1390. Four Divisors

Solved

Medium | Topics | Companies | Hint

Given an integer array `nums`, return *the sum of divisors of the integers in that array that have exactly four divisors*. If there is no such integer in the array, return `0`.

Example 1:

```
Input: nums = [21,4,7]
Output: 32
Explanation:
21 has 4 divisors: 1, 3, 7, 21
4 has 3 divisors: 1, 2, 4
7 has 2 divisors: 1, 7
The answer is the sum of divisors of 21 only.
```

Example 2:

```
Input: nums = [21,21]
Output: 64
```

Example 3:

```
Input: nums = [1,2,3,4,5]
Output: 0
```

Constraints:

- `1 <= nums.length <= 10^4`
- `1 <= nums[i] <= 10^5`

Python:

```
class Solution:
    def sumFourDivisors(self, nums: list[int]) -> int:
```

```
        res = 0
        for n in nums:
            val = self.sumOne(n)
            if val != -1:
                res += val
        return res

    def sumOne(self, n: int) -> int:
        p = round(n ** (1/3))
        if p ** 3 == n and self.isPrime(p):
            return 1 + p + p*p + p*p*p

        for i in range(2, int(n ** 0.5) + 1):
            if n % i == 0:
                a, b = i, n // i
                if a != b and self.isPrime(a) and self.isPrime(b):
                    return 1 + a + b + n
                return -1
        return -1

    def isPrime(self, x: int) -> bool:
        if x < 2:
            return False
        for i in range(2, int(x ** 0.5) + 1):
            if x % i == 0:
                return False
        return True
```

JavaScript:

```
var sumFourDivisors = function(nums) {
    let res = 0;

    for (let n of nums) {
        let val = sumOne(n);
        if (val !== -1) res += val;
    }
    return res;
};

function sumOne(n) {
    let p = Math.round(Math.cbrt(n));
    if (p * p * p === n && isPrime(p)) {
        return 1 + p + p*p + p*p*p;
    }
```

```
    for (let i = 2; i * i <= n; i++) {
        if (n % i === 0) {
            let a = i, b = n / i;
            if (a !== b && isPrime(a) && isPrime(b)) {
                return 1 + a + b + n;
            }
            return -1;
        }
    }
    return -1;
}

function isPrime(x) {
    if (x < 2) return false;
    for (let i = 2; i * i <= x; i++) {
        if (x % i === 0) return false;
    }
    return true;
}
```

Java:

```
class Solution {
    public int sumFourDivisors(int[] nums) {
        int res = 0;

        for(int num : nums){
            if(sumOne(num) != -1)res+=sumOne(num);
        }

        return res;
    }

    private int sumOne(int n) {
        // Case 1: p^3
        int p = (int) Math.round(Math.cbrt(n));
        if ((long) p * p * p == n && isPrime(p)) {
            return 1 + p + p * p + p * p * p;
        }

        // Case 2: p * q
        for (int i = 2; i * i <= n; i++) {
            if (n % i == 0) {
                int a = i;
```

```
            int b = n / i;
            if (a != b && isPrime(a) && isPrime(b)) {
                return 1 + a + b + n;
            }
            return -1;
        }
    }
    return -1;
  }

  private boolean isPrime(int x) {
    if (x < 2) return false;
    for (int i = 2; i * i <= x; i++) {
        if (x % i == 0) return false;
    }
    return true;
  }
}
```