Problem 1985: Find the Kth Largest Integer in the Array

Problem Information

Difficulty: Medium
Acceptance Rate: 0.00%
Paid Only: No

Problem Description

You are given an array of strings

nums

and an integer

k

. Each string in

nums

represents an integer without leading zeros.

Return

the string that represents the

k

th

largest integer

in

nums

.

Note

: Duplicate numbers should be counted distinctly. For example, if

nums

is

["1","2","2"]

,

"2"

is the first largest integer,

"2"

is the second-largest integer, and

"1"

is the third-largest integer.

Example 1:

Input:

nums = ["3","6","7","10"], k = 4

Output:

"3"

Explanation:

The numbers in nums sorted in non-decreasing order are ["3","6","7","10"]. The 4

th

largest integer in nums is "3".

Example 2:

Input:

nums = ["2","21","12","1"], k = 3

Output:

"2"

Explanation:

The numbers in nums sorted in non-decreasing order are ["1","2","12","21"]. The 3

rd

largest integer in nums is "2".

Example 3:

Input:

nums = ["0","0"], k = 2

Output:

"0"

Explanation:

The numbers in nums sorted in non-decreasing order are ["0","0"]. The 2 nd

largest integer in nums is "0".

Constraints:

1 <= k <= nums.length <= 10

4

1 <= nums[i].length <= 100

nums[i]

consists of only digits.

nums[i]

will not have any leading zeros.

Code Snippets

C++:

```
class Solution {
public:
string kthLargestNumber(vector<string>& nums, int k) {

}
};
```

Java:

```
class Solution {
public String kthLargestNumber(String[] nums, int k) {

}
}
```

Python3:

```
class Solution:
def kthLargestNumber(self, nums: List[str], k: int) -> str:
```

Python:

```
class Solution(object):
def kthLargestNumber(self, nums, k):
"""
:type nums: List[str]
:type k: int
:rtype: str
"""
```

JavaScript:

```
/**
* @param {string[]} nums
* @param {number} k
* @return {string}
*/
var kthLargestNumber = function(nums, k) {

};
```

TypeScript:

```
function kthLargestNumber(nums: string[], k: number): string {

};
```

C#:

```
public class Solution {
public string KthLargestNumber(string[] nums, int k) {

}
}
```

C:

```
char* kthLargestNumber(char** nums, int numsSize, int k) {

}
```

Go:

```
func kthLargestNumber(nums []string, k int) string {

}
```

Kotlin:

```
class Solution {
fun kthLargestNumber(nums: Array<String>, k: Int): String {

}
}
```

Swift:

```
class Solution {
func kthLargestNumber(_ nums: [String], _ k: Int) -> String {

}
}
```

Rust:

```
impl Solution {
pub fn kth_largest_number(nums: Vec<String>, k: i32) -> String {

}
}
```

Ruby:

```
# @param {String[]} nums
# @param {Integer} k
# @return {String}
def kth_largest_number(nums, k)

end
```

PHP:

```
class Solution {

/**
* @param String[] $nums
* @param Integer $k
* @return String
*/
function kthLargestNumber($nums, $k) {

}
}
```

Dart:

```
class Solution {
String kthLargestNumber(List<String> nums, int k) {

}
}
```

Scala:

```
object Solution {
def kthLargestNumber(nums: Array[String], k: Int): String = {

}
}
```

Elixir:

```
defmodule Solution do
@spec kth_largest_number(nums :: [String.t], k :: integer) :: String.t
def kth_largest_number(nums, k) do

end
end
```

Erlang:

```
-spec kth_largest_number(Nums :: [unicode:unicode_binary()], K :: integer())
-> unicode:unicode_binary().
```

```
kth_largest_number(Nums, K) ->
.
```

Racket:

```
(define/contract (kth-largest-number nums k)
(-> (listof string?) exact-integer? string?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Find the Kth Largest Integer in the Array
* Difficulty: Medium
* Tags: array, string, sort, queue, heap
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

class Solution {
public:
string kthLargestNumber(vector<string>& nums, int k) {

}
};
```

Java Solution:

```
/**
* Problem: Find the Kth Largest Integer in the Array
* Difficulty: Medium
* Tags: array, string, sort, queue, heap
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
```

```
*/

class Solution {
public String kthLargestNumber(String[] nums, int k) {

}
}
```

Python3 Solution:

```
"""
Problem: Find the Kth Largest Integer in the Array
Difficulty: Medium
Tags: array, string, sort, queue, heap

Approach: Use two pointers or sliding window technique
Time Complexity: O(n) or O(n log n)
Space Complexity: O(1) to O(n) depending on approach
"""

class Solution:
def kthLargestNumber(self, nums: List[str], k: int) -> str:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def kthLargestNumber(self, nums, k):
"""
:type nums: List[str]
:type k: int
:rtype: str
"""
```

JavaScript Solution:

```
/**
* Problem: Find the Kth Largest Integer in the Array
* Difficulty: Medium
* Tags: array, string, sort, queue, heap
```

```
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

/**
* @param {string[]} nums
* @param {number} k
* @return {string}
*/
var kthLargestNumber = function(nums, k) {

};
```

TypeScript Solution:

```
/**
* Problem: Find the Kth Largest Integer in the Array
* Difficulty: Medium
* Tags: array, string, sort, queue, heap
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

function kthLargestNumber(nums: string[], k: number): string {

};
```

C# Solution:

```
/*
* Problem: Find the Kth Largest Integer in the Array
* Difficulty: Medium
* Tags: array, string, sort, queue, heap
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
```

```
*/

public class Solution {
public string KthLargestNumber(string[] nums, int k) {

}
}
```

C Solution:

```
/*
* Problem: Find the Kth Largest Integer in the Array
* Difficulty: Medium
* Tags: array, string, sort, queue, heap
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

char* kthLargestNumber(char** nums, int numsSize, int k) {

}
```

Go Solution:

```
// Problem: Find the Kth Largest Integer in the Array
// Difficulty: Medium
// Tags: array, string, sort, queue, heap
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

func kthLargestNumber(nums []string, k int) string {

}
```

Kotlin Solution:

```
class Solution {
fun kthLargestNumber(nums: Array<String>, k: Int): String {

}
}
```

Swift Solution:

```
class Solution {
func kthLargestNumber(_ nums: [String], _ k: Int) -> String {

}
}
```

Rust Solution:

```
// Problem: Find the Kth Largest Integer in the Array
// Difficulty: Medium
// Tags: array, string, sort, queue, heap
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

impl Solution {
pub fn kth_largest_number(nums: Vec<String>, k: i32) -> String {

}
}
```

Ruby Solution:

```
# @param {String[]} nums
# @param {Integer} k
# @return {String}
def kth_largest_number(nums, k)

end
```

PHP Solution:

```
class Solution {

/**
* @param String[] $nums
* @param Integer $k
* @return String
*/
function kthLargestNumber($nums, $k) {

}
}
```

Dart Solution:

```
class Solution {
String kthLargestNumber(List<String> nums, int k) {

}
}
```

Scala Solution:

```
object Solution {
def kthLargestNumber(nums: Array[String], k: Int): String = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec kth_largest_number(nums :: [String.t], k :: integer) :: String.t
def kth_largest_number(nums, k) do

end
end
```

Erlang Solution:

```
-spec kth_largest_number(Nums :: [unicode:unicode_binary()], K :: integer())
-> unicode:unicode_binary().
kth_largest_number(Nums, K) ->
```

```
.
```

Racket Solution:

```
(define/contract (kth-largest-number nums k)
(-> (listof string?) exact-integer? string?)
)
```