1. <https://leetcode.com/problems/power-of-three/>

Code

class Solution {

    public boolean isPowerOfFour(int n) {

        if(n>1 && n%3==0)

            return isPowerOfFour(n/3);

        else if(n==1)

            return true;

        return false;

    }

}

A screenshot of a computer

Description automatically generated with medium confidence

1. <https://leetcode.com/problems/power-of-four/>

Code:

class Solution {

    public boolean isPowerOfFour(int n) {

        if(n>1 && n%4==0)

            return isPowerOfFour(n/4);

        else if(n==1)

            return true;

        return false;

    }

}

A screenshot of a computer

Description automatically generated with medium confidence

1. <https://leetcode.com/problems/super-pow/>

This code cant be done using just recursion we will use eulers method

Code:

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;

}

}

1. [https://practice.geeksforgeeks.org/problems/missing-number-in-array1416/1?page=1&curated[]=1&sortBy=submissions](https://practice.geeksforgeeks.org/problems/missing-number-in-array1416/1?page=1&curated%5b%5d=1&sortBy=submissions)

Code:

class Solution {

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

// Your Code Here

int sum = n\*(n+1)/2;

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

sum-=array[i];

return sum;

}

}

A screenshot of a test

Description automatically generated with medium confidence