Array : Video - 83

Leetcode
- 3034
- 3036
Medium
Hard

Weekly Contest - 384

Facebook
Instagram → codestorywithMIK

(Twitter) → CSwithMIK

codestorywithMIK →

Brute - I. → II (TLE).

3034. Number of Subarrays That Match a Pattern I

Medium | Topics | Companies | Hint

You are given a **0-indexed** integer array `nums` of size `n`, and a **0-indexed** integer array `pattern` of size `m` consisting of integers `-1`, `0`, and `1`.

A subarray `nums[i..j]` of size `m + 1` is said to match the `pattern` if the following conditions hold for each element `pattern[k]`:

- `nums[i + k + 1] > nums[i + k]` if `pattern[k] == 1`.
- `nums[i + k + 1] == nums[i + k]` if `pattern[k] == 0`.
- `nums[i + k + 1] < nums[i + k]` if `pattern[k] == -1`.

→ inc.
→ equal
← dec.

{1, -1, 0}

Return *the* ***count*** *of subarrays in* `nums` *that match the* `pattern`.

Constraints:

- `2 <= n == nums.length <= 100`
- `1 <= nums[i] <= 10^9`
- `1 <= m == pattern.length < n`
- `-1 <= pattern[i] <= 1`

PART-I

Constraints:

- `2 <= n == nums.length <= 10^6`
- `1 <= nums[i] <= 10^9`
- `1 <= m == pattern.length < n`
- `-1 <= pattern[i] <= 1`

PART-II

अरे भाई भाई भाई!

Example :- nums = {1, 2, 3, 4, 5, 6}, n = 6

i <= n-m-1

Pattern = {1, 1}, m = 2

nums[i+K+1] nums[i+K]

4 > 3

Output :- 4

Understand Problem

nums = {1, 2, 3, 4, 5, 6} (i = 4), n = 6

Pattern = {1, 1} (indices 0, 1), m = 2

K = 0

1 + 1
1 + 1

i = 0

Pattern[K] = 1 (K = 1)

```
if( nums[i+K+1] > nums[i+K] ) {
    K++;
}
```

Pattern[K] = 0

```
if( nums[i+K+1] == nums[i+K] ) {
    K++;
}
```

Patt[K] = -1

```
if( nums[i+K+1] < nums[i+K] ) {
}
```

Optimal Approach

Knuth-Morris-Pratt KMP String Matching Algorithm | Search Pattern | GFG POTD

[redacted] • 6 hours ago

bhaiya aapki wajah se easy question me atakne wale ne aaj hard question(3036. Number of Subarrays That Match a Pattern II) bna liya ...thanks for everything in coding bhaiya ...please continue this playlist ...

Translate to English

nums = {1, 2, 3, 4, 5, 6}, n = 6

Pattern = {1, 1}, m = 2

Rule-1

{ 1 , 1 , 1 , 1 , 1 } → size = n-1

nums = { 1, 4, 4, 1, 3, 5, 5, 3 }

Pattern = { 1, 0, -1 }, m = 3

Output = 2

KMP / Z ~ -

T.C = O(m+n)

Pattern = "abc" → m

txt = "bcdeabcdabc"

↳ n