

Minimum Sum

🔽 solved by	Senan
🔽 Platform	GeeksForGeeks
🔗 difficulty	Medium
≡ tags	LogicString Manipulation
🗨 language	C++
📅 solved on	@09/11/2024
🔗 link	https://www.geeksforgeeks.org/problems/minimum-sum4058/1
☑ Completion	✓

Intuition

The problem requires finding the smallest possible sum by arranging and adding numbers formed from an array of integers. To achieve the minimum sum, the numbers are sorted in descending order to balance the sum evenly across digits, minimizing the total by pairing large numbers together.

Approach

- Sort the Array:** Start by sorting the array in descending order. This ensures that larger numbers are added first, distributing values in the most significant positions, resulting in a smaller sum.
- Pair Digits:** Traverse the sorted array, taking two elements at a time. Add these two elements along with any carry from the previous step to calculate the current "units" digit and a new carry for the next iteration.
- Construct the Answer String:** Store each calculated units digit as a character in the `answer` string.
- Handle Remaining Carry:** If a carry is left after processing all pairs, append it to the result.
- Remove Leading Zeros:** Trim any trailing zeros added during calculation and resize the string as needed.
- Reverse and Return:** Finally, reverse the `answer` string to obtain the smallest sum in the correct order.

Complexity

Time Complexity:

- $O(n \log n)$: Sorting the array takes $O(n \log n)$.
- $O(n)$: Iterating through the array to construct the result.

Therefore, the total time complexity is $O(n \log n)$.

Space Complexity:

- $O(n)$: For storing the result string.

Code

```

class Solution {
public:
    string minSum(vector<int> &arr) {
        sort(arr.rbegin(), arr.rend());

        string answer(arr.size(), ' ');

        int idx = 0, carry = 0, units = 0;

        for(int i = 0; i < arr.size(); i += 2) {
            int sum = carry + arr[i];

            if(i + 1 < arr.size()) sum += arr[i + 1];
            units = sum % 10;
            carry = sum / 10;

            answer[idx++] = units + '0';
        }

        if(carry) answer[idx++] = carry + '0';

        while(idx > 1 && answer[idx - 1] == '0') idx--;
        answer.resize(idx);

        reverse(answer.begin(), answer.end());

        return answer;
    }
};

```