

Problem 2344: Minimum Deletions to Make Array Divisible

Problem Information

Difficulty: Hard

Acceptance Rate: 58.84%

Paid Only: No

Tags: Array, Math, Sorting, Heap (Priority Queue), Number Theory

Problem Description

You are given two positive integer arrays `nums` and `numsDivide`. You can delete any number of elements from `nums`.

Return `the minimum` number of deletions such that the `smallest` element in `nums` `divides` all the elements of `numsDivide`. If this is not possible, return `-1`.

Note that an integer `x` divides `y` if `y % x == 0`.

Example 1:

Input: `nums = [2,3,2,4,3]`, `numsDivide = [9,6,9,3,15]` **Output:** 2 **Explanation:** The smallest element in `[2,3,2,4,3]` is 2, which does not divide all the elements of `numsDivide`. We use 2 deletions to delete the elements in `nums` that are equal to 2 which makes `nums = [3,4,3]`. The smallest element in `[3,4,3]` is 3, which divides all the elements of `numsDivide`. It can be shown that 2 is the minimum number of deletions needed.

Example 2:

Input: `nums = [4,3,6]`, `numsDivide = [8,2,6,10]` **Output:** -1 **Explanation:** We want the smallest element in `nums` to divide all the elements of `numsDivide`. There is no way to delete elements from `nums` to allow this.

Constraints:

`1 <= nums.length, numsDivide.length <= 105` `1 <= nums[i], numsDivide[i] <= 109`

Code Snippets

C++:

```
class Solution {  
public:  
    int minOperations(vector<int>& nums, vector<int>& numsDivide) {  
  
    }  
};
```

Java:

```
class Solution {  
    public int minOperations(int[] nums, int[] numsDivide) {  
  
    }  
}
```

Python3:

```
class Solution:  
    def minOperations(self, nums: List[int], numsDivide: List[int]) -> int:
```