Problem 3269: Constructing Two Increasing Arrays

Problem Information

Difficulty: Hard
Acceptance Rate: 62.45%
Paid Only: Yes
Tags: Array, Dynamic Programming

Problem Description

Given 2 integer arrays `nums1` and `nums2` consisting only of 0 and 1, your task is to calculate the **minimum** possible **largest** number in arrays `nums1` and `nums2`, after doing the following.

Replace every 0 with an _even positive integer_ and every 1 with an _odd positive integer_. After replacement, both arrays should be **increasing** and each integer should be used **at most** once.

Return the _minimum possible largest number_ after applying the changes.

Example 1:

Input: nums1 = [], nums2 = [1,0,1,1]

Output: 5

Explanation:

After replacing, `nums1 = []`, and `nums2 = [1, 2, 3, 5]`.

Example 2:

Input: nums1 = [0,1,0,1], nums2 = [1,0,0,1]

Output: 9

Explanation:

One way to replace, having 9 as the largest element is `nums1 = [2, 3, 8, 9]`, and `nums2 = [1, 4, 6, 7]`.

Example 3:

Input: nums1 = [0,1,0,0,1], nums2 = [0,0,0,1]

Output: 13

Explanation:

One way to replace, having 13 as the largest element is `nums1 = [2, 3, 4, 6, 7]`, and `nums2 = [8, 10, 12, 13]`.

Constraints:

* `0 <= nums1.length <= 1000` * `1 <= nums2.length <= 1000` * `nums1` and `nums2` consist only of 0 and 1.

Code Snippets

C++:

```
class Solution {
public:
int minLargest(vector<int>& nums1, vector<int>& nums2) {

}
};
```

Java:

```
class Solution {
public int minLargest(int[] nums1, int[] nums2) {
}
}
```

Python3:

```
class Solution:
def minLargest(self, nums1: List[int], nums2: List[int]) -> int:
```