알고리즘과게임콘텐츠

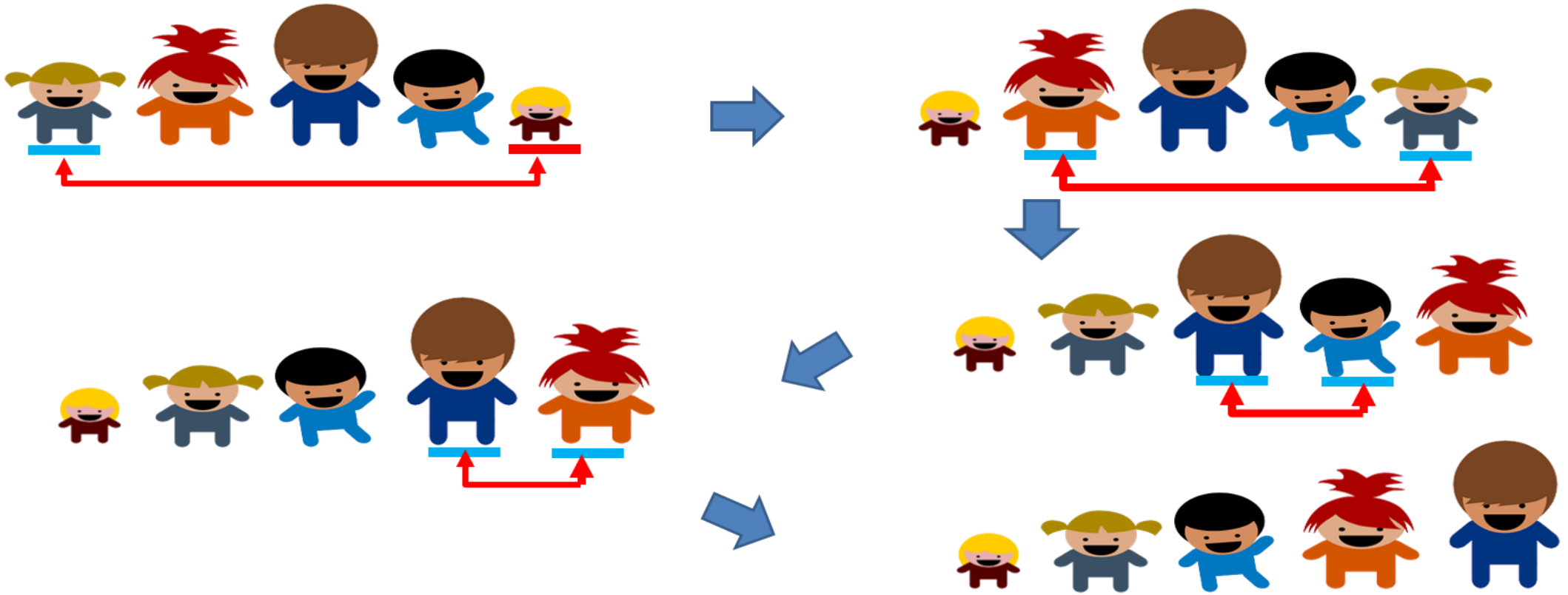
8장 자료정렬기법_선택_삽입_퀵

학습 목표

1. 삽입 정렬 기법의 개념을 이해 할 수 있다.
2. 삽입 정렬의 알고리즘을 이해할 수 있다.
3. 퀵 정렬 기법의 개념을 이해 할 수 있다.
4. 퀵 정렬의 알고리즘을 이해할 수 있다.

선택 정렬(SELECTION SORT) 개념

- 정렬되지 않은 자료 중 최소값(최대값)을 찾아 왼쪽부터 순서대로 정렬하는 알고리즘

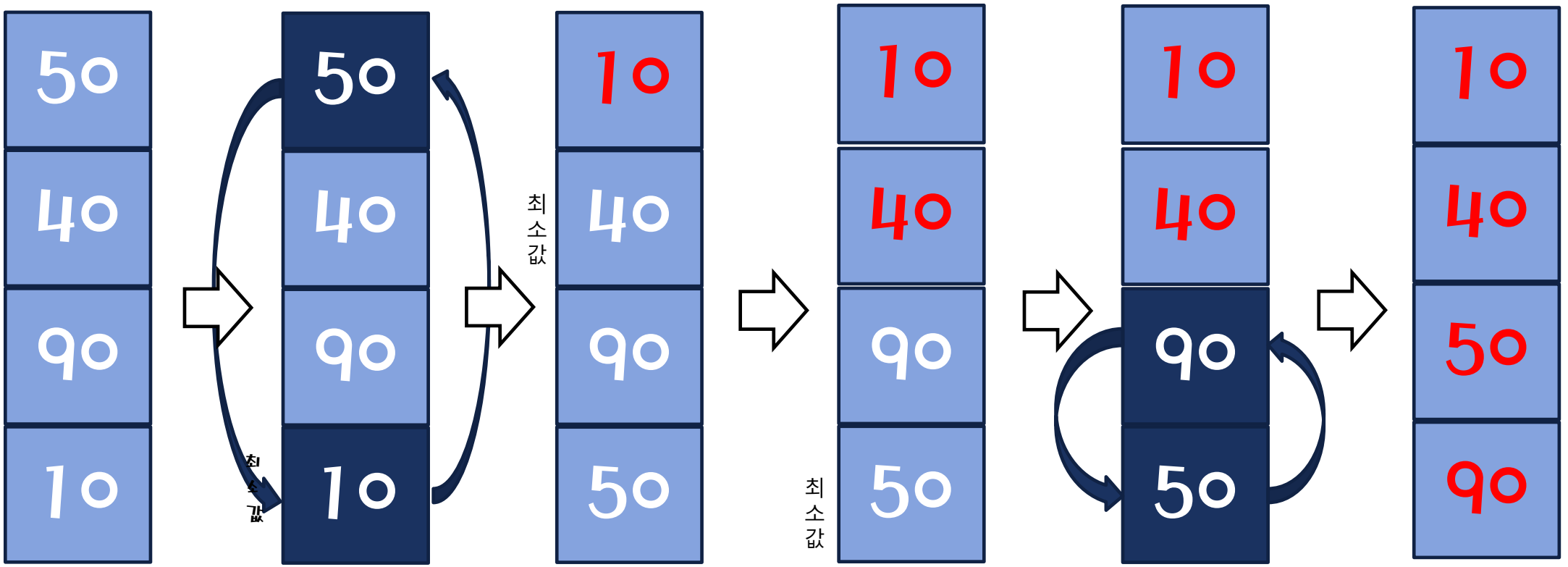


선택 정렬(SELECTION SORT) 개념

- 기본 전략(오름차순)
  - 입력된 자료 중 최소값을 선택하여 가장 앞쪽에 자료와 자리를 바꾸고, 다음엔 가장 앞쪽에 자료를 제외한 나머지 자료에서 최소값을 선택하여, 두번째 자리의 자료와 자리를 바꿈
  - 이러한 방식으로 마지막에 2개의 자료 중 최소값을 선택하여 자리를 바꿈으로써 정렬을 마침

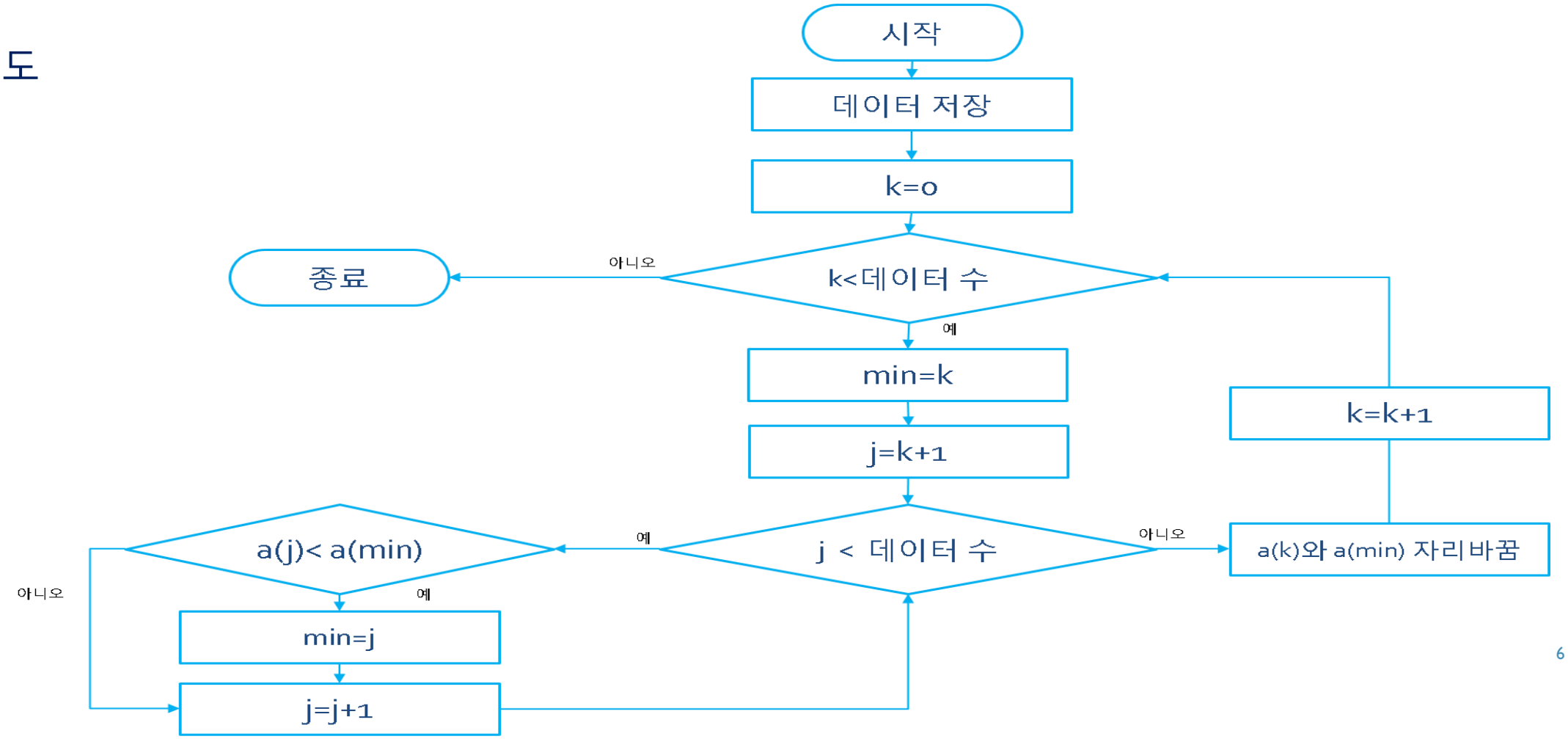
선택 정렬(SELECTION SORT) 개념

- 기본 전략(오름차순)



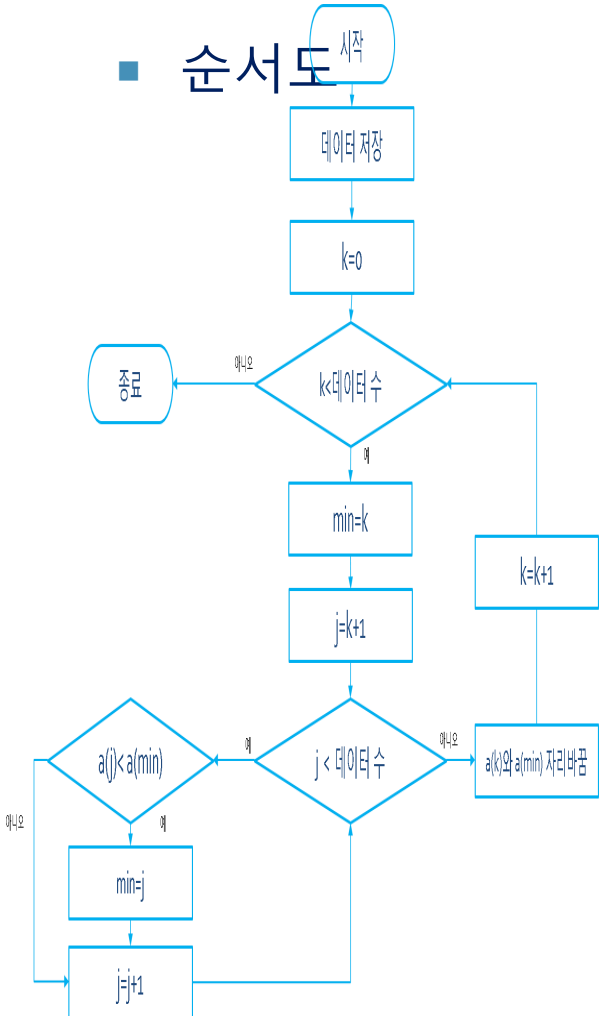
선택 정렬(SELECTION SORT) 개념

- 순서도



선택 정렬(SELECTION SORT) 개념

- 순서도




```
for k in range (0, n-1) :
    min = k
    for j in rage (k+1,  n) :
        if a[j] < a[min]:
            min = j
    temp = a[k]
    a[k] = a[min]
    a[min] = temp
```

5, 8, 2, 6 data의 경우

k=0

① min=k

② j=k+1 (0+1)

1) j<4

2) a[1]<a[0] (8<5)

3) 아니면 j=2

4) 2<4

5) a[2]<a[0]

6) min=2

7) j=3

8) 3<4

9) a(3)<a(2)

10) j=4

11) a(k)와 a(min) 값자리 바꿈

선택 정렬(SELECTION SORT) 개념

- 코딩
  - a배열에 n개의 데이터가 저장되었을 때

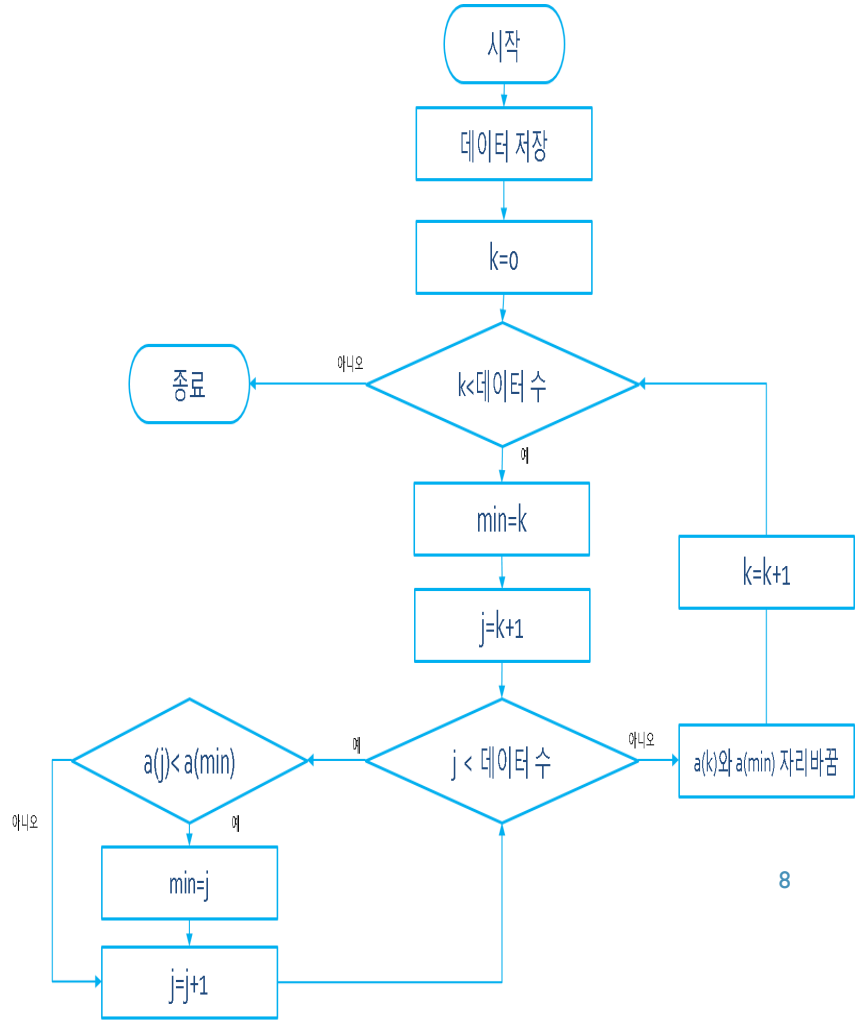
```
for k in range (0, n-1) :
    min = k
    for j in rage (k+1,  n) :
        if a[j] < a[min]:
            min = J
    temp = a[k]
    a[k] = a[min]
    a[min] = temp
```

원본 : 5 8 2 6

2 8 5 6

2 5 8 6

2 5 6 8



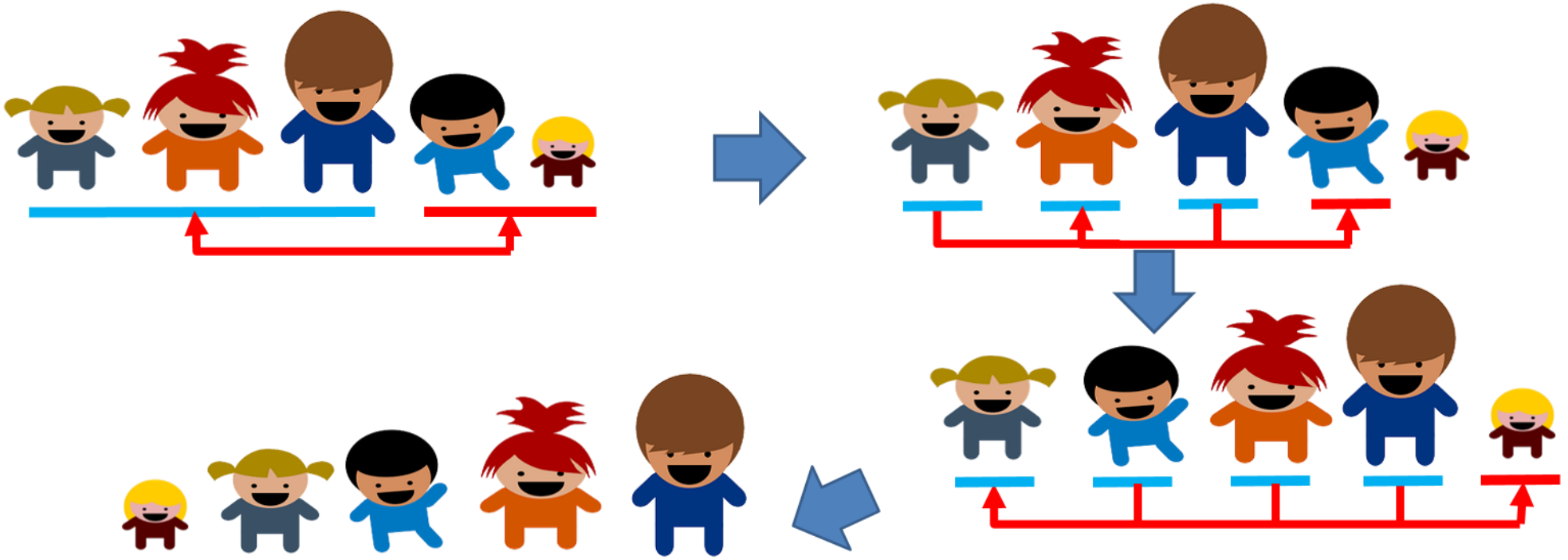
선택 정렬(SELECTION SORT) 개념

- 선택정렬의 장단점

장점	• 자료의 양이 적을 때 아주 좋은 성능을 나타냄 • 정렬을 위한 비교횟수는 많으나, 교환횟수는 적음 • 정렬도중 멈춰도 부분 정렬 값을 얻을 수 있음
단점	• 가장 작은 값과 현재 값을 교환하는 방식이라 현재 값이 뒤쪽 어디로 갈지 알 수 없으므로 안정성이 없음 • 많은 비교를 거치므로 대량의 자료에 대해서는 속도가 느려짐 • 한번에 한 개의 값만 정렬 되므로 다른 정렬기법에 비해 성능이 떨어짐

삽입정렬(INSERTION SORT) 개념

- 자료를 정렬된 부분과 정렬 안 된 부분으로 나누고, 정렬 안 된 부분의 가장 왼쪽 자료를, 정렬된 부분의 적절한 위치에 삽입하는 과정을 반복하는 알고리즘

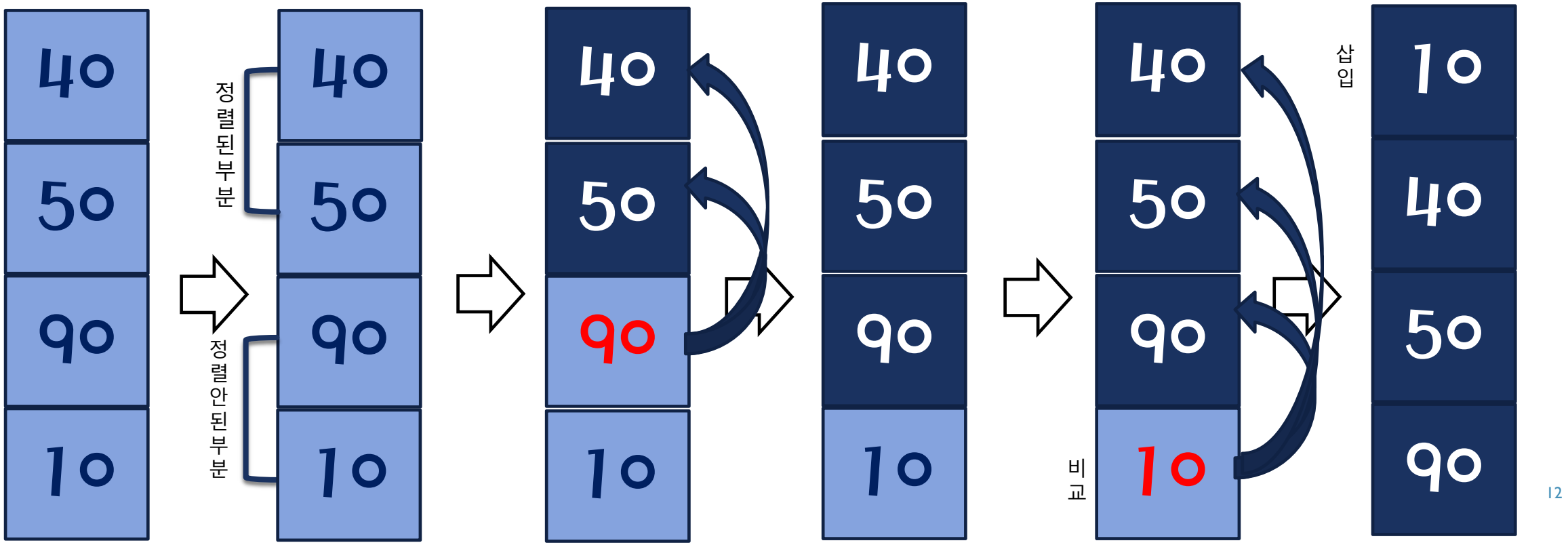


삽입정렬(INSERTION SORT) 개념

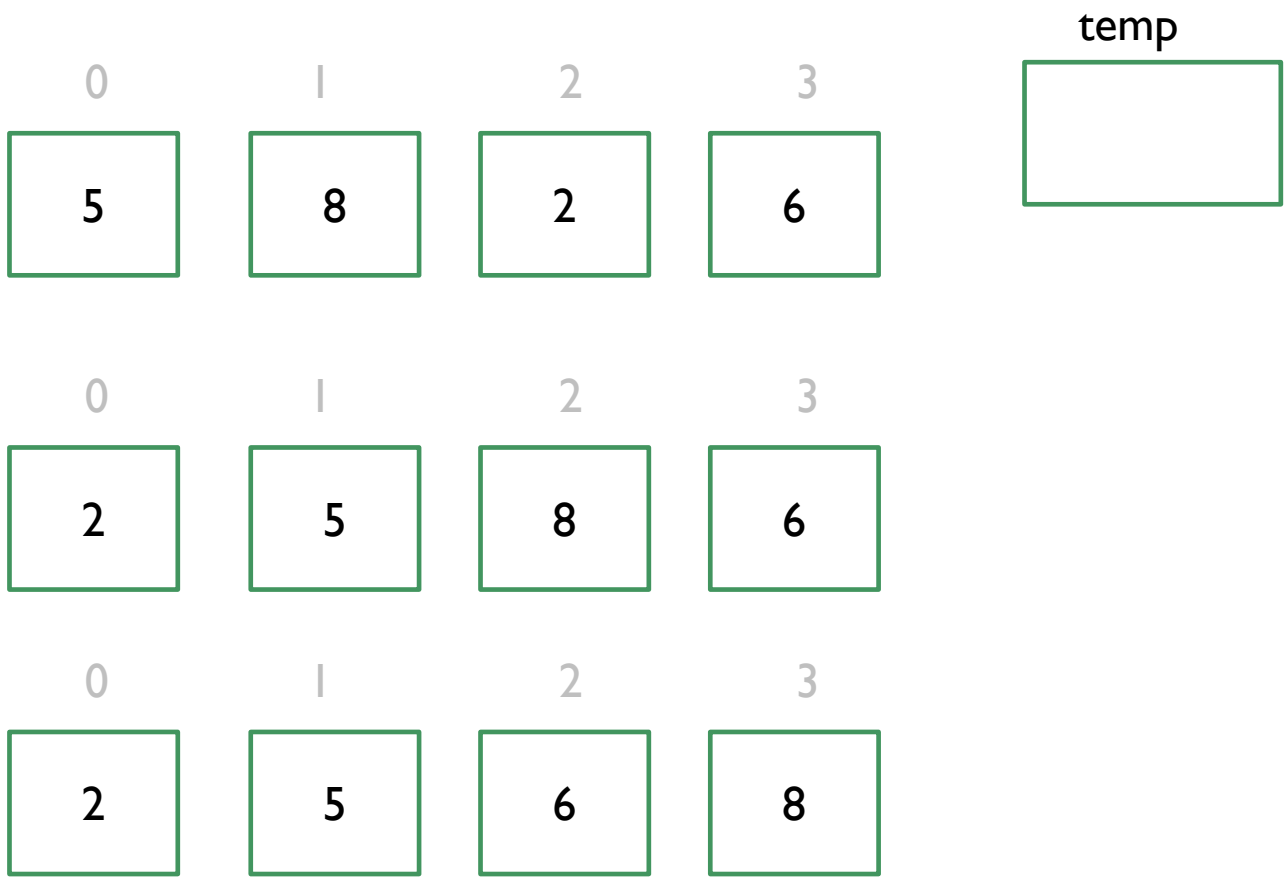
- 기본 전략
  - 정렬 안 된 부분의 자료가 정렬된 부분에 삽입됨으로써, 정렬된 부분의 자료가 늘어나고, 동시에 정렬이 안 된 부분의 자료는 줄어 듬
  - 이를 반복하여 수행하면, 마지막 정렬이 안 된 부분에 자료는 정렬된 부분으로 모두 옮겨지게 됨으로써 전체 자료가 정렬되게 됨

삽입정렬(INSERTION SORT) 개념

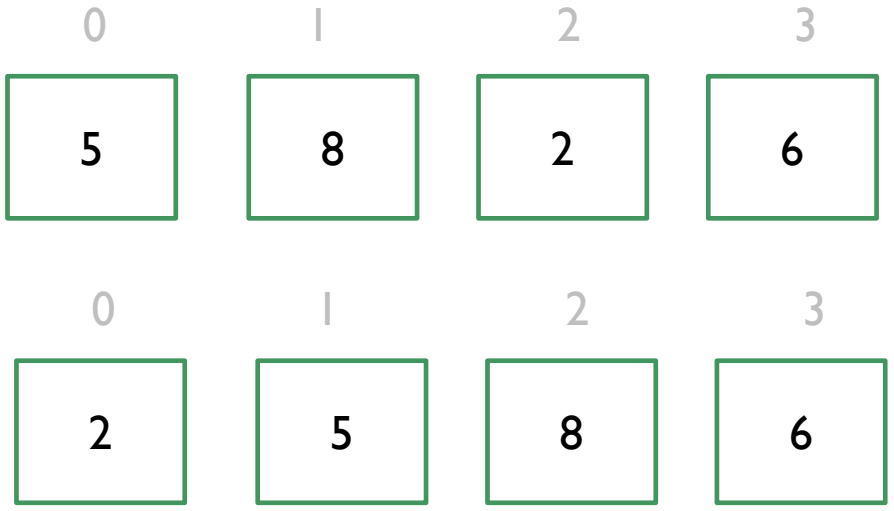
- 기본 전략



삽입정렬(INSERTION SORT) 개념



삽입정렬(INSERTION SORT) 개념



….

temp

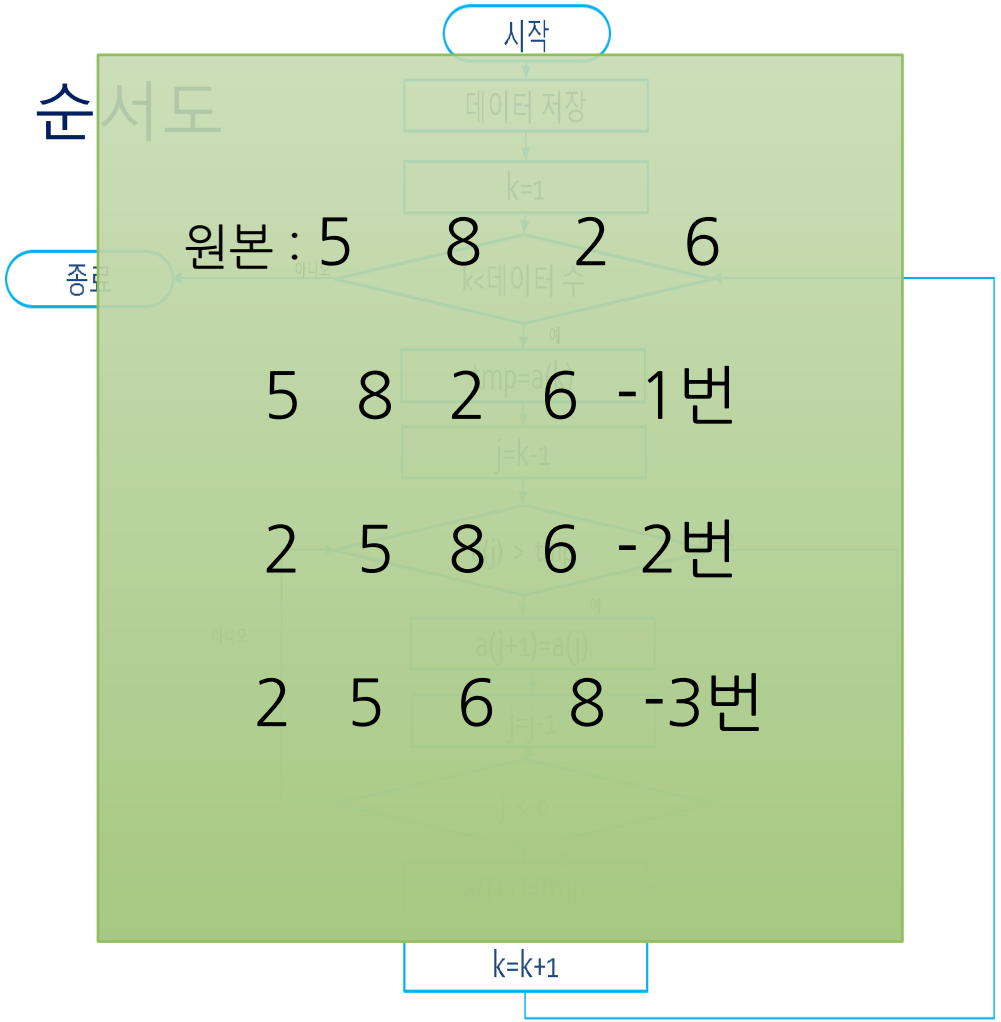
5, 8, 2, 6

1) k=1
2) k<4 (1<4)
3) temp=a(1) (temp=8)
4) j=k-1 (j=0)
5) a(0)>8 (5>8)
6) 아니오
7) a(1)=temp (a(1)=8)
8) k=2
9) k<4 (2<4)
10) temp=a(2) (temp=2)
11) j=2-1 (j=1)
12) a(1)>2 (8>2)
13) 그러면
14) a(j+1)=a(j) (8을 a(2)번 방에 넣음)
15) j가 0 보다 작아
16) 그렇지 않으면
17) a(0)>temp (5>2)
18) a(1)=a(0) (5를 a(1)방에 넣음)
19) a(0)=temp

….

삽입정렬(INSERTION SORT) 개념

- 순서도



5, 8, 2, 6

1) k=1
2) k<4 (1<4)
3) temp=a(1) (temp=8)
4) J=k-1 (j=0)
5) a(0)>8 (5>8)
6) 아니오
7) a(1)=temp (a(1)=8)
8) k=2
9) k<4 (2<4)
10) temp=a(2) (temp=2)
11) j=2-1 (j=1)
12) a(1)>2 (8>2)
13) 그러면
14) a(j+1) = a(j) (8을 a(2)번 방에 넣음
15) j가 0 보다 작아
16) 그렇지 않으면
17) a(0)>temp (5>2)
18) a(1)=a(0) (5를 a(1)방에 넣음)
19) a(0)=temp

....

삽입정렬(INSERTION SORT) 개념

- 코딩
  - a배열에 n개의 데이터가 저장되었을 때

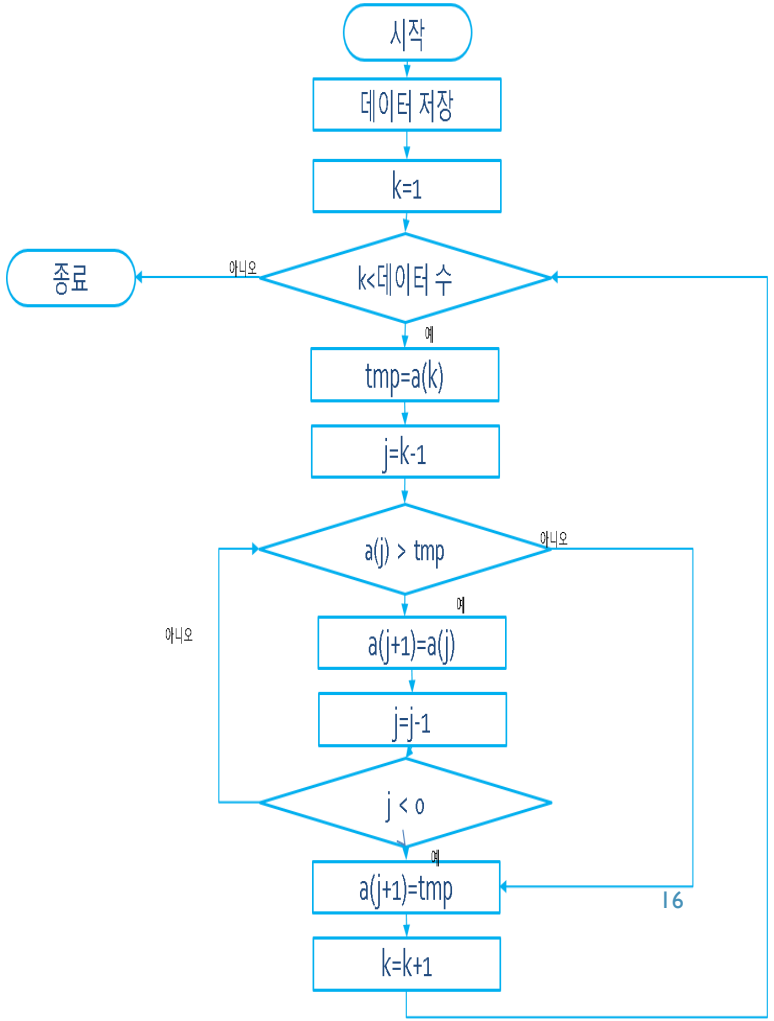
```
for k in range (1, len(a)):
    temp=a[k]
    j=k-1
    while j >= 0 and a[j] > temp :
        a[j+1] = a[j]
        j=j-1
    a[j+1]=temp
```

원본 : 5 8 2 6

5 8 2 6 -1번

2 5 8 6 -2번

2 5 6 8 -3번

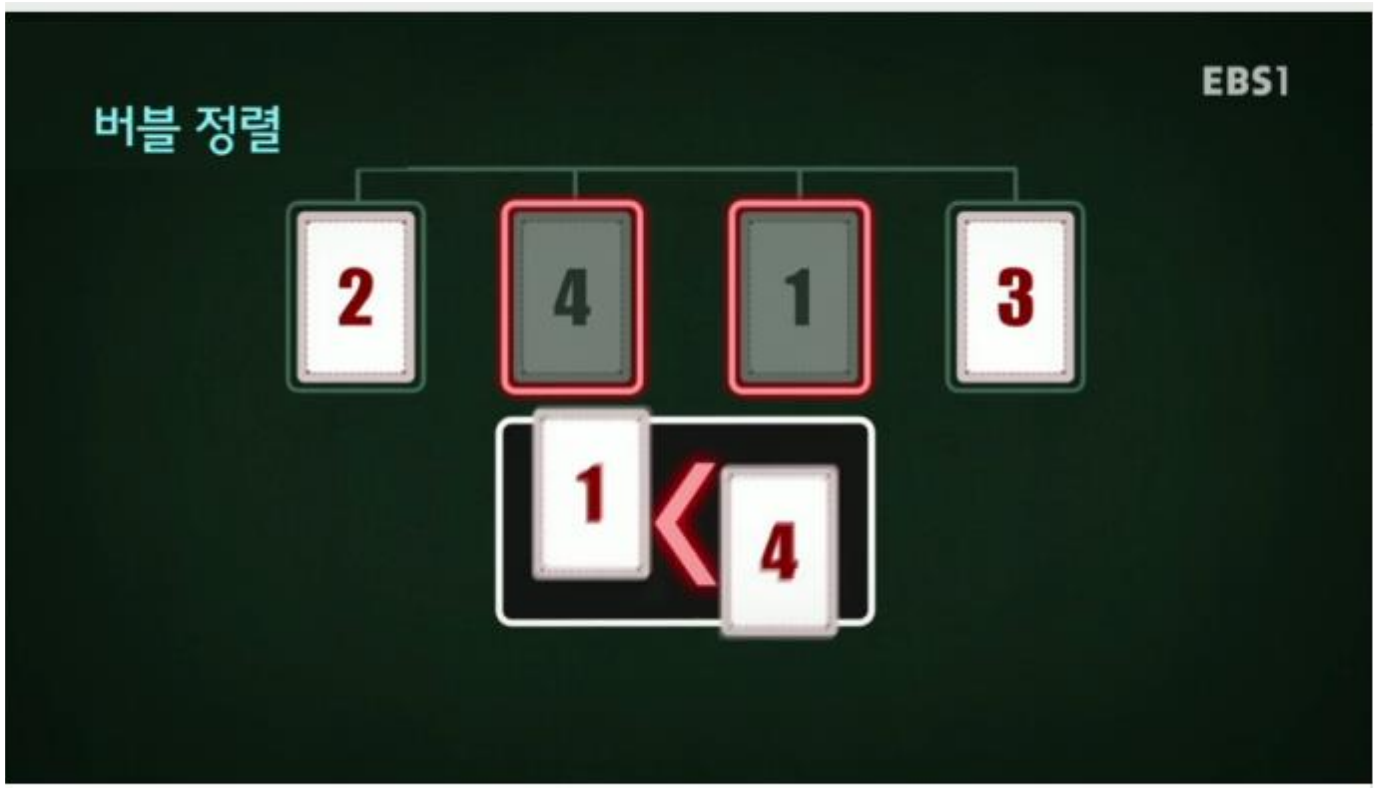


삽입정렬(INSERTION SORT) 개념

- 삽입정렬의 장단점

장점	• 데이터의 양이 적을 때는 성능이 우수함. • 자료의 대부분이 이미 정렬 되어 있는 경우 효율적임
단점	• 정렬된 자료에 새로운 값을 삽입하는 경우 이미 정렬된 자료들이 이동해야 하는 경우가 발생하므로 안정성이 낮음

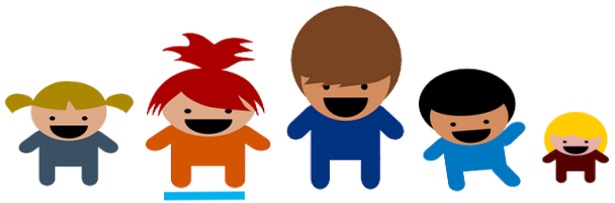
버블_삽입 정렬 비교

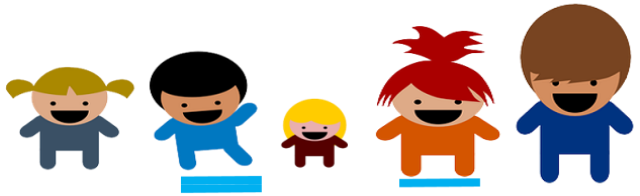


동영상 4:17

퀵 정렬(QUICK SORT) 개념

- 기준값을 기준으로 작거나 같은 값을 지닌 자료는 앞으로, 큰 값을 지닌 자료는 뒤로 가도록 하여 작은 값을 갖는 자료와 큰 값을 갖는 자료로 분리해가며 정렬하는 방법

① 

② 

③ 

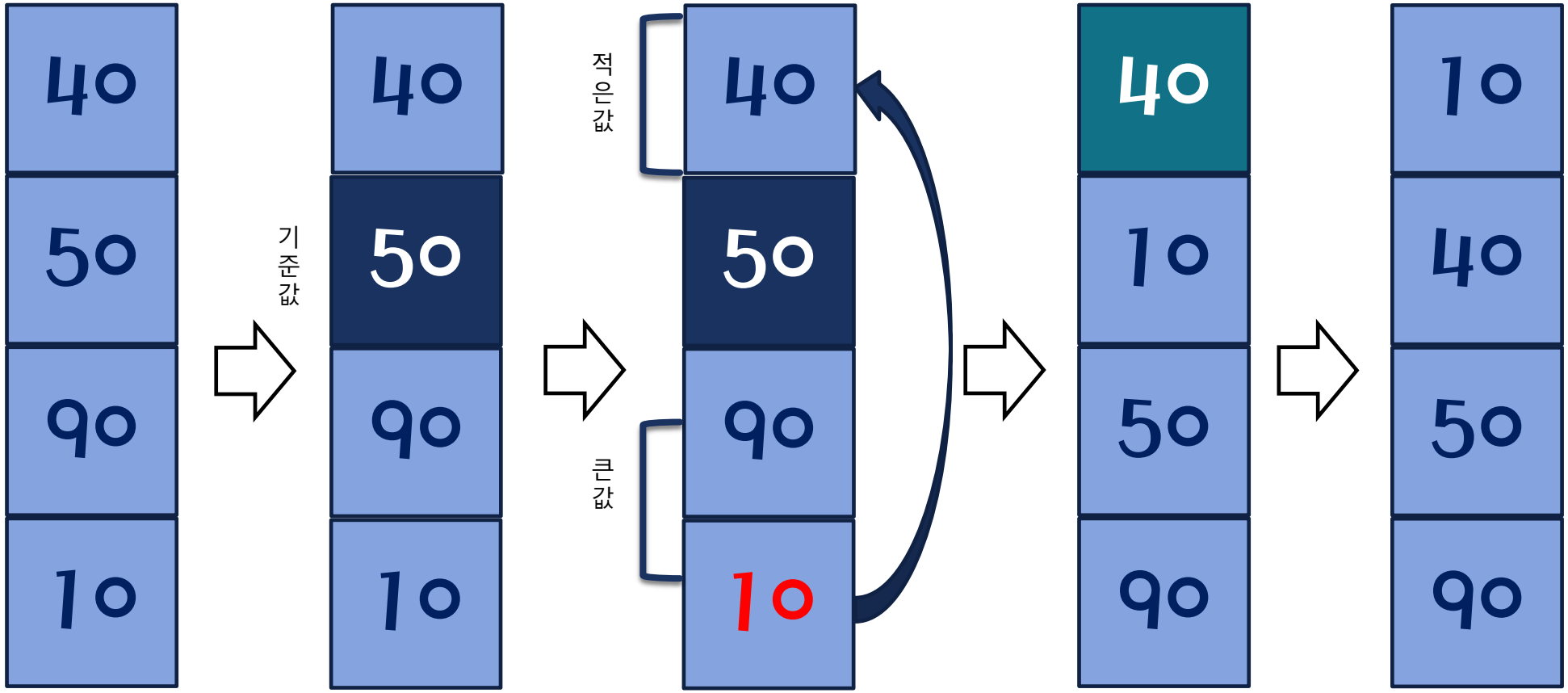
④ 

퀵 정렬(QUICK SORT) 개념

- 기본 전략
  - 기준값을 중심으로 왼쪽에 작은 값의 자료, 오른쪽에 큰 값의 자료로 나눔
  - 작은 값의 자료들 중, 기준값을 중심으로 다시 왼쪽에 작은 값의 자료, 오른쪽에 큰값의 자료로 나눔
  - 큰값의 자료들 중, 기준값을 중심으로 다시 왼쪽에 작은 값의 자료, 오른쪽에 큰값의 자료로 나눔
  - 작은 값을 갖는 자료와 큰값을 갖는 자료를 분리해 가며 정렬을 반복 함(분할정복)

퀵 정렬(QUICK SORT) 개념

- 기본 전략



퀵 정렬(QUICK SORT) 개념

- 퀵정렬의 장단점

장점	• 일반적으로 다른 정렬 기법보다 수행속도가 빠름
단점	• 기준값의 선택에 따라 다양한 성능을 나타낼 수 있음

자료 정렬 기법_삽입_퀵

- 삽입 정렬 기법 개념 및 알고리즘
- 퀵 정렬 기법 개념 및 알고리즘