5959. Minimum Operations to Make the Array K-Increasing

My Submissions (/contest/weekly-contest-272/problems/minimum-operations-to-make-the-array-k-increasing/submissions/)

Back to Contest (/contest/weekly-contest-272/)

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Hard

You are given a **0-indexed** array `arr` consisting of `n` positive integers, and a positive integer `k`.

The array `arr` is called **K-increasing** if `arr[i-k] <= arr[i]` holds for every index `i`, where `k <= i <= n-1`.

- For example, `arr = [4, 1, 5, 2, 6, 2]` is K-increasing for `k = 2` because:
  - `arr[0] <= arr[2] (4 <= 5)`
  - `arr[1] <= arr[3] (1 <= 2)`
  - `arr[2] <= arr[4] (5 <= 6)`
  - `arr[3] <= arr[5] (2 <= 2)`
- However, the same `arr` is not K-increasing for `k = 1` (because `arr[0] > arr[1]`) or `k = 3` (because `arr[0] > arr[3]`).

In one **operation**, you can choose an index `i` and **change** `arr[i]` into **any** positive integer.

Return *the **minimum number of operations** required to make the array K-increasing for the given* `k`.

Example 1:

```
Input: arr = [5,4,3,2,1], k = 1
Output: 4
Explanation:
For k = 1, the resultant array has to be non-decreasing.
Some of the K-increasing arrays that can be formed are [5,6,7,8,9], [1,1,1,1,1], [2,2,3,4,4]. All of them require 4 ope
It is suboptimal to change the array to, for example, [6,7,8,9,10] because it would take 5 operations.
It can be shown that we cannot make the array K-increasing in less than 4 operations.
```

Example 2:

```
Input: arr = [4,1,5,2,6,2], k = 2
Output: 0
Explanation:
This is the same example as the one in the problem description.
Here, for every index i where 2 <= i <= 5, arr[i-2] <= arr[i].
Since the given array is already K-increasing, we do not need to perform any operations.
```

Example 3:

```
Input: arr = [4,1,5,2,6,2], k = 3
Output: 2
Explanation:
Indices 3 and 5 are the only ones not satisfying arr[i-3] <= arr[i] for 3 <= i <= 5.
One of the ways we can make the array K-increasing is by changing arr[3] to 4 and arr[5] to 5.
The array will now be [4,1,5,4,6,5].
Note that there can be other ways to make the array K-increasing, but none of them require less than 2 operations.
```

Constraints:

- `1 <= arr.length <= 10^5`
- `1 <= arr[i], k <= arr.length`

JavaScript

```
function Bisect() {
    return { insort_right, insort_left, bisect_left, bisect_right }
    function insort_right(a, x, lo = 0, hi = null) {
        lo = bisect_right(a, x, lo, hi);
        a.splice(lo, 0, x);
    }
    function bisect_right(a, x, lo = 0, hi = null) { // > upper_bound
        if (lo < 0) throw new Error('lo must be non-negative');
        if (hi == null) hi = a.length;
        while (lo < hi) {
            let mid = parseInt((lo + hi) / 2);
            x < a[mid] ? hi = mid : lo = mid + 1;
        }
        return lo;
    }
    function insort_left(a, x, lo = 0, hi = null) {
        lo = bisect_left(a, x, lo, hi);
        a.splice(lo, 0, x);
    }
    function bisect_left(a, x, lo = 0, hi = null) { // >= lower_bound
        if (lo < 0) throw new Error('lo must be non-negative');
        if (hi == null) hi = a.length;
        while (lo < hi) {
            let mid = parseInt((lo + hi) / 2);
            a[mid] < x ? lo = mid + 1 : hi = mid;
        }
        return lo;
    }
}

const kIncreasing = (A, k) => {
    let n = A.length;
    let res = 0, t = Math.ceil(n / k);
    // pr(n, k, "t", t);
    let d = [];
    for (let start = 0; start < k; start++) {
        let re = [], i;
        for (i = start; i + k < n; i += k) re.push(A[i]);
        re.push(A[i])
        d.push(re);
    }
    // pr(d);
    for (const a of d) res += LIS(a);
    return n - res;
};

const LIS = (a) => {
    // pr(a);
    let bi = new Bisect(), dp = [], n = a.length;
    for (const x of a) {
        let idx = bi.bisect_right(dp, x);
        if (idx == n) {
            dp.push(x);
        } else {
            dp[idx] = x;
        }
    }
    // pr(dp);
    return dp.length;
};
```

Custom Testcase Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/603863702/) More Details > (/submissions/detail/603863702/)

Share your acceptance!

1