Leetcode-

1.

Power of three:

class Solution {

    public boolean isPowerOfThree(int n) {

        if(n==1) return true;

        if(n%3 !=0 || n<=0) return false;

        return isPowerOfThree(n/3);

    }

}

A screenshot of a computer

Description automatically generated

2:

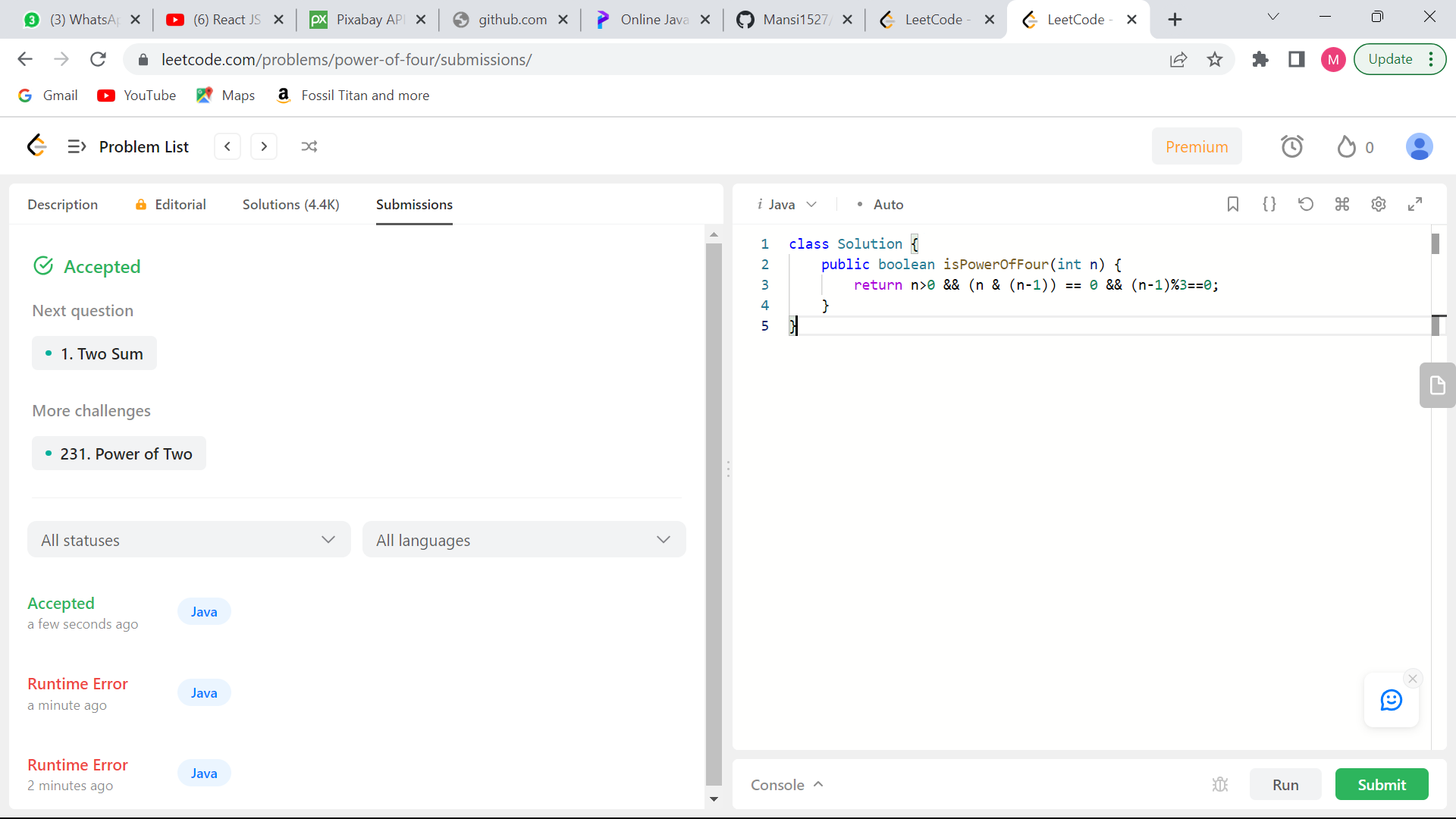
class Solution {

    public boolean isPowerOfFour(int n) {

        return n>0 && (n & (n-1)) == 0 && (n-1)%3==0;

    }

}



3:

class Solution {

    int base = 1337;

    public int superPow(int a, int[] b) {

        return superPow(a, b, b.length-1);

    }

    public int superPow(int a, int[] b, int index) {

        if (index == 0)

            return pow(a, b[index]) % base;

        int pre = superPow(a, b, index-1);

        return (pow(pre, 10) \* pow(a, b[index])) % base;

    }

    public int pow(int x, int n) {

        if (n == 0)

            return 1;

        if (n == 1)

            return x % base;

        int half = pow(x, n/2);

        if(n%2==0)

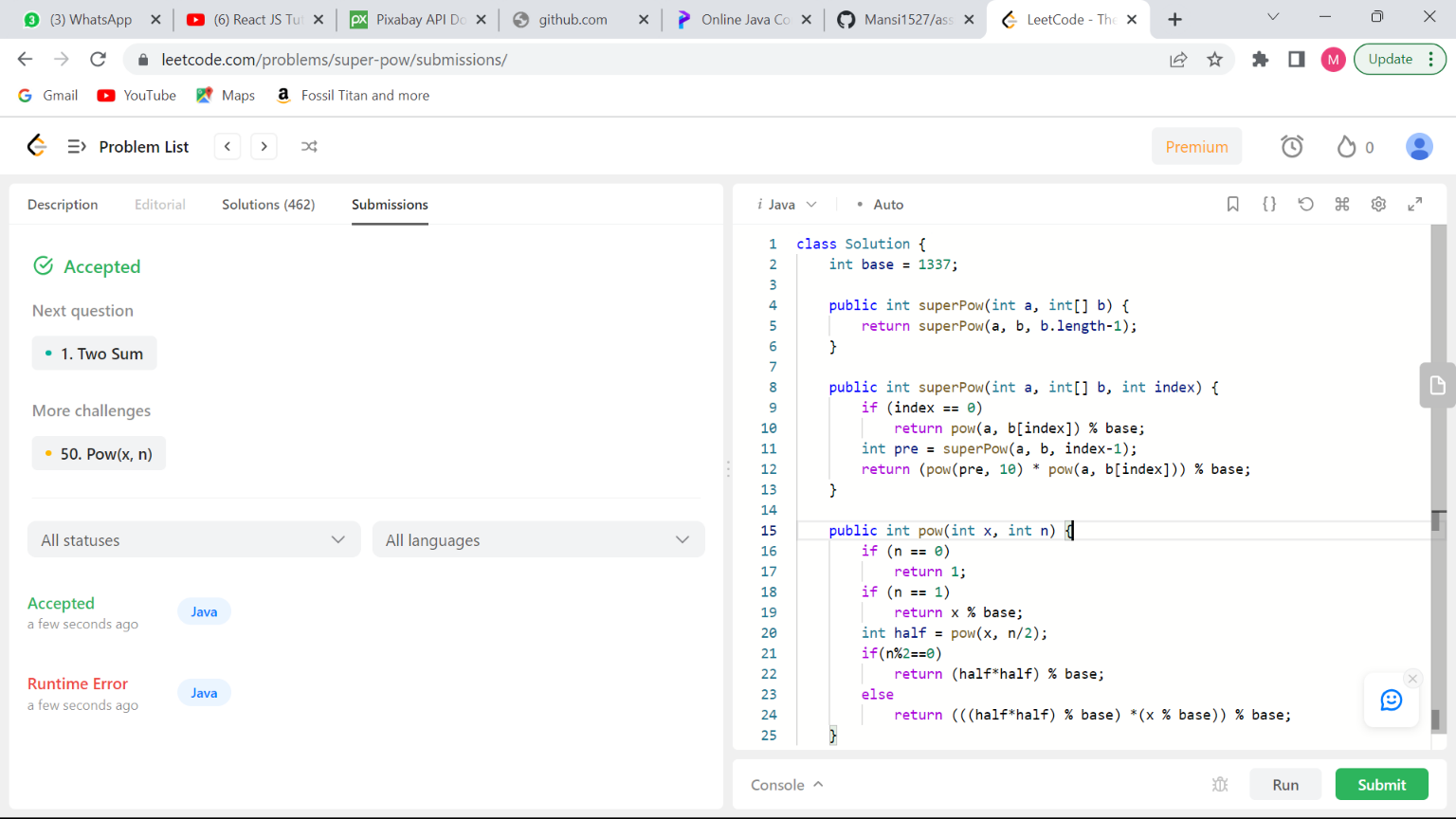
            return (half\*half) % base;

        else

            return (((half\*half) % base) \*(x % base)) % base;

    }

}



4:

class Solution {

int missingNumber(int array[], int n) {

// Your Code Here

int mis = 1;

Arrays.sort(array);

for(int i = 0 ; i <= array.length-1 ; i++){

if(array[i] == mis){

mis++;

}

}

return mis;

}

}

