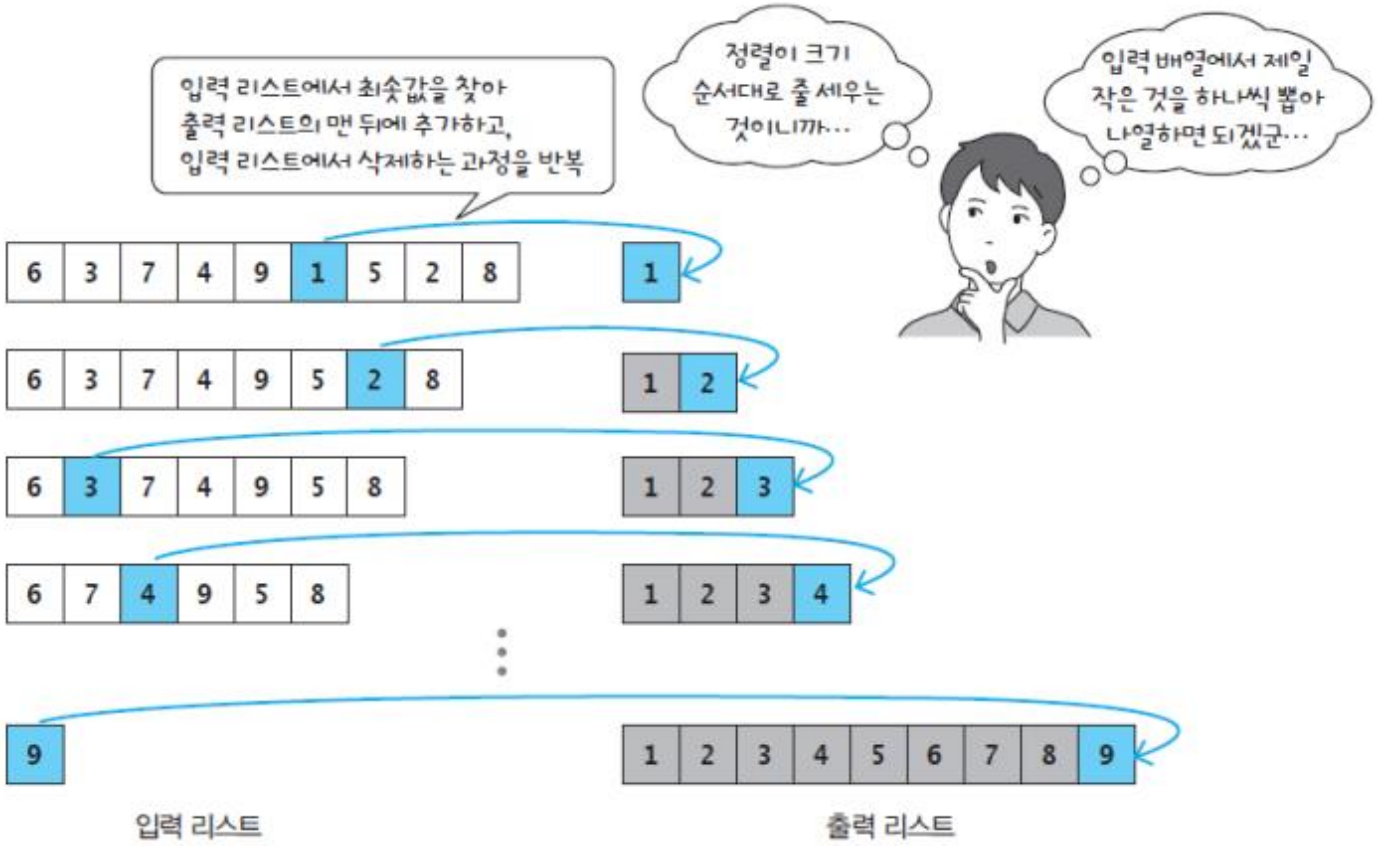
[Week 9] Practice

Sort

Problem 1

Selection sort



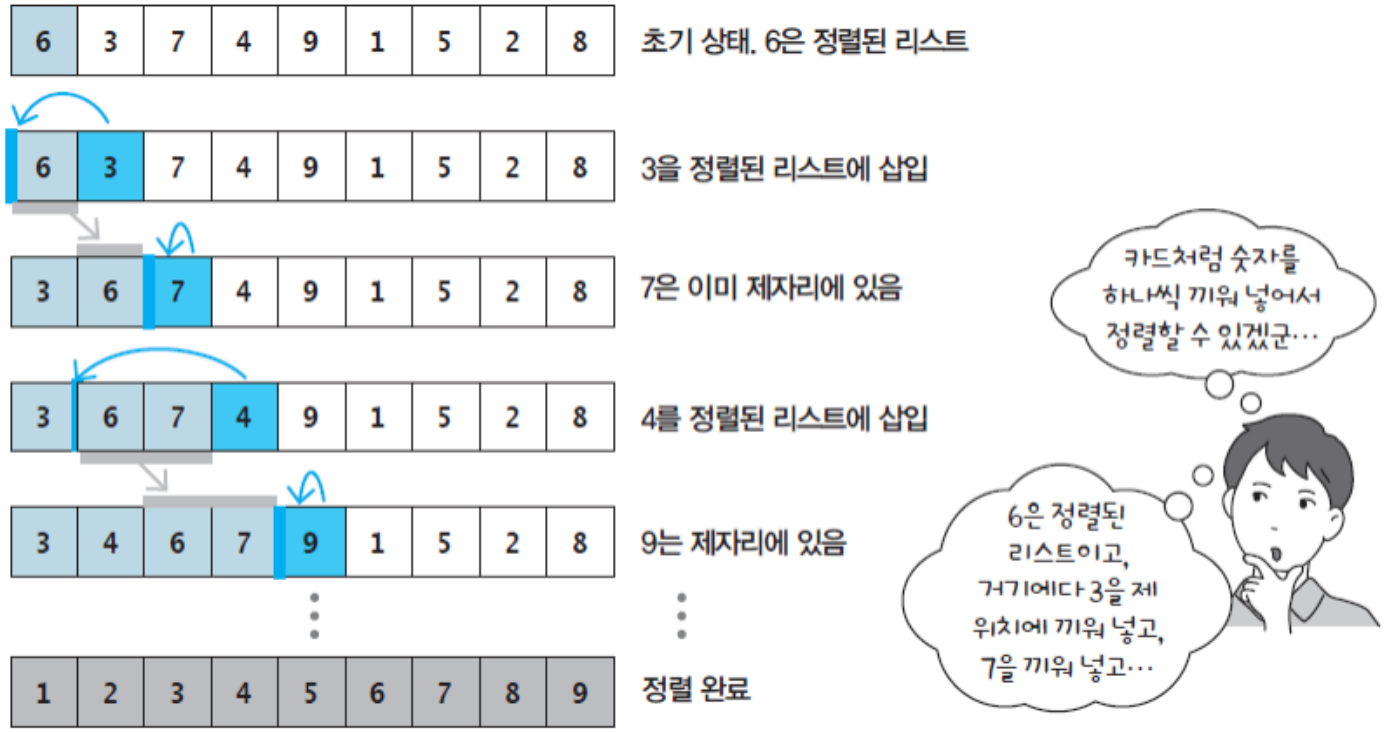
[Code] basic/main.py

```python
def selection_sort(arr):
    n = len(arr)
    # i가 0부터 n-1까지 순회하도록 range 완성
    for i in range(''' block '''):
        min_idx = i
        for j in range(i + 1, n):
            # 최소값 인덱스 갱신 로직 구현
            ########################################
            #                block                 #
            ########################################
        # 스왑(swap) 구문 구현
        ''' block '''
    return arr

def insertion_sort(arr):
    n = len(arr)
    # i가 1부터 n-1까지 순회하도록 range 완성
    for i in range(''' block '''):
        key = arr[i]
        j = i - 1
        # key보다 큰 요소를 뒤로 이동시키는 조건 구현
        while j >= 0 and ''' block ''':
            ########################################
            #                block                 #
            ########################################
        # key 삽입 구문 구현
        ''' block '''
    return arr
```

Problem 1

Insertion sort



[Code] basic/main.py

```python
def selection_sort(arr):
    n = len(arr)
    # i가 0부터 n-1까지 순회하도록 range 완성
    for i in range(''' block '''):
        min_idx = i
        for j in range(i + 1, n):
            # 최소값 인덱스 갱신 로직 구현
            ########################################
            #                block                 #
            ########################################
        # 스왑(swap) 구문 구현
        ''' block '''
    return arr

def insertion_sort(arr):
    n = len(arr)
    # i가 1부터 n-1까지 순회하도록 range 완성
    for i in range(''' block '''):
        key = arr[i]
        j = i - 1
        # key보다 큰 요소를 뒤로 이동시키는 조건 구현
        while j >= 0 and ''' block ''':
            ########################################
            #                block                 #
            ########################################
        # key 삽입 구문 구현
        ''' block '''
    return arr
```

Problem 2

- Selection sort:
  오름차순 선택정렬
- Insertion sort:
  오름차순 삽입정렬
- Frequency based selection sort:
  빈도 내림차순 우선, 값 오름차순 선택정렬
- Frequency based insertion sort:
  빈도 내림차순 우선, 값 오름차순 삽입정렬

```python
def frequency_calculation(arr):
    # 빈도 계산
    freq = {}
    for x in arr:
        freq[x] = freq.get(x, 0) + 1
    return freq
```

- 빈도 계산 함수

[Code] advanced/main.py

```python
def frequency_based_selection_sort(arr):
    freq = frequency_calculation(arr)
    n = len(arr)
    # i가 0부터 n-1까지 순회하도록 range 완성
    for i in range(''' block '''):
        best = i
        # j가 i+1부터 n-1까지 순회하도록 range 완성
        for j in range(''' block '''):
            # 빈도 내림차순 우선 · 값 오름차순 조건 기준으로 best 갱신 로직 구현
            ########################################
            #               block                  #
            ########################################
        # 스왑 구문 구현
        ''' block '''
    return arr

def frequency_based_insertion_sort(arr):
    freq = frequency_calculation(arr)
    n = len(arr)
    # i가 1부터 n-1까지 순회하도록 range 완성
    for i in range(''' block '''):
        key = arr[i]
        j = i - 1
        # 빈도 내림차순 우선 · 값 오름차순 조건 기준으로 삽입 위치 찾기
        while j >= 0 and (''' block '''):
            ########################################
            #               block                  #
            ########################################
        # key 삽입 구문 구현
        ''' block '''
    return arr
```

Problem Output

Problem 1 (basic)

```
selection_sort: [1, 2, 3, 4, 5, 6, 7, 8, 9]
insertion_sort: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Problem 2 (advanced)

```
selection_sort: [1, 1, 1, 2, 2, 2, 2, 3, 3, 5, 5, 5, 6]
insertion_sort: [1, 1, 1, 2, 2, 2, 2, 3, 3, 5, 5, 5, 6]
frequency_based_selection_sort: [2, 2, 2, 2, 1, 1, 1, 5, 5, 5, 3, 3, 6]
frequency_based_insertion_sort: [2, 2, 2, 2, 1, 1, 1, 5, 5, 5, 3, 3, 6]
```