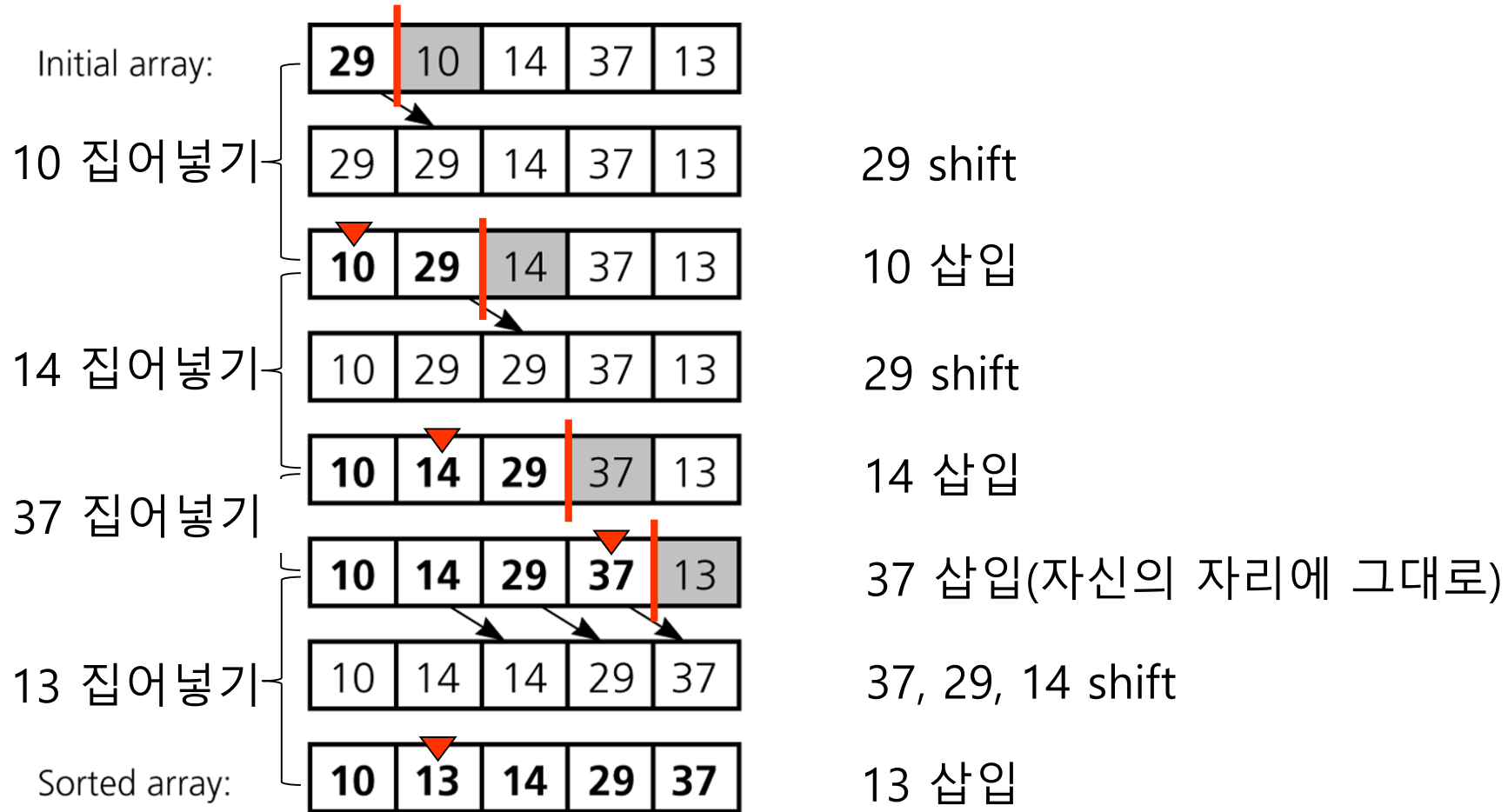


# 삽입정렬 실습

백석대학교 강윤희



# 삽입정렬



- ✓ 수행 시간:  $\Theta(n^2)$
- Worst case:  $1+2+\dots+(n-2)+(n-1)$
  - Average case:  $\frac{1}{2} (1+2+\dots+(n-2)+(n-1))$
- ✓ Best case:  $\Theta(n)$



insertionSort(A[], n)   ▷ A[1 ... n]을 정렬한다

$$\{$$

```

for  $i \leftarrow 2$  to  $n$ 
    ----- ①

```

A[1 ...  $i$ ]의 적당한 자리에 A[ $i$ ]를 삽입한다; ----- ②

$$\}$$

✓ 수행 시간:

- ①의 **for** 루프는 n-1 번 반복

— ②의 삽입은 최악의 경우  $i-1$ 회 비교

$$\frac{1}{2} \left( \frac{n \times (n-1)}{2} \right) = \frac{n^2 - n}{4} = \frac{n^2}{4}$$

11/22/21

✓ Worst case:  $1+2+\dots+(n-2)+(n-1) = \Theta(n^2)$

✓ Average case:  $\frac{1}{2} (1+2+\dots+(n-2)+(n-1)) = \Theta(n^2)$

- ✓ Best case:  $1+1+\dots+1+1 = \Theta(n)$

2024/10/27

$$n-1$$


```

void insertion_sort(int A[], int size)
{
    int i, j, k;
    int newValue;
    for (i = 1; i < size; i++) 2 4 2 2 4 4 2 1 < 2
    {
        newValue = A[i];
        j = i;
        while (j > 0 && A[j - 1] > newValue) {
            A[j] = A[j - 1];
            j--;
        }
        A[j] = newValue;
        printf("Partial Result of list taken by insertion sort :Wn");
        for (k = 0; k < size; k++)
            printf("%d ", A[k]);
        printf("Wn");
    }
}

```

