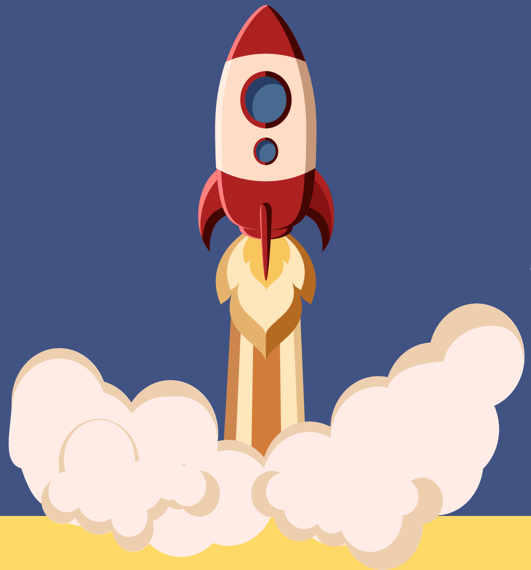


质数约数题目评讲

chenguo

问题1：素数距离

A 素数距离

时间限制：- MS　空间限制：- KB

问题描述

给定两个整数L, R, 求闭区间$[L, R]$中相邻质数差值最小的数对和差值最大的数对。当存在多个时, 输出靠前的素数对。

输入格式

输入包括两个数$L, R(1 \le L < R < 2^{31}, R - L \le 10^6)$

输出格式

见样例

样例输入 1

```
2 17
```

样例输出 1

```
2,3 are closest, 7,11 are most distant.
```

样例输入 2

```
14 17
```

样例输出 2

```
There are no adjacent primes.
```

问题1：素数距离

- 分析：
- 由于数据范围很大，无法生成 $[1,R]$ 中的所有素数。
- 使用筛法求出 $[2,\sqrt{R}]$ 之间的所有素数，对于每个素数 p ，把 $[L,R]$ 中能被 p 整除的数标记，即标记 $i\times p\left(\left\lceil\frac{L}{p}\right\rceil \le i \le \left\lfloor\frac{R}{p}\right\rfloor\right)$ 为合数。
- 将筛出的素数进行相邻两两比较，找出差最大的即可。

```cpp
#include <bits/stdc++.h>
#define ll long long
using namespace std;
const int MAXN = 1e6 + 10;
const int MAX = 1e5;
int prime[MAX], tag[MAX], vis[MAXN], tot;
void get_prime(void){
    for(int i = 2; i < MAX; i++){
        if(!tag[i]){
            prime[tot++] = i;
            for(int j = 2; j * i < MAX; j++){
                tag[j * i] = 1;
            }
        }
    }
}
int main(){
    get_prime();
    ll l, r;
    scanf("%lld%lld", &l, &r);
    memset(vis, 0, sizeof(vis));
    for(int i = 0; i < tot; i++){
        ll a = (l + prime[i] - 1) / prime[i];
        ll b = r / prime[i];
        for(int j = max(2LL, a); j <= b; j++){ // 筛[l, r]内的合数
            vis[prime[i] * j - l] = 1; //减个l方便标记,输出答案时加回去即可
        }
    }
    if(l == 1) vis[0] = 1; // 注意这个1并不是素数
    ll cnt = -1, sol1 = MAXN, sol2 = 0, x1, y1, x2, y2;
    for(int i = 0; i <= r - l; i++){
        if(vis[i] == 0){
            if(cnt != -1){
                if(sol1 > i - cnt){
                    x1 = cnt;
                    y1 = i;
                    sol1 = i - cnt;
                }
                if(sol2 < i - cnt){
                    x2 = cnt;
                    y2 = i;
                    sol2 = i - cnt;
                }
            }
            cnt = i;
        }
    }
    if(sol2 == 0) puts("There are no adjacent primes.");
    else printf("%lld,%lld are closest, %lld,%lld are most distant.\n", x1 + l, y1 + l, x2 + l, y2 + l);
    return 0;
}
```

问题2：轻拍牛头

C 轻拍牛头

时间限制：- MS 空间限制：- KB

评测说明：1s 256MB

问题描述

今天是贝茜的生日，为了庆祝自己的生日，贝茜邀你来玩一个游戏。

贝茜让 N 头奶牛坐成一个圈。除了 1 号与 N 号奶牛外，i 号奶牛与 $i-1$ 号和 $i+1$ 号奶牛相邻，N 号奶牛与 1 号奶牛相邻。农夫约翰用很多纸条装满了一个桶，每一张包含了一个1到 10^6 的数字。

接着每一头奶牛 i 从桶中取出一张纸条 A_i，每头奶牛轮流走一圈，同时拍打所有「编号是 A_i 的约数」的牛，然后走回到原来的位置。牛们希望你帮助他们确定，每一头奶牛需要拍打的牛。

输入格式

第一行包含一个整数 N；接下来第二到第$N+1$ 行每行包含一个整数 A_i。对于全部数据，$1 \leq N \leq 10^5$。

输出格式

第一到第 N 行，第 i 行的输出表示第 i 头奶牛要拍打的牛数量。

问题2： 轻拍牛头

对于每个a[i]，它对于所有ans[j]的贡献为1（j%i=0）
然后我们可以用一种类似于筛法的方法对每个a[i]进行处理

问题2: 轻拍牛头

参考代码：

```cpp
#include <bits/stdc++.h>
using namespace std;
int n, mx;
int a[100005], cnt[1000005], s[1000005];
int main() {
    scanf("%d", &n);
    for (int i = 1; i <= n; i++) {
        scanf("%d", &a[i]);
        cnt[a[i]]++;
        mx = max(a[i], mx);
    }
    for (int i = 1; i <= mx; i++)
        if (cnt[i])
            for (int j = i; j <= mx; j += i) s[j] += cnt[i];
    for (int i = 1; i <= n; i++) printf("%d\n", s[a[i]] - 1);
    return 0;
}
```

问题3： Sherlock和他的女朋友

E Sherlock和他的女朋友

时间限制：- MS　空间限制：- KB

评测说明：1s 256MB

问题描述

Sherlock 有了一个新女友（这太不像他了！）。情人节到了，他想送给女友一些珠宝当做礼物。

他买了 n 件珠宝。第 i 件的价值是 $i+1$。那就是说，珠宝的价值分别为 $2, 3, 4, \cdots, n+1$。

Watson 挑战 Sherlock，让他给这些珠宝染色，使得一件珠宝的价格是另一件的质因子时，两件珠宝的颜色不同。并且，Watson 要求他最小化颜色的使用数。

请帮助 Sherlock 完成这个简单的任务。

输入格式

只有一行一个整数 n，表示珠宝件数。

$1 \le n \le 10^5$

输出格式

第一行一个整数 k，表示最少的染色数；

第二行n 个整数，表示第 1 到第 n 件珠宝被染成的颜色。若有多种答案，输出任意一种。

问题3： Sherlock和他的女朋友

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef int ll;
const int N = 100005;
int n;
bool Bo[N];
int Prime[N];
inline void Get_Prime() {
    for (int i = 2; i <= n; i++) {
        if (!Bo[i])
            Prime[++*Prime] = i;
        for (int j = 1; j <= *Prime && Prime[j] * i <= n; j++) {
            Bo[Prime[j] * i] = 1;
            if (i % Prime[j] == 0)
                break;
        }
    }
    return;
}
```

```cpp
int main() {
    scanf("%d", &n);
    n++;
    if (n == 2) {
        printf("1\n1\n");
        return 0;
    }
    if (n == 3) {
        printf("1\n1 1\n");
        return 0;
    }
    Get_Prime();
    puts("2");
    for (int i = 2; i <= n; i++) {
        if(Bo[i]) printf("2\n");
        else printf("1\n");
    }
    return 0;
}
```

问题4：不定方程

F 不定方程

时间限制：- MS　空间限制：- KB

评测说明：1s 256MB

问题描述

求不定方程：

$$\frac{1}{x}+\frac{1}{y}=\frac{1}{n!}$$

的正整数解 (x,y)的数目。

输入格式

一个整数 n。

对于 30% 的数据，$n \le 100$；

对于全部数据，$1 \le n \le 10^6$ 。

输出格式

一个整数，表示有多少对(x,y)满足题意。答案对 10^9+7 取模。

样例输入

```
2
```

样例输出

```
3
```

问题4：不定方程

$\frac{1}{x}+\frac{1}{y}=\frac{1}{n!}$
$\frac{xy}{x+y}=n!$
$xy=n!(x+y)$
$-n!(x+y)+xy=0 \longleftrightarrow (n!x+n!y)-xy=0$

对数学敏感的同学相信写到这一步时就已经可以发现一些东西了,这个式子是十字相乘法拆开括号的后两项!

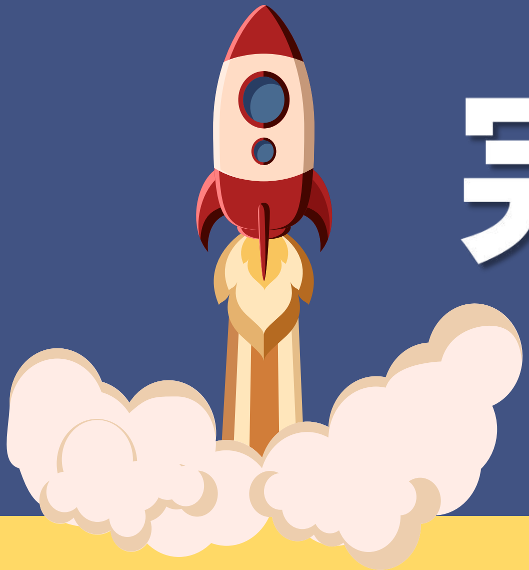
若设$a=-n!, b=x, c=y,$ 则$a(b+c)+bc=0,$ 等式两边同时加上a^2
则$a^2+a(b+c)+bc=0, (a+c)(a+b)=0$

于是有$(x-n!)(y-n!)=(n!)^2$
也就是说$(x-n!)|(n!)^2$
那么,$(x-n!)$等价于$(n!)^2$的因子,又由于$(x-n!)$和x的个数相等,那么x的个数和$(n!)^2$的因子的个数一一对应

问题4： 不定方程

参考代码：

```
#include <bits/stdc++.h>
using namespace std;
const int mod = 1e9 + 7;
const int N = 1e6 + 7;
long long c[N], ans = 1;
int n, isprime[N], prime[N];
void divide(int x) {
    for (int i = 1; prime[i] * prime[i] <= x; i++) {
        while (x % prime[i] == 0) {
            x /= prime[i];
            c[prime[i]]++;
        }
    }
    if (x != 1)
        c[x]++;
}
int main() {
    cin >> n;
    for (int i = 2; i <= n; i++) {
        if (!isprime[i])
            prime[++prime[0]] = i;
        for (int j = 1; j <= prime[0] && i * prime[j] < n; ++j) {
            isprime[i * prime[j]] = 1;
            if (i % prime[j] == 0)
                break;
        }
    }
    for (int i = 1; i <= n; i++) divide(i);
    for (int i = 1; i <= prime[0]; i++) ans = (ans * (c[prime[i]] << 1 | 1)) % mod;
    cout << ans << endl;
    return 0;
}
```



完！

以梦为码 心之所往