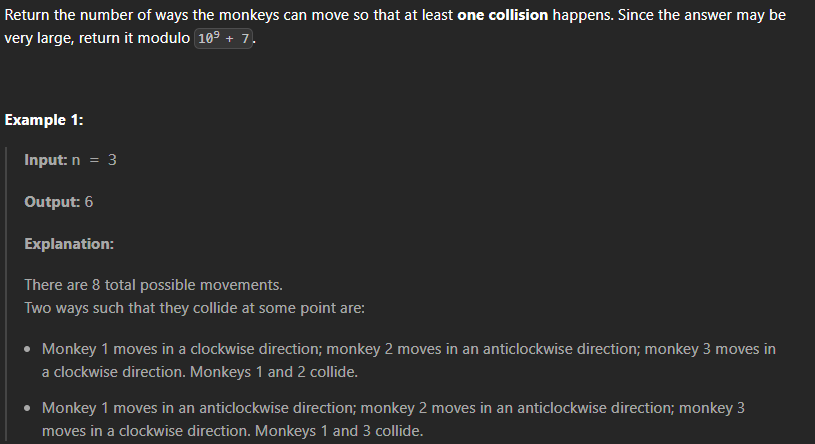
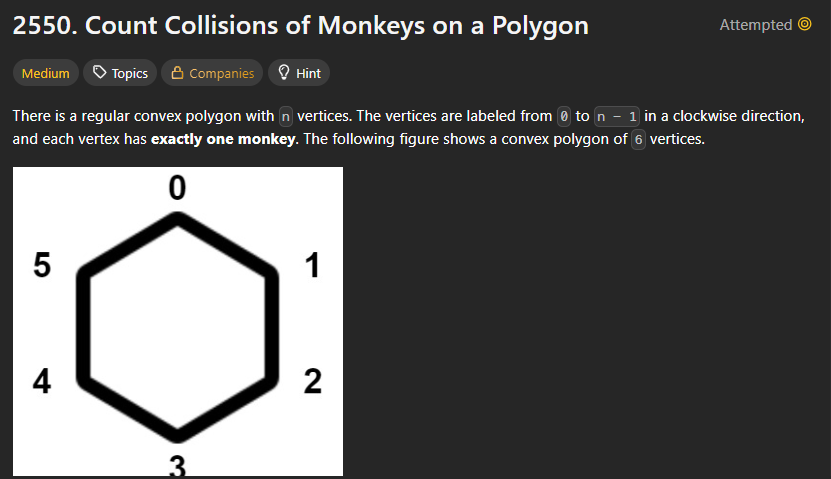
**LeetCode Problem 2550**

****

**Code snippet: O(log n)**

class Solution {

public:

    int monkeyMove(int n) {

        //brute force approach

        // we need to find 2^n means power function is ultimately going to be used.

        int x = 2;

        int result = 1;

        while(n!= 0){

            if(n%2 == 0){

                x = x \* x; // square the base

                n = n /2; // half the power

            }

            else{

                //if n is odd

                n = n -1 ;

                result = result \* x;

            }

        }

        return (result - 2);

    }

};

**Key learnings:**

1. There is some problem in constraints , so learning that part.
2. In c language , only zero is considered as false , negatives and positives both are true.

**New & Correct Code**

class Solution {

public:

    int monkeyMove(int n) {

        // we need to find 2^n means power function is ultimately going to be

        // used.

        long long x = 2;

        long long result = 1;

        const int MOD = 1e9 + 7;

        while (n != 0) {

            if (n % 2 == 0) {

                x = (x \* x) % MOD; // square the base

                n = n / 2; // half the power

            } else {

                // if n is odd

                n = n - 1;

                result = (result \* x) % MOD;

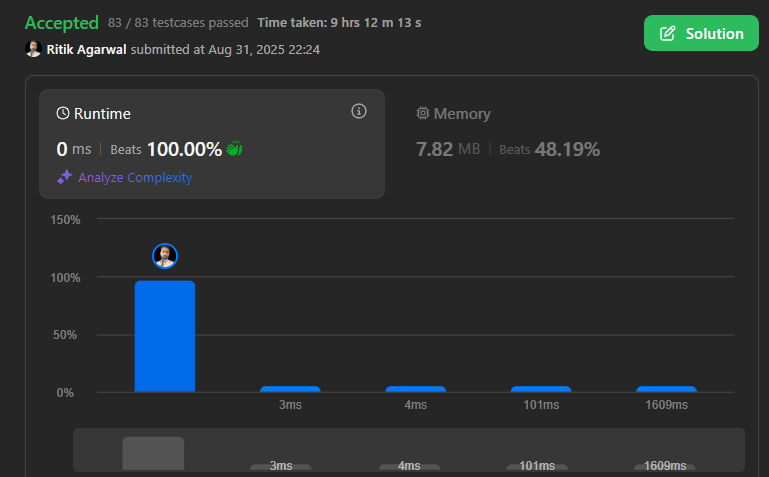
            }

        }

        return ((result - 2 + MOD) % MOD);

    }

};

****