1868. Product of Two Run-Length Encoded Arrays Premium

Medium | Topics | Companies | Hint

Run-length encoding is a compression algorithm that allows for an integer array `nums` with many segments of **consecutive repeated** numbers to be represented by a (generally smaller) 2D array `encoded`. Each `encoded[i] = [val_i, freq_i]` describes the `i^th` segment of repeated numbers in `nums` where `val_i` is the value that is repeated `freq_i` times.

- For example, `nums = [1,1,1,2,2,2,2,2]` is represented by the **run-length encoded** array `encoded = [[1,3],[2,5]]`. Another way to read this is "three `1`'s followed by five `2`'s".

The **product** of two run-length encoded arrays `encoded1` and `encoded2` can be calculated using the following steps:

1. **Expand** both `encoded1` and `encoded2` into the full arrays `nums1` and `nums2` respectively.
2. Create a new array `prodNums` of length `nums1.length` and set `prodNums[i] = nums1[i] * nums2[i]`.
3. **Compress** `prodNums` into a run-length encoded array and return it.

You are given two **run-length encoded** arrays `encoded1` and `encoded2` representing full arrays `nums1` and `nums2` respectively. Both `nums1` and `nums2` have the **same length**. Each `encoded1[i] = [val_i, freq_i]` describes the `i^th` segment of `nums1`, and each `encoded2[j] = [val_j, freq_j]` describes the `j^th` segment of `nums2`.

Return *the **product** of* `encoded1` *and* `encoded2`.

Note: Compression should be done such that the run-length encoded array has the **minimum** possible length.

Example 1:

> **Input:** encoded1 = [[1,3],[2,3]], encoded2 = [[6,3],[3,3]]
> **Output:** [[6,6]]
> **Explanation:** encoded1 expands to [1,1,1,2,2,2] and encoded2 expands to [6,6,6,3,3,3].
> prodNums = [6,6,6,6,6,6], which is compressed into the run-length encoded array [[6,6]].

Example 2:

> **Input:** encoded1 = [[1,3],[2,1],[3,2]], encoded2 = [[2,3],[3,3]]
> **Output:** [[2,3],[6,1],[9,2]]
> **Explanation:** encoded1 expands to [1,1,1,2,3,3] and encoded2 expands to [2,2,2,3,3,3].
> prodNums = [2,2,2,6,9,9], which is compressed into the run-length encoded array [[2,3],[6,1],[9,2]].

Constraints:

- `1 <= encoded1.length, encoded2.length <= 10^5`
- `encoded1[i].length == 2`
- `encoded2[j].length == 2`
- `1 <= val_i, freq_i <= 10^4` for each `encoded1[i]`.
- `1 <= val_j, freq_j <= 10^4` for each `encoded2[j]`.
- The full arrays that `encoded1` and `encoded2` represent are the same length.

Seen this question in a real interview before? 1/5

Yes No

Accepted 50.3K | Submissions 85.5K | Acceptance Rate 58.8%

Topics

Array Two Pointers

Companies

0 - 3 months

Meta 7

6 months ago

Amazon 2 Yandex 2

Hint 1

Keep track of the indices on both RLE arrays and join the parts together.

Hint 2

What is the maximum number of segments if we took the minimum number of elements left on both the current segments every time?

Discussion (10)