

Problem 2342: Max Sum of a Pair With Equal Sum of Digits

Problem Information

Difficulty: Medium

Acceptance Rate: 65.94%

Paid Only: No

Tags: Array, Hash Table, Sorting, Heap (Priority Queue)

Problem Description

You are given a **0-indexed** array `nums` consisting of **positive** integers. You can choose two indices `i` and `j`, such that `i != j`, and the sum of digits of the number `nums[i]` is equal to that of `nums[j]`.

Return the **maximum** value of `nums[i] + nums[j]` that you can obtain over all possible indices `i` and `j` that satisfy the conditions. If no such pair of indices exists, return -1.

Example 1:

Input: `nums = [18,43,36,13,7]` **Output:** 54 **Explanation:** The pairs (i, j) that satisfy the conditions are: - (0, 2), both numbers have a sum of digits equal to 9, and their sum is 18 + 36 = 54. - (1, 4), both numbers have a sum of digits equal to 7, and their sum is 43 + 7 = 50. So the maximum sum that we can obtain is 54.

Example 2:

Input: `nums = [10,12,19,14]` **Output:** -1 **Explanation:** There are no two numbers that satisfy the conditions, so we return -1.

Constraints:

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

Code Snippets

C++:

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

Java:

```
class Solution {  
    public int maximumSum(int[] nums) {  
  
    }  
}
```

Python3:

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