Problem 3533: Concatenated Divisibility

Problem Information

Difficulty: Hard
Acceptance Rate: 30.00%
Paid Only: No
Tags: Array, Dynamic Programming, Bit Manipulation, Bitmask

Problem Description

You are given an array of positive integers `nums` and a positive integer `k`.

A permutation of `nums` is said to form a **divisible concatenation** if, when you _concatenate_ _the decimal representations_ of the numbers in the order specified by the permutation, the resulting number is **divisible by** `k`.

Return the **lexicographically smallest** permutation (when considered as a list of integers) that forms a **divisible concatenation**. If no such permutation exists, return an empty list.

Example 1:

Input: nums = [3,12,45], k = 5

Output: [3,12,45]

Explanation:

Permutation | Concatenated Value | Divisible by 5 ---|---|--- [3, 12, 45] | 31245 | Yes [3, 45, 12] | 34512 | No [12, 3, 45] | 12345 | Yes [12, 45, 3] | 12453 | No [45, 3, 12] | 45312 | No [45, 12, 3] | 45123 | No The lexicographically smallest permutation that forms a divisible concatenation is `[3,12,45]`.

Example 2:

Input: nums = [10,5], k = 10

Output: [5,10]

Explanation:

Permutation | Concatenated Value | Divisible by 10 ---|---|--- [5, 10] | 510 | Yes [10, 5] | 105 | No The lexicographically smallest permutation that forms a divisible concatenation is `[5,10]`.

Example 3:

Input: nums = [1,2,3], k = 5

Output: []

Explanation:

Since no permutation of `nums` forms a valid divisible concatenation, return an empty list.

Constraints:

* `1 <= nums.length <= 13` * `1 <= nums[i] <= 105` * `1 <= k <= 100`

Code Snippets

C++:

```
class Solution {
public:
vector<int> concatenatedDivisibility(vector<int>& nums, int k) {

}
};
```

Java:

```
class Solution {
public int[] concatenatedDivisibility(int[] nums, int k) {

}
}
```

Python3:

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