H   PRACTICE   COMPETE   JOBS   LEADERBOARD   🔍 Search   💬 🔔   **H** pruthvishalcodi1 ⌄

All Contests > ALCoding Summer Long Challenge 1 > Fibonacci pairs

Fibonacci pairs   🔒 locked

H by pruthvishalcodi1

Problem	Submissions	Leaderboard	Discussions	Editorial

Statistics
Difficulty: Hard
Required Knowledge: Math
Publish Date: Jun 24 2019

Editorial by PruthvishE

using,GCD(Fib1, Fib2) = Fib(GCD(i, j)).

Refer to find about some math optimization techinques for various commom math operations.

Using this operations we can implement a efficient solution to the problem.

Set by PruthvishE

Problem Setter's code :

```cpp
#include <iostream>
#include <cstdio>

using namespace std;

const int MAX = 1000005;
const int MODULO = 1000000007;
const int PISANO_PERIOD = 2000000016;

long long getPisanoPeriod(long long m) {
    long long a = 0, b = 1, c = 1;
    for (long long i = 0; ; i++) {
        c = (a + b) % m;
        a = b;
        b = c;
        if (a == 0 && b == 1) return i + 1;
    }
    return -1;
}

int N;
long long K;
int data[MAX], counts[MAX], dcounts[MAX];
long long pairs[MAX];

int power(int a, long long b, int modulo) {
    if (b == 0) return 1 % modulo;
    if (b & 1) return 1LL * a * power(a, b - 1, modulo) % modulo;
    int half = power(a, b >> 1, modulo);
    return 1LL * half * half % modulo;
}

struct matrix{
    int a[2][2];
    matrix() {
    }
};
matrix operator * (matrix a, matrix b) {
    matrix c;
    for (int i = 0; i < 2; i ++) {
        for (int j = 0; j < 2; j ++) {
            c.a[i][j] = 0;
            for (int k = 0; k < 2; k ++) {
                c.a[i][j] = (c.a[i][j] + (1LL * a.a[i][k] * b.a[k][j] % MODULO)) % MODULO;
```

```cpp
                }
            }
        }
        return c;
    }
    matrix power(matrix a, int p) {
        if (p == 1) {
            return a;
        }
        if (p & 1) {
            return power(a, p - 1) * a;
        }
        matrix half = power(a, p >> 1);
        return half * half;
    }
    int NthFibonacci(int N) {
        matrix fib;
        fib.a[0][0] = 1;
        fib.a[0][1] = 1;
        fib.a[1][0] = 1;
        fib.a[1][1] = 0;
        matrix result = power(fib, N);
        return result.a[0][1];
    }

    int main() {


        scanf("%d%lld", &N, &K);
        for (int i = 1; i <= N; i ++) {
            scanf("%d", &data[i]);
            counts[data[i]] ++;
        }
        for (int i = 1; i < MAX; i ++) {
            for (int j = i; j < MAX; j += i) dcounts[i] += counts[j];
        }
        for (int i = MAX - 1; i > 0; i --) {
            pairs[i] = 1LL * dcounts[i] * (dcounts[i] - 1) / 2;
            for (int j = i + i; j < MAX; j += i) pairs[i] -= pairs[j];
        }

        int result = 0;
        for (int i = 1; i < MAX; i ++) {
            if (!pairs[i]) continue;
            pairs[i] %= MODULO;
            int fibonacciID = power(i, K, PISANO_PERIOD);
            result = (result + 1LL * pairs[i] * NthFibonacci(fibonacciID) % MOD
ULO) % MODULO;
        }
        cout << result << endl;

        return 0;
    }
```