

NOI2016 循环之美

2017-05-21 19:29:17

Problem

求:

$$\sum_{i=1}^n \sum_{j=1}^m \left[\frac{i}{j} k \right]$$

Analysis

$\frac{x}{y}$ 是 k 进制纯循环小数当且仅当:

$$\exists l : \frac{x \times (k^l - 1)}{y} \in \mathbb{Z} \mid (x, y) = 1, x \times k^l \equiv x \pmod{y}$$

由于 $(x, y) = 1$, 所以 $k^l \equiv 1 \pmod{y}$, 则必有 $(y, k) = 1$

问题转化为求:

$$\begin{aligned} & \sum_{i=1}^n \sum_{j=1}^m [(i, j) = 1] [(j, k) = 1] \\ &= \sum_{i=1}^n \sum_{j=1}^m [(j, k) = 1] \sum_{d \mid i, d \mid j} \mu(d) \\ &= \sum_{d=1}^n [(d, k) = 1] \mu(d) \left\lfloor \frac{n}{d} \right\rfloor \sum_{j=1}^{\lfloor \frac{m}{d} \rfloor} [(j, d) = 1] \end{aligned}$$

首先观察式子的后部分，考虑如何求：

$$f(n) = \sum_{i=1}^n [(i, k) = 1]$$

根据 $\gcd(a+b, b) = \gcd(a, b)$, $f(n) = \lfloor \frac{n}{k} \rfloor f(k) + f(n \bmod k)$

然后要计算的是：

$$\begin{aligned} s(n, k) &= \sum_{d=1}^n [(d, k) = 1] \mu(d) \\ &= \sum_{d=1}^n \mu(d) \sum_{x|d} \mu(x) \\ &= \sum_{x|k} \mu(x) \sum_{d=1}^{\lfloor \frac{n}{x} \rfloor} \mu(d \cdot x) \\ &= \sum_{x|k} \mu^2(x) \sum_{d=1}^{\lfloor \frac{n}{x} \rfloor} \mu(d) \\ &= \sum_{x|k} \mu^2(x) s(\lfloor \frac{n}{x} \rfloor, x) \end{aligned}$$

那么 $s(n, k)$ 就可以递归计算了，边界条件是 $k = 1$.

当 $k = 1$ 的时候，这东西就叫做杜教筛。我们有：

$$\sum_{d|n} \mu(d) = [n = 1]$$

那么就不难得到：

$$\begin{aligned} &\sum_{i=1}^n \mu(i) \\ &= 1 + \sum_{i=2}^n \mu(i) \\ &= 1 - \sum_{i=2}^n \sum_{d|i} \mu(d) \\ &= 1 - \sum_{k=2}^n \sum_{d=1}^{\lfloor \frac{n}{k} \rfloor} \mu(d) \end{aligned}$$

至此，原式转化为一个可以递归求解的式子，顺便贴一个杜教筛教程。

Code

```
#include<bits/stdc++.h>
using namespace std;

typedef long long LL;
typedef pair<int, int> pii;

const int oo = 0x3f3f3f3f;
const int maxn = 1e6 + 10;

template <typename T> bool chkmax(T& a, T b) { return a < b ? a = b, 1 : 0; }
template <typename T> bool chkmin(T& a, T b) { return a > b ? a = b, 1 : 0; }

#define fst first
#define snd second
#define debug(x) cerr << #x << ":" << (x) << endl
#define REP(i, a, b) for(int i = (a), i##end = (b); i < i##end; ++i)
#define DREP(i, a, b) for(int i = (a)-1, i##bgn = (b); i >= i##bgn; --i)

template<typename T> T read() {
    T n = 0, f = 1;
    char ch = getchar();
    for( ;!isdigit(ch); ch = getchar()) if(ch == '-') f = -1;
    for( ; isdigit(ch); ch = getchar()) n = n * 10 + ch - 48;
    return n * f;
}

bool isprime[maxn];
LL prime[maxn], mu[maxn], smu[maxn], pcnt;

void sieve() {
    memset(isprime, 1, sizeof isprime);
```

```

mu[1] = 1;
for(LL i = 2; i < maxn; i++) {
    if(isprime[i]) {
        mu[i] = -1;
        prime[pcnt++] = i;
    }
    static LL d;
    for(LL j = 0; j < pcnt && (d = i * prime[j]) < maxn; j++) {
        isprime[d] = 0;
        if(i % prime[j] == 0) break;
        mu[d] = -mu[i];
    }
}for(int i = 1; i < maxn; i++) smu[i] = smu[i-1] + mu[i];
}

```

```

namespace Hash_Map {
    static const int mod = 1000007;
    static const int maxnode = 8000000 + 10;

    int cnt = 0;
    LL f[maxnode], to[maxnode];
    int st[mod + 5], nxt[maxnode];

    LL find(LL state) {
        int x = state % mod;
        for(int i = st[x]; i; i = nxt[i])
            if(to[i] == state) return i;
        return -1;
    }

    void insert(LL state, LL v) {
        int x = state % mod;

```

```

        to[++cnt] = state;
        nxt[cnt] = st[x];
        st[x] = cnt; f[cnt] = v;
    }
}

LL calc(int n, int k) {
    if(n == 0 || (k == 1 && n < maxn))
        return smu[n];

    LL val = 1LL*n*2005 + k;
    LL cur = Hash_Map::find(val);
    if(cur >= 0) return Hash_Map::f[cur];

    LL ans = 0;
    if(k == 1) {
        ans = 1;

        int j;
        for(int i = 2; i <= n; i = j+1) {
            j = n / (n/i);
            ans -= (j-i+1) * calc(n/i, k);
        }
    }else {
        for(int i = 1; i*i <= k; i++) if(k % i == 0) {
            if(mu[i]) ans += calc(n/i, i);

            if(k != i*i && mu[k/i])
                ans += calc(n/(k/i), k/i);
        }
    }
    Hash_Map::insert(val, ans);
}

```

```

        return ans;
    }

    int n, m, k;
    int cnt[10000 + 10];

    LL f(int a) {
        return a / k * cnt[k] + cnt[a % k];
    }

    LL solve() {
        for(int i = 1; i <= k; i++)
            cnt[i] = cnt[i-1] + (__gcd(i, k) == 1);

        LL j = 0, now = 0, lst = 0, ans = 0;
        for(int i = 1; i <= min(n, m); i = j + 1) {
            j = min(n/(n/i), m/(m/i));

            now = calc(j, k);
            ans += (now - lst) * (n/i) * f(m/i);
            lst = now;
        }
        return ans;
    }

    int main() {
#ifdef ONLINE_JUDGE
        freopen("data.txt", "r", stdin);
        freopen("ans.txt", "w", stdout);
#endif
        sieve();

        n = read<int>();
        m = read<int>();

```

```
k = read<int>();

printf("%lld\n", solve());
return 0;
}
```