5984. Minimum Sum of Four Digit Number After Splitting Digits

My Submissions (/contest/biweekly-contest-71/problems/minimum-sum-of-four-digit-number-after-splitting-digits/submissions/)

Back to Contest (/contest/biweekly-contest-71/)

You are given a **positive** integer `num` consisting of exactly four digits. Split `num` into two new integers `new1` and `new2` by using the **digits** found in `num`. **Leading zeros** are allowed in `new1` and `new2`, and **all** the digits found in `num` must be used.

- For example, given `num = 2932`, you have the following digits: two `2`'s, one `9` and one `3`. Some of the possible pairs `[new1, new2]` are `[22, 93]`, `[23, 92]`, `[223, 9]` and `[2, 329]`.

Return *the **minimum** possible sum of* `new1` *and* `new2`.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Easy

Example 1:

```
Input: num = 2932
Output: 52
Explanation: Some possible pairs [new1, new2] are [29, 23], [223, 9], etc.
The minimum sum can be obtained by the pair [29, 23]: 29 + 23 = 52.
```

Example 2:

```
Input: num = 4009
Output: 13
Explanation: Some possible pairs [new1, new2] are [0, 49], [490, 0], etc.
The minimum sum can be obtained by the pair [4, 9]: 4 + 9 = 13.
```

Constraints:

- `1000 <= num <= 9999`

JavaScript

```javascript
const sortstr = (s) => s.split("").sort((x, y) => x.localeCompare(y)).join("");

const minimumSum = (x) => {
    let s = x + '';
    s = sortstr(s);
    let res1 = (s[0] + s[2] - '0')  + (s[1] + s[3] - '0');
    let res2 = (s[0] + s[3] - '0')  + (s[1] + s[2] - '0');
    return Math.min(res1, res2);
};
```

Custom Testcase Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/635012407/) More Details > (/submissions/detail/635012407/)

Share your acceptance!