

정렬되지 않은 배열에서 k번째 큰 요소를 추출하라.

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

Output: 5

1.정렬

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

2.heapq

import heapq

```
class Solution:
    def findKthLargest(self, nums: List[int], k: int) -> int:
        nums = [[-n, n] for n in nums]
        heapq.heapify(nums)
        for _ in range(k):
            result = heapq.heappop(nums)[1]
        return result
```

import heapq

```
class Solution:
    def findKthLargest(self, nums: List[int], k: int) -> int:
        heap = []
        for n in nums:
            heapq.heappush(heap, -n)
        for _ in range(k - 1):
            heapq.heappop(heap)
        return -heapq.heappop(heap)
```

import heapq

```
class Solution:
    def findKthLargest(self, nums: List[int], k: int) -> int:
        heapq.heapify(nums)
        for _ in range(len(nums) - k):
            heapq.heappop(nums)
        return heapq.heappop(nums)
```

import heapq

```
class Solution:
    def findKthLargest(self, nums: List[int], k: int) -> int:
        return heapq.nlargest(k, nums)[-1]
```

▸ nlargest, nsmallest