

Problem 2578: Split With Minimum Sum

Problem Information

Difficulty: Easy

Acceptance Rate: 72.88%

Paid Only: No

Tags: Math, Greedy, Sorting

Problem Description

Given a positive integer `num`, split it into two non-negative integers `num1` and `num2` such that:

* The concatenation of `num1` and `num2` is a permutation of `num`. * In other words, the sum of the number of occurrences of each digit in `num1` and `num2` is equal to the number of occurrences of that digit in `num`. * `num1` and `num2` can contain leading zeros.

Return the **minimum** possible sum of `num1` and `num2`.

Notes:

* It is guaranteed that `num` does not contain any leading zeros. * The order of occurrence of the digits in `num1` and `num2` may differ from the order of occurrence of `num`.

Example 1:

Input: `num = 4325` **Output:** 59 **Explanation:** We can split 4325 so that `num1` is 24 and `num2` is 35, giving a sum of 59. We can prove that 59 is indeed the minimal possible sum.

Example 2:

Input: `num = 687` **Output:** 75 **Explanation:** We can split 687 so that `num1` is 68 and `num2` is 7, which would give an optimal sum of 75.

Constraints:

```
*`10 <= num <= 109`
```

Code Snippets

C++:

```
class Solution {  
public:  
    int splitNum(int num) {  
  
    }  
};
```

Java:

```
class Solution {  
    public int splitNum(int num) {  
  
    }  
}
```

Python3:

```
class Solution:  
    def splitNum(self, num: int) -> int:
```