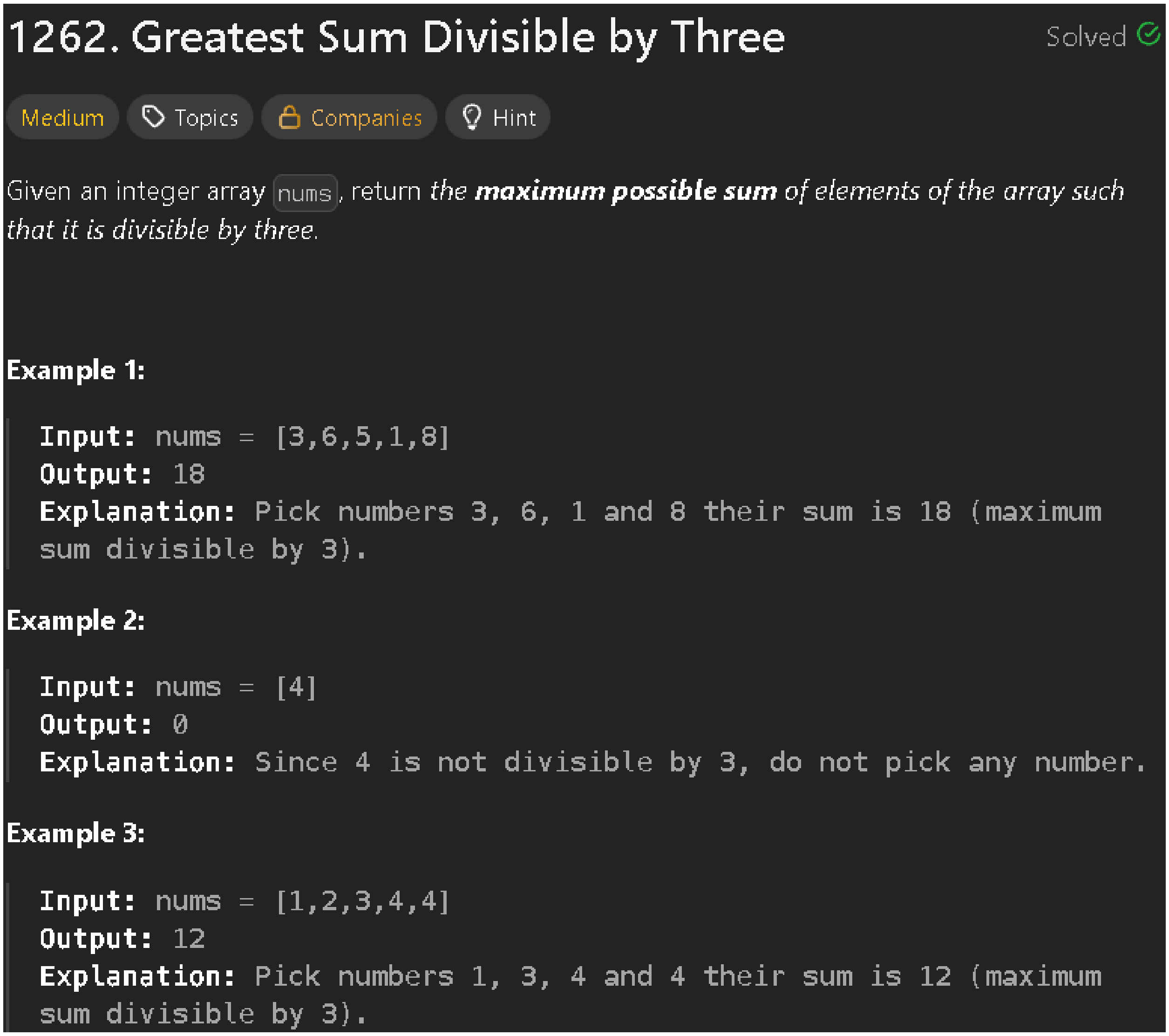


1262. Greatest Sum Divisible by Three

Given an integer array `nums`, return *the* ***maximum possible sum*** *of elements of the array such that it is divisible by three.*

Example 1:

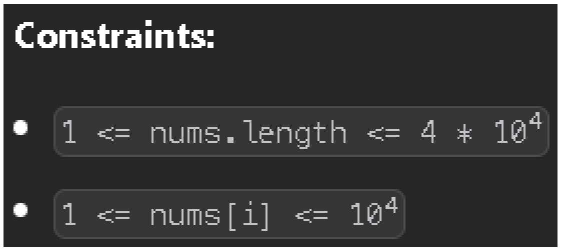
```
Input: nums = [3,6,5,1,8]
Output: 18
Explanation: Pick numbers 3, 6, 1 and 8 their sum is 18 (maximum
sum divisible by 3).
```

Example 2:

```
Input: nums = [4]
Output: 0
Explanation: Since 4 is not divisible by 3, do not pick any number.
```

Example 3:

```
Input: nums = [1,2,3,4,4]
Output: 12
Explanation: Pick numbers 1, 3, 4 and 4 their sum is 12 (maximum
sum divisible by 3).
```



Constraints:

- $1 <= \text{nums.length} <= 4 * 10^4$
- $1 <= \text{nums[i]} <= 10^4$

Python:

```
class Solution:
    def maxSumDivThree(self, nums: List[int]) -> int:
        sum_ = 0

        min1 = min2 = float('inf')
        min11 = min22 = float('inf')
        for x in nums:
            sum_ += x
            r = x % 3

            if r == 1:
                if x < min1:
                    min11, min1 = min1, x
                elif x < min11:
                    min11 = x
            elif r == 2:
                if x < min2:
                    min22, min2 = min2, x
                elif x < min22:
                    min22 = x

        rem = sum_ % 3

        if rem == 0:
            return sum_

        if rem == 1:
            r1 = min1
            r2 = min2 + min22 if min2 < float('inf') and min22 < float('inf') else float('inf')
            remove = min(r1, r2)
            return 0 if remove == float('inf') else sum_ - remove
        else:
            r1 = min2
            r2 = min1 + min11 if min1 < float('inf') and min11 < float('inf') else float('inf')
            remove = min(r1, r2)
            return 0 if remove == float('inf') else sum_ - remove
```

JavaScript:

```
/**
 * @param {number[]} nums
 * @return {number}
 */
var maxSumDivThree = function(nums) {
    let sum = 0;
    const INF = 1e9;
    let r1_min1 = INF, r1_min2 = INF;
    let r2_min1 = INF, r2_min2 = INF;

    for (const x of nums) {
        sum += x;
        const r = x % 3;
        if (r === 1) {
            if (x < r1_min1) {
                r1_min2 = r1_min1;
                r1_min1 = x;
            } else if (x < r1_min2) {
                r1_min2 = x;
            }
        } else if (r === 2) {
            if (x < r2_min1) {
                r2_min2 = r2_min1;
                r2_min1 = x;
            } else if (x < r2_min2) {
                r2_min2 = x;
            }
        }
    }

    const mod = sum % 3;
    if (mod === 0) return sum;

    let removeCost = 1e18;

    if (mod === 1) {
        if (r1_min1 !== INF) removeCost = Math.min(removeCost, r1_min1);
        if (r2_min2 !== INF) removeCost = Math.min(removeCost, r2_min1 + r2_min2);
    } else {
        if (r2_min1 !== INF) removeCost = Math.min(removeCost, r2_min1);
        if (r1_min2 !== INF) removeCost = Math.min(removeCost, r1_min1 + r1_min2);
    }

    if (removeCost >= 1e18) return 0;
    return sum - removeCost;
};
```

Java:

```
class Solution {
    public int maxSumDivThree(int[] nums) {
        int sum = 0;

        int min1 = Integer.MAX_VALUE;
        int min2 = Integer.MAX_VALUE;
        int min11 = Integer.MAX_VALUE;
        int min22 = Integer.MAX_VALUE;
        for (int x : nums) {
            sum += x;
            int r = x % 3;

            if (r == 1) {
                if (x < min1) { min11 = min1; min1 = x; }
                else if (x < min11) min11 = x;
            }
            else if (r == 2) {
                if (x < min2) { min22 = min2; min2 = x; }
                else if (x < min22) min22 = x;
            }
        }

        int rem = sum % 3;

        if (rem == 0) return sum;

        if (rem == 1) {
            int remove1 = min1;
            int remove2 = (min2 == Integer.MAX_VALUE || min22 == Integer.MAX_VALUE)
                            ? Integer.MAX_VALUE : min2 + min22;
            int remove = Math.min(remove1, remove2);
            return (remove == Integer.MAX_VALUE) ? 0 : sum - remove;
        }
        else {
            int remove1 = min2;
            int remove2 = (min1 == Integer.MAX_VALUE || min11 == Integer.MAX_VALUE)
                            ? Integer.MAX_VALUE : min1 + min11;
            int remove = Math.min(remove1, remove2);
            return (remove == Integer.MAX_VALUE) ? 0 : sum - remove;
        }
    }
}
```