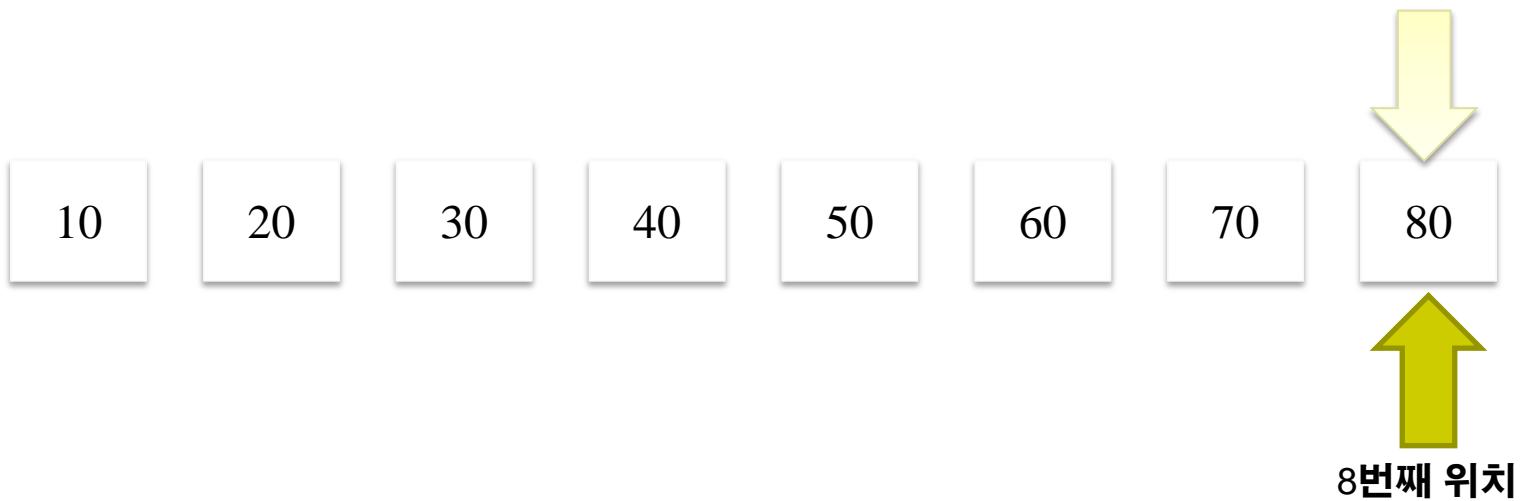




8번째 작은 키 값



8번째 작은 키 값



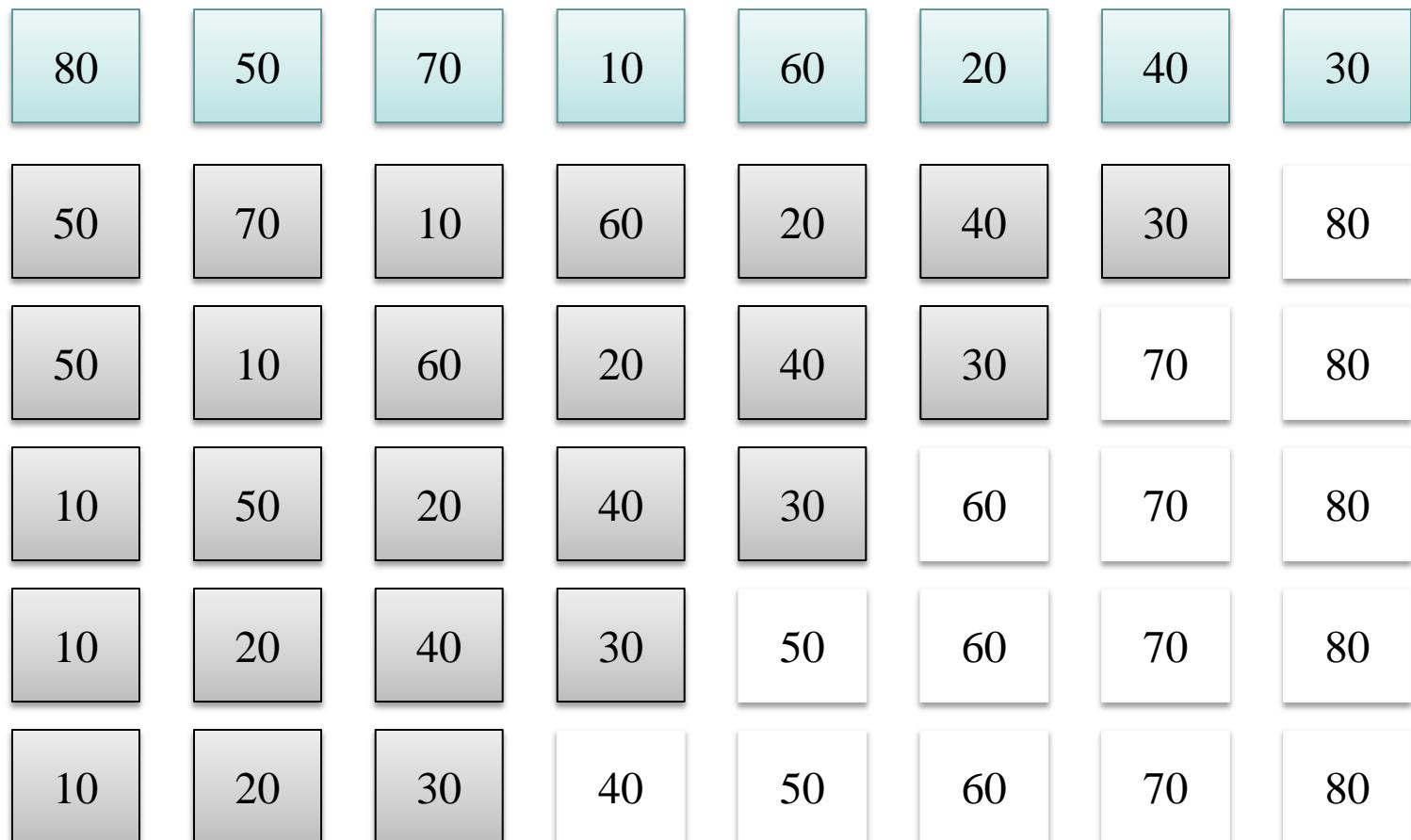
2. 선택 정렬 (2/2)

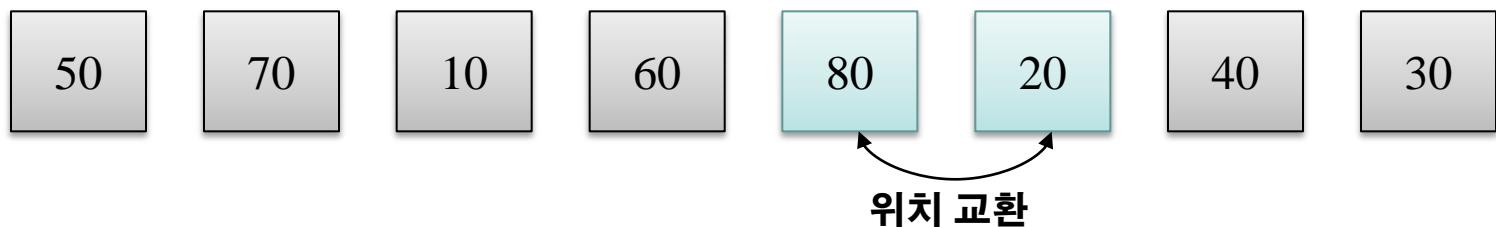
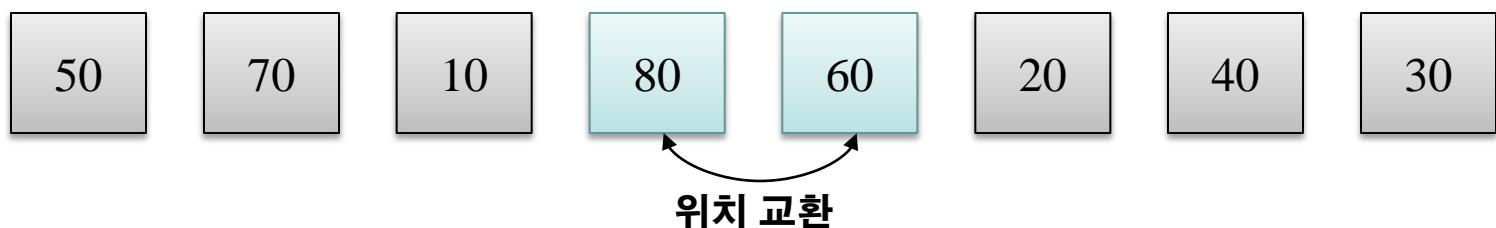
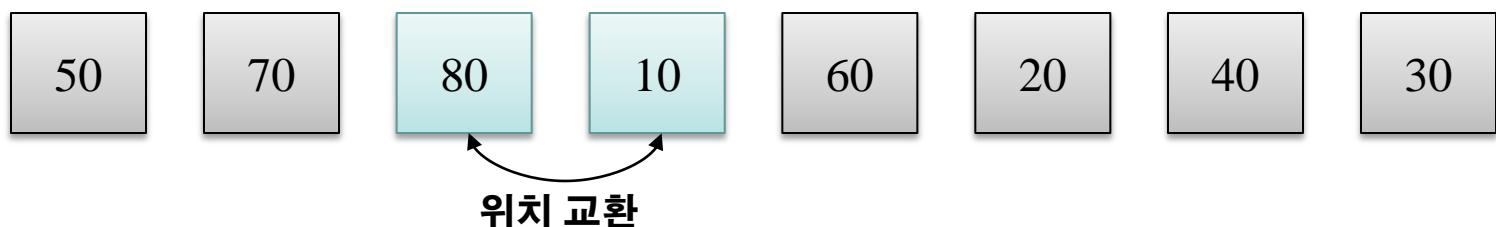
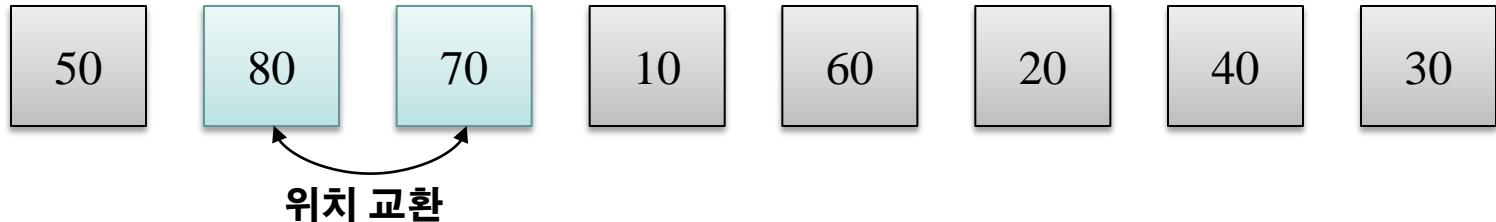
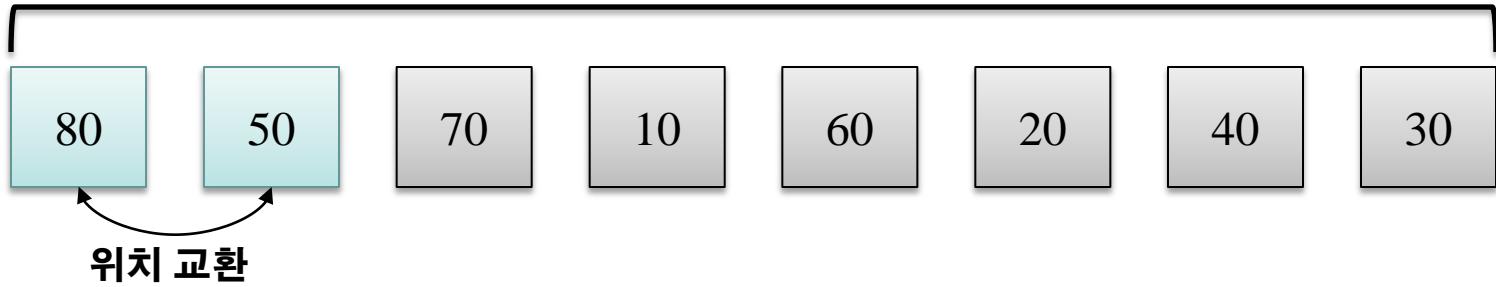
- 특성
 - 비교 연산 횟수
 - $O((n-1) + (n-2) + \dots + 3 + 2 + 1) = O(n(n-1) / 2) = O(n^2)$
 - 이동 연산 횟수
 - $O(3(n-1)) = O(n)$
 - 최선, 평균, 최악
 - 최선, 평균, 최악: $O(n^2 + n) = O(n^2)$
 - 정렬의 안정성
 - 불안정

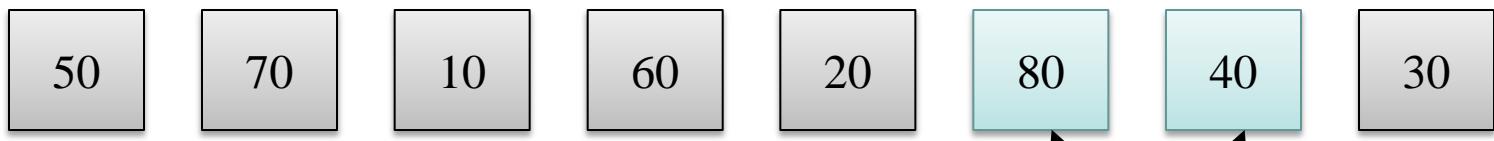
3. 버블 정렬 (1/2)

- 버블 정렬(Bubble Sort)

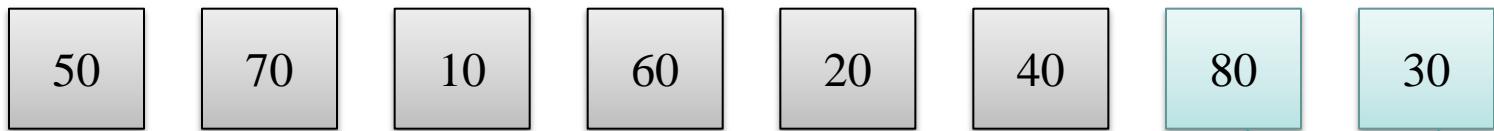
- 정렬되지 않은 자료를 대상으로 인접 두 개 자료의 값을 비교하여 위치 교환







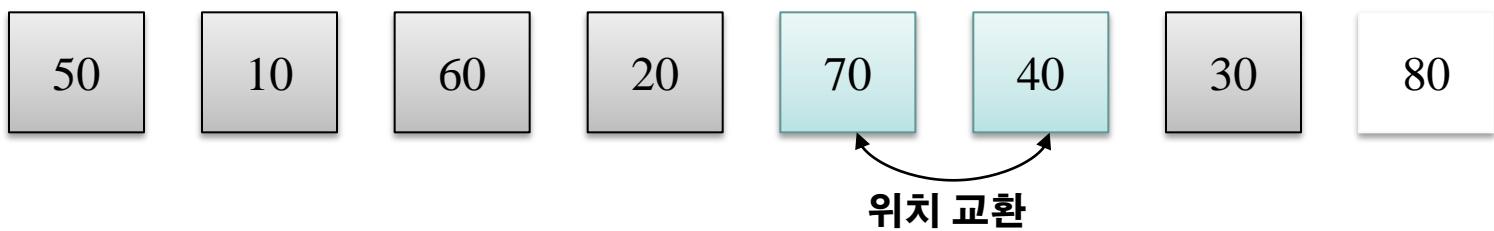
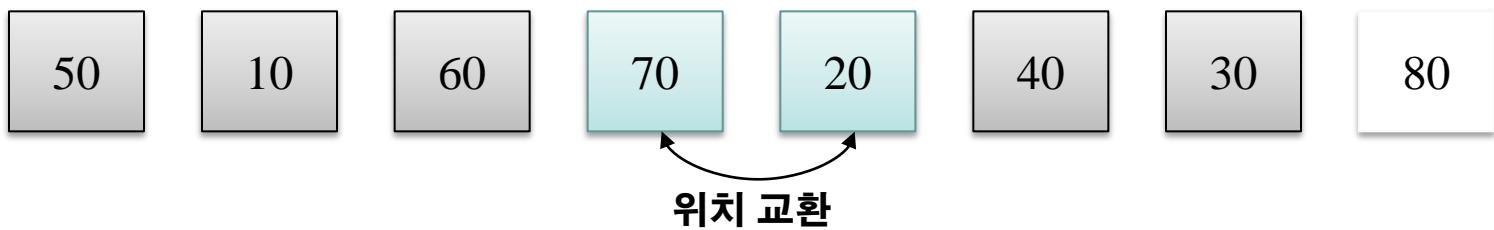
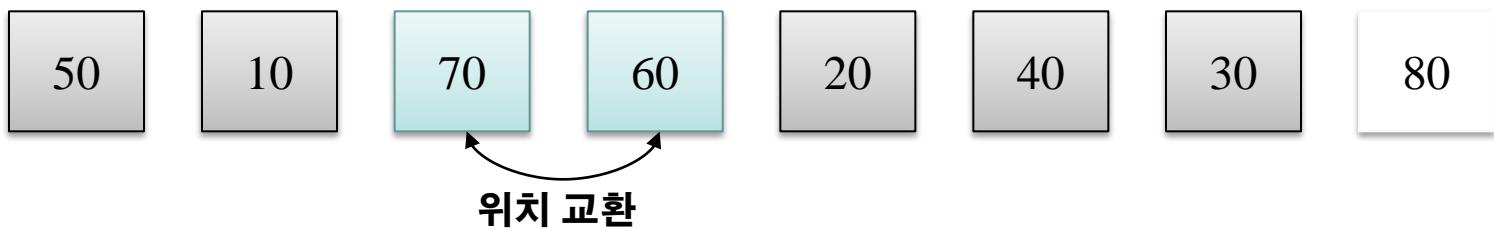
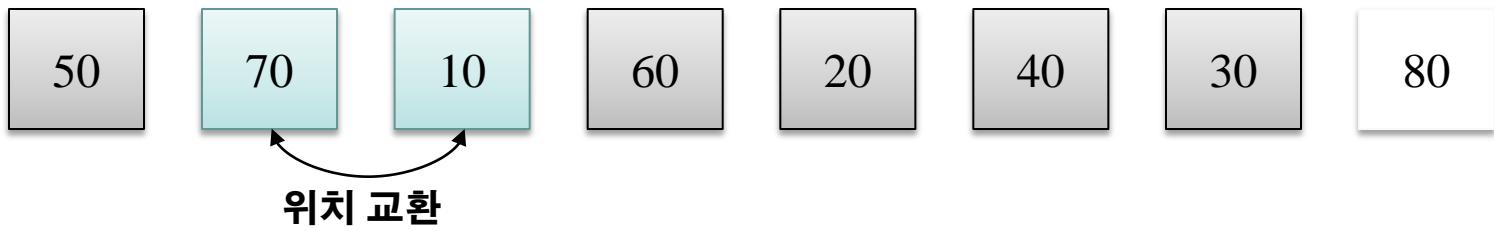
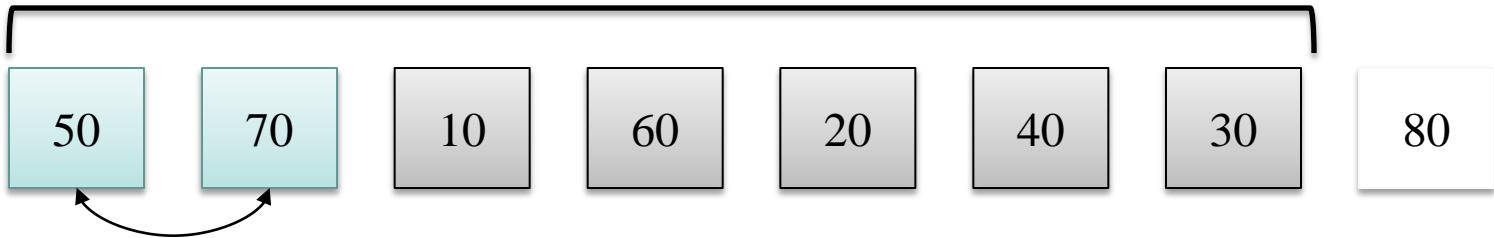
위치 교환



위치 교환

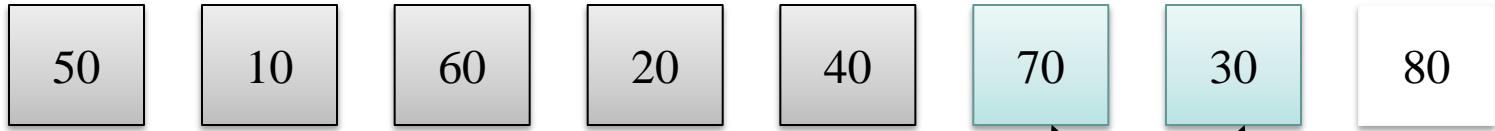


정렬된 자료

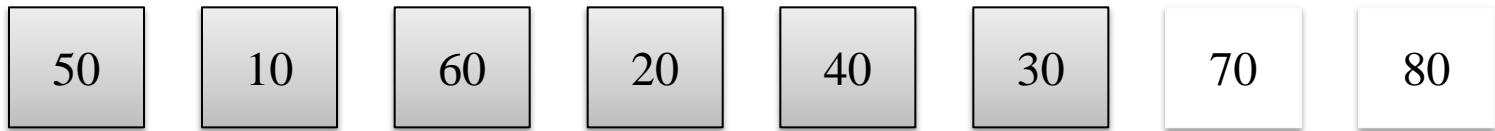




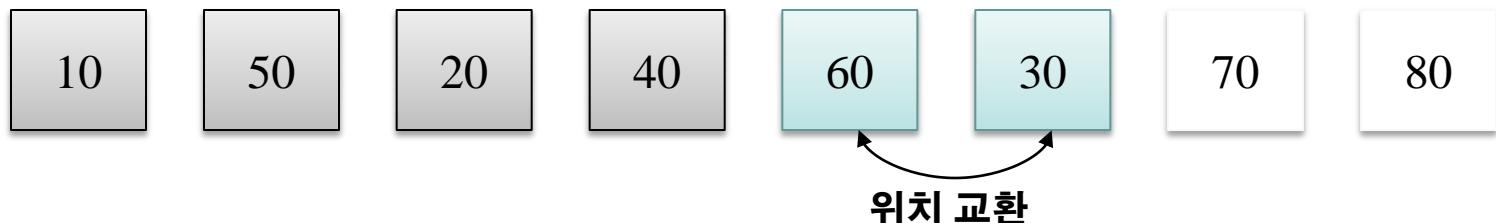
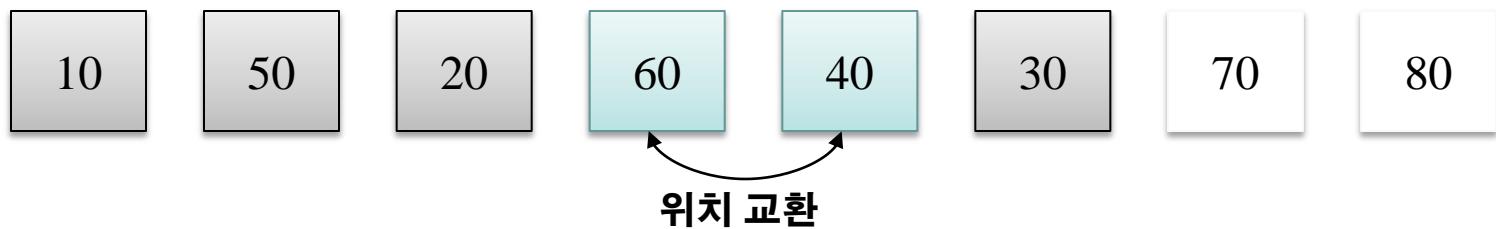
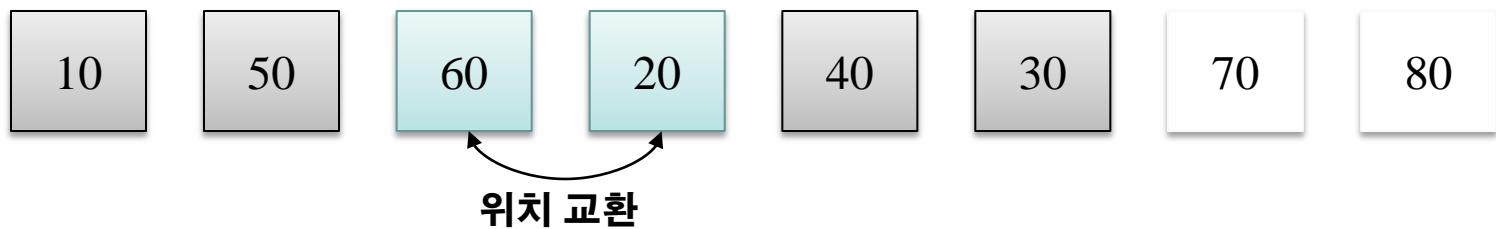
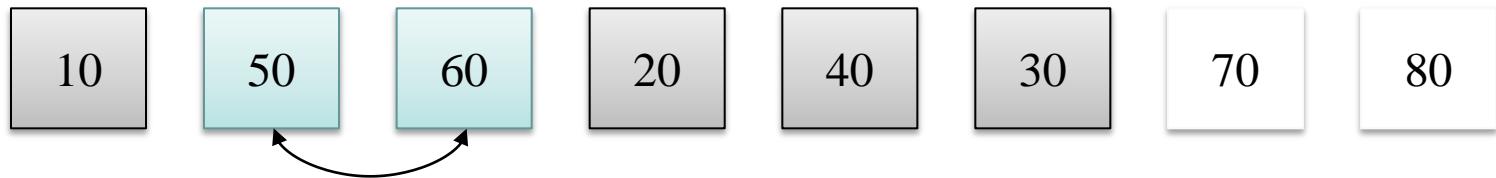
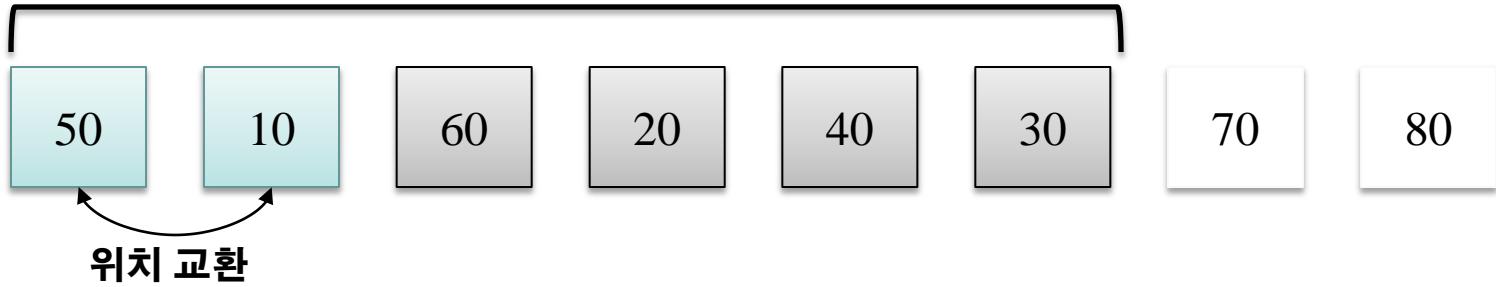
위치 교환



위치 교환



정렬된 자료



10

50

20

40

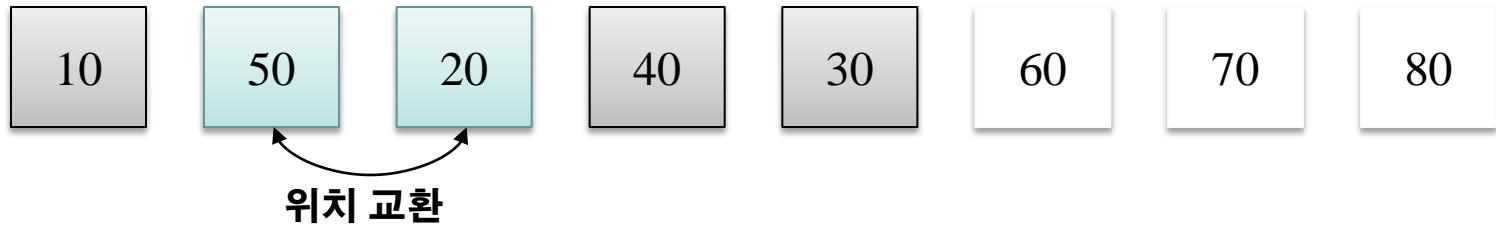
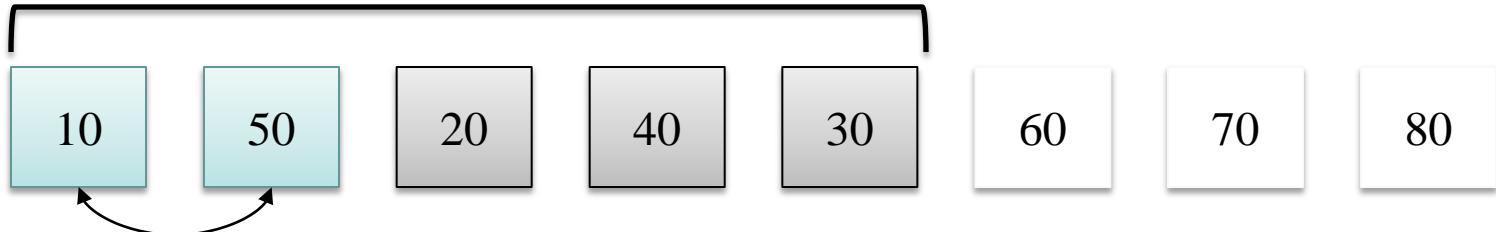
30

60

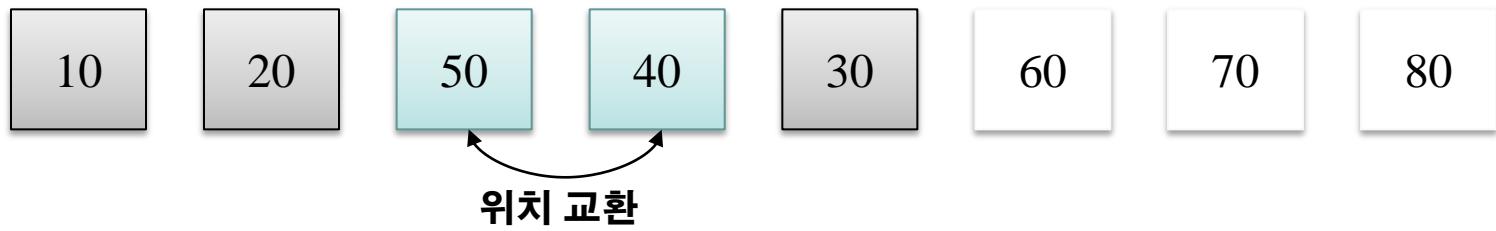
70

80

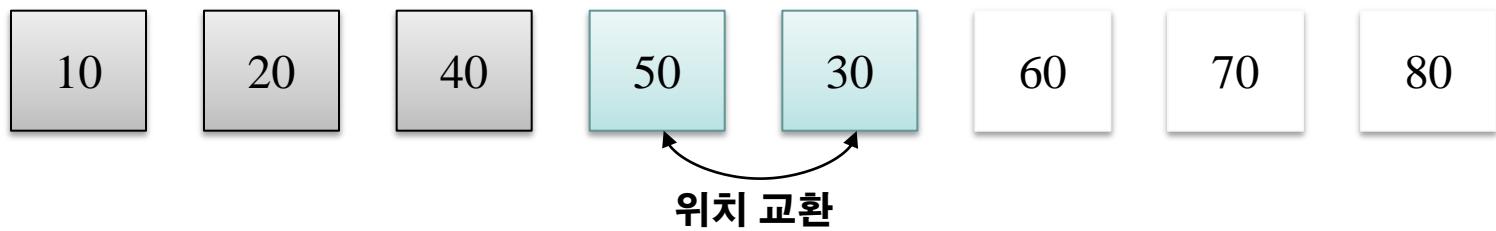
정렬된 자료



위치 교환

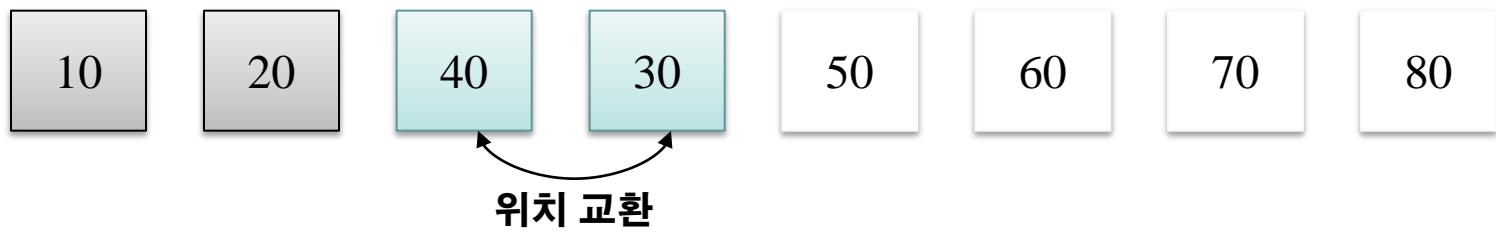
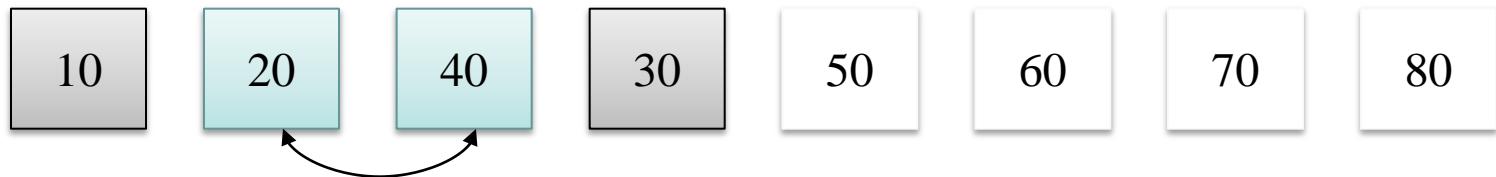
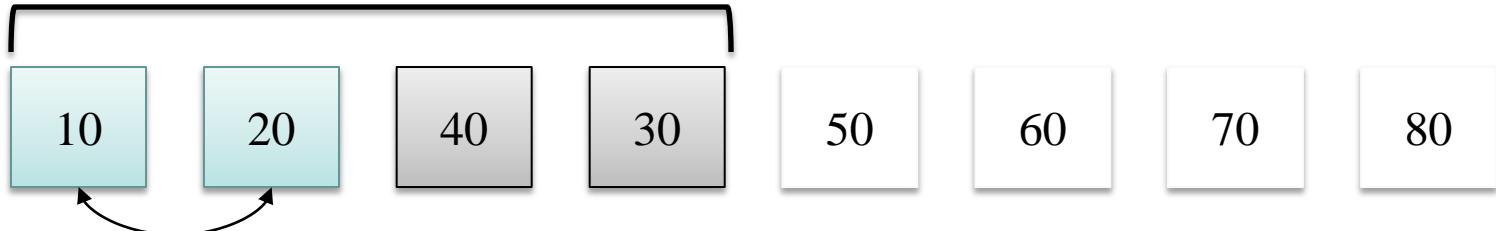


위치 교환



위치 교환





정렬된 자료



정렬된 자료



정렬된 자료

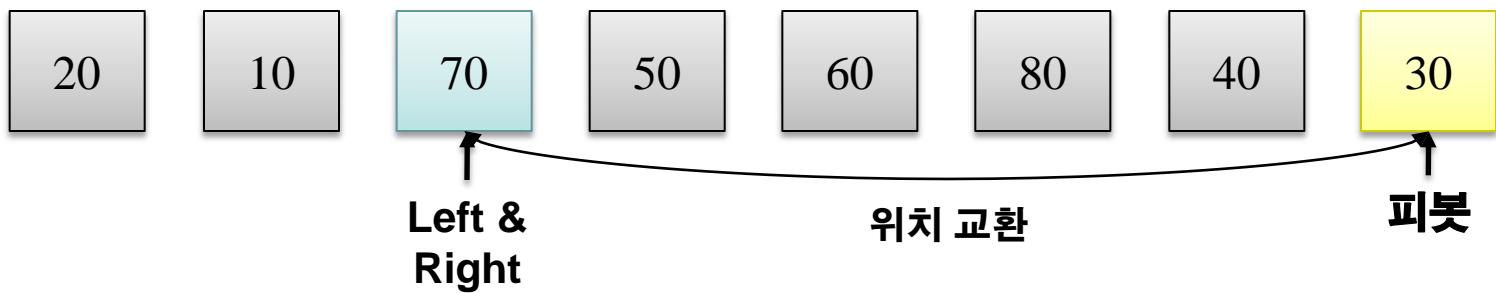
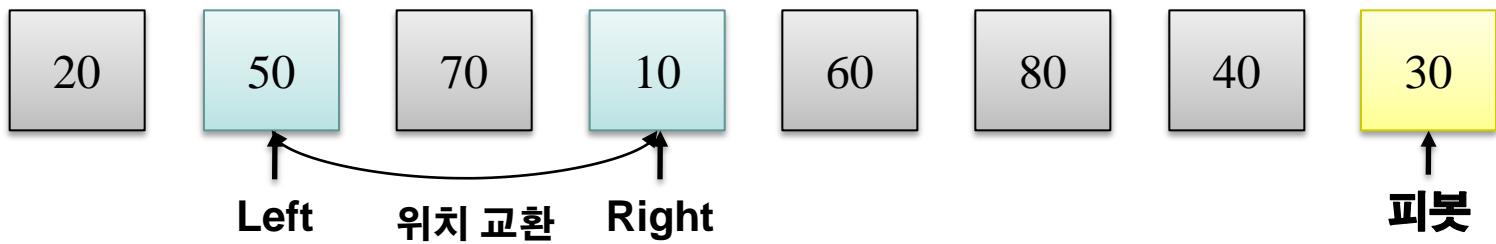
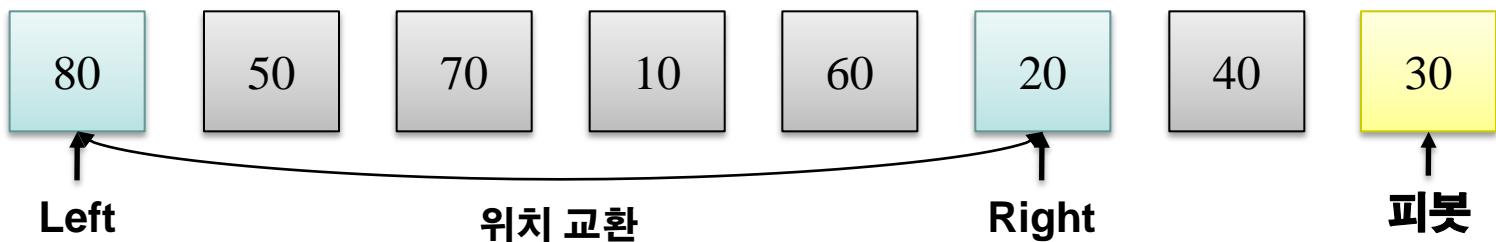
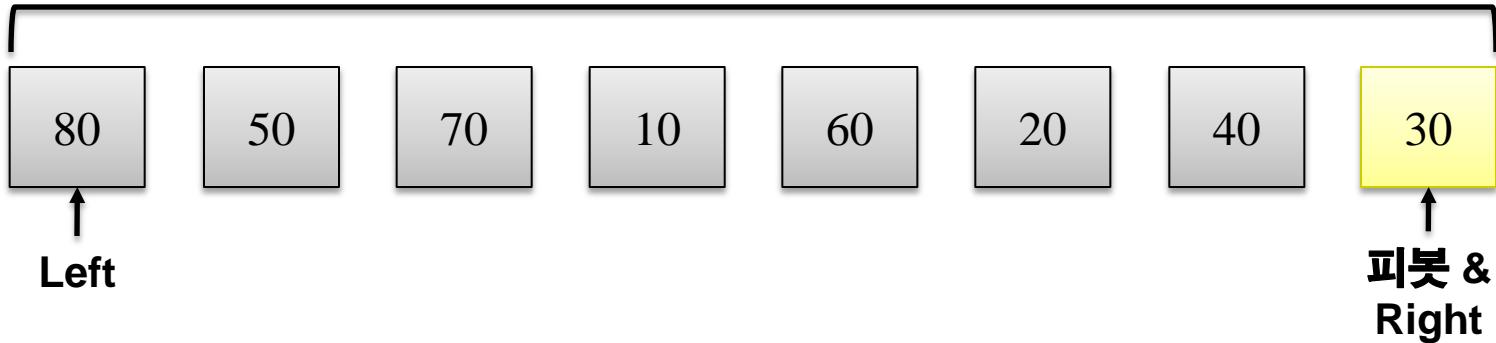


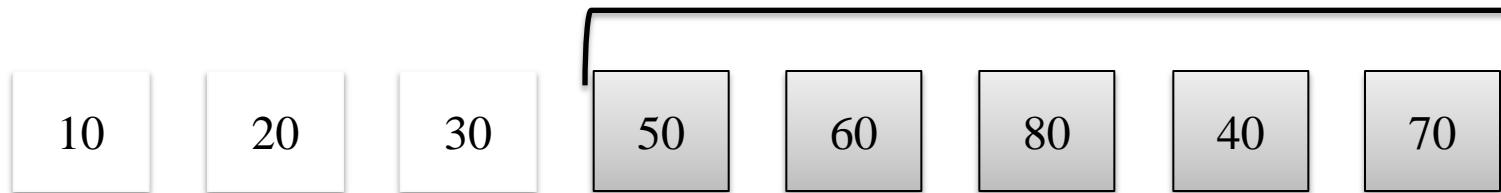
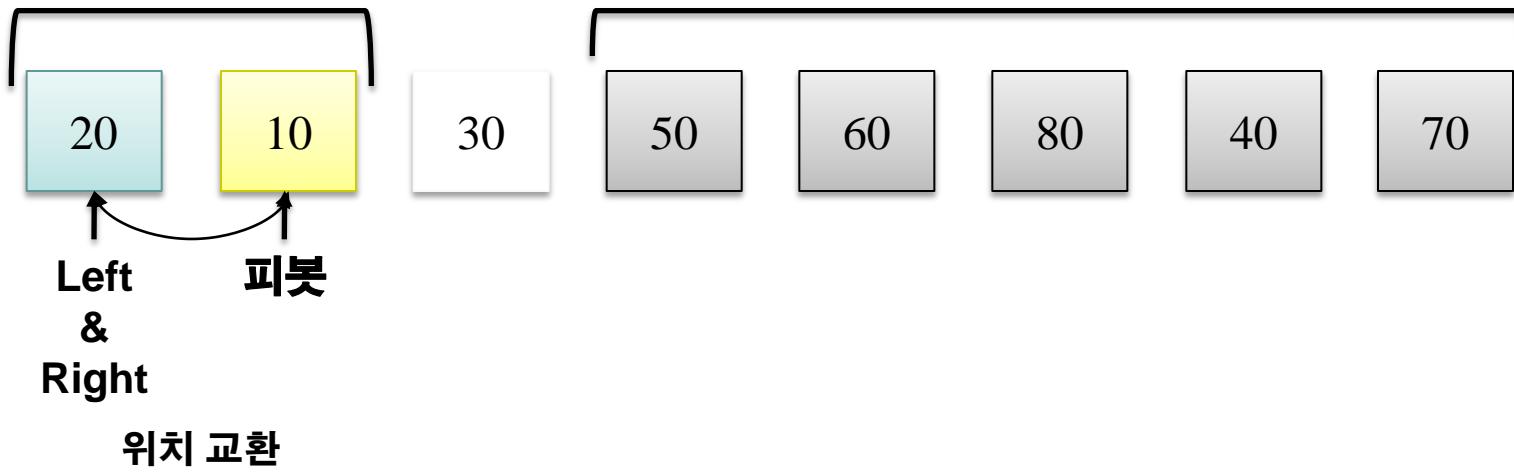
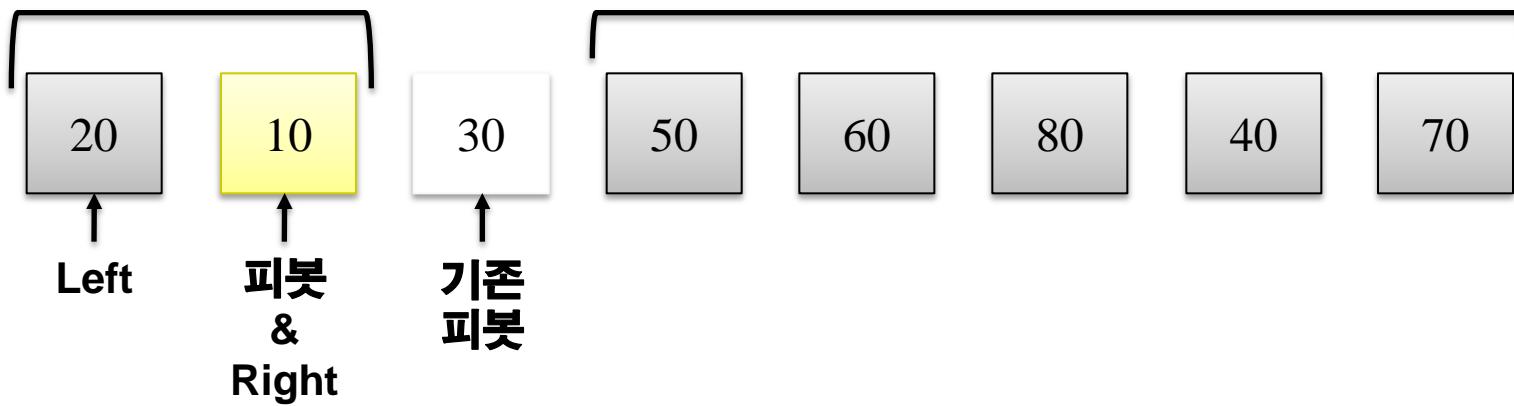
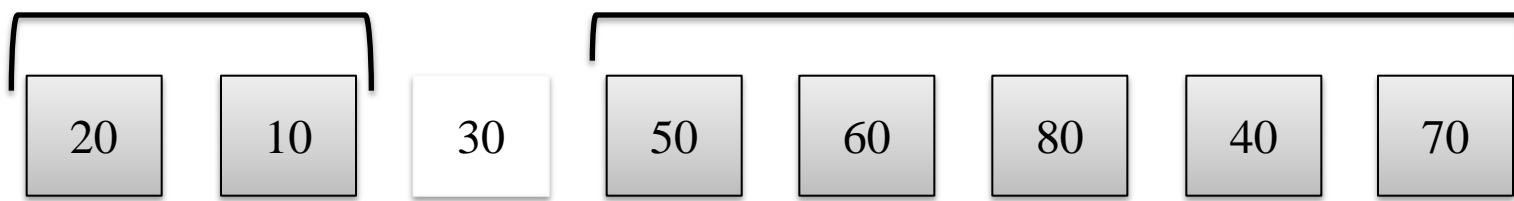
3. 버블 정렬 (2/2)

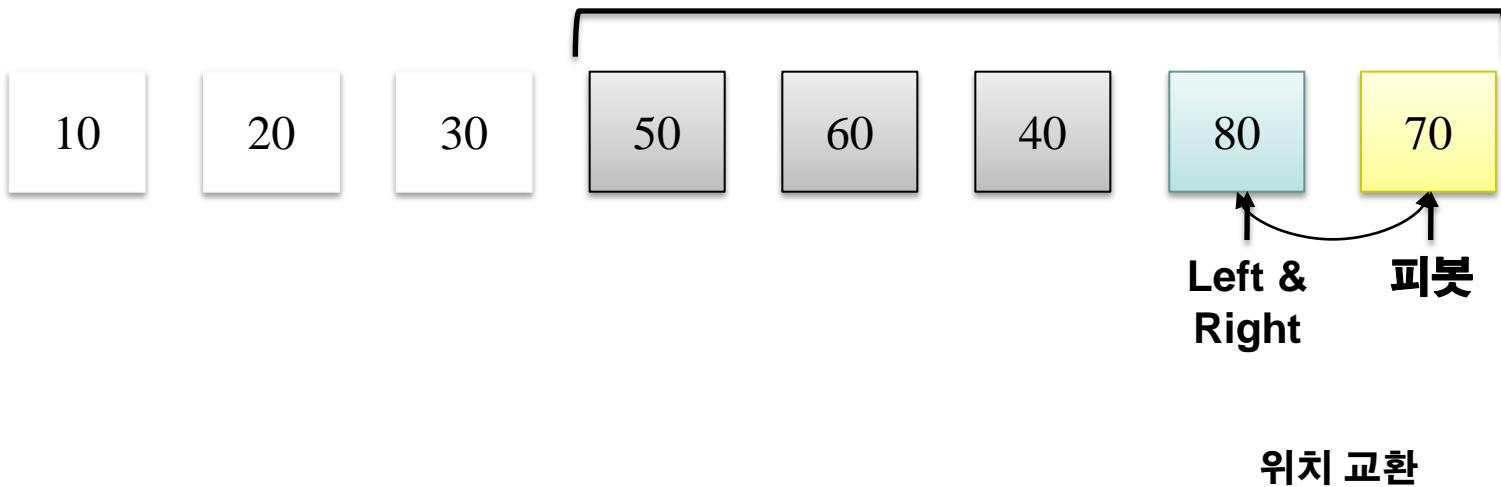
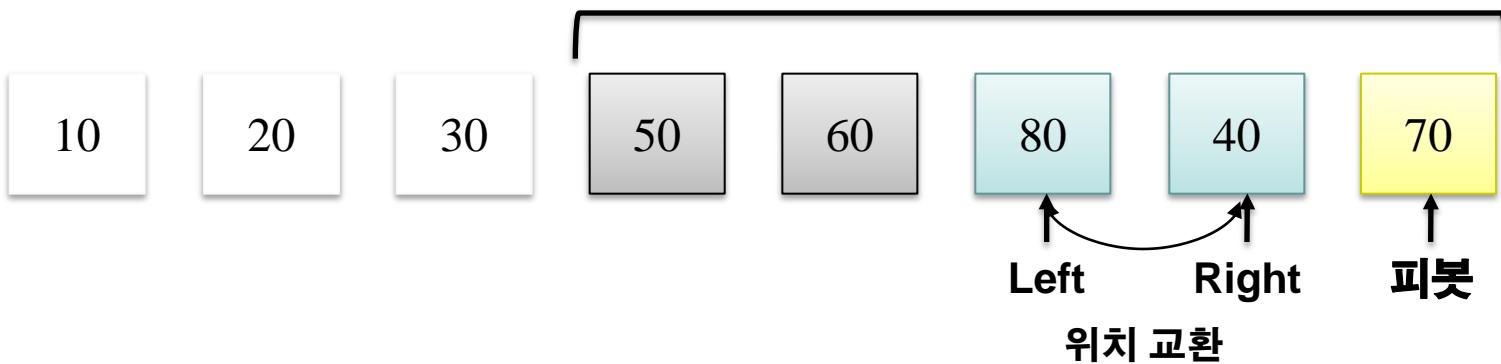
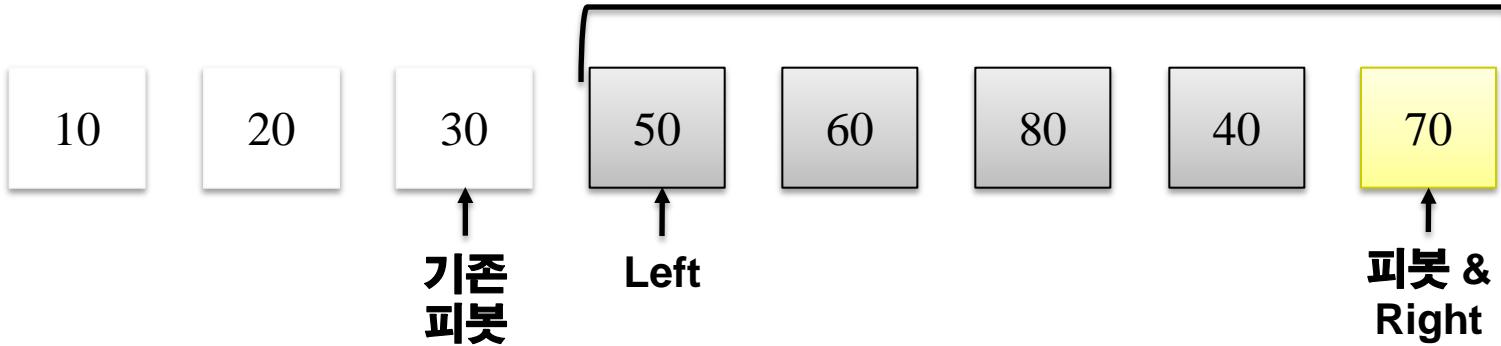
- 특성
 - 비교 연산 횟수
 - $O((n-1) + (n-2) + \dots + 3 + 2 + 1) = O(n(n-1) / 2) = O(n^2)$
 - 이동 연산 횟수
 - 최선: $O(0) = 0$
 - 평균 및 최악: $O(n^2)$
 - 최선, 평균, 최악
 - 최선, 평균, 최악: $O(n^2)$
 - 정렬의 안정성
 - 안정성 유지

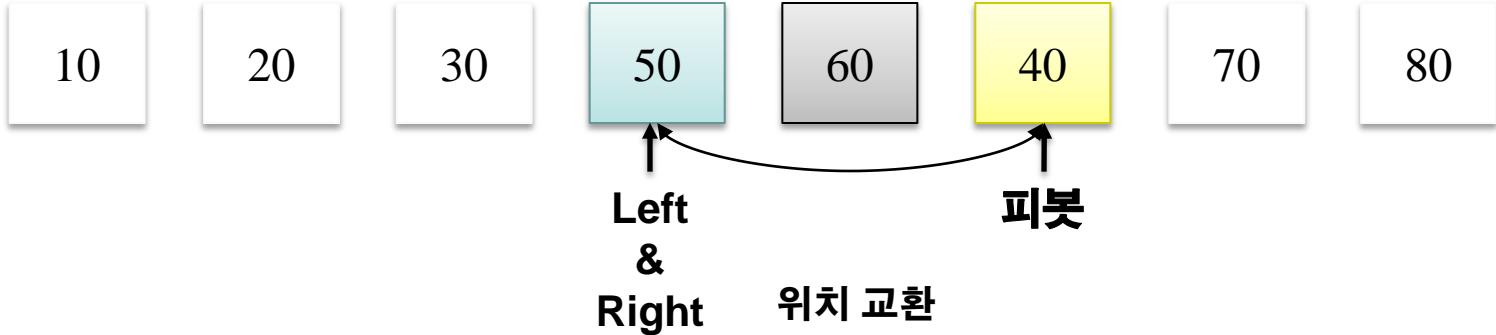
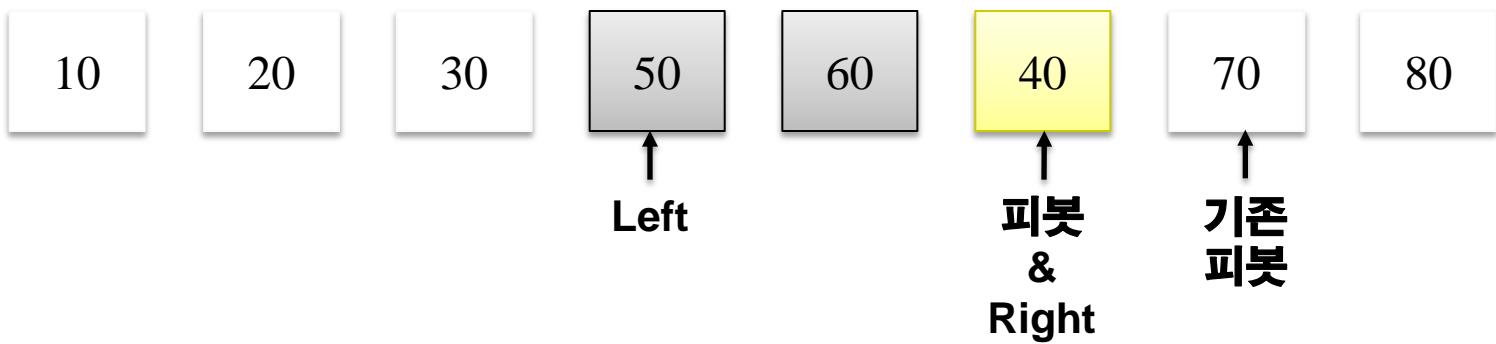
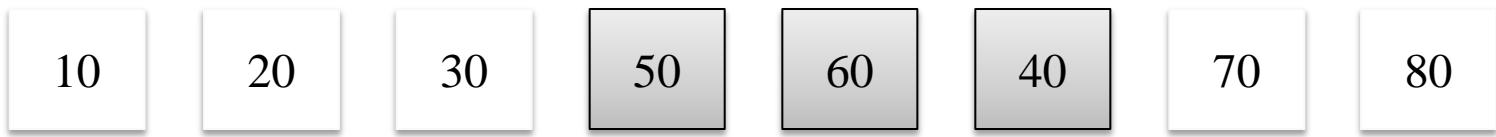
4. 퀵 정렬 (1/2)

- 퀵 정렬(Quick Sort)
 - 중심 값(피봇: Pivot)을 기준으로 두 자료의 키 값을 비교하여 위치를 교환
 - Left: 피봇보다 큰 자료 찾음, 원쪽->오른쪽, Right까지 이동 가능
 - Right: 피봇보다 작은 자료 찾음, 오른쪽->원쪽, Left보다 원쪽으로 이동할 수 없음











↑
**기준
피봇**

↑
Left

↑
**피봇
&
Right**



↑
**Left &
Right**

↑
피봇

위치 교환

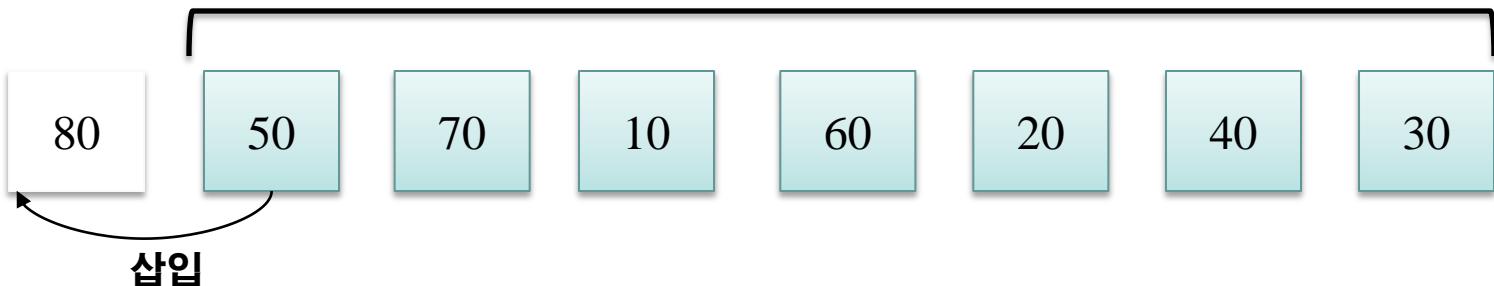


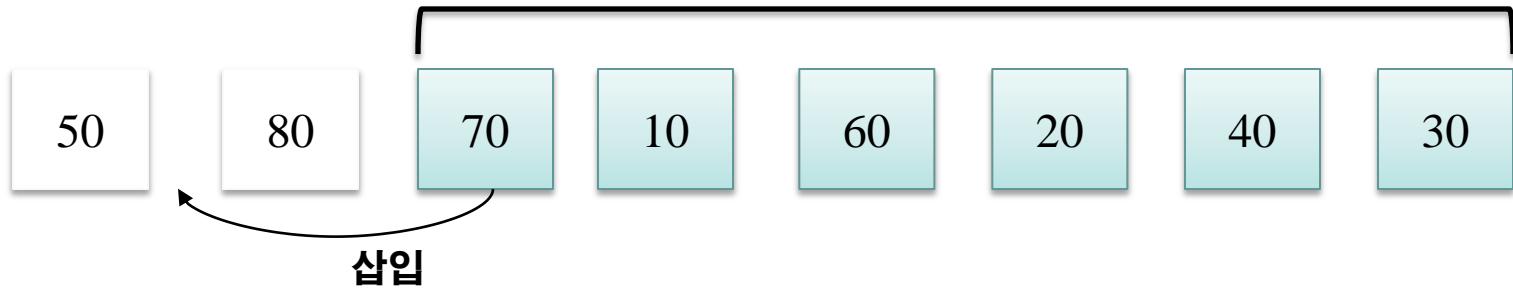
4. 퀵 정렬 (2/2)

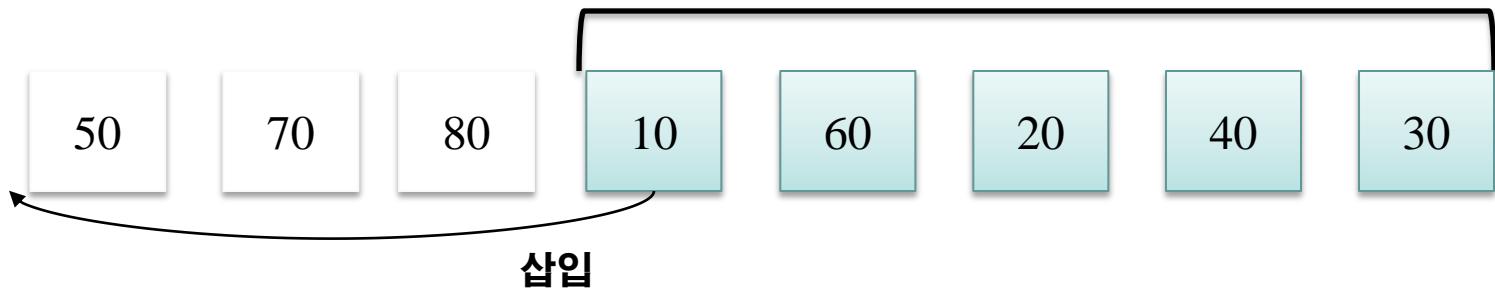
- 특성
 - 비교 연산 횟수
 - $O(n \log_2 n)$
 - 이동 연산 횟수
 - 최선, 평균: $O(n \log_2 n)$
 - 최악: $O(n^2)$
 - 최선, 평균, 최악
- 정렬의 안정성
 - 불안정 정렬
- Cf) 기존 자료 중 중간 값(Median)을 피봇으로 선택하여 퀵 연산을 수행하는 방법

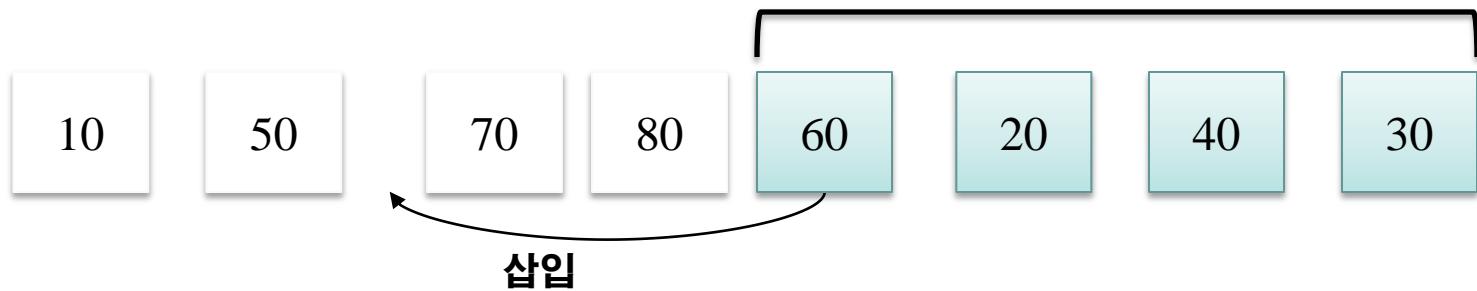
5. 삽입 정렬 (1/2)

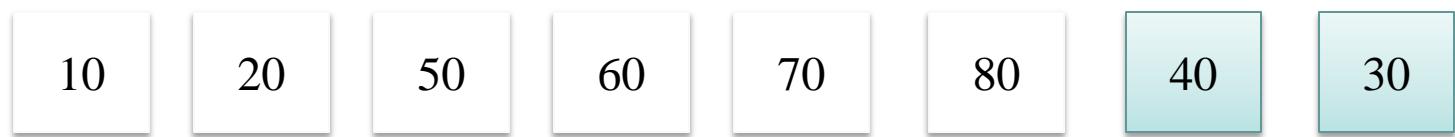
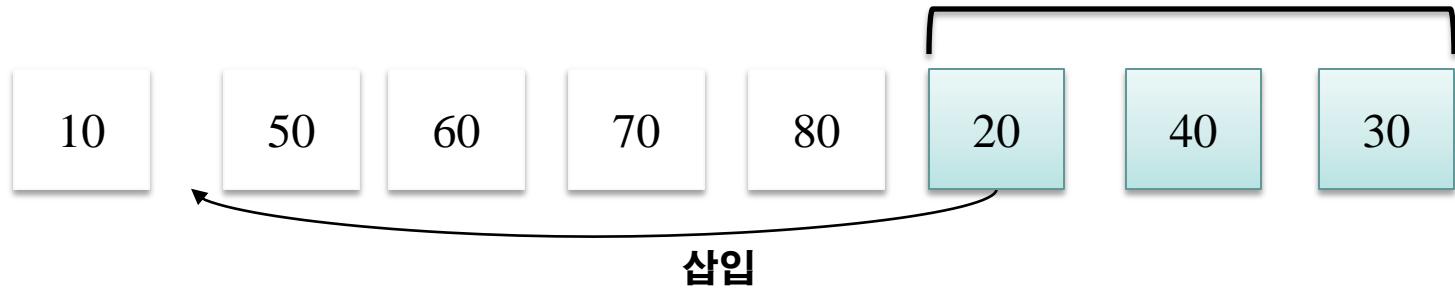
- 삽입 정렬(Insertion Sort)
 - 기존에 정렬된 부분 집합에 정렬할 자료의 위치를 찾아 삽입하는 정렬

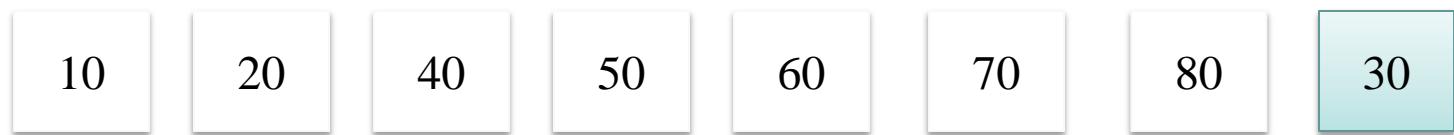
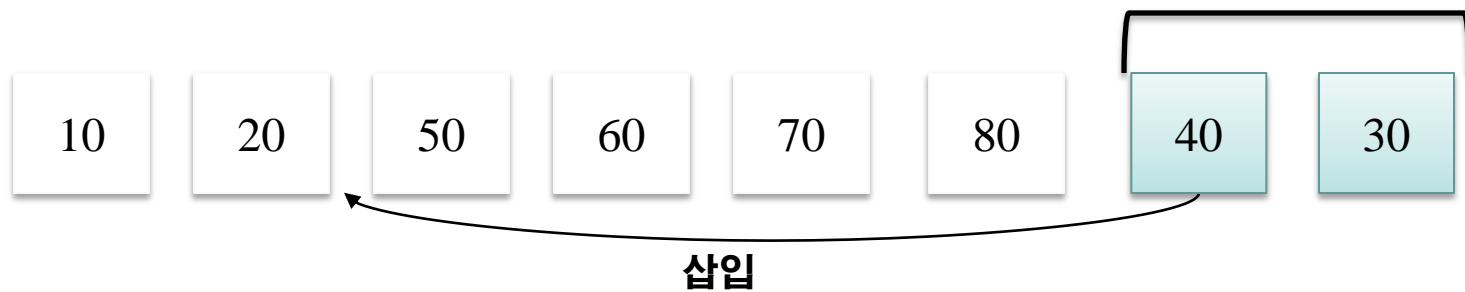


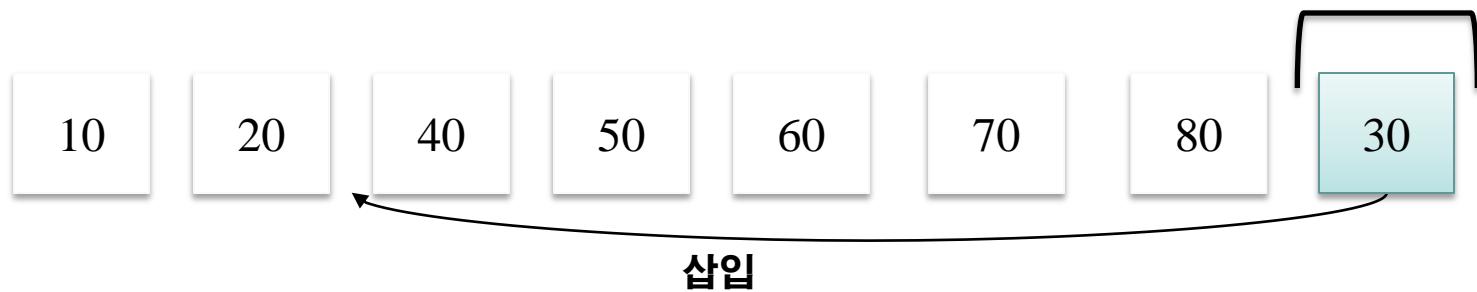










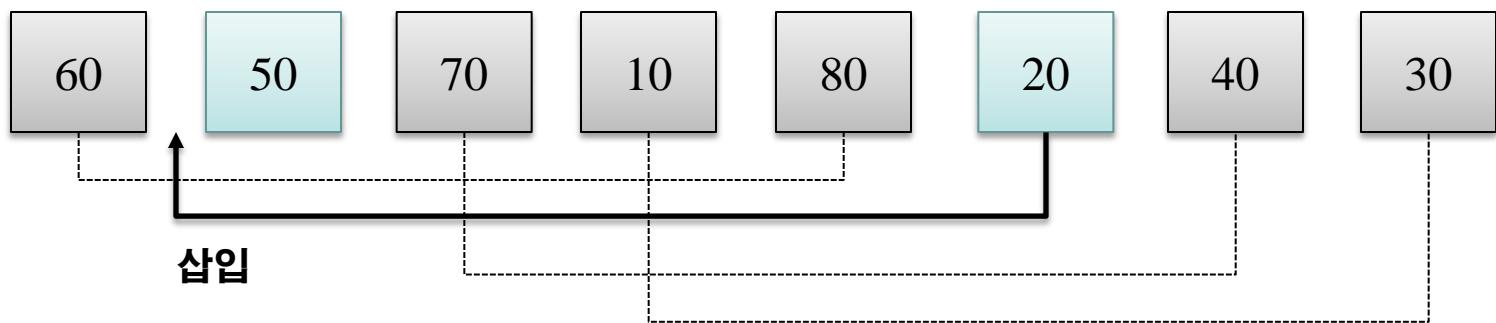
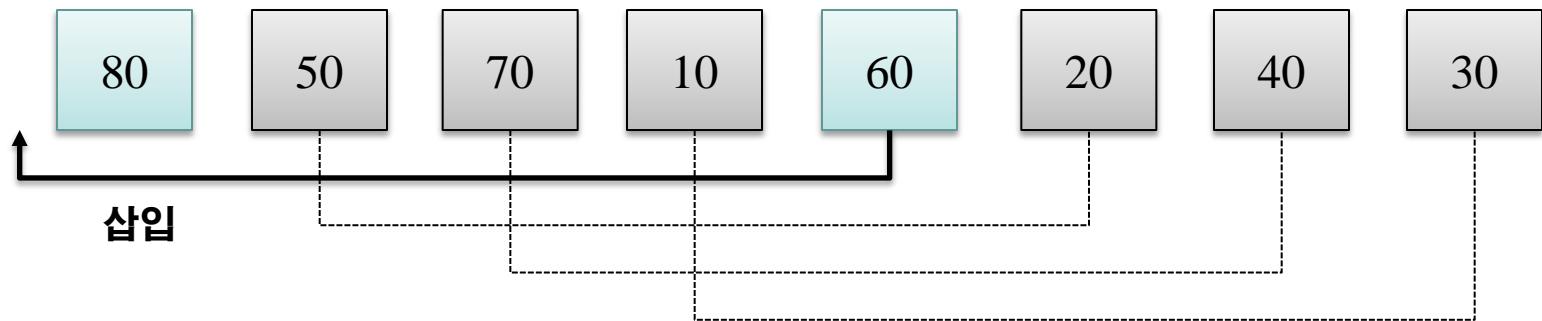
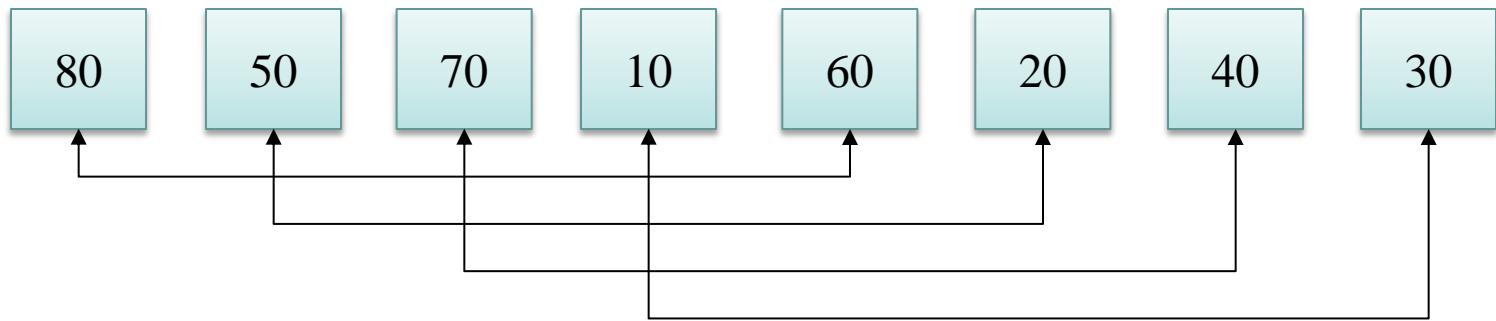


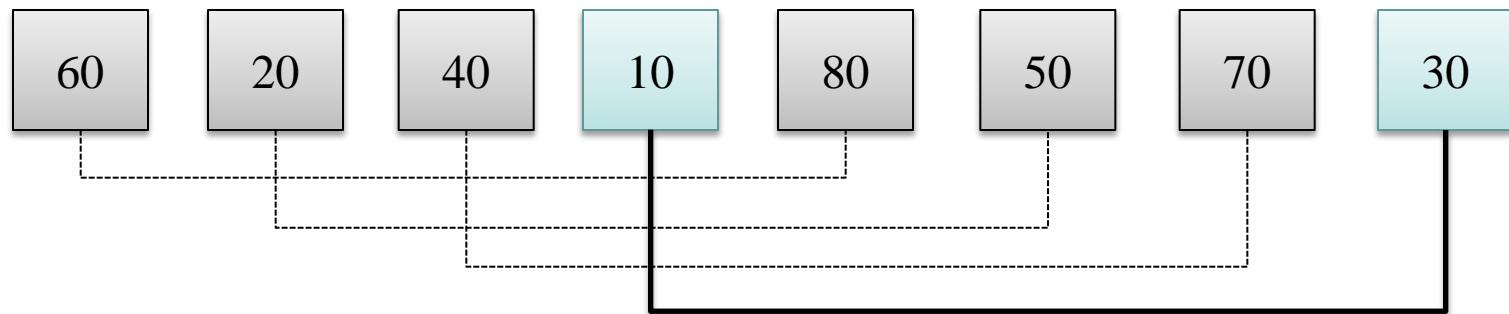
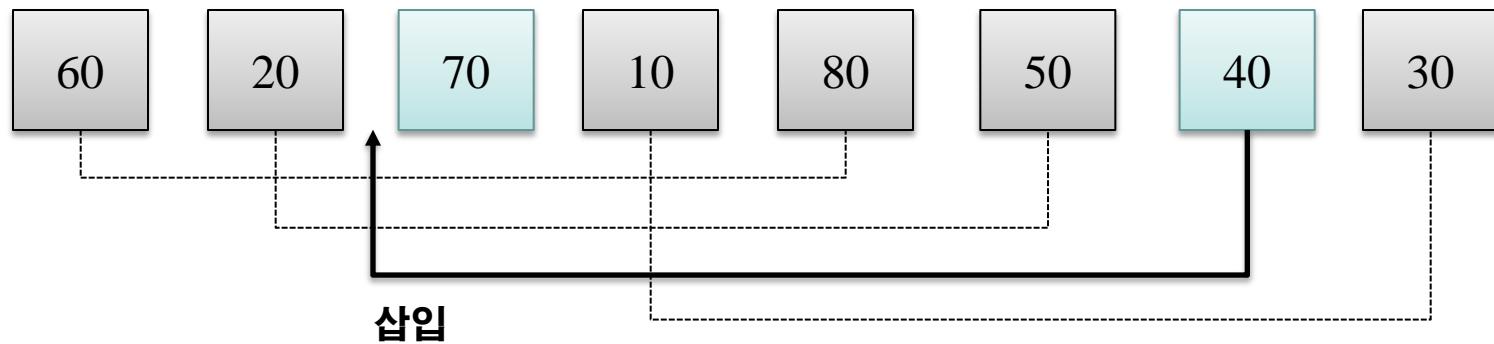
5. 삽입 정렬 (2/2)

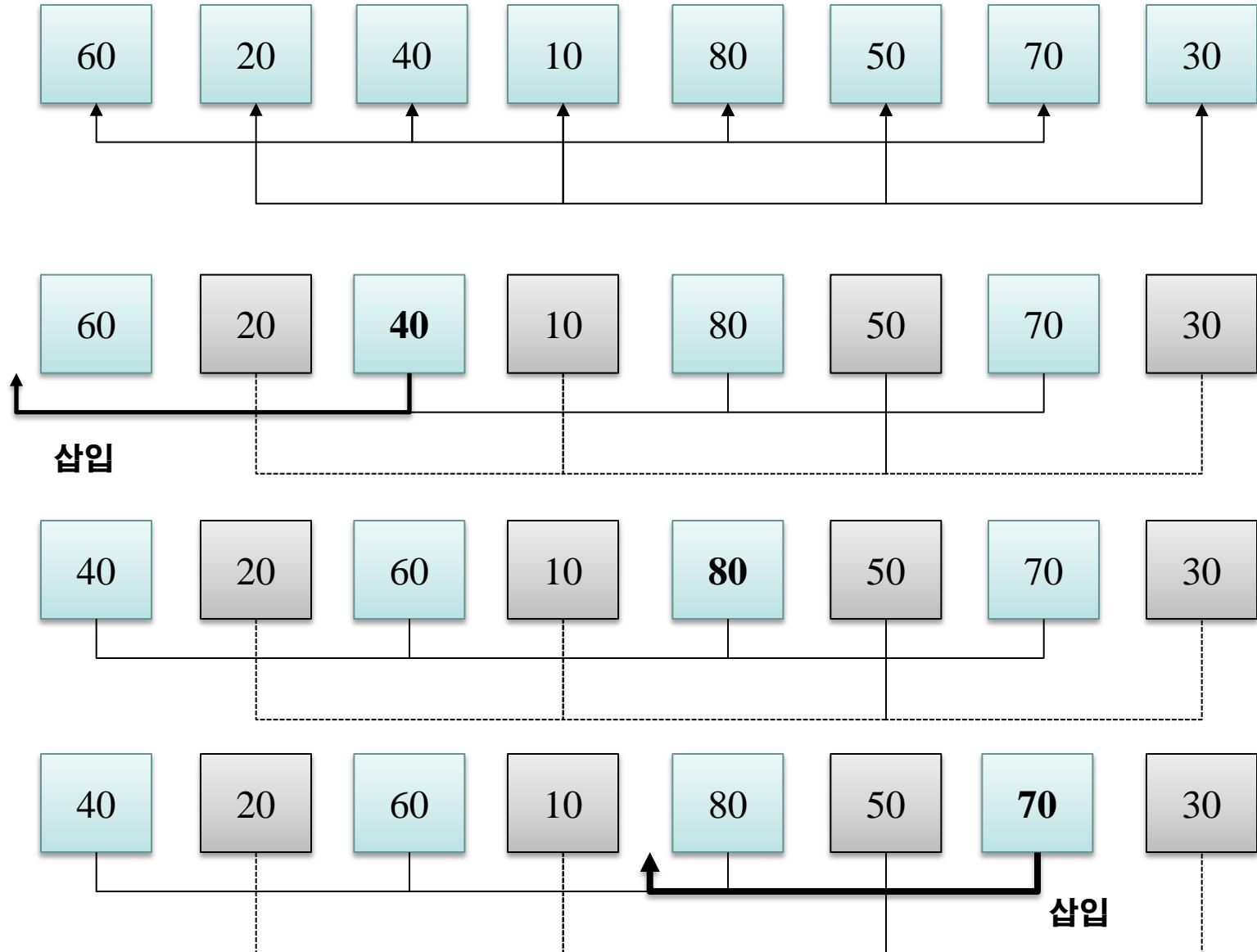
- 특성
 - 비교 연산 횟수
 - $O(1 + 2 + 3 + \dots + (n-2) + (n-1)) = n(n-1) / 2 = O(n^2)$
 - 이동 연산 횟수
 - 최선: $O(n)$
 - 평균 및 최악 $O(n^2)$
 - 최선, 평균, 최악
 - 최선: $O(n)$
 - 평균 및 최악 $O(n^2)$
 - 정렬의 안정성
 - 안정성 유지

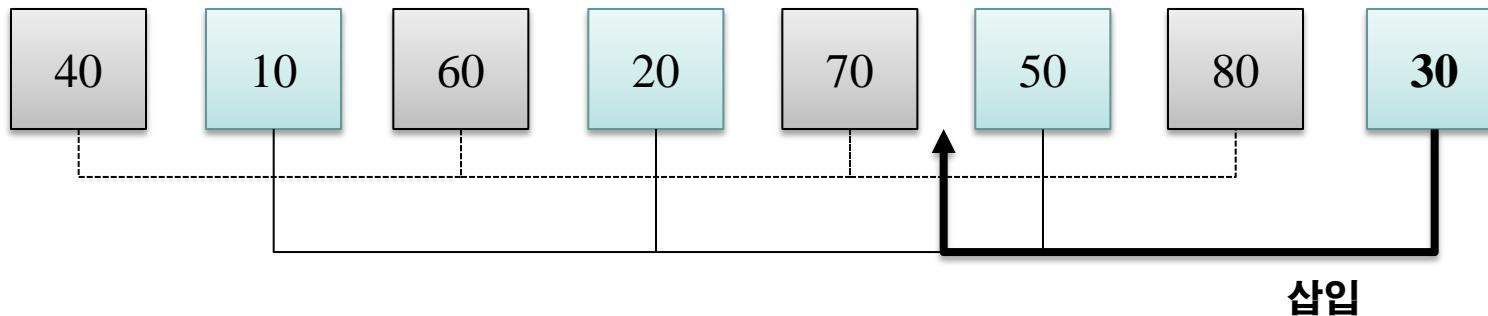
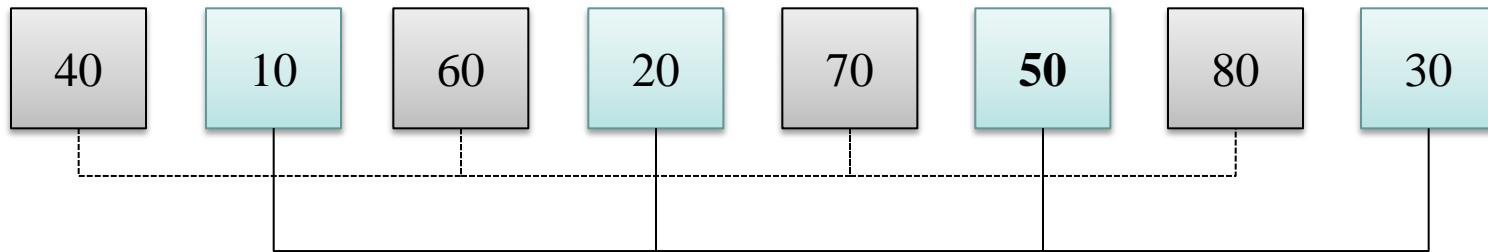
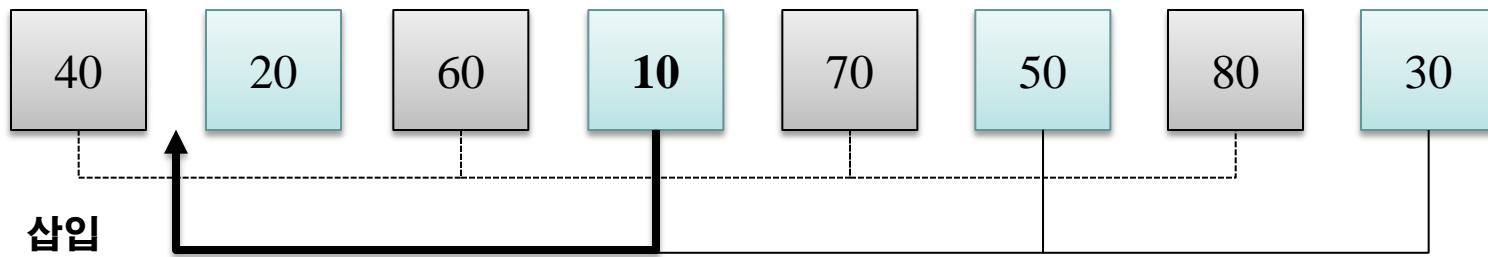
6. 셀 정렬 (1/2)

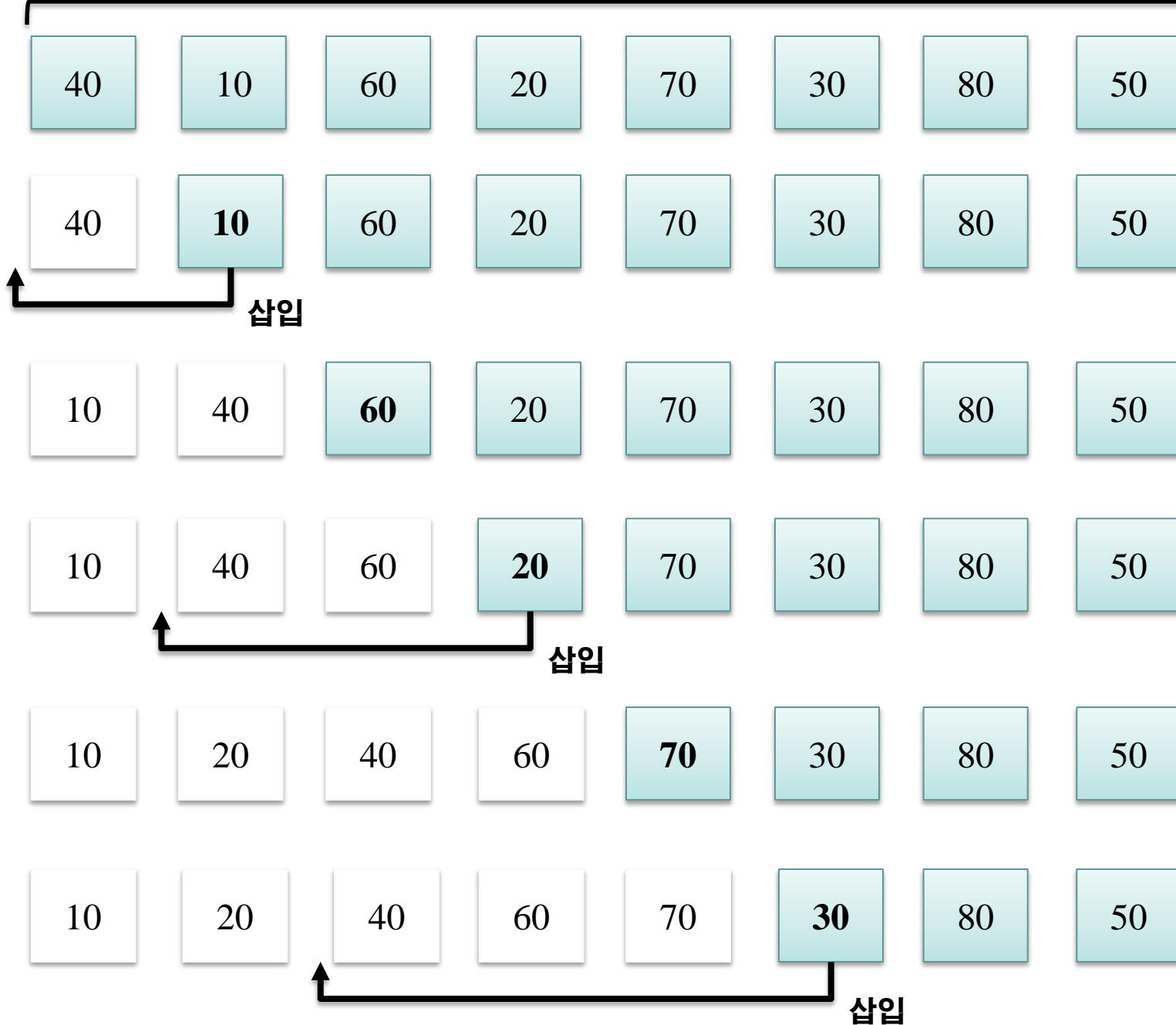
- 셀 정렬(Shell Sort)
 - 일정한 간격으로 여러 개의 부분 집합으로 나눈 다음, 각 정렬에 대해서 삽입 정렬을 수행

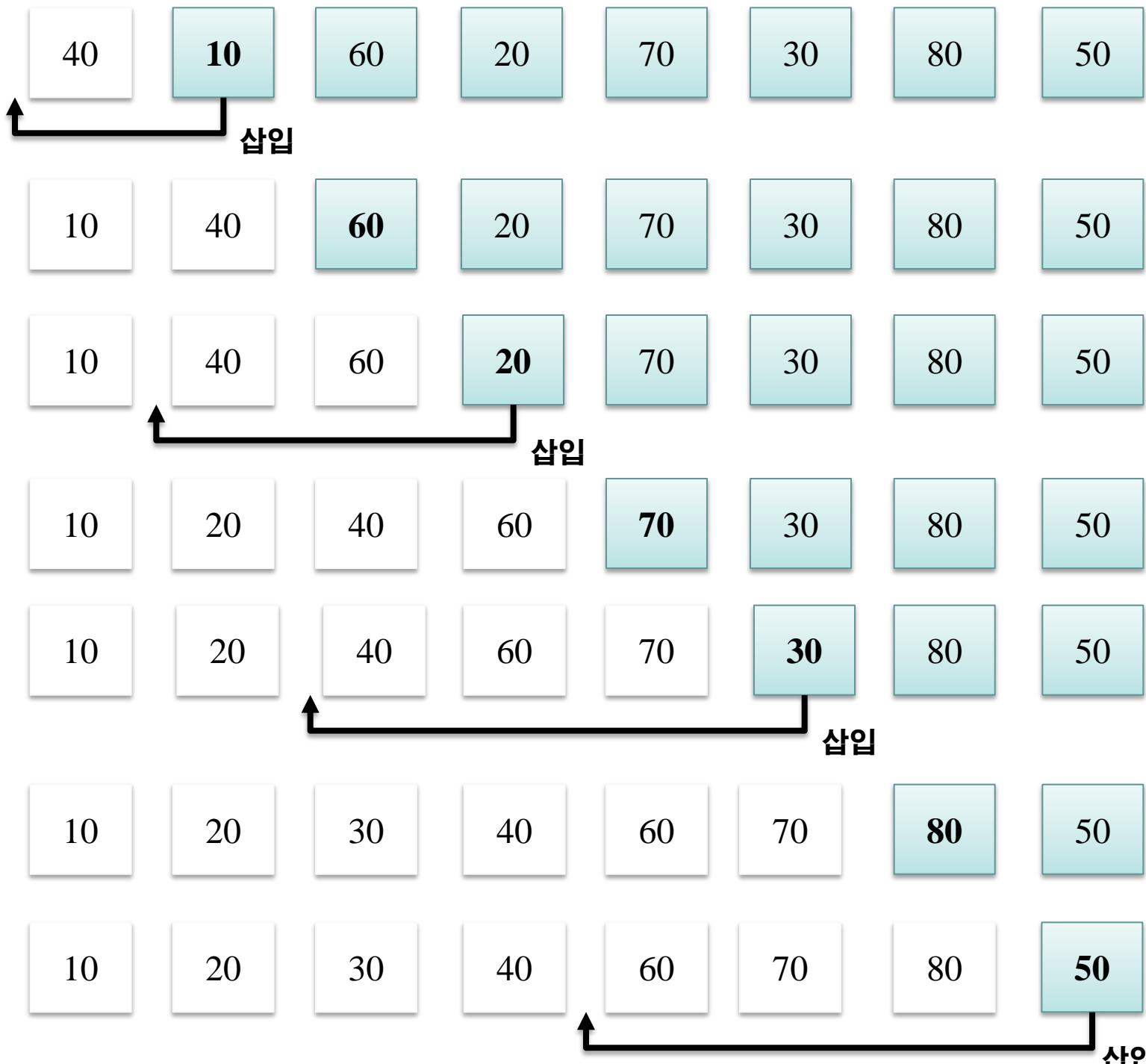


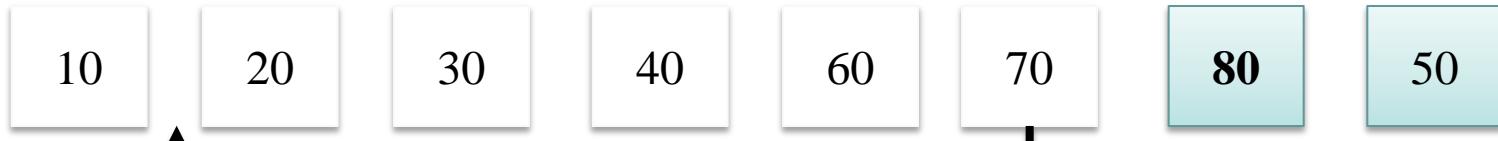




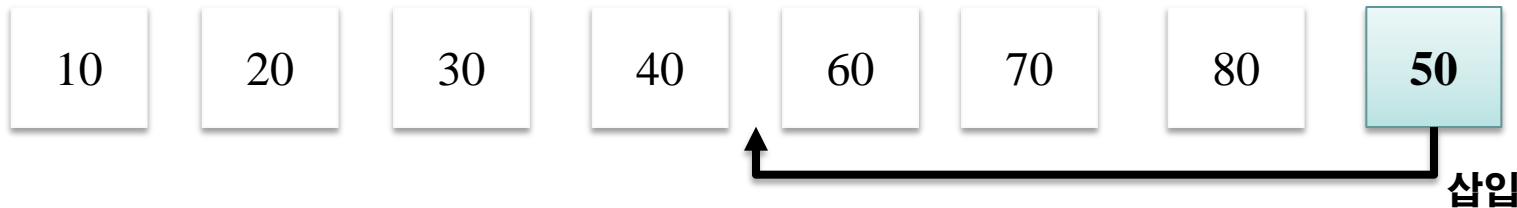




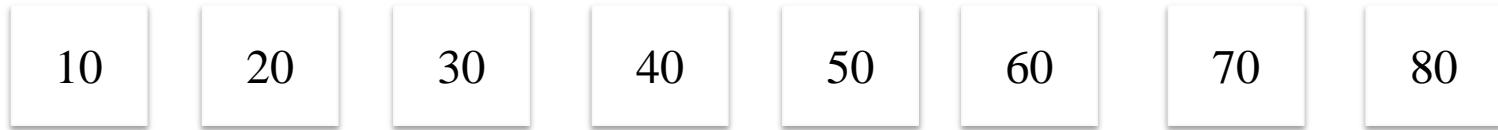




삽입



삽입



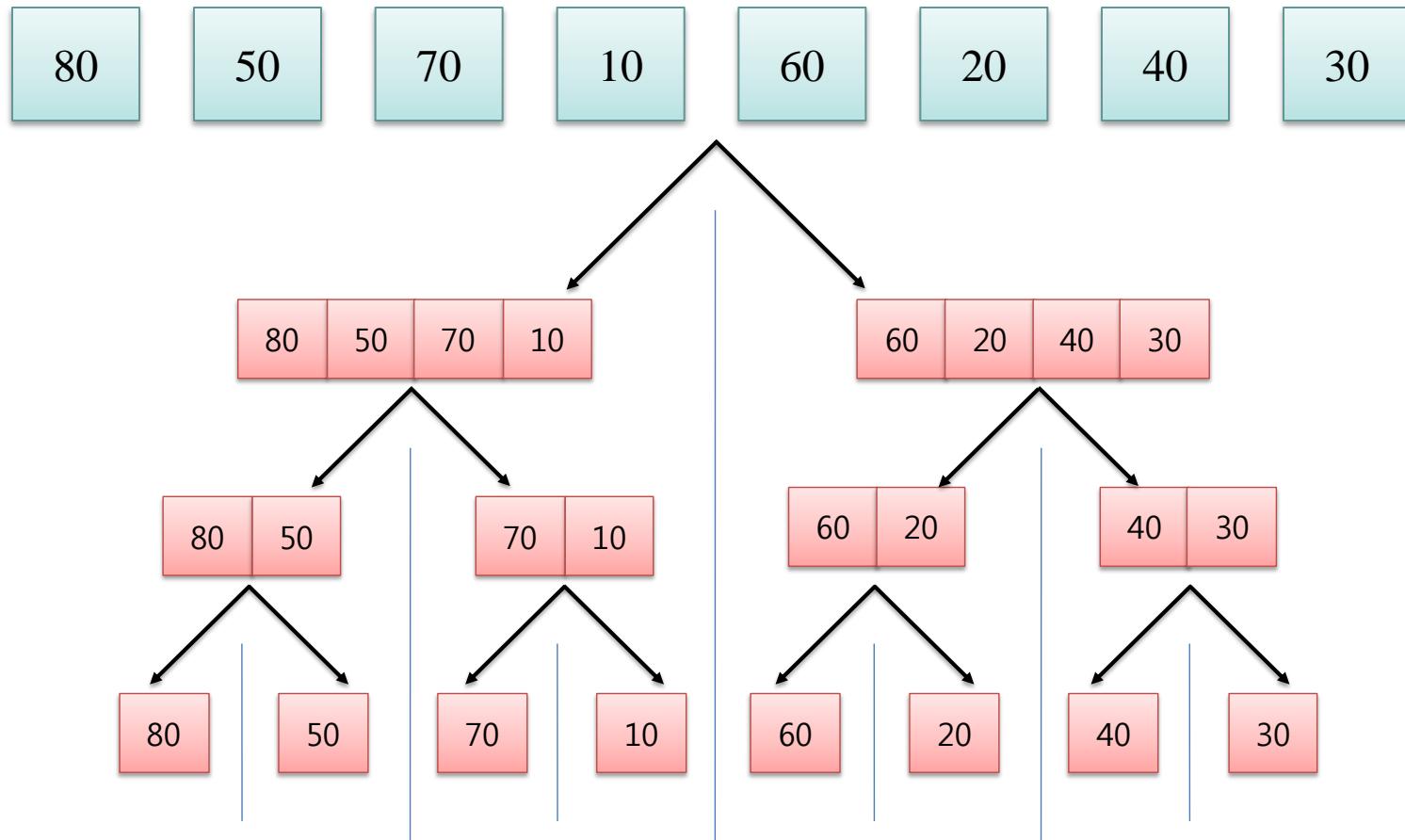
6. 셀 정렬 (2/2)

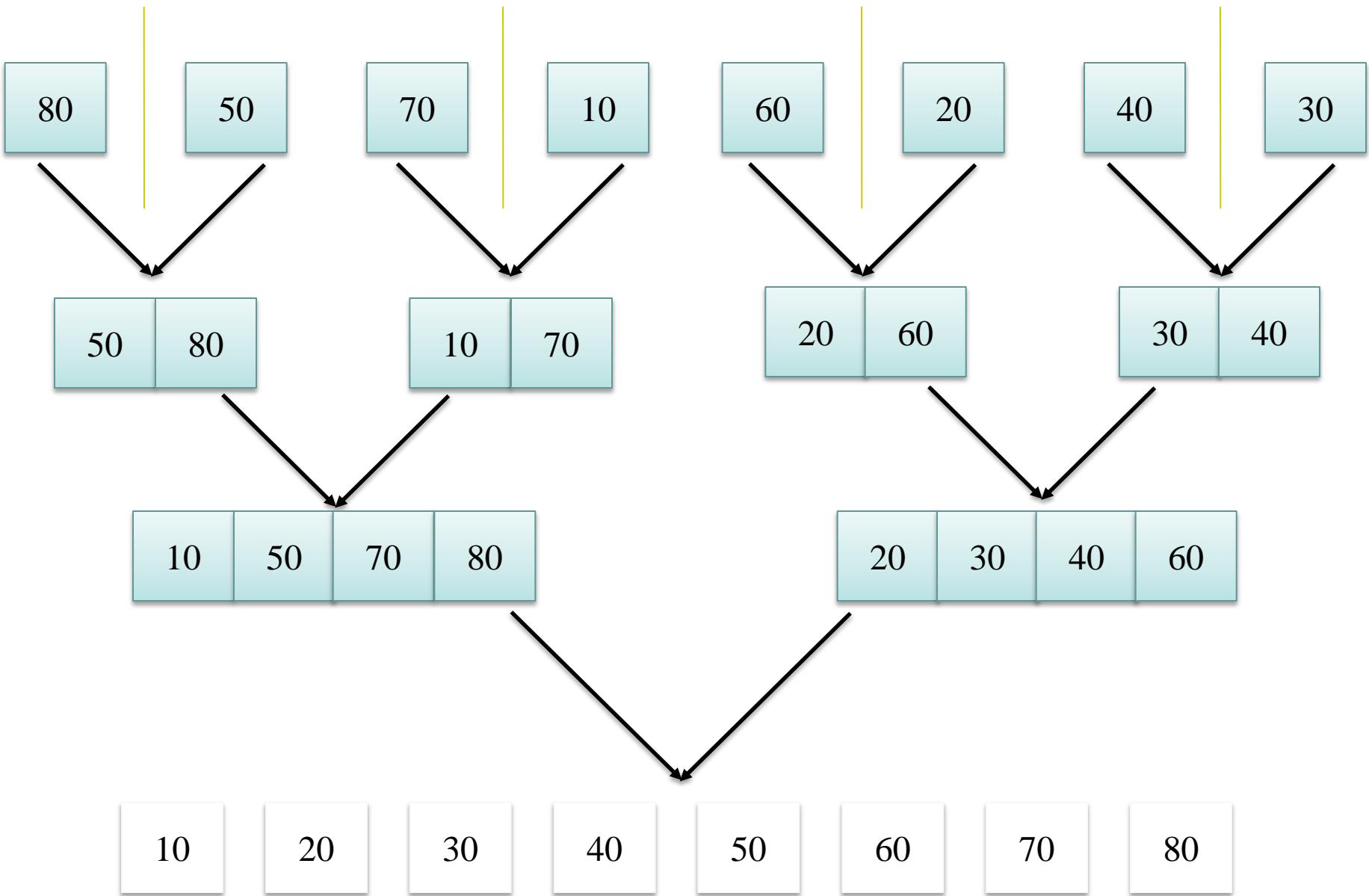
- 특성
 - 최선, 평균, 최악
 - 최선: $O(n)$
 - 평균: $O(n^{1.25})$
 - 최악: $O(n^2)$
 - 정렬의 안정성
 - 불안정 정렬

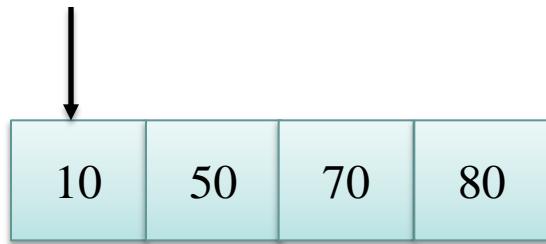
7. 병합 정렬 (1/2)

- 병합 정렬(Merge Sort)

- 같은 개수의 원소를 가지는 부분 집합으로 기존 자료를 분할(divide)하고 분할된 각 부분 집합을 병합

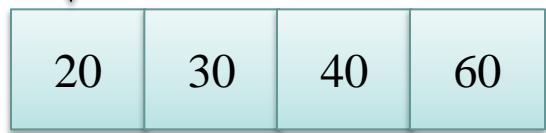






10	50	70	80
----	----	----	----

10



20	30	40	60
----	----	----	----

20 30 40 50 60 70 80



10	50	70	80
----	----	----	----



20	30	40	60
----	----	----	----



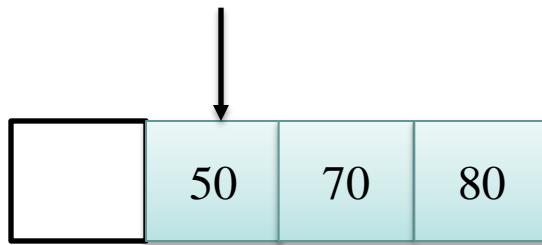
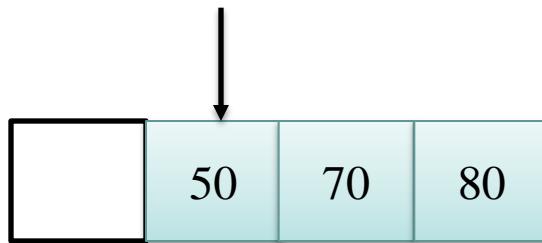
	50	70	80
--	----	----	----

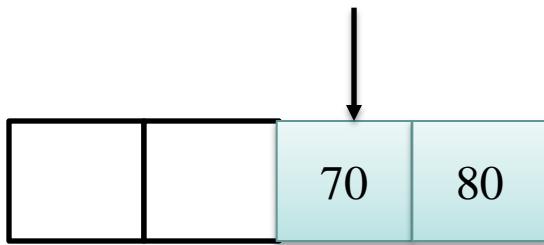
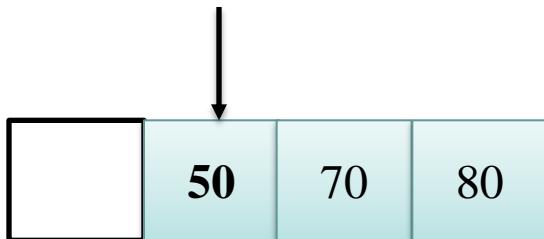


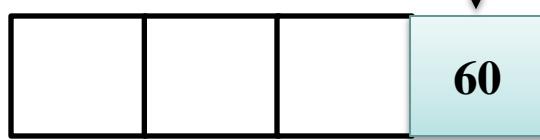
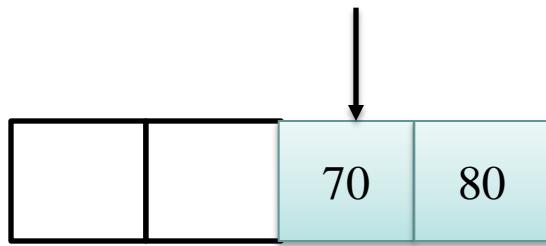
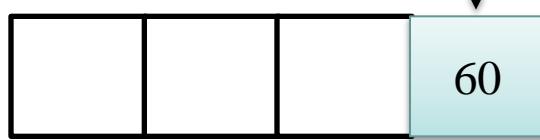
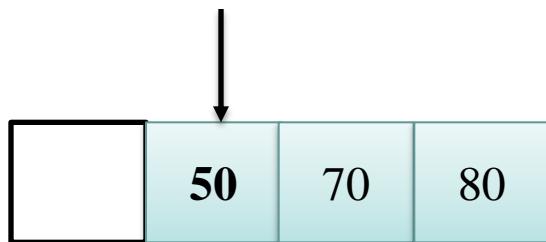
20	30	40	60
----	----	----	----

10

10 20







70

80

10

20

30

40

50

60

70



80

10

20

30

40

50

60

70

80

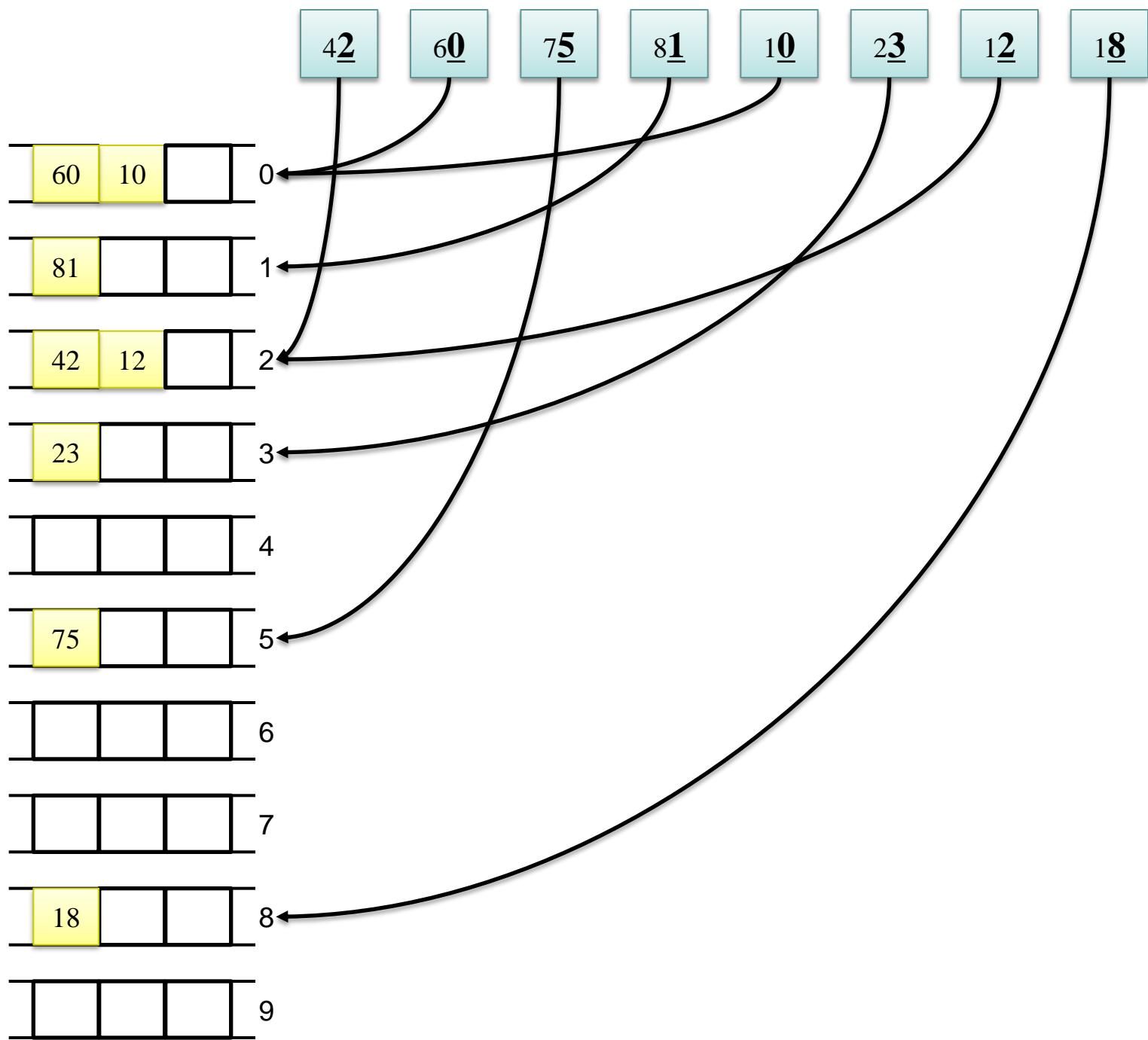


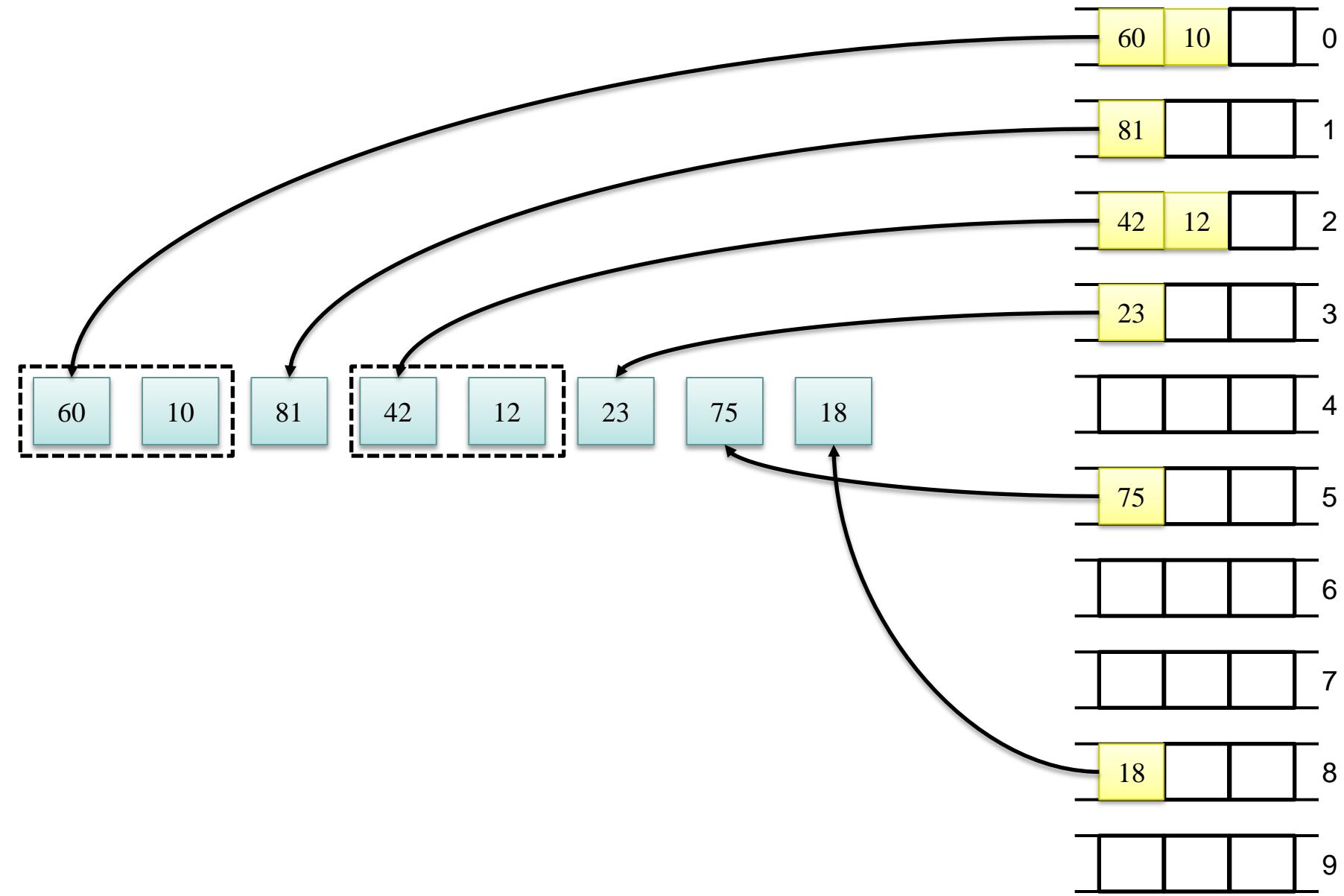
7. 병합 정렬 (2/2)

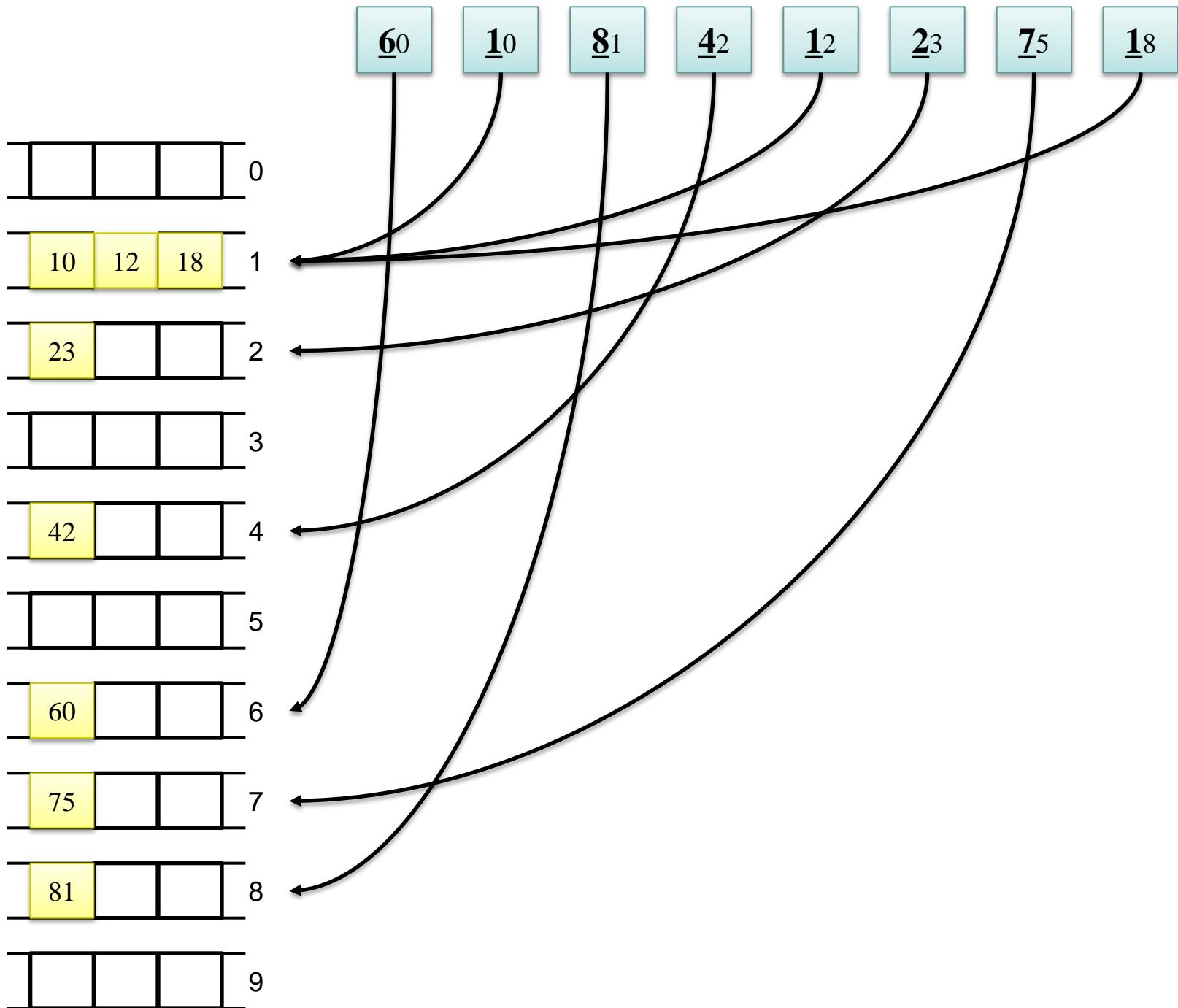
- 특성
 - 최선, 평균, 최악
 - 이동 및 비교 연산 횟수(최선, 평균, 최악): $O(n \log n)$
 - 최선, 평균, 최악: $O(n \log n)$
 - 추가 메모리 공간이 필요하며, 자료의 이동 횟수가 많다는 단점
 - 연결 리스트 이용
 - 정렬의 안정성
 - 안정성 유지

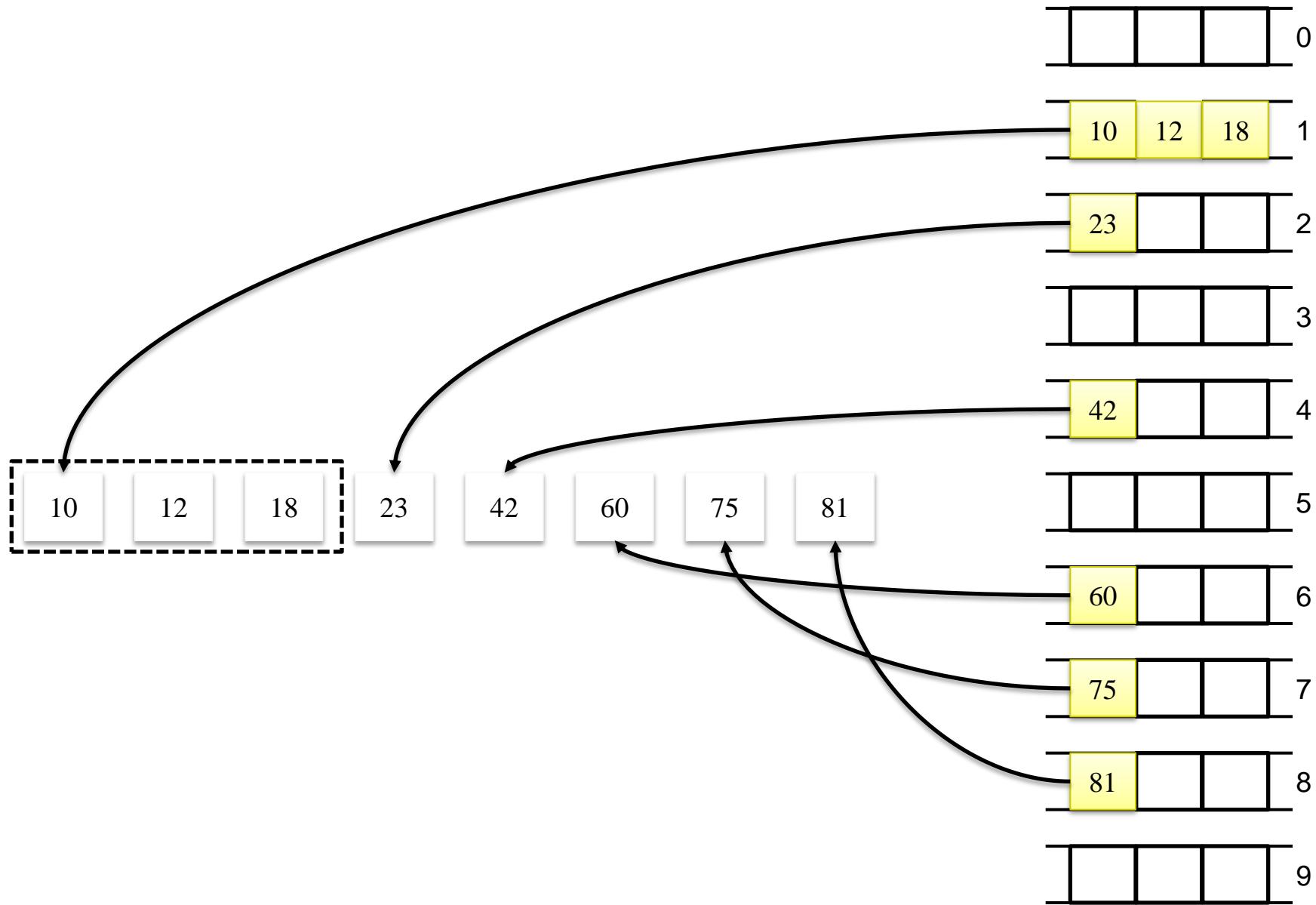
8. 기수 정렬 (1/3)

- 기수 정렬(Radix Sort)
 - 기수(Radix): 숫자의 자릿수
 - 키 값의 자릿수에 따라 자료를 분배(Distribution)하는 방식을 통해 정렬
 - 버킷(Bucket)이라 불리는 자료 보관 큐에 자료를 분배하고 다시 이를 꺼내는 연산







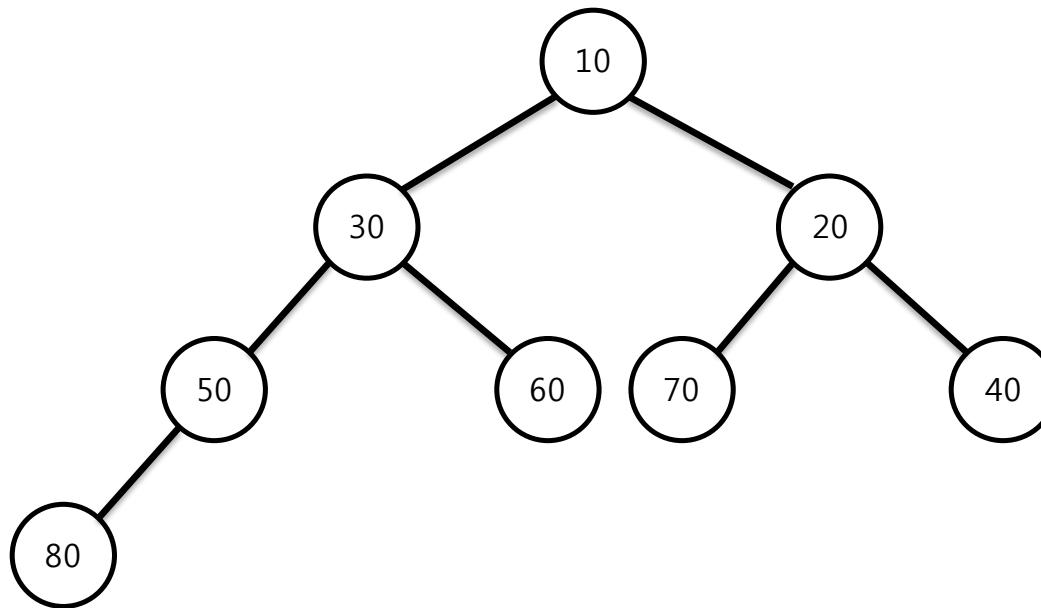


8. 기수 정렬 (3/3)

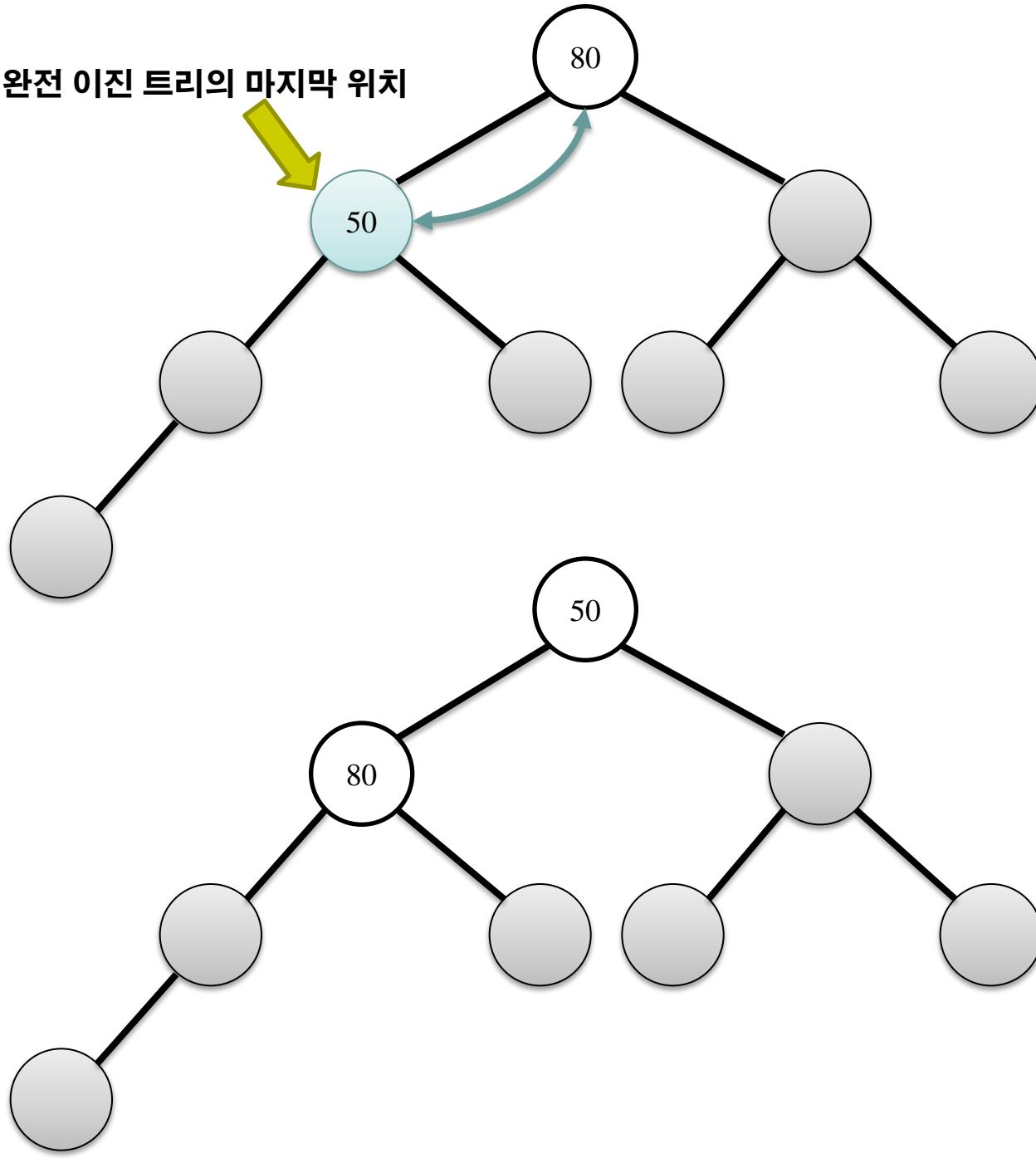
- 특성
 - 비교 연산 횟수
 - 없음
 - 이동 연산 횟수
 - 최선, 평균, 최악: $O(d*n)$
 - 최선, 평균, 최악
 - 최선, 평균, 최악: $O(d*n)$
 - 정렬의 안정성
 - 유지
 - 숫자 자료, 문자 자료?

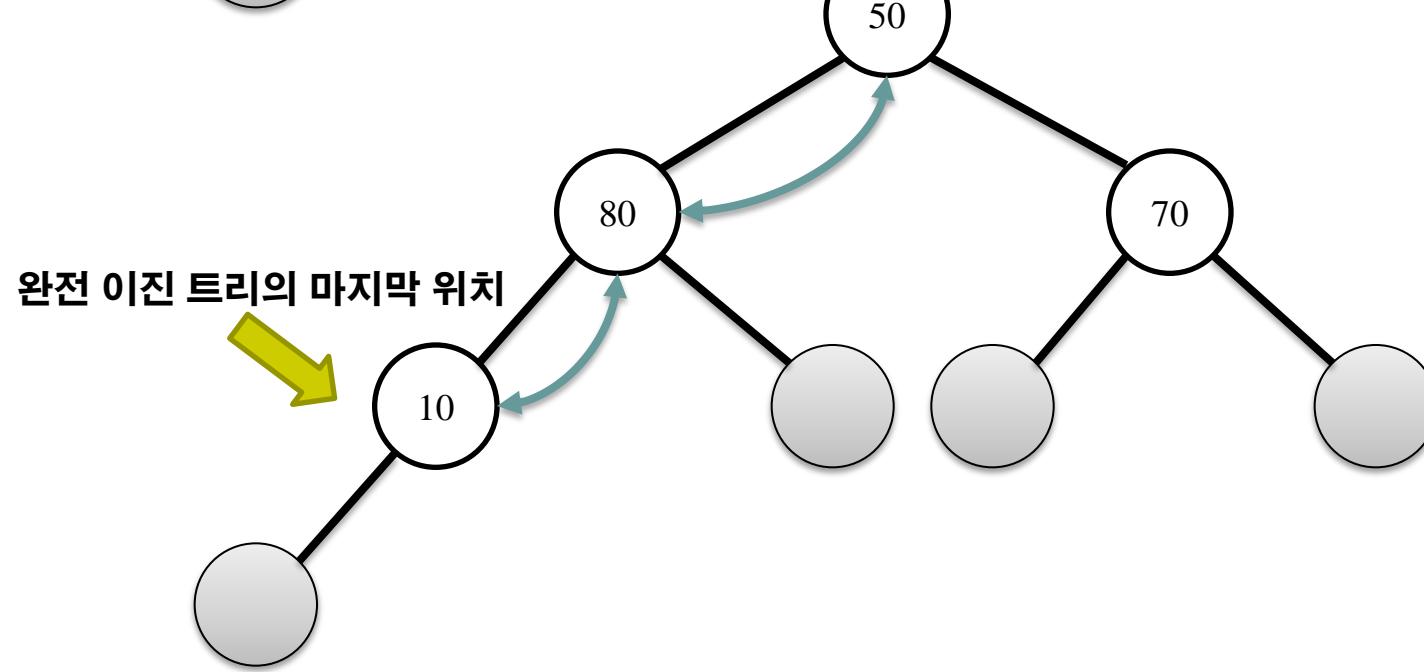
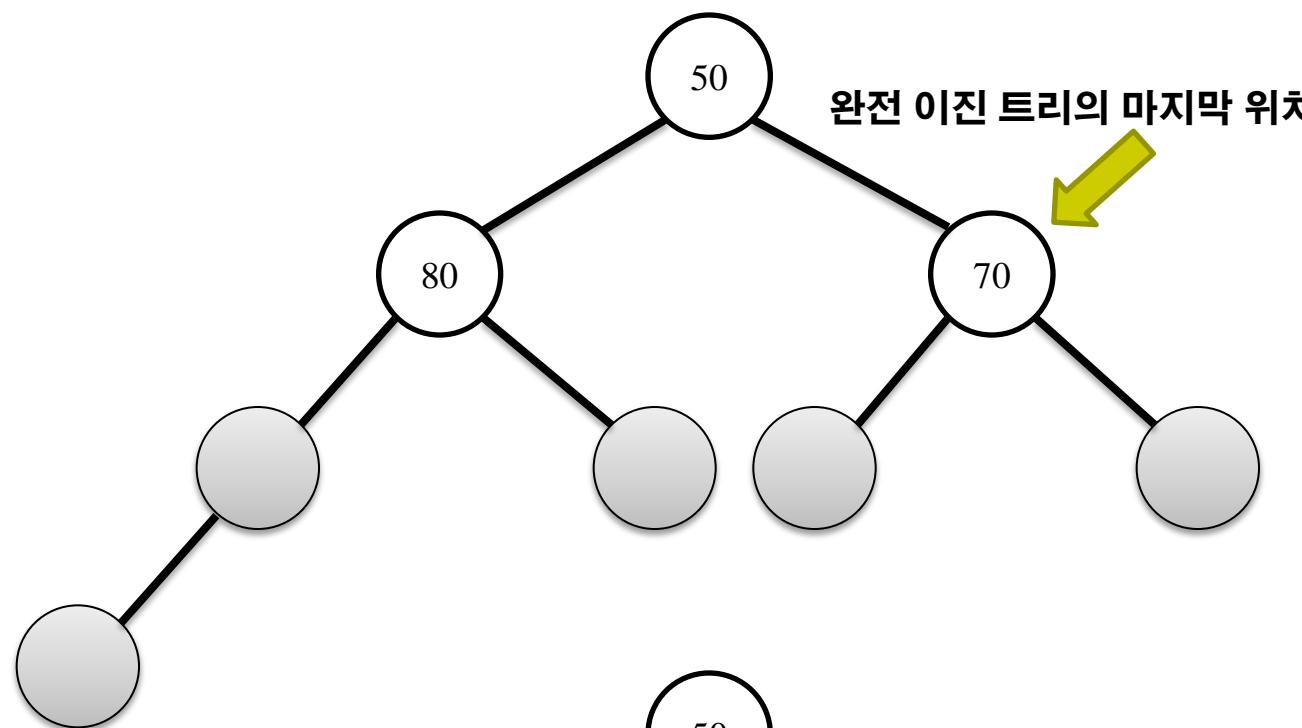
9. 히프 정렬 (1/2)

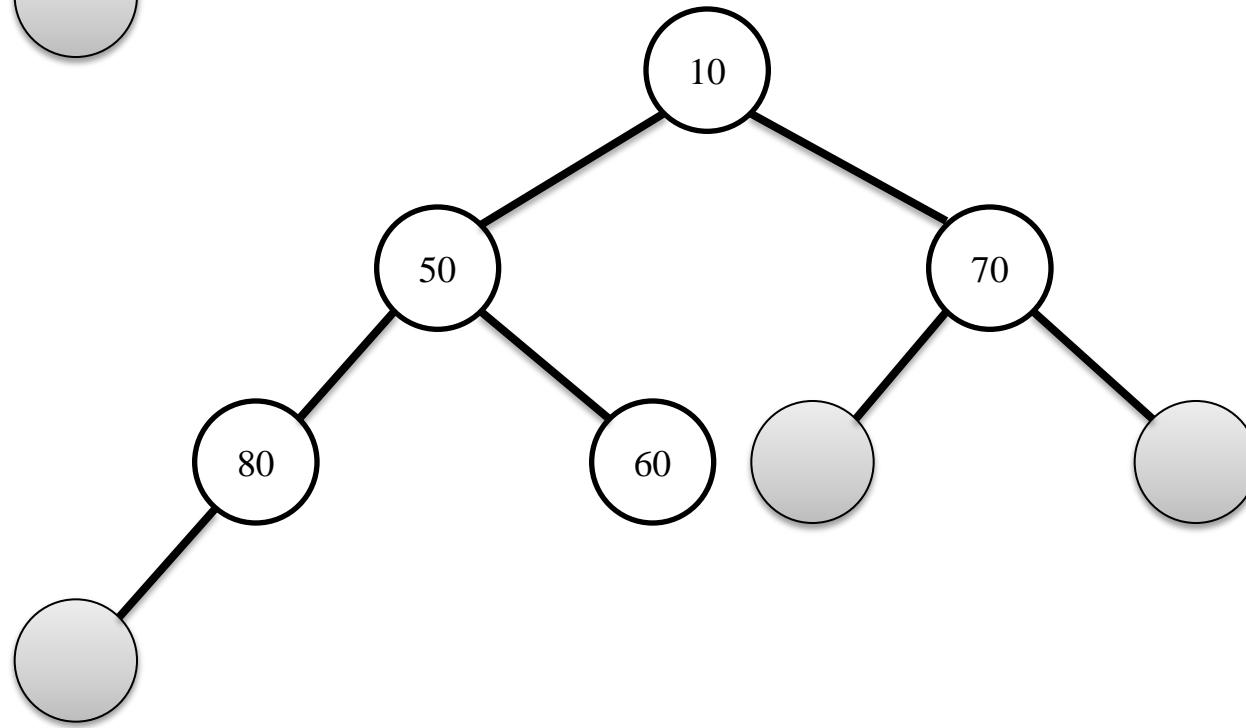
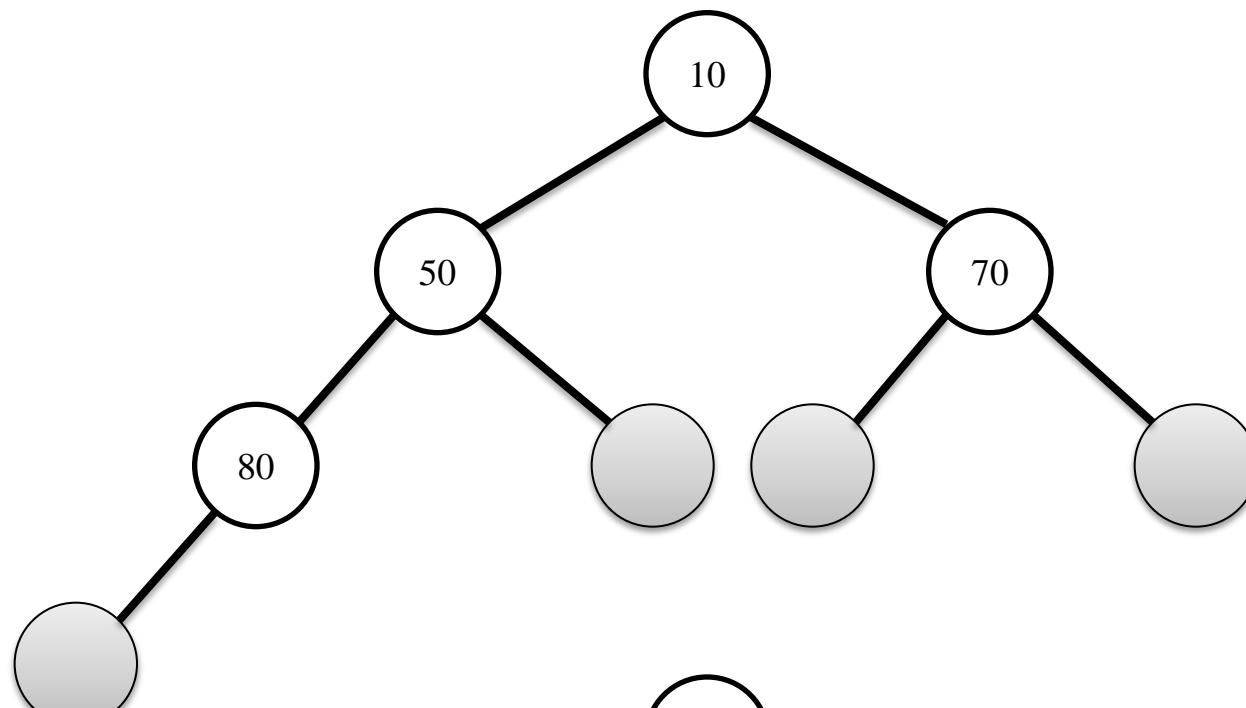
- 히프 정렬(Heap Sort)
 - 히프 자료 구조를 이용하여 정렬

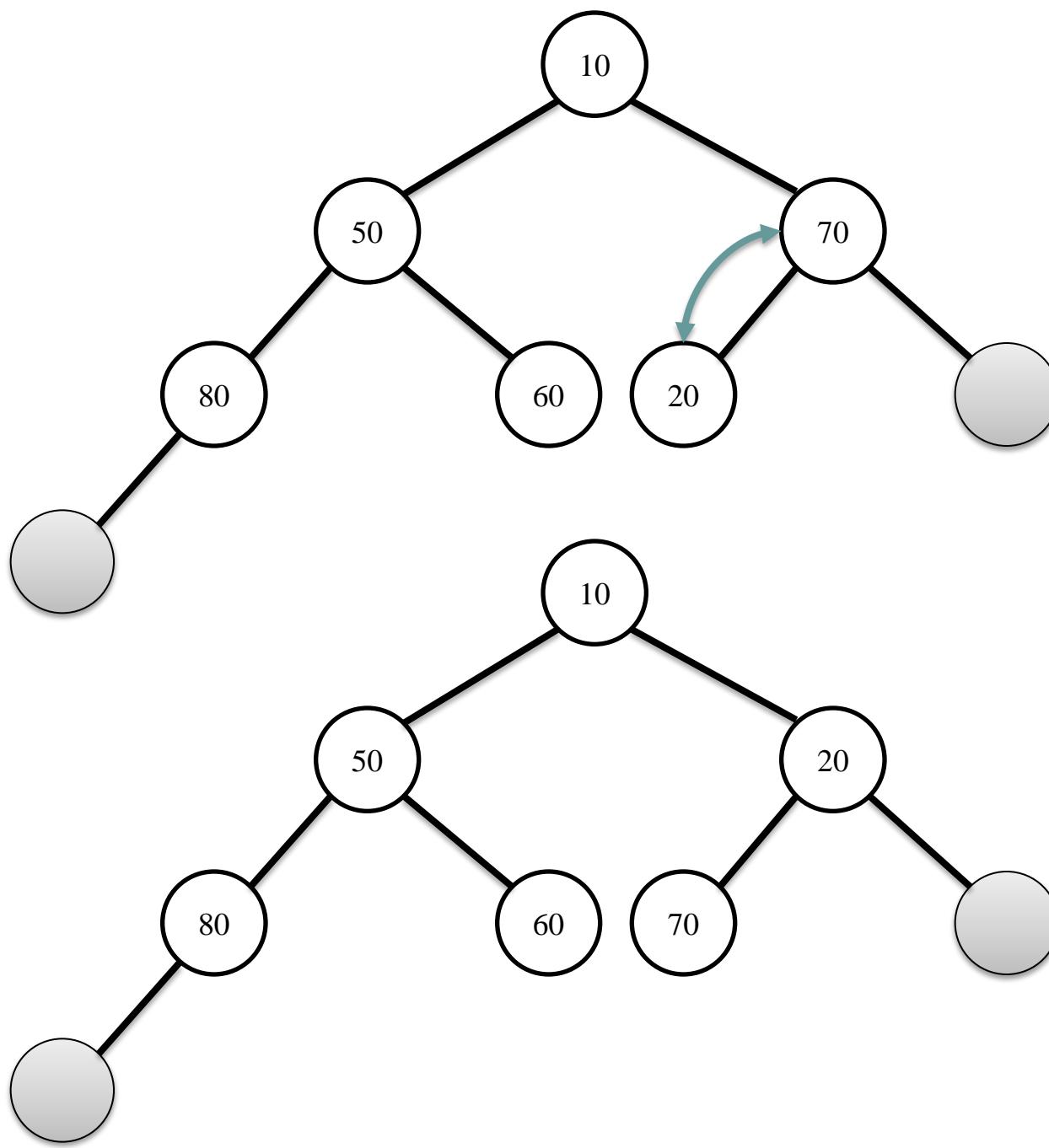


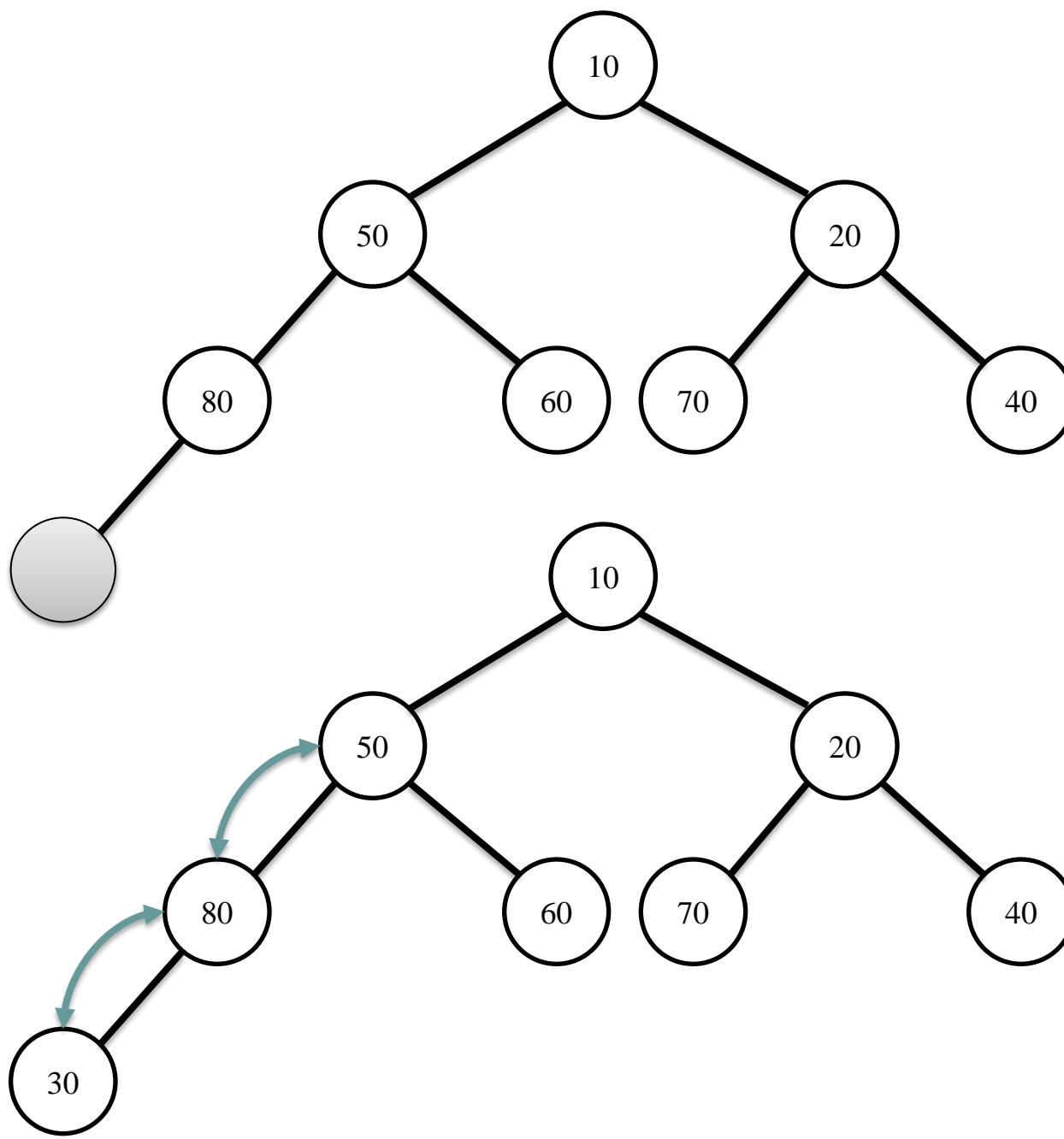
완전 이진 트리의 마지막 위치

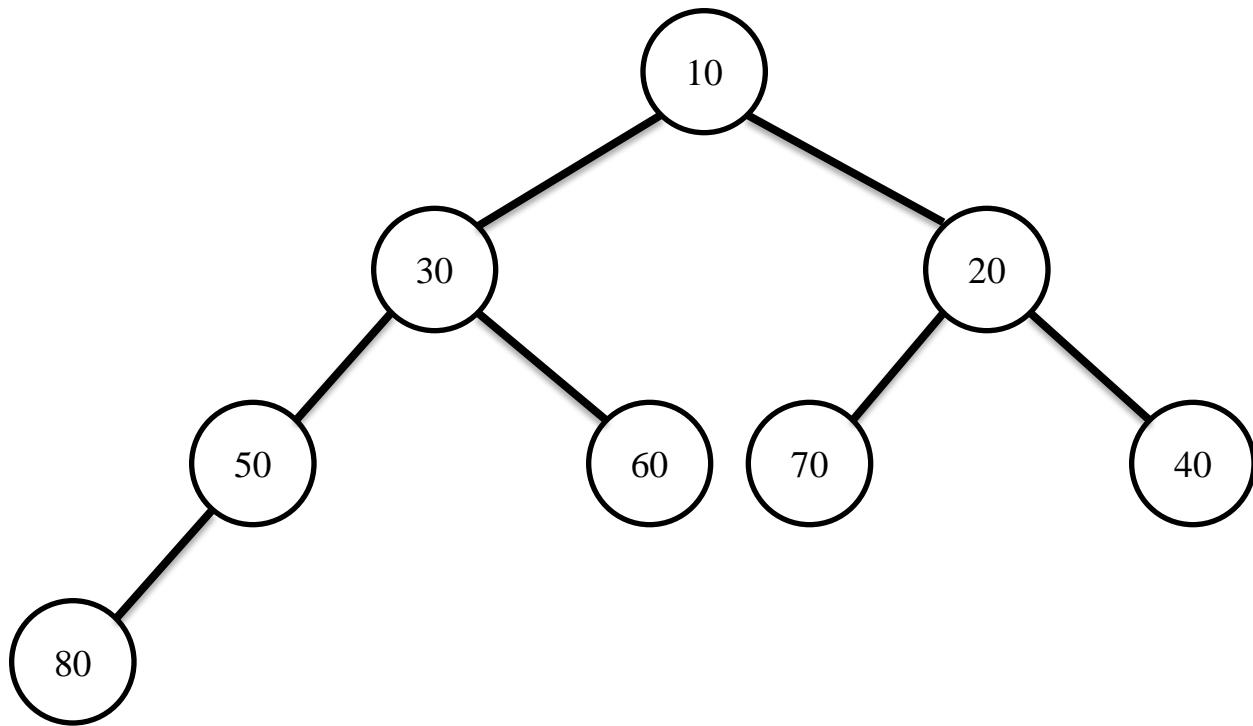


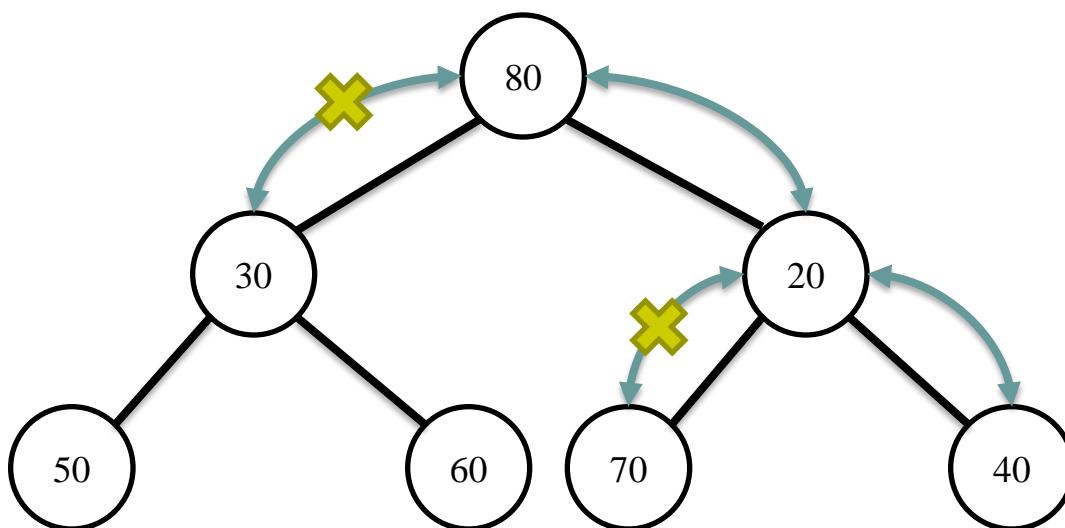
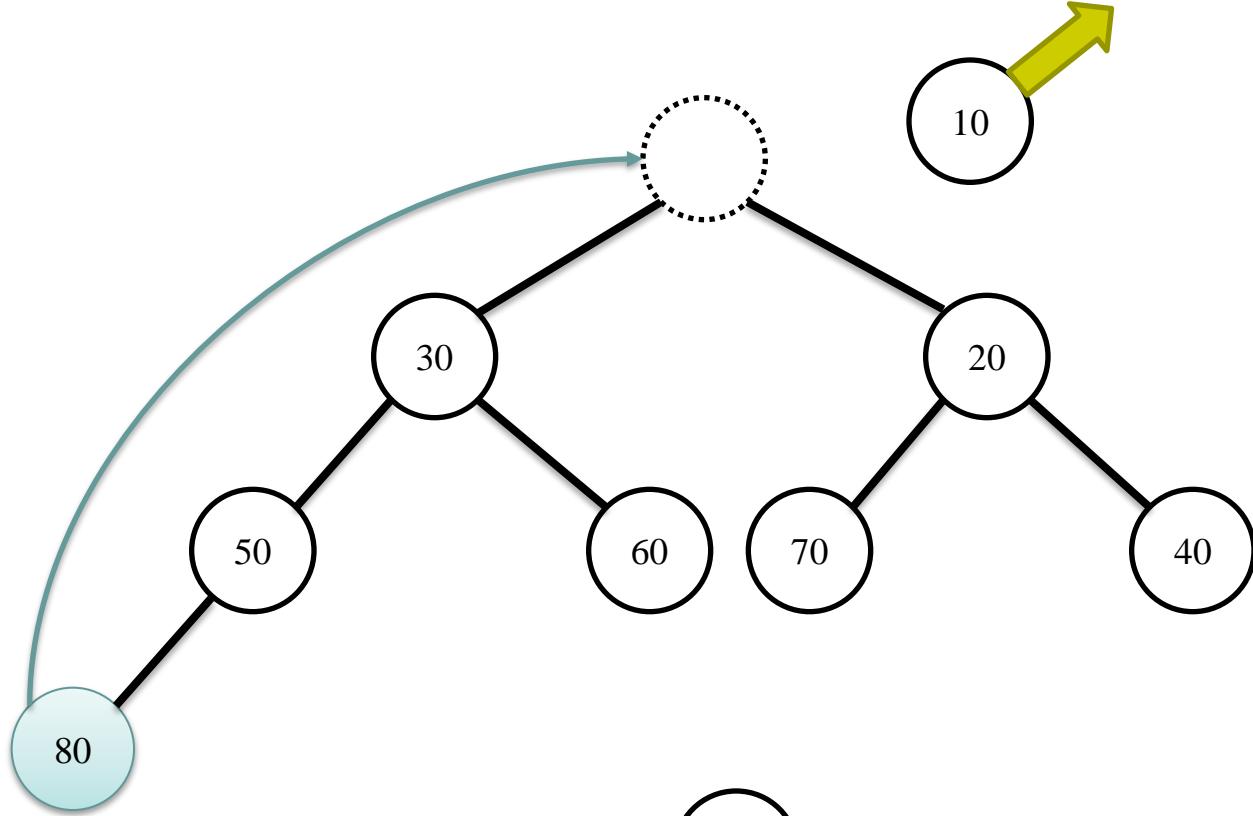


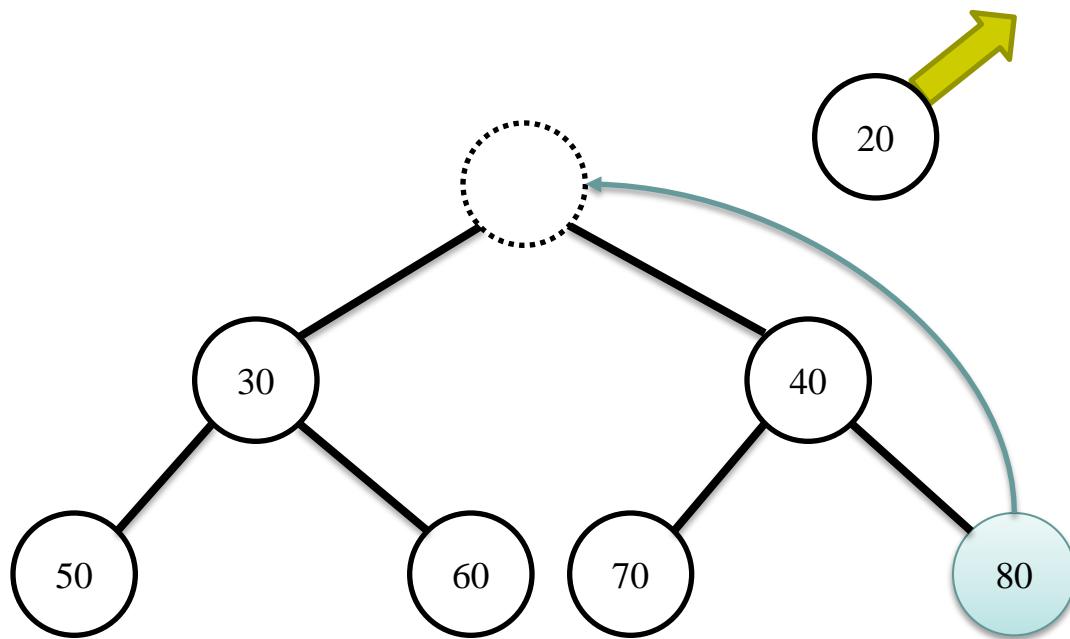
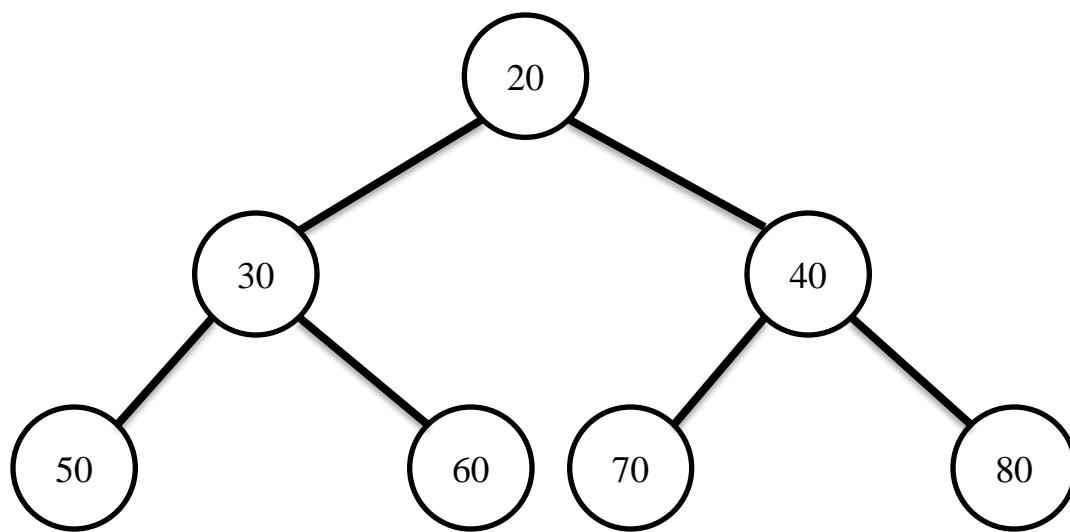


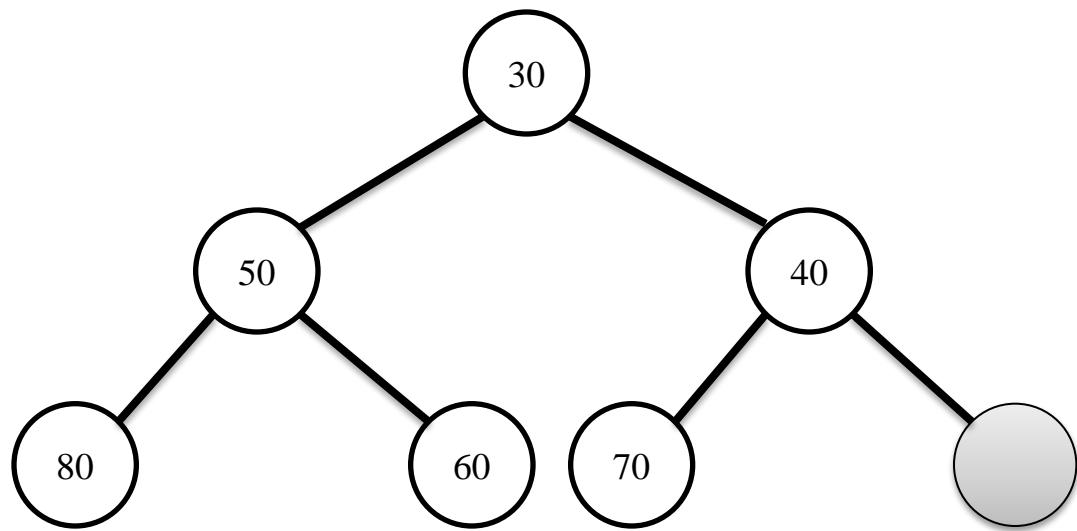
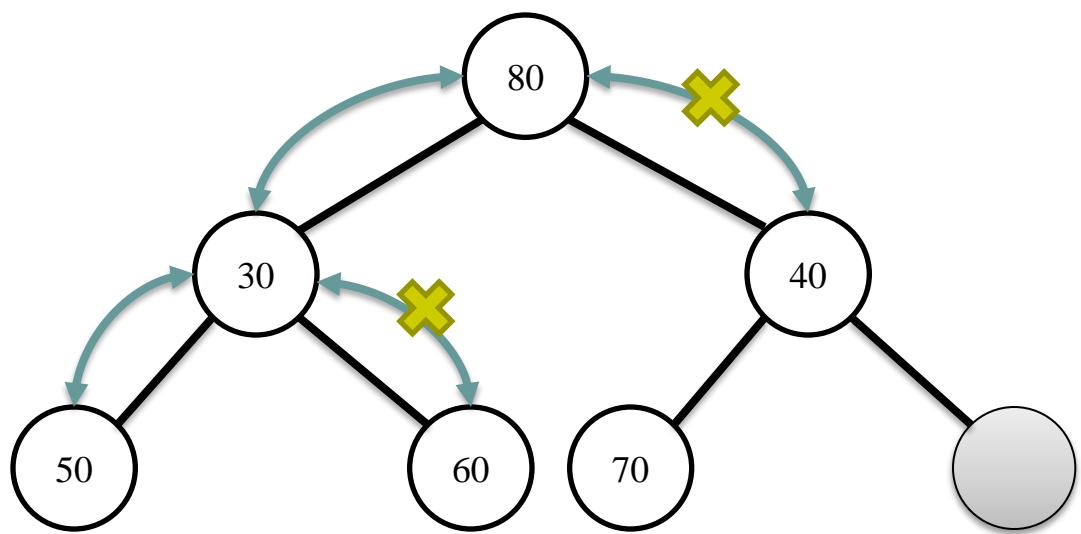


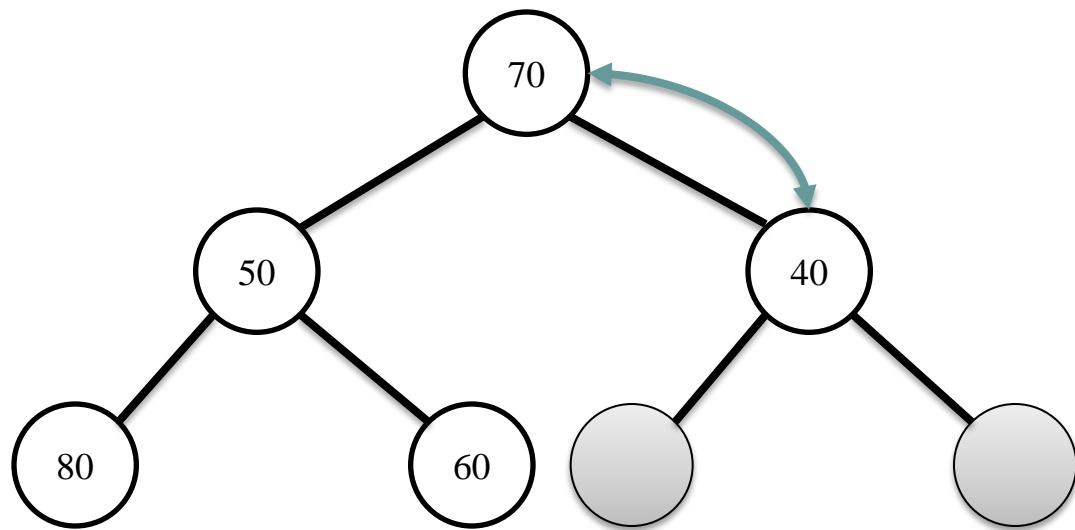
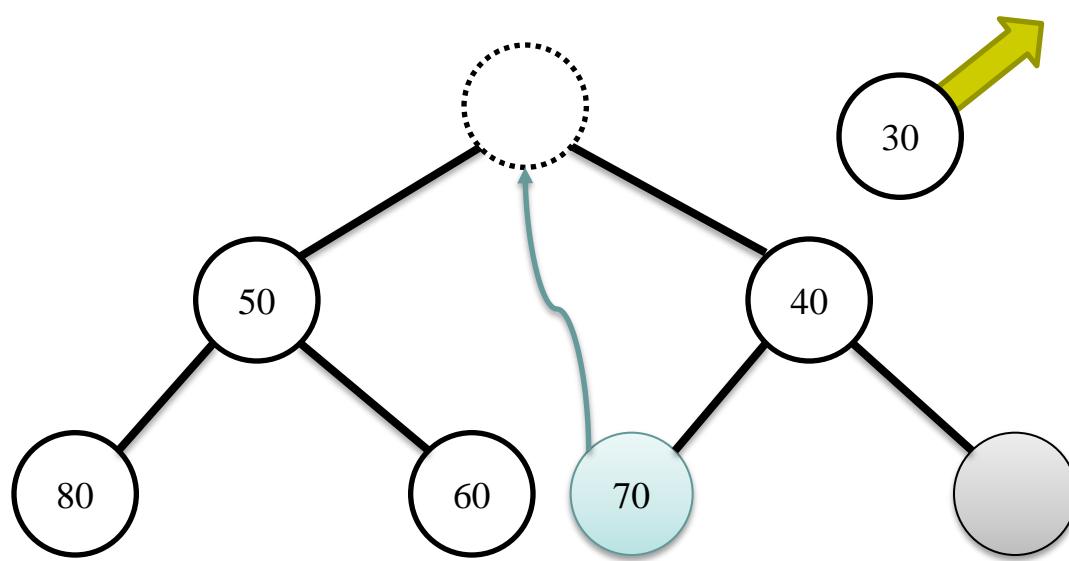


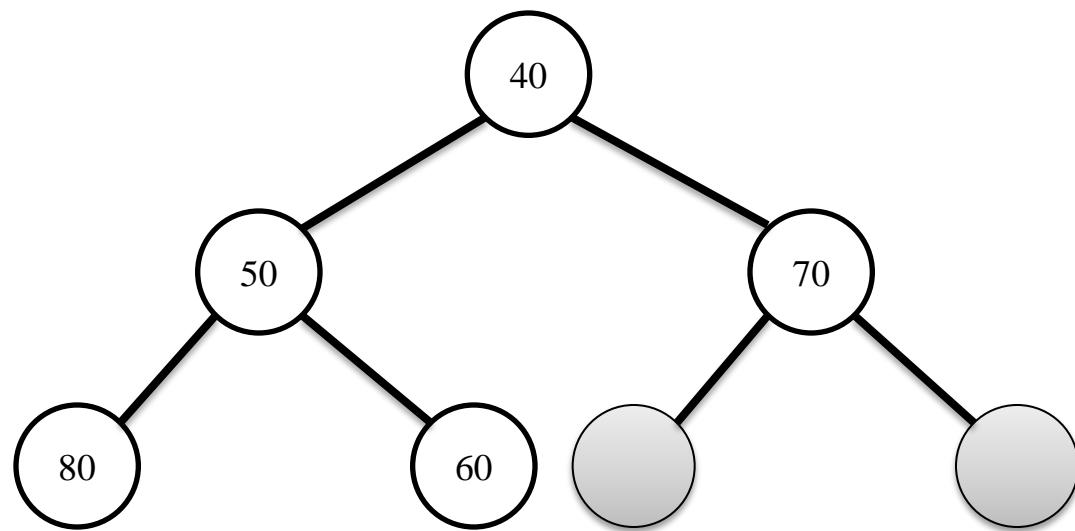


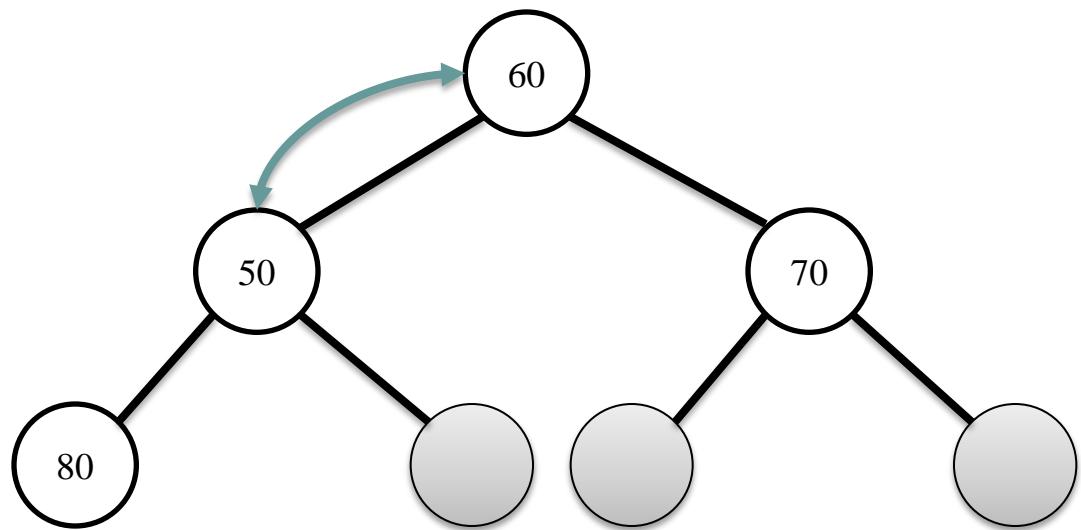
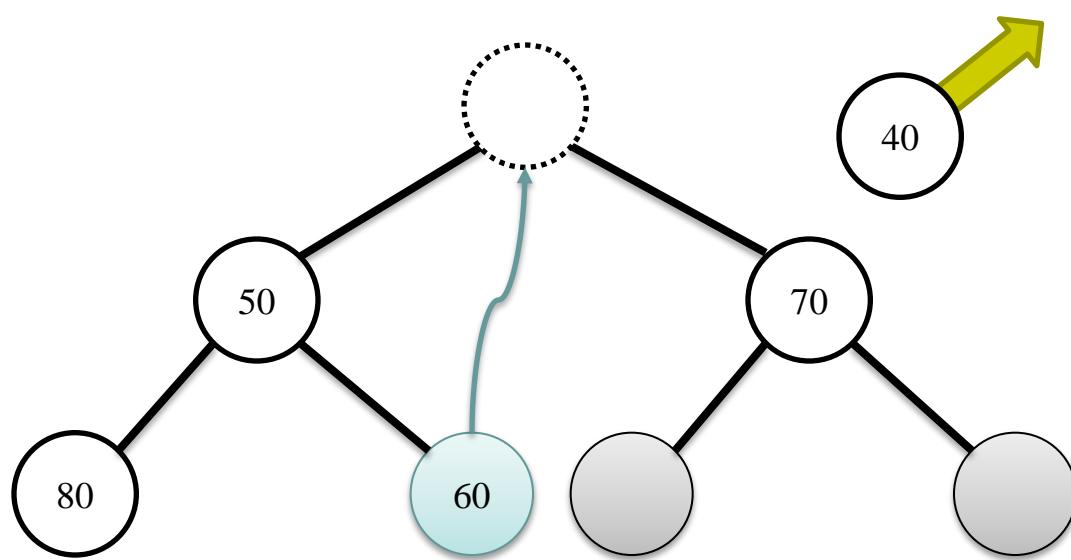


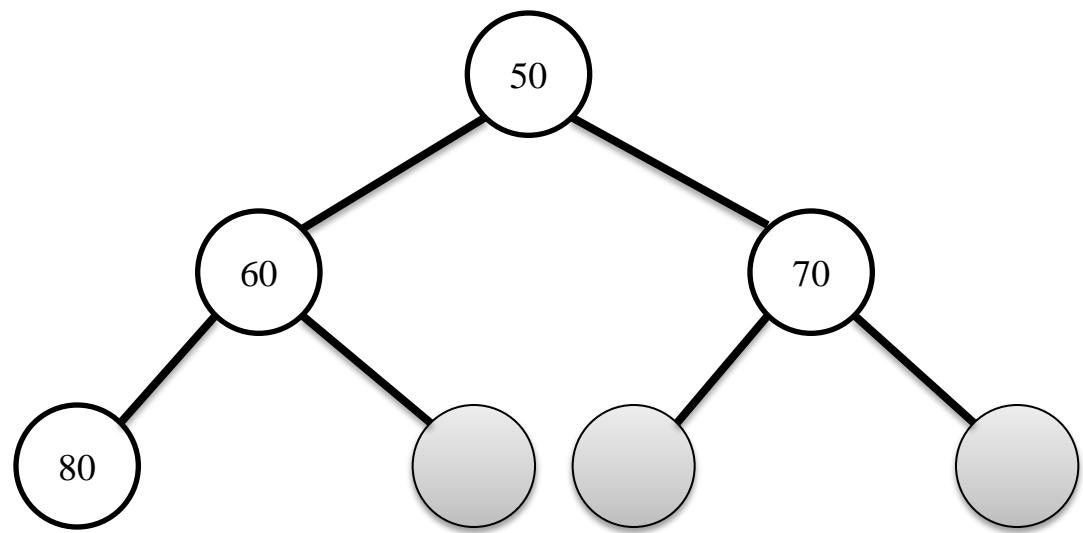


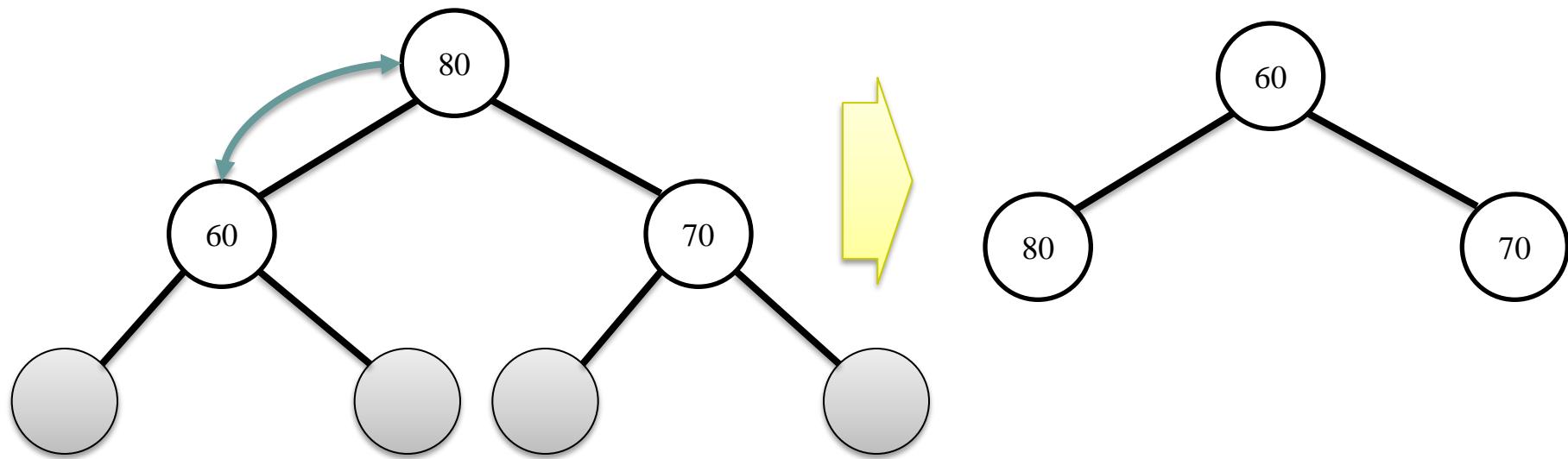
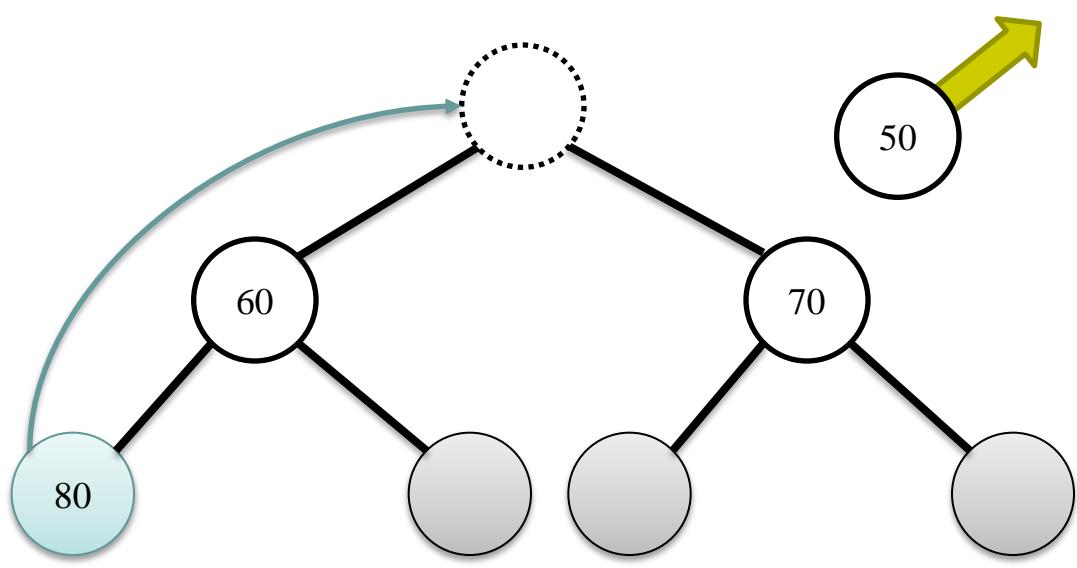


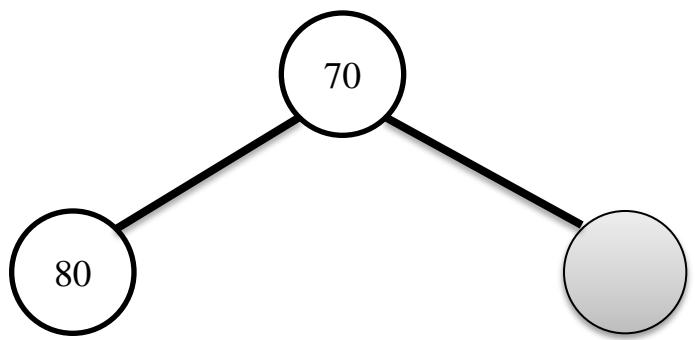
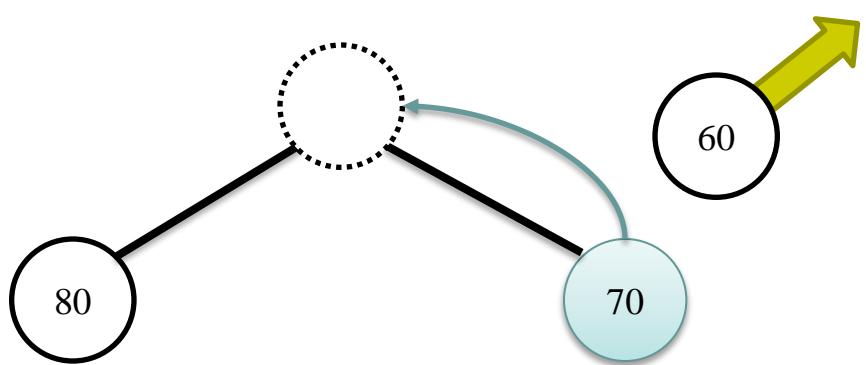


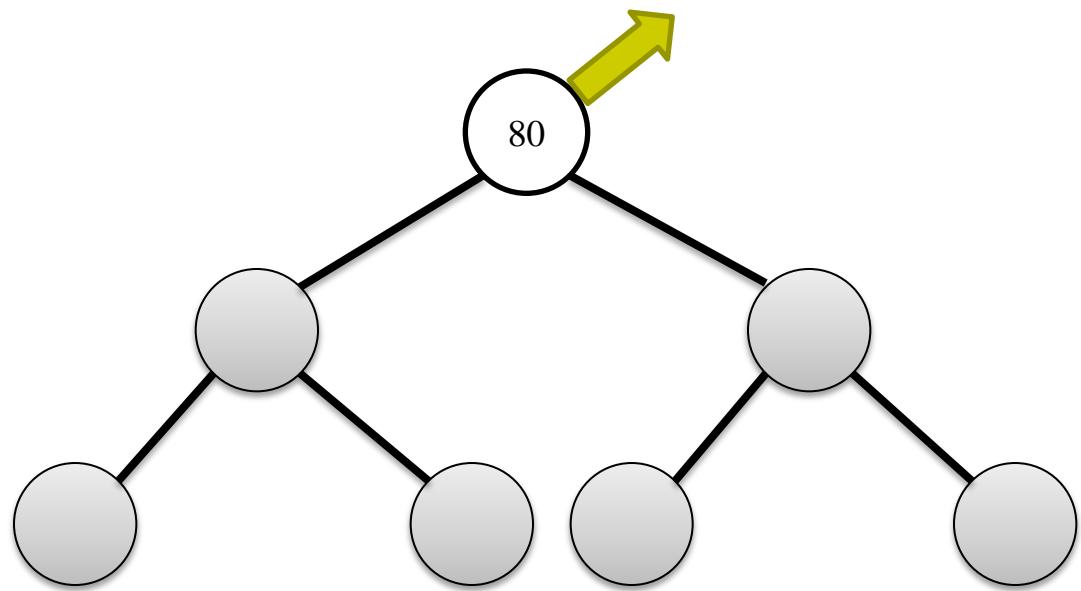
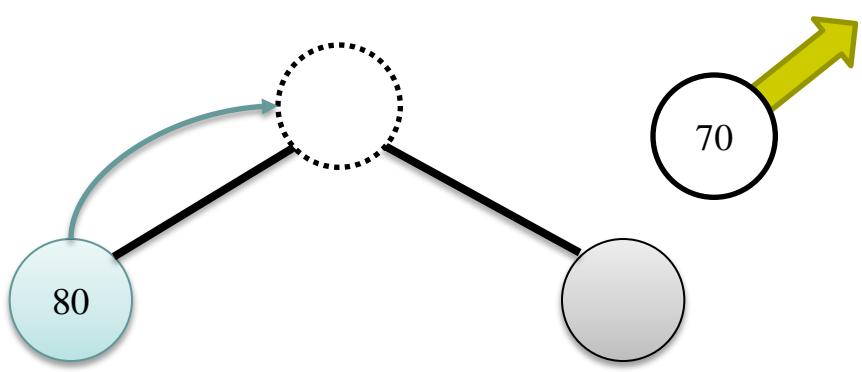












9. 히프 정렬 (2/2)

- 특성
 - 최선, 평균, 최악
 - 최선, 평균, 최악: $O(n \log 2n)$
- 정렬의 안정성
 - 불안정 정렬

	최선	평균	최악	안정성 여부	특징
선택 정렬	$O(n^2)$	$O(n^2)$	$O(n^2)$	불안정	장점: 이동 연산의 횟수 $O(n)$ 단점: 느리다
버블 정렬	$O(n^2)$	$O(n^2)$	$O(n^2)$	안정	단점: 느리다 & 이동 연산의 횟수 $O(n^2)$
퀵 정렬	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n^2)$	불안정	단점: 기준 자료의 정렬 정도에 따라 최악 $O(n^2)$
삽입 정렬	$O(n)$	$O(n^2)$	$O(n^2)$	안정	장점: 기준 자료의 정렬 정도에 따라 최선 $O(n)$
셸 정렬	$O(n)$	$O(n^{1.25})$	$O(n^2)$	불안정	삽입 정렬을 개선
병합 정렬	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	안정	장점: 정렬 전 자료 상태에 영향 적음 단점: 추가 메모리 공간 필요
기수 정렬	$O(d * n)$	$O(d * n)$	$O(d * n)$	안정	장점: 우수한 정렬 효율성 단점: 문자 키인 경우 적용이 어렵고 버킷 저장을 위한 추가 메모리 필요
히프 정렬	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	불안정	장점: 우수한 정렬 효율성 단점: 추가 메모리 필요

이번 장에서는

- 정렬의 종류
- 선택 정렬
- 버블 정렬
- 퀵 정렬
- 삽입 정렬
- 셀 정렬
- 병합 정렬
- 기수 정렬
- 히프 정렬