# 고 급 문 제 해 결

## 문제 11.8

## 4월 14일 발표 자료

2018117543 한재성

# 문제 11.8:
# find kth largest element in an array

# 문제 11.8:
# find kth largest element in an array

12.8 FIND THE *k*TH LARGEST ELEMENT

Many algorithms require as a subroutine the computation of the $k$th largest element of an array. The first largest element is simply the largest element. The $n$th largest element is the smallest element, where $n$ is the length of the array.

For example, if the input array $A = \langle 3, 2, 1, 5, 4 \rangle$, then $A[3]$ is the first largest element in $A$, $A[0]$ is the third largest element in $A$, and $A[2]$ is the fifth largest element in $A$.

**Example 1:**

```
Input: nums = [3,2,1,5,6,4], k = 2
Output: 5
```

**Example 2:**

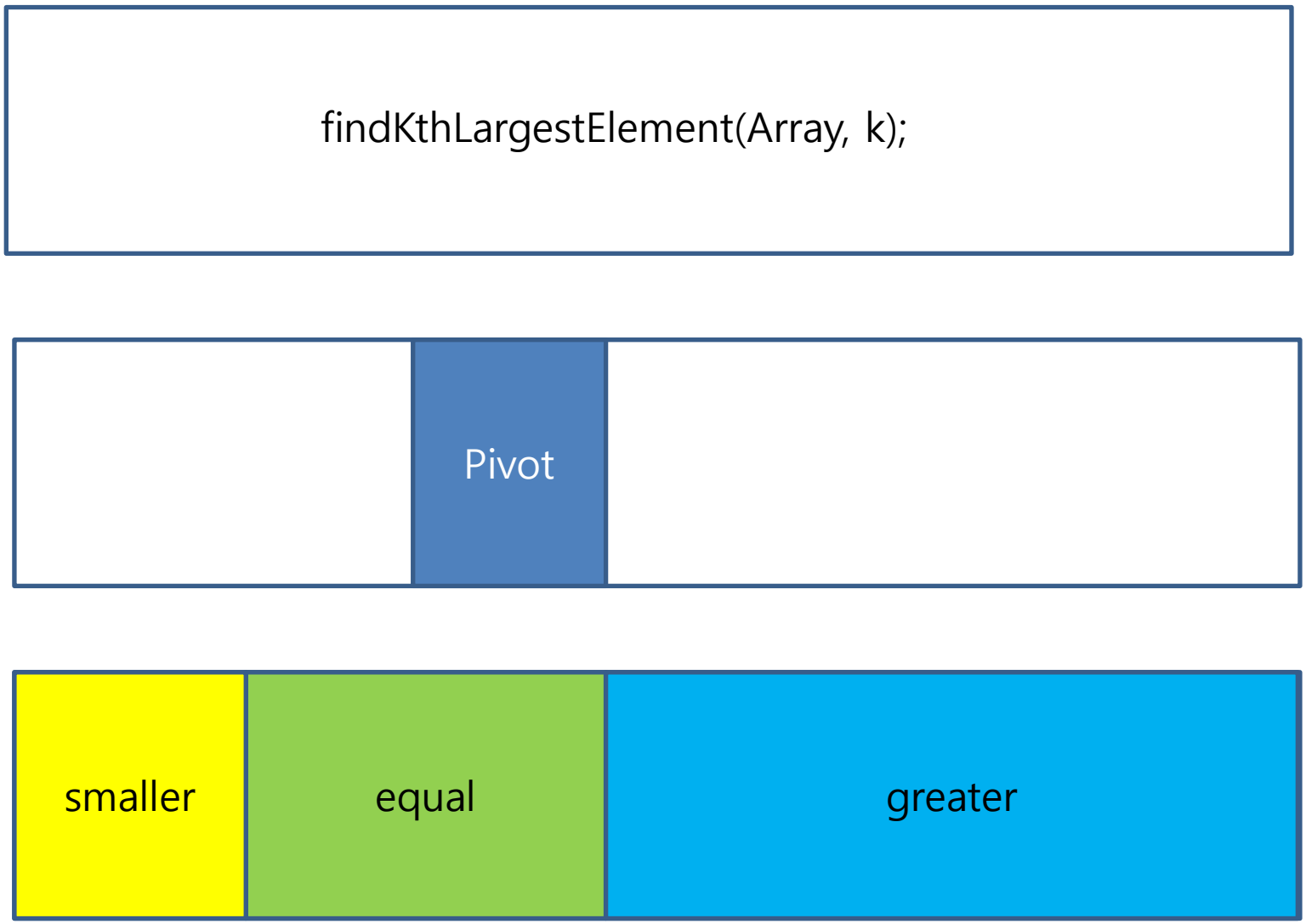
```
Input: nums = [3,2,3,1,2,4,5,5,6], k = 4
Output: 4
```

# 빠르고 간단하게!

```python
class Solution:
    def findKthLargest(self, nums: List[int], k: int) -> int:
        nums.sort()
        return nums[-k]
```

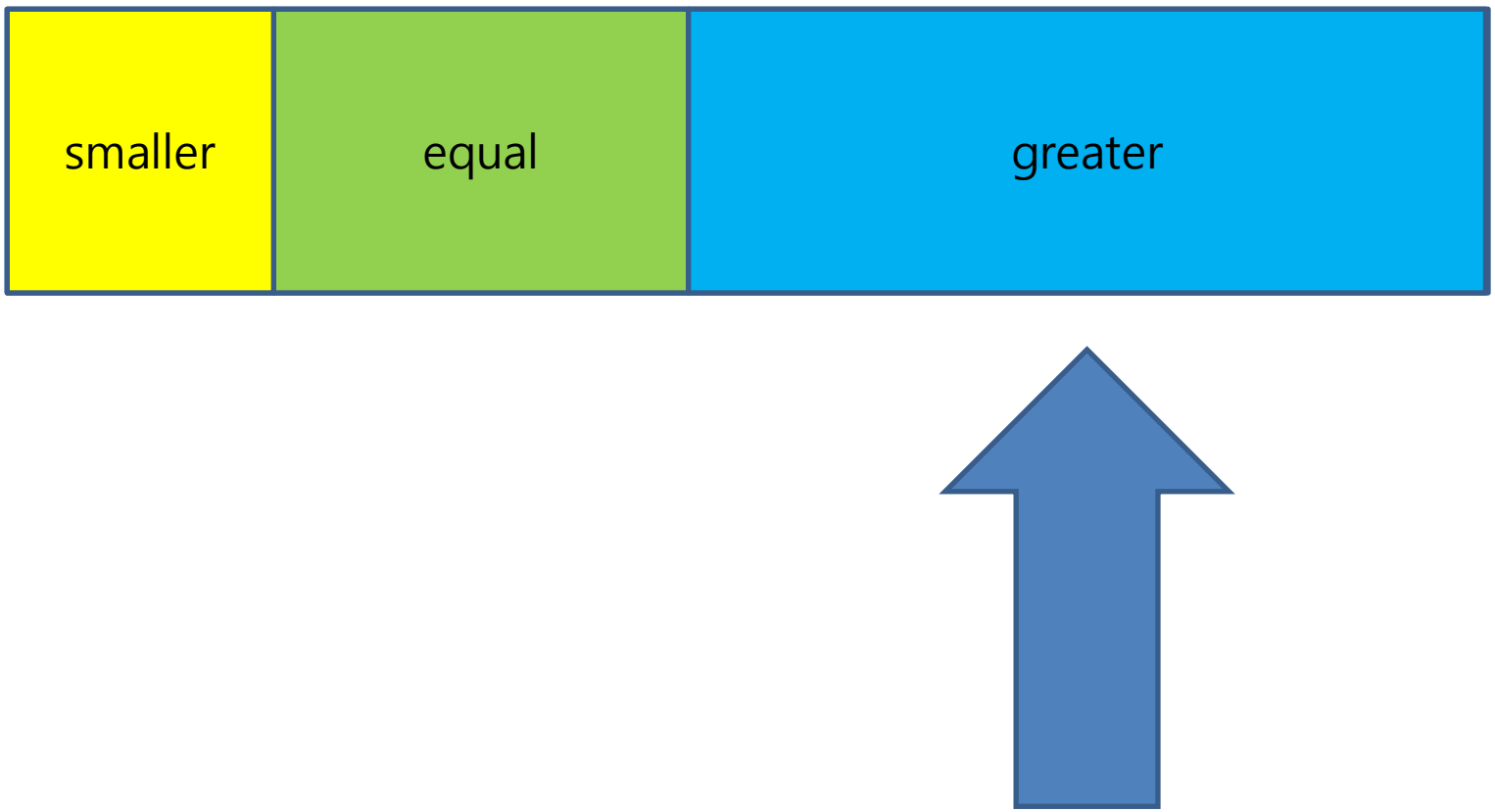
- Sort : O(nlogn)
- Heap : O(nlogn)
- ..

# Quickselect algorithm



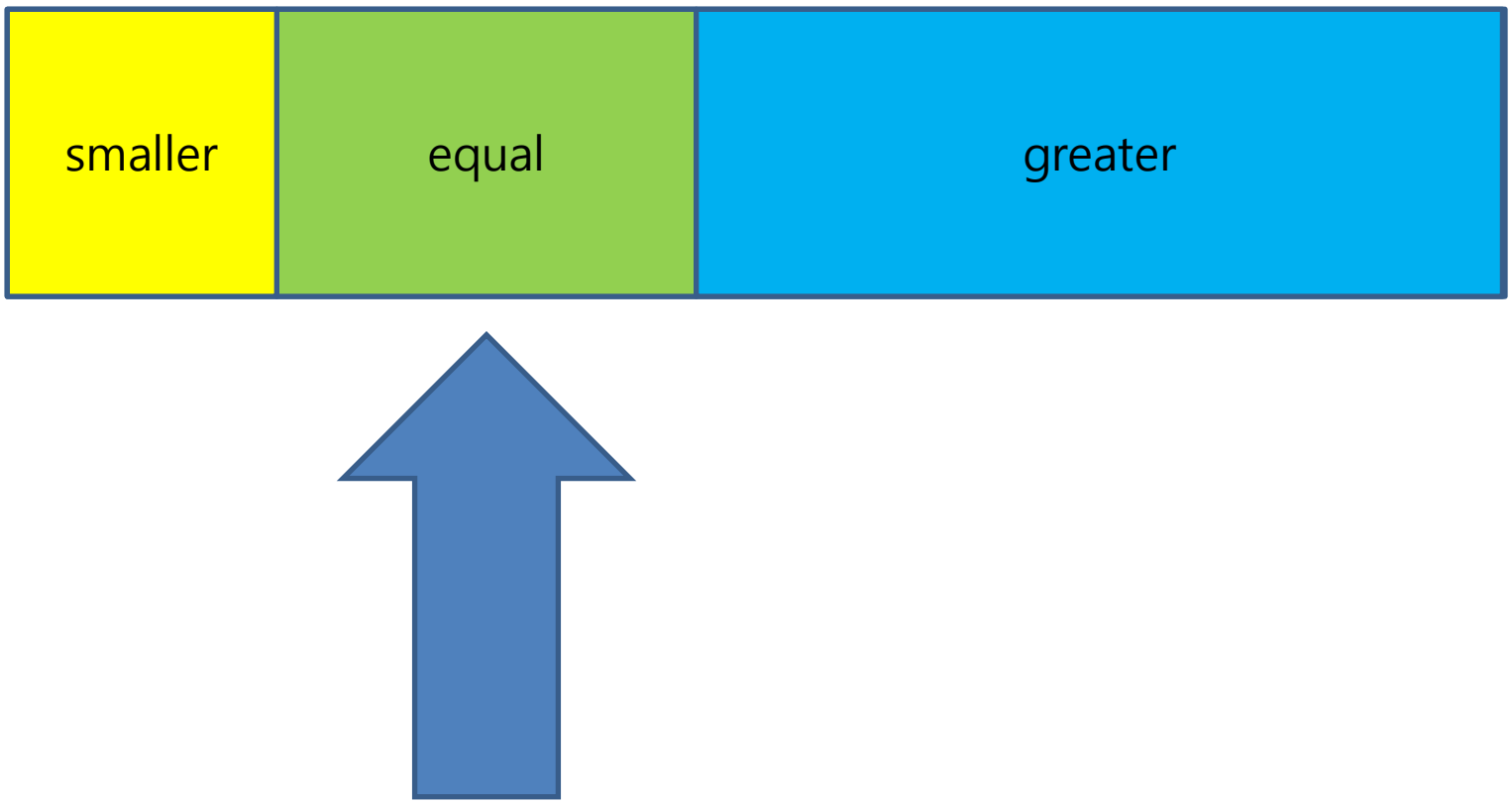

주의: 각 리스트는 정렬하지 않습니다.

# Quickselect algorithm



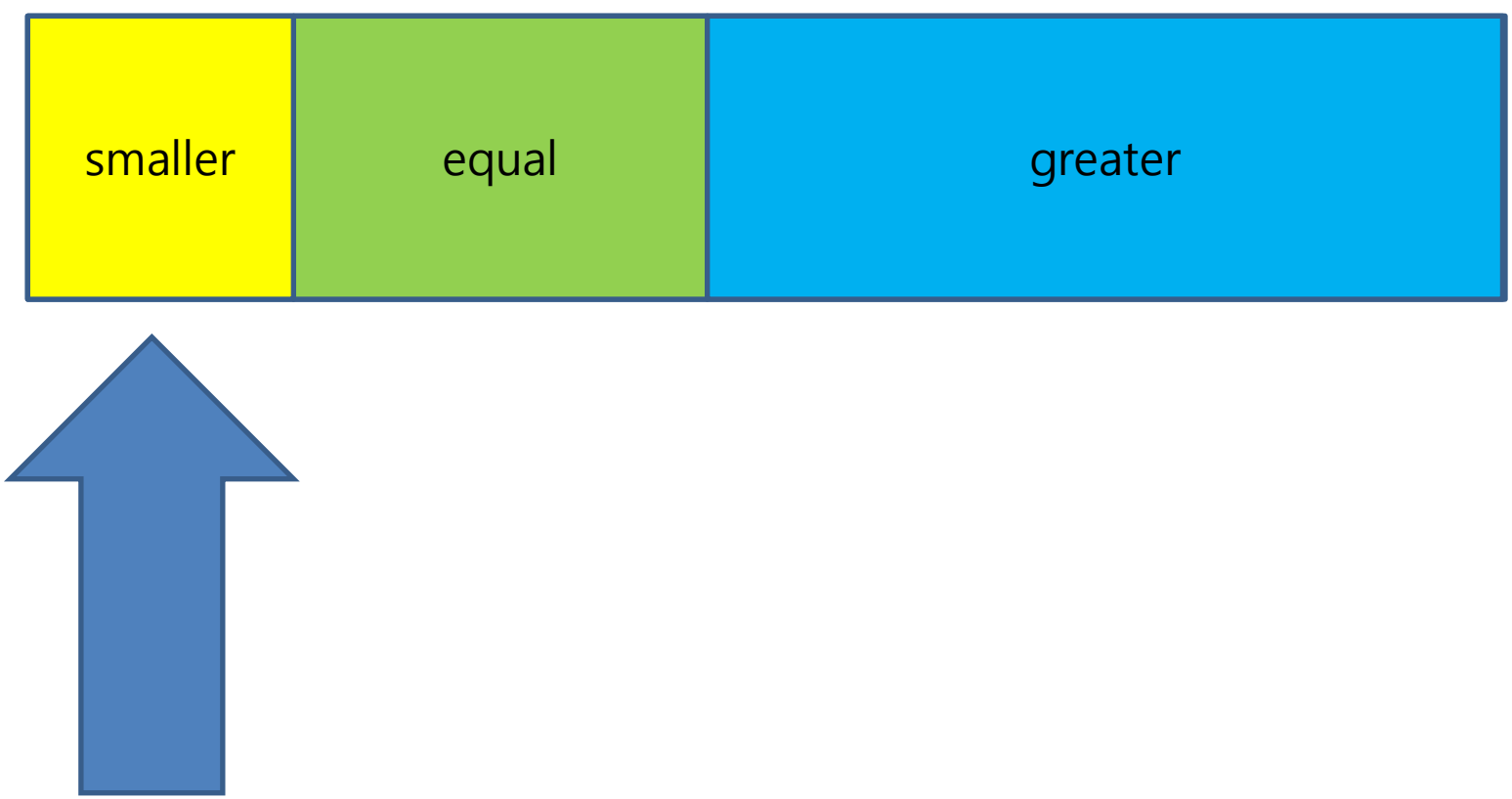

K 가 x번째: x <= len(greater)로 크다면
findKthLargestElement(greater, k)

# Quickselect algorithm




K 가 x번째: x <= len(greater) + len(equal)
로 크다면
findKthLargestElement(equal, k)

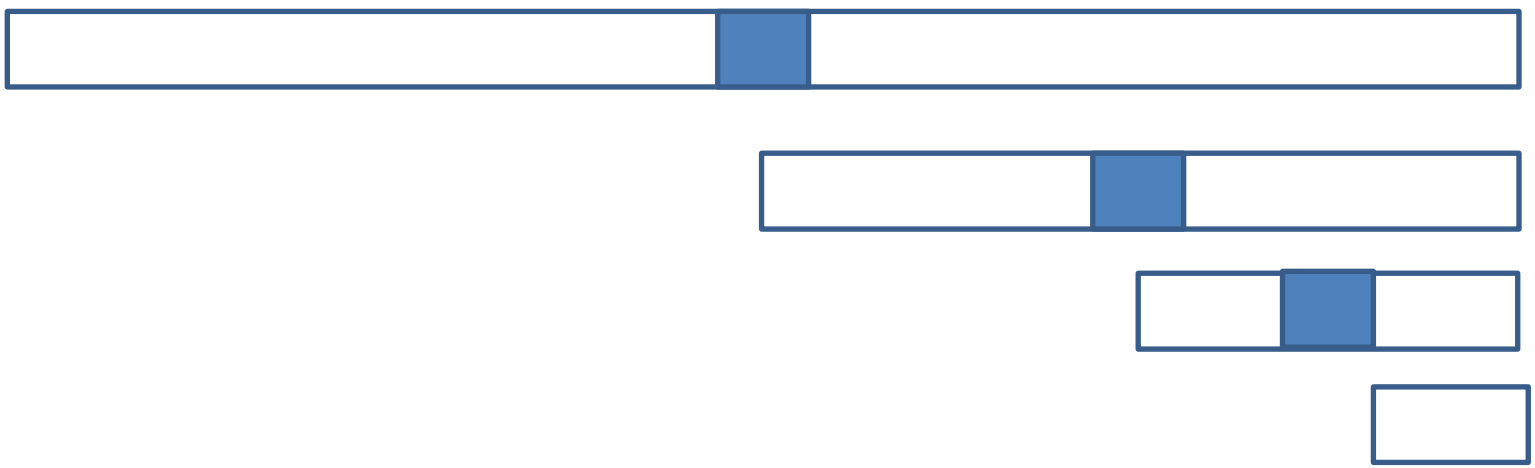
# Quickselect algorithm




K 가 x번째: x >= len(greater) + len(equal)
로 크다면
findKthLargestElement(smaller, k)

# Quickselect algorithm

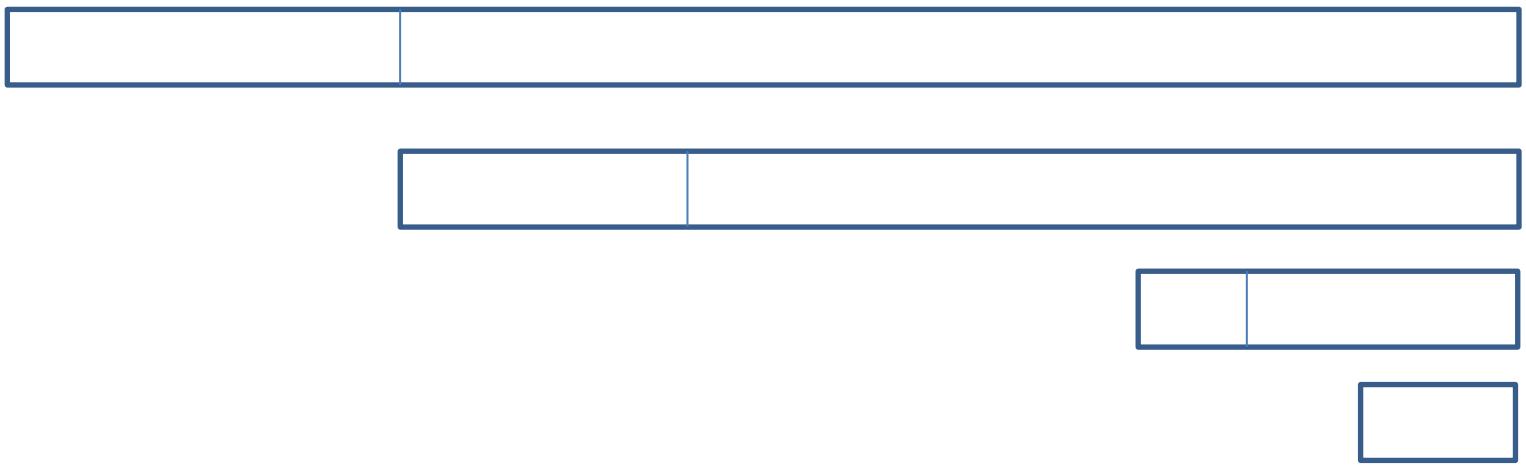
피벗을 너무 잘 골라서 항상 N/2 씩 쪼개어지는 경우



$$N + \frac{N}{2} + \frac{N}{4} + \cdots + 1 = N(1 + \frac{1}{2} + \frac{1}{4} + \cdots) = 2N$$

# Quickselect algorithm

피벗을 적당히 잘 골라서 항상 3/4N 씩 쪼개어지는 경우



$$N(1+\tfrac{3}{4}+(\tfrac{3}{4})^2+\cdots)=N\tfrac{1}{1-\frac{3}{4}}=4N$$

# Quickselect algorithm

최악의 경우 :


$$N + (N-1) + (N-2) + \cdots = \mathcal{O}(N^2)$$

# Quickselect algorithm

문제 해결 방법: Quick Sort 동일

정렬된 상태 input array에서의 문제점 해결?

Using random pivoting

```
RANDOMIZED-PARTITION(A, p, r)
1  i = RANDOM(p, r)
2  exchange A[r] with A[i]
3  return PARTITION(A, p, r)


RANDOMIZED-QUICKSORT(A, p, r)
1  if p < r
2       q = RANDOMIZED-PARTITION(A, p, r)
3       RANDOMIZED-QUICKSORT(A, p, q − 1)
4       RANDOMIZED-QUICKSORT(A, q + 1, r)
```

# Quickselect algorithm

```python
class Solution:
    def findKthLargest(self, nums: List[int], k: int) -> int:
        """Quickselect"""
        pivot = choice(nums)

        greater, equal, smaller = [], [], []
        for n in nums:
            if n > pivot:
                greater.append(n)
            elif n == pivot:
                equal.append(n)
            else:
                smaller.append(n)

        if len(greater) >= k:
            return self.findKthLargest(greater, k)

        if len(greater) + len(equal) >= k:
            return equal[0]

        return self.findKthLargest(smaller, k - len(greater) - len(equal))
```

# Summary

- **Problem 5.18**
- **Quickselect Algorithm (using Random Pivoting )**

# 들어 주셔서 감사합니다