Chapter. 05

고급 정렬 알고리즘

핵심 유형 문제풀이

FAST CAMPUS
ONLINE
유형별 문제풀이

강사. 나동빈

Chapter. 05

고급 정렬 알고리즘(핵심 유형 문제풀이)

FAST CAMPUS
ONLINE
나동빈 강사.

I 혼자 힘으로 풀어 보기

수 정렬하기 2: https://www.acmicpc.net/problem/2751

문제 제목: 수 정렬하기 2

문제 난이도: 하(Easy)

문제 유형: 정렬

추천 풀이 시간: 20분

수 정렬하기 2: https://www.acmicpc.net/problem/2751

I 문제 풀이 핵심 아이디어

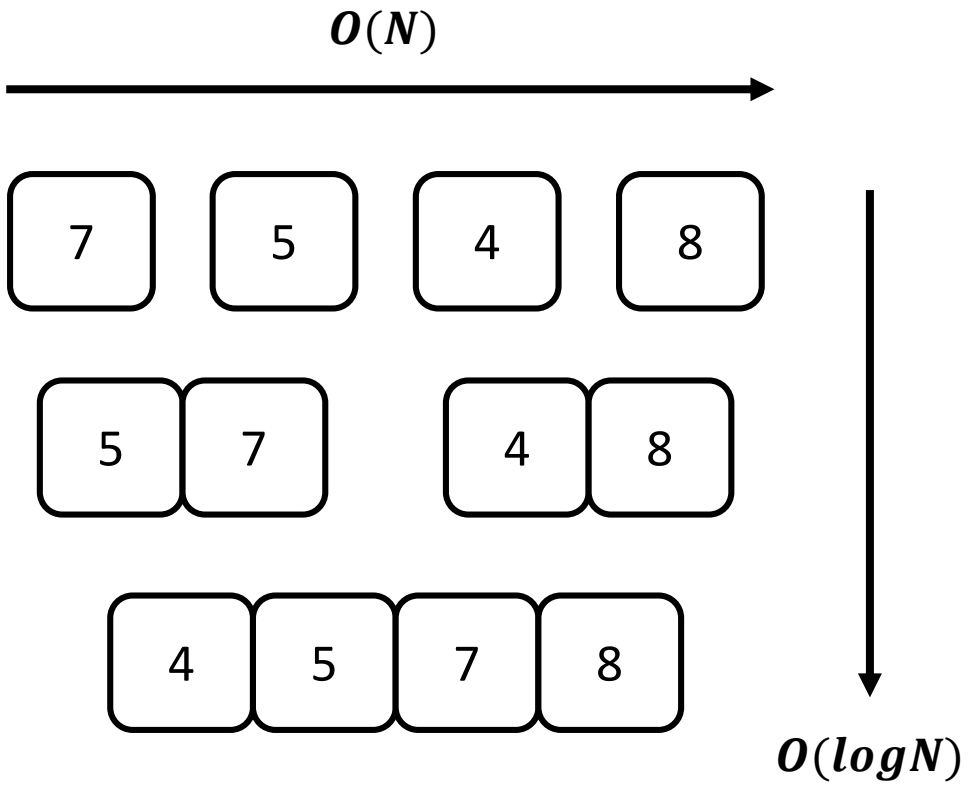
- 데이터의 개수가 최대 1,000,000개입니다.
- 시간 복잡도 $O(NlogN)$의 정렬 알고리즘을 이용해야 합니다.
- 고급 정렬 알고리즘 (병합 정렬, 퀵 정렬, 힙 정렬 등) 을 이용하여 문제를 해결할 수 있습니다.
- 혹은 파이썬의 **기본 정렬 라이브러리**를 이용하여 문제를 풀 수 있습니다.
- 메모리가 허용된다면, 되도록 Python 3보다는 PyPy 3를 선택하여 코드를 제출합니다.

수 정렬하기 2: https://www.acmicpc.net/problem/2751

I 문제 풀이 핵심 아이디어

병합 정렬(Merge Sort) 알고리즘

- 분할 정복 (Divide & Conquer) 방식을 이용합니다.
- 절반씩 합치면서 정렬하면, 전체 리스트가 정렬됩니다.
- 시간 복잡도 $O(NlogN)$을 보장합니다.

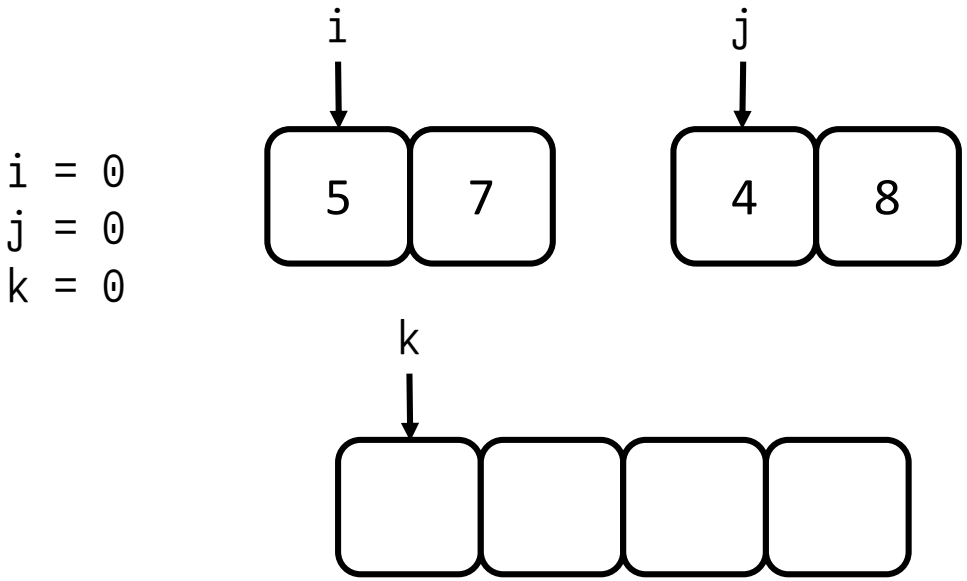


수 정렬하기 2: https://www.acmicpc.net/problem/2751

I 문제 풀이 핵심 아이디어

병합 정렬(Merge Sort) 알고리즘: 병합 과정

- 병합할 때는 리스트의 앞 원소부터 차례대로 채워 넣습니다.



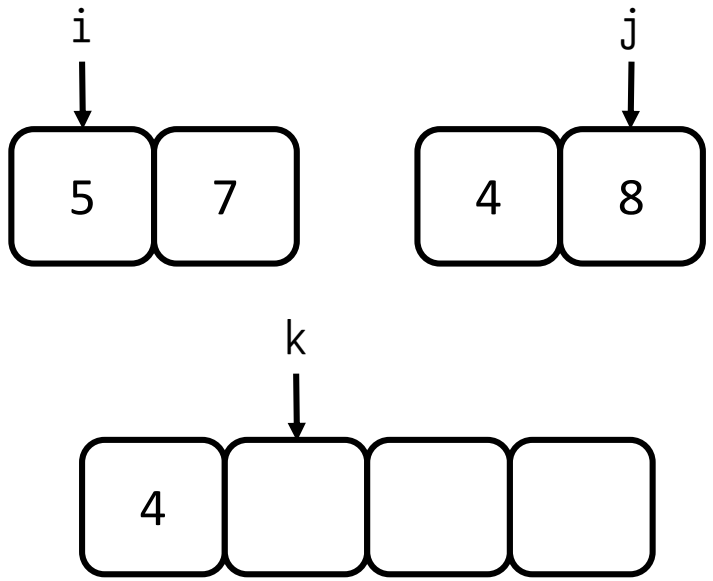
수 정렬하기 2: https://www.acmicpc.net/problem/2751

I 문제 풀이 핵심 아이디어

병합 정렬(Merge Sort) 알고리즘: 병합 과정

- 병합할 때는 리스트의 앞 원소부터 차례대로 채워 넣습니다.

```
i = 0
j = 1
k = 1
```



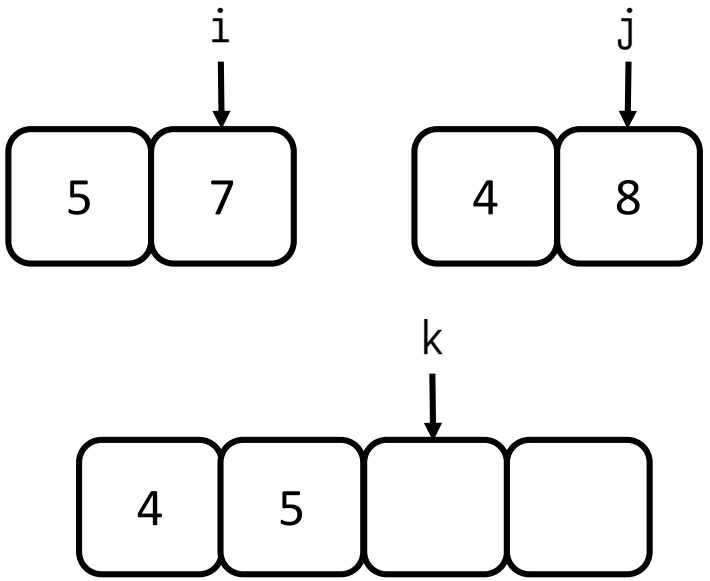
수 정렬하기 2: https://www.acmicpc.net/problem/2751

I 문제 풀이 핵심 아이디어

병합 정렬(Merge Sort) 알고리즘: 병합 과정

- 병합할 때는 리스트의 앞 원소부터 차례대로 채워 넣습니다.

```
i = 1
j = 1
k = 2
```

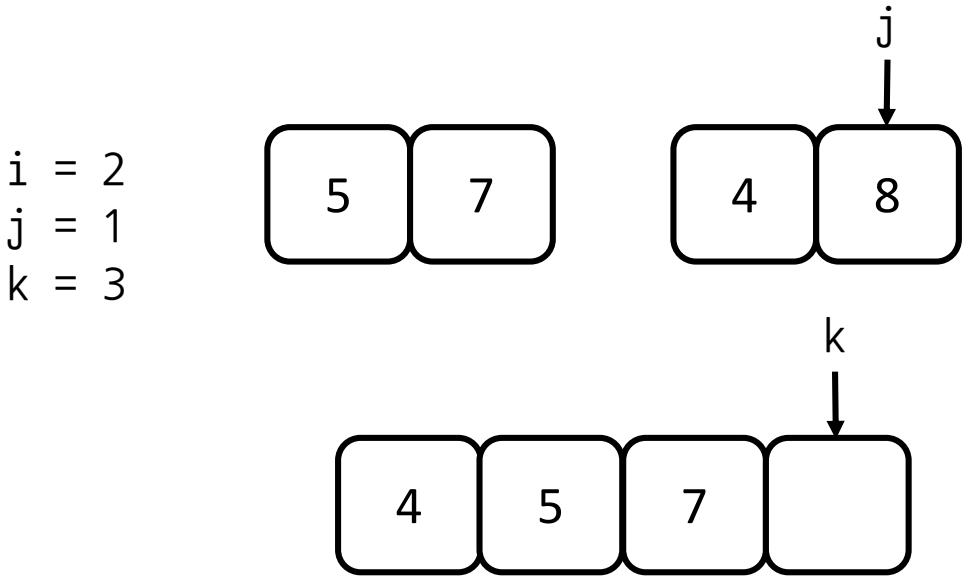


수 정렬하기 2: https://www.acmicpc.net/problem/2751

I 문제 풀이 핵심 아이디어

병합 정렬(Merge Sort) 알고리즘: 병합 과정

- 병합할 때는 리스트의 앞 원소부터 차례대로 채워 넣습니다.



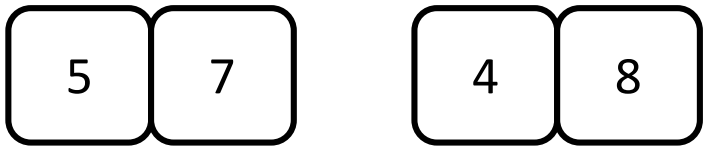
수 정렬하기 2: https://www.acmicpc.net/problem/2751

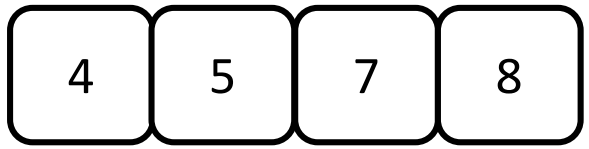
I 문제 풀이 핵심 아이디어

병합 정렬(Merge Sort) 알고리즘: 병합 과정

- 병합할 때는 리스트의 앞 원소부터 차례대로 채워 넣습니다.

```
i = 2
j = 2
k = 4
```





수 정렬하기 2: https://www.acmicpc.net/problem/2751

| 소스코드 ①

```python
def merge_sort(array):
    if len(array) <= 1:
        return array
    mid = len(array) // 2
    left = merge_sort(array[:mid])
    right = merge_sort(array[mid:])
    i, j, k = 0, 0, 0
    while i < len(left) and j < len(right):
        if left[i] < right[j]:
            array[k] = left[i]
            i += 1
        else:
            array[k] = right[j]
            j += 1
        k += 1
    if i == len(left):
        while j < len(right):
            array[k] = right[j]
            j += 1
            k += 1
    elif j == len(right):
        while i < len(left):
            array[k] = left[i]
            i += 1
            k += 1
    return array
```

```python
n = int(input())
array = []

for _ in range(n):
    array.append(int(input()))

array = merge_sort(array)

for data in array:
    print(data)
```

소스코드 ②

수 정렬하기 2: https://www.acmicpc.net/problem/2751

```python
n = int(input())
array = []

for _ in range(n):
    array.append(int(input()))

array = sorted(array)

for data in array:
    print(data)
```

I 혼자 힘으로 풀어 보기

K 번째 수: https://www.acmicpc.net/problem/11004

문제 제목: K 번째 수

문제 난이도: 중(Medium)

문제 유형: 정렬

추천 풀이 시간: 25분

K 번째 수: https://www.acmicpc.net/problem/11004

I 문제 풀이 핵심 아이디어

- 데이터의 개수가 최대 5,000,000개입니다.
- 시간 복잡도 $O(NlogN)$의 정렬 알고리즘을 이용해야 합니다.
- 고급 정렬 알고리즘 (병합 정렬, 퀵 정렬, 힙 정렬 등) 을 이용하여 문제를 해결할 수 있습니다.
- 혹은 파이썬의 **기본 정렬 라이브러리**를 이용하여 문제를 풀 수 있습니다.
- **시간적 이점을 위하여** PyPy 3를 선택하여 코드를 제출합니다.

K 번째 수: https://www.acmicpc.net/problem/11004

| 소스코드 ①

```python
def merge_sort(array):
    if len(array) <= 1:
        return array
    mid = len(array) // 2
    left = merge_sort(array[:mid])
    right = merge_sort(array[mid:])
    i, j, k = 0, 0, 0
    while i < len(left) and j < len(right):
        if left[i] < right[j]:
            array[k] = left[i]
            i += 1
        else:
            array[k] = right[j]
            j += 1
        k += 1
    if i == len(left):
        while j < len(right):
            array[k] = right[j]
            j += 1
            k += 1
    elif j == len(right):
        while i < len(left):
            array[k] = left[i]
            i += 1
            k += 1
    return array
```

```python
n, k = map(int, input().split())
array = list(map(int, input().split()))

array = merge_sort(array)

print(array[k - 1])
```

| 소스코드 ②

K 번째 수: https://www.acmicpc.net/problem/11004

```
n, k = map(int, input().split())
array = list(map(int, input().split()))

array = sorted(array)

print(array[k - 1])
```