Problem 3109: Find the Index of Permutation

Problem Information

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

Problem Description

Given an array

perm

of length

n

which is a permutation of

[1, 2, ..., n]

, return the index of

perm

in the

lexicographically sorted

array of all of the permutations of

[1, 2, ..., n]

.

Since the answer may be very large, return it

modulo

10

9

+ 7

.

Example 1:

Input:

perm = [1,2]

Output:

0

Explanation:

There are only two permutations in the following order:

[1,2]

,

[2,1]

And

[1,2]

is at index 0.

Example 2:

Input:

perm = [3,1,2]

Output:

4

Explanation:

There are only six permutations in the following order:

[1,2,3]

,

[1,3,2]

,

[2,1,3]

,

[2,3,1]

,

[3,1,2]

,

[3,2,1]

And

[3,1,2]

is at index 4.

Constraints:

1 <= n == perm.length <= 10

5

perm

is a permutation of

[1, 2, ..., n]

.

Code Snippets

C++:

```
class Solution {
public:
int getPermutationIndex(vector<int>& perm) {

}
};
```

Java:

```
class Solution {
public int getPermutationIndex(int[] perm) {

}
}
```

Python3:

```
class Solution:
def getPermutationIndex(self, perm: List[int]) -> int:
```

Python:

```
class Solution(object):
def getPermutationIndex(self, perm):
"""
:type perm: List[int]
:rtype: int
"""
```

JavaScript:

```
/**
* @param {number[]} perm
* @return {number}
*/
var getPermutationIndex = function(perm) {

};
```

TypeScript:

```
function getPermutationIndex(perm: number[]): number {

};
```

C#:

```
public class Solution {
public int GetPermutationIndex(int[] perm) {

}
}
```

C:

```
int getPermutationIndex(int* perm, int permSize) {

}
```

Go:

```
func getPermutationIndex(perm []int) int {
```

```
}
```

Kotlin:

```
class Solution {
fun getPermutationIndex(perm: IntArray): Int {

}
}
```

Swift:

```
class Solution {
func getPermutationIndex(_ perm: [Int]) -> Int {

}
}
```

Rust:

```
impl Solution {
pub fn get_permutation_index(perm: Vec<i32>) -> i32 {

}
}
```

Ruby:

```
# @param {Integer[]} perm
# @return {Integer}
def get_permutation_index(perm)

end
```

PHP:

```
class Solution {

/**
* @param Integer[] $perm
* @return Integer
*/
```

```
function getPermutationIndex($perm) {

}
}
```

Dart:

```
class Solution {
int getPermutationIndex(List<int> perm) {

}
}
```

Scala:

```
object Solution {
def getPermutationIndex(perm: Array[Int]): Int = {

}
}
```

Elixir:

```
defmodule Solution do
@spec get_permutation_index(perm :: [integer]) :: integer
def get_permutation_index(perm) do

end
end
```

Erlang:

```
-spec get_permutation_index(Perm :: [integer()]) -> integer().
get_permutation_index(Perm) ->
.
```

Racket:

```
(define/contract (get-permutation-index perm)
(-> (listof exact-integer?) exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Find the Index of Permutation
* Difficulty: Medium
* Tags: array, tree, graph, sort, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

class Solution {
public:
int getPermutationIndex(vector<int>& perm) {

}
};
```

Java Solution:

```
/**
* Problem: Find the Index of Permutation
* Difficulty: Medium
* Tags: array, tree, graph, sort, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

class Solution {
public int getPermutationIndex(int[] perm) {

}
}
```

Python3 Solution:

```
"""
Problem: Find the Index of Permutation
Difficulty: Medium
Tags: array, tree, graph, sort, search

Approach: Use two pointers or sliding window technique
Time Complexity: O(n) or O(n log n)
Space Complexity: O(h) for recursion stack where h is height
"""

class Solution:
def getPermutationIndex(self, perm: List[int]) -> int:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def getPermutationIndex(self, perm):
"""
:type perm: List[int]
:rtype: int
"""
```

JavaScript Solution:

```
/**
* Problem: Find the Index of Permutation
* Difficulty: Medium
* Tags: array, tree, graph, sort, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

/**
* @param {number[]} perm
* @return {number}
*/
var getPermutationIndex = function(perm) {
```

```
};
```

TypeScript Solution:

```
/**
* Problem: Find the Index of Permutation
* Difficulty: Medium
* Tags: array, tree, graph, sort, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

function getPermutationIndex(perm: number[]): number {

};
```

C# Solution:

```
/*
* Problem: Find the Index of Permutation
* Difficulty: Medium
* Tags: array, tree, graph, sort, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

public class Solution {
public int GetPermutationIndex(int[] perm) {

}
}
```

C Solution:

```
/*
* Problem: Find the Index of Permutation
* Difficulty: Medium
```

```
* Tags: array, tree, graph, sort, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

int getPermutationIndex(int* perm, int permSize) {

}
```

Go Solution:

```
// Problem: Find the Index of Permutation
// Difficulty: Medium
// Tags: array, tree, graph, sort, search
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(h) for recursion stack where h is height

func getPermutationIndex(perm []int) int {

}
```

Kotlin Solution:

```
class Solution {
fun getPermutationIndex(perm: IntArray): Int {

}
}
```

Swift Solution:

```
class Solution {
func getPermutationIndex(_ perm: [Int]) -> Int {

}
}
```

Rust Solution:

```
// Problem: Find the Index of Permutation
// Difficulty: Medium
// Tags: array, tree, graph, sort, search
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(h) for recursion stack where h is height

impl Solution {
pub fn get_permutation_index(perm: Vec<i32>) -> i32 {

}
}
```

Ruby Solution:

```
# @param {Integer[]} perm
# @return {Integer}
def get_permutation_index(perm)

end
```

PHP Solution:

```
class Solution {

/**
* @param Integer[] $perm
* @return Integer
*/
function getPermutationIndex($perm) {

}
}
```

Dart Solution:

```
class Solution {
int getPermutationIndex(List<int> perm) {
```

```
}
}
```

Scala Solution:

```
object Solution {
def getPermutationIndex(perm: Array[Int]): Int = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec get_permutation_index(perm :: [integer]) :: integer
def get_permutation_index(perm) do

end
end
```

Erlang Solution:

```
-spec get_permutation_index(Perm :: [integer()]) -> integer().
get_permutation_index(Perm) ->
.
```

Racket Solution:

```
(define/contract (get-permutation-index perm)
(-> (listof exact-integer?) exact-integer?)
)
```