积性函数

2020-12-21 | 2022-10-23 | OI & ACM | 13
21k | 19 分钟

积性函数

定义

1. 若 $f(n)$ 的定义域为正整数域，值域为复数，即 $f:Z+\to C$，则称 $f(n)$为 **数论函数。**
2. 若 $f(n)$ 为数论函数，且 $f(1)=1$，对于互质的正整数 p,q 有 $f(p\cdot q)=f(p)\cdot f(q)$，则称其为 **积性函数。**
3. 若 $f(n)$ 为积性函数，且对于任意正整数 p,q 都有 $f(p\cdot q)=f(p)\cdot f(q)$，则称其为 **完全积性函数。**

性质

1. 对于任意积性函数 $f(1)=1$
2. 对于任意积性函数 f,g，$f\cdot g$ 依然是积性函数

积性函数的例子

1. 除数函数 $\sigma_k(n)=\sum_{d|n}d^k$
2. 约数个数函数 $\sigma_0(n)=\tau(n)=\sum_{d|n}1$
3. 约数和函数 $\sigma(n)=\sum_{d|n}d$
4. 欧拉函数 $\varphi(n)$
5. 莫比乌斯函数 $\mu(n)$
6. 元函数 $\epsilon(n)=[n=1]$ 完全积性
7. 恒等函数 $I(n)=1$ 完全积性
8. 单位函数 $id(n)=n$ 完全积性
9. 幂函数 $id^k(n)=n^k$ 完全积性
10.

欧拉函数

定义

对于正整数 n，$\varphi(n)$ 的等于 1 到 $n-1$ 中与 n 互质的数的个数，规定 $\varphi(1)=1$

通项公式：$\varphi(n)=n\prod_{i=1}^{k}(1-\frac{1}{p_i})$，其中 n 的质因数分解形式为 $n=\prod_{i=1}^{k}p_i^{a_i}$

性质

以下不特加说明，默认 p 为素数

1. $\varphi(p)=p-1$
2. $\varphi(p^k)=p^k-p^{k-1}=(p-1)p^{k-1}$

   证明：

   小于 p^k 的数一共有 p^k-1 个，其中不与 p^k 互素的数的个数为 $1p,2p,\cdots,(p^{k-1}-1)p$，一共 $p^{k-1}-1$ 个
3. 令 n 的质因数分解形式为 $n=\prod_{i=1}^{k}p_i^{a_i}$，则 $\varphi(n)=n\prod_{i=1}^{k}(1-\frac{1}{p_i})$

   证明：

   $\varphi(n)=\prod_{i=1}^{k}\varphi(p_i^{a_i})=n\prod_{i=1}^{k}(1-\frac{1}{p_i})$
4. 欧拉定理：如果 a,m 互质，则一定有 $a^{\varphi(m)}\equiv 1(\mathrm{mod}\ m)$
5. 费马小定理：$a^{p-1}\equiv 1(\mathrm{mod}\ p)$
6. $\sum_{i=1}^{n}i[(n,i)=1]=\frac{[n=1]+\varphi(n)}{2}$

   证明：

   如果 $(n,i)=1$，则 $(n,n-i)=1$

   由此可知，与 n 互质的数成对存在，并且先加等于 n
7. 若 $p|n$，则 $\varphi(n\cdot p)=p\cdot\varphi(n)$，否则 $\varphi(n\cdot p)=(p-1)\cdot\varphi(n)$

   参考通项公式
8. $\varphi(ab)=\varphi(a)\varphi(b)\frac{(a,b)}{\varphi((a,b))}$

   若 $(a,b)=1$，则结论显然，所以下面只考虑 a 和 b 不互质的情况

   我们考虑将 $\varphi(n)$ 写成 $\varphi(n)=n\prod_{i=1}^{m}(1-\frac{1}{p_i})$

那么 $\varphi(a)\varphi(b)$ 相当于 $\varphi(ab)\prod_{p|(a,b)}(1-\frac{1}{p})$

而 $\frac{(a,b)}{\varphi((a,b))}$ 正好等于后面那个多余的东西的倒数

9. 狄利克雷卷积的结果

   $\sum_{d|n}\varphi(d)=n$

线性筛求欧拉函数

```
void init_phi(int n) {
    phi[1] = 1;
    for (int i = 2; i <= n; ++i) {
        if (!isp[i]) pri[++cnt] = i, phi[i] = i - 1;
        for (int j = 1; j <= cnt && i * pri[j] <= n; ++j) {
            isp[i * pri[j]] = 1;
            if (i % pri[j] == 0) { phi[i * pri[j]] = phi[i] * pri[j]; break; }
            phi[i * pri[j]] = phi[i] * (pri[j] - 1);
        }
    }
}
```

莫比乌斯函数

定义

$$\mu(n)=\begin{cases}1, & n=1\\ (-1)^k, & n=\prod_{i=1}^{k}p_i\\ 0\end{cases}$$

性质

1. $\sum_{i=1}^{n}\mu^2(i)=\sum_{i=1}^{\sqrt{n}}\mu(i)\lfloor\frac{n}{i^2}\rfloor$

   证明：

   注意到上式左边是求无平方因子的数的个数，我们令 $f(n)$ 表示最大的整除 n 的平方因子

   那么左式等价于 $\sum_{i=1}^{n}[f(i)=1]=\sum_{d|f(i)}\mu(d)$

   我们注意到当 d 含有平方因子的时候 $\mu(d)=0$，当且仅当 d 不含平方因子的时候 $\mu(d)\neq 0$

   那么这时候一定有 $d^2|f(i)$，我们知道 $\frac{i}{f(i)}$ 一定不含平方因子，所以可以直接 $d^2|f(i)\Rightarrow d^2|i$

那么我们有 $\sum_{i=1}^{n}[f(i)=1]=\sum_{i=1}^{n}\sum_{d|f(i)}\mu(d)=\sum_{i=1}^{n}\sum_{d^2|i}\mu(d)=\sum_{i=1}^{\sqrt{n}}\mu(i)\sum_{d^2|i}1=\sum_{i=1}^{\sqrt{n}}\mu(i)\lfloor\frac{n}{i^2}\rfloor$

2. 狄利克雷卷积的结果

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

   $\sum_{d|n}\mu(d)\frac{n}{d}=\varphi(n)$

狄利克雷卷积

定义

设 f,g 是两个数论函数，则 $f\circ g=\sum_{d|n}f(d)g(\frac{n}{d})$

性质

1. 交换律
2. 结合律
3. 加法的分配率 $$
4. 两个积性函数的狄利克雷卷积仍然是积性函数
5. 积性函数的逆元仍然是积性函数
6. 积性函数相乘仍然是积性函数
7. 如果 f 为完全积性函数，$(f\cdot g)\circ(f\cdot h)=f\cdot(g\circ h)$，

逆元

对于数论函数 f，若存在 g，使得 $f\circ g=\epsilon$，则称 g 是 f 的逆元

对于数论函数 f，当 $f(1)$ 不为 1 时，存在逆元 $g(n)=\frac{1}{f(1)}([n=1]-\sum_{d|n,d\neq 1}f(d)g(\frac{n}{d}))$

常用的狄利克雷卷积

1. $\mu\circ I=\epsilon$
2. $\varphi\circ I=id$
3. $\mu\circ id=\varphi$
4. $I\circ I=\tau,id\circ I=\sigma,\sigma_k=id_k\circ I$

狄利克雷前缀和

大概就是令 $s_n=\sum_{d|n}a_d$，我们称 s 是 a 的狄利克雷前缀和，从狄利克雷卷积的角度来考虑，这个东西就是 $1\circ a$，从高维前缀和的角度考虑，这东西就是以每个质因子为一个维度的前缀和，然后这东西可以做到 $O(n\log\log n)$

```
for (int i = 1; i <= cnt; ++i)
    for (int j = 1; pri[i] <= n / j; ++j) s[pri[i] * j] += s[j];
```

线性筛

如果一个积性函数在素数的答案可以简单求得，且其最小质因数的次数不为一的时候的答案也可以简单计算，那么我们可以直接用线性筛 $O(n)$ 来求这个积性函数，以 $\varphi(n)$ 为例

```
void init_phi(int n) {
    phi[1] = 1;
    for (int i = 2; i <= n; ++i) {
        if (!isp[i]) pri[++cnt] = i, phi[i] = i - 1;
        for (int j = 1; j <= cnt && i * pri[j] <= n; ++j) {
            isp[i * pri[j]] = 1;
            if (i % pri[j] == 0) { phi[i * pri[j]] = phi[i] * pri[j]; break; }
            phi[i * pri[j]] = phi[i] * (pri[j] - 1);
        }
    }
}
```

否则就要麻烦一点，先求出这个积性函数在 p^c 的答案，然后再利用积性函数的性质求出所有位置的答案

以 $f(n)=\sum_{d|n}d\varphi(d)$ 为例，其中 a_n 表示 n 去掉其最小质因子所有次方后的值

注意到这样做的时间复杂度仍然是 $O(n)$

```
void init_isp(int n) {
    for (int i = 2; i <= n; ++i) {
        if (!isp[i]) pri[++cnt] = i;
        for (int j = 1; j <= cnt && i * pri[j] <= n; ++j) {
            isp[i * pri[j]] = 1;
            if (i % pri[j] == 0) { a[i * pri[j]] = a[i]; break; }
            a[i * pri[j]] = i;
        }
    } h[1] = 1;
    for (int i = 1; i <= cnt; ++i)
        for (ll j = pri[i]; j <= n; j *= pri[i])
            h[j] = h[j / pri[i]] + (j - j / pri[i]) * j;
    for (int i = 2; i <= n; ++i)
        if (a[i] > 1) h[i] = h[i / a[i]] * h[a[i]];
}
```

杜教筛

众所周知，积性函数前缀和可以 $O(n)$ 预处理，但如果数据范围高达 10^9 的话就无计可施了，这个时候就只能用杜教筛了

不妨设所求为 $S(n)=\sum_{i=1}^{n}f(i)$，我们根据玄学选取另一积性函数 $g(i)$

$$\begin{aligned}\sum_{t=1}^{n}(f\circ g)(t)&=\sum_{t=1}^{n}\sum_{d|t}g(d)f(\frac{t}{d})\\&=\sum_{d=1}^{n}\sum_{d|t}g(d)f(\frac{t}{d})\\&=\sum_{d=1}^{n}g(d)\sum_{d|t}f(\frac{t}{d})\\&=\sum_{d=1}^{n}g(d)\sum_{k=1}^{\lfloor\frac{n}{d}\rfloor}f(k)\\&=\sum_{d=1}^{n}g(d)S(\lfloor\frac{n}{d}\rfloor)\end{aligned}$$

即

$$\begin{aligned}\sum_{t=1}^{n}(f\circ g)(t)&=\sum_{t=1}^{n}g(t)S(\lfloor\frac{n}{t}\rfloor)\\g(1)S(n)&=\sum_{t=1}^{n}(f\circ g)(t)-\sum_{t=2}^{n}g(t)S(\lfloor\frac{n}{t}\rfloor)\end{aligned}$$

如果我们能快速的计算 g 的前缀和以及 $f\circ g$ 的值，那么我们可以用数论分块做到 $O(n^{\frac{2}{3}})$ 的复杂度

关于杜教筛的一些时间复杂度：

1. 预处理前 $n^{\frac{2}{3}}$ 项，时间复杂度 $O(n^{\frac{2}{3}})$
2. 不预处理前 $n^{\frac{2}{3}}$ 项，时间复杂度 $O(n^{\frac{3}{4}})$
3. 数论分块套数论分块，不预处理时间复杂度 $O(n^{\frac{3}{4}})$，预处理时间复杂度 $O(n^{\frac{2}{3}})$
4. 数论分块套杜教筛，时间复杂度 $O(n^{\frac{2}{3}})$

以 $\sum_{i=1}^{n}\varphi(i)$ 为例，对于两个函数我们选择的 g 均为恒等函数 1

```
namespace D_Seive {
    const int N = 1000000; // n ^ 2/3
    ll _g[maxn], n; bool vis[maxn]; int sn; // n ^ 1/2

    void init(ll _n) {
        n = _n; sn = sqrt(n);
        for (int i = 1; i <= 2 * sn; ++i) vis[i] = 0;
    }
```

```
    int get(ll x) { return x <= sn ? x : sn * 2 - n / x + 1; }
    ll calc(ll n) {
        if (n <= N) return g[n];
        int id = get(n);
        if (vis[id]) return _g[id]; ll ans = F1(n); // n * (n + 1) / 2
        for (ll l = 2, r; l <= n; l = r + 1) {
            r = n / (n / l);
            ans = (ans - (r - l + 1)  * calc(n / l)) % p;
        }
        return vis[id] = 1, _g[id] = ans;
    }
}
```

有的时候，$unordered\_map$ 比用 Div_Hash 还要快

Min_25筛

简介

Min_25 筛是一种能够快速求解积性函数 $f(x)$ 的前缀和 $\sum_{i=1}^{n} f(i)$ 的筛法，前提是 $f(p)$ 是关于 p 的多项式，同时 $f(p^k)$ 可以快速计算

时间复杂度为 $O(\frac{n^{\frac{3}{4}}}{\log n})$，空间复杂度 $O(\sqrt{n})$

推导过程

下文中，我们令 $f(i)$ 为在素数处和原函数相同的完全积性函数，这样做的原因是 g 数组我们只需要用到 $g(n,|P|)$

$\sum_{i=1}^{n} f(i) = f(1) + \sum_{p\in P\wedge p\le n} f(p) + \sum_{p\in P\wedge p^x\le n\wedge p\le\sqrt{n}} f(p^x)(\sum_{i\le\lfloor\frac{n}{p^x}\rfloor\wedge LPF(i)>p} f(i) + [x>1])$

大概就是将 $f(x)$ 的前缀和分成三部分，$f(1)$、质数以及合数的情况，至于合数的情况，我们是通过枚举最小质因子来枚举合数的

至于如何计算这几个部分，我们考虑构造函数 $g(n,m)$，$g(n,m) = \sum_{i=1}^{n}[i\in P\vee LPF(i)>P_m]f(i)$

这个东西就是我们只算所有的素数以及最小质因子大于第 m 个素数的数

容易得到 $g(n,m)$ 的递推式

$$g(n,m) = \begin{cases} g(n,m-1), & P_m^2 > n \\ g(n,m-1) - f(P_m)(g(\lfloor\frac{n}{P_m}\rfloor, m-1) - g(P_{m-1}, m-1)), & P_m^2 \le n \end{cases}$$

简单来讲就是 $g(n,m)$ 一定是 $g(n,m-1)$ 减掉最小质因数为 P_m 的合数的贡献，因为 $f(i)$ 是完全积性函数，所以我们考虑将 P_m 提出来，需要注意的是我们需要将素数的贡献减掉

然后我们再构造一个函数，$S(n,m)=\sum_{i=1}^{n}[LPF(i)>P_m]i^k$，那么我们可以将 S 和 g 以某种关系连接起来，另外为了方便，我们把 $g(n,|P|)=\sum_{p\in P}^{n}p^k$ 简写成 $g(n)$

容易得到 $S(n,m)=g(n)-\sum_{i=1}^{m-1}P_i^k+\sum_{P_i^x\le n\wedge i>m}(P_i^x)^k(S(\lfloor\frac{n}{P_i^x}\rfloor,i)+[x>1])$

大概就是分成两部分，第一部分是大于 P_m 的质数，另一部分是最小质因子大于 P_m 的合数

最后的答案显然就是 $S(n,0)$

模板

求积性函数 $f(x)$ 的前缀和，其中 $f(p^x)=p^x$

```
ll id1[maxn], id2[maxn]; int Sn;
int get_id(ll x) { return x <= Sn ? id1[x] : id2[n / x]; }

ll g[maxn], w[maxn], inv2; int num;
void init_min25(ll n) {
    inv2 = pow_mod(2, p - 2);
    for (ll l = 1, r; l <= n; l = r + 1) {
        r = n / (n / l); w[++num] = n / l; ll t = n / l % p;
        g[num] = t * (t + 1) % p * inv2 % p - 1; // 注意需要把 f(1) 减掉
        if (w[num] <= Sn) id1[w[num]] = num;
        else id2[n / w[num]] = num;
    }
    for (int i = 1; i <= cnt; ++i)
        for (int j = 1; j <= num && 1ll * pri[i] * pri[i] <= w[j]; ++j) {
            g[j] = (g[j] - pri[i] * (g[get_id(w[j] / pri[i])] - sp[i - 1])) % p;
        }
}

ll S(ll n, int m) {
    if (pri[m] >= n) return 0;
    ll ans = (g[get_id(n)] - sp[m]) % p;
    for (int i = m + 1; i <= cnt && 1ll * pri[i] * pri[i] <= n; ++i)
        for (ll x = 1, mul = pri[i]; mul <= n; mul *= pri[i], ++x)
            ans = (ans + mul * (S(n / mul, i) + (x > 1))) % p;
    return ans;
}
```

powerful number筛

简介

powerful number：每个质因子次数都不为 1 的数，$\le n$ 的 powerful number 只有 $O(\sqrt{n})$

设 $f(x)$ 为要求的积性函数，我们需要构造一个积性函数 $g(x)$ 需要满足 $g(p)=f(p)$，同时我们需要求一个积性函数 $h(x)$，满足 $f=g\circ h$，在素数处我们有 $f(p)=g(1)h(p)+g(p)h(1)$，因为我们有 $g(1)=h(1)=1,g(p)=f(p)$，所以我们可以得到 $h(p)=0$，由于 h 是积性函数，那么当且仅当 x 为 $powerful\ number$ 时，$h(x)$ 才不为 0，那么 f 的前缀和可以写成 $\sum_{i=1}^{n}f(i)=\sum_{i=1}^{n}h(i)\sum_{j=1}^{\lfloor\frac{n}{i}\rfloor}g(j)$，这个东西需要我们能够快速求 $g(x)$ 的前缀和，一般情况下我们需要使用 $min25$ 或者杜教筛来求 $g(x)$ 的前缀和，关于 $h(x)$ 我们直接枚举所有 $powerful\ number$ 即可，所以关于 $h(x)$ 我们只需要知道 $h(p^x)$ 的值，一般有两种方法，第一种是手动求一下，第二种就是递推一下，因为我们显然有 $h(p^k)=f(p^k)-\sum_{i=0}^{k-1}h(p^i)g(p^{k-i})$

模板

```
ll dfs(int k, ll m, ll h) { // 枚举所有 powerful number
    ll ans = h * D_Seive::sg(n / m) % p;
    for (ll res = n / m; 1ll * pri[k] * pri[k] <= res; ++k)
        for (ll e = 2, t = 1ll * pri[k] * pri[k]; t <= res; t *= pri[k], ++e)
            ans = (ans + dfs(k + 1, m * t, h * ch(pri[k], e, t) % p)) % p;
    return ans;
}
```

递推 $h(x)$ 并不影响复杂度

```
ll dfs(int k, ll m, ll h) {
    ll ans = h * D_Seive::sg(n / m) % p;
    for (ll res = n / m; 1ll * pri[k] * pri[k] <= res; ++k) {
        vector<ll> _h { 1, 0 }, g { 1, cg(pri[k], pri[k]) };
        for (ll e = 2, t = 1ll * pri[k] * pri[k]; t <= res; t *= pri[k], ++e) {
            _h.push_back(cf(t)); g.push_back(cg(pri[k], t));
            for (ll o = 0; o < e; ++o) _h[e] = (_h[e] - _h[o] * g[e - o]) % p;
            ans = (ans + dfs(k + 1, m * t, h * _h[e] % p)) % p;
        }
    }
    return ans;
}
```

例题

1. 简要题意：求 $\sum_{i=1}^{n}\sum_{j=1}^{m}[(i,j)=k]$

   $n,m,k\leq 5\times 10^4$

   简要题解：$\sum_{i=1}^{n}\sum_{j=1}^{m}[(i,j)=k]=\sum_{d=1}^{\lfloor\frac{n}{k}\rfloor}\mu(d)\lfloor\frac{n}{dk}\rfloor\lfloor\frac{m}{dk}\rfloor$

   $O(n)$ 预处理后可以做到单次 $O(\sqrt{n})$

   Luogu P3455 [POI2007]ZAP-Queries

2. 简要题意：$\sum_p \sum_{i=1}^n \sum_{j=1}^m [(i,j)=p]$

   $T=10^4, n,m \le 10^7$

   简要题解：$\sum_p \sum_{i=1}^n \sum_{j=1}^m [(i,j)=p] = \sum_{T=1}^n \lfloor \frac{n}{T} \rfloor \lfloor \frac{m}{T} \rfloor \sum_{p|T} \mu(\frac{T}{p})$

   $O(n)$ 预处理后可以做到单次 $O(\sqrt{n})$

   Luogu P2257 YY的GCD

3. 简要题意：$\sum_{i=1}^n \sum_{j=1}^m (i,j)^k$

   $n,m \le 5 \times 10^6, T \le 2 \times 10^3$

   简要题解：$\sum_{i=1}^n \sum_{j=1}^m (i,j)^k = \sum_{T=1}^n \lfloor \frac{n}{T} \rfloor \lfloor \frac{m}{T} \rfloor \sum_{t|T} t^k \mu(\frac{T}{t})$

   $O(n)$ 预处理后可以做到单次 $O(\sqrt{n})$

   Luogu P4449 于神之怒加强版

4. 简要题意：$\sum_{i=1}^n \sum_{j=1}^m [i,j]$

   $n,m \le 10^7$

   简要题解：$\sum_{i=1}^n \sum_{j=1}^m [i,j] = \sum_{T=1}^n \frac{\lfloor \frac{n}{T} \rfloor (\lfloor \frac{n}{T} \rfloor +1)}{2} \frac{\lfloor \frac{m}{T} \rfloor (\lfloor \frac{m}{T} \rfloor +1)}{2} T \sum_{d|T} d\mu(d)$

   $O(n)$ 预处理后可以做到单次 $O(\sqrt{n})$

   Luogu P1829 [国家集训队]Crash的数字表格 / JZPTAB

5. 简要题意：$\sum_{i=1}^n \sum_{j=1}^n [a_i,a_j]$

   $n \le 2 \times 10^5, a_i \le 10^6$

   简要题意：$\sum_{i=1}^n \sum_{j=1}^n [a_i,a_j] = \sum_{T=1}^N \frac{1}{T} (\sum_{T|a_i} a_i)^2 \sum_{d|T} d\mu(d)$

   可以做到 $O(n \log n)$

   AtCoder AGC038C LCMs

6. 简要题意：给定 n，求所有长度为 n 的 01 串的价值总和，一个 01 串的价值定义为 $\lfloor \frac{n}{k} \rfloor$，其中 k 是这个 01 串的最小周期

   $n \le 10^9$

简要题解：

$\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor}\lfloor\frac{n}{i}\rfloor f(i)+2^n-\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor}f(i)=2^n+\sum_{i=1}^{\lfloor\frac{n}{2}\rfloor}(\lfloor\frac{n}{i}\rfloor-1)f(i)$

$2^x=\sum_{d|x}f(d)$，也就是说 $(f\circ 1)(n)=2^n$，可以直接上杜教筛

2021杭电多校4 C Cycle Binary

7. 简要题意：定义积性函数 $f(x)$，且 $f(p^x)=(p^x)^2-p^x$，求 $\sum_{i=1}^n f(i)$

   $n\le 10^{10}$

   简要题解：直接上 Min_25 筛

   Luogu P5325 【模板】Min_25筛

8. 简要题意：求 $1\sim n$ 的素数个数

   $n\le 10^{11}$

   简要题解：我们令 $f(x)=1$，这东西显然是完全积性函数，然后直接 Min_25 筛即可

   Loj 6235 区间素数个数

9. 简要题意：给定 n，求 $\frac{1}{2}\sum_{i=1}^n\sigma_0^2(i)-\sigma_0(i)$

   $n\le 10^{10}$

   简要题解：注意到 $\sum_{i=1}^n\sigma_0(i)=\sum_{i=1}^n\lfloor\frac{n}{i}\rfloor$，对于 $\sum_{i=1}^n\sigma_0^2(i)$ 我们可以考虑用 Min_25 筛

   我们令 $f(x)=\sigma_0(x)^2$，这个东西显然是积性函数，且 $f(p^x)=(x+1)^2$

   Loj 6682 梦中的数论

10. 简要题意：求 $\sum_{i=1}^n\sum_{j=1}^m\tau(i)\tau(j)\tau((i,j))$

   $n,m\le 2\times 10^6$

   简要题解：$\sum_{i=1}^n\sum_{j=1}^m\tau(i)\tau(j)\tau((i,j))=\sum_{T=1}^n\sum_{i=1}^{\lfloor\frac{n}{T}\rfloor}\tau(iT)\sum_{j=1}^{\lfloor\frac{m}{T}\rfloor}\tau(jT)\sum_{t|T}\tau(t)\mu(\frac{T}{t})$

   我们考虑后面那个东西 $\sum_{d|n}\tau(d)\mu(\frac{n}{d})$，这个东西就是 $I\circ I\circ\mu$ 显然等于 I

   那么我们只需要求 $\sum_{T=1}^n\sum_{i=1}^{\lfloor\frac{n}{T}\rfloor}\tau(iT)\sum_{j=1}^{\lfloor\frac{m}{T}\rfloor}\tau(jT)$

   注意到这个东西就是 $\sum_{T=1}^n(\sum_{d|i}^n\tau(d))(\sum_{d|i}^m\tau(d))$，这是个狄利克雷后缀和的形式，可以做到

$O(n\log\log n)$

[Luogu P6810 「MCOI-02」Convex Hull 凸包]

11. 简要题意：求 $\sum_{i=1}^n\sum_{j=1}^n(i+j)^kf((i,j))(i,j)$，其中 $f(n)$ 当且仅当 n 无平方因子时为 1

    $n\le 5\times 10^6,k\le 10^{18}$

    简要题解：$\sum_{i=1}^n\sum_{j=1}^n(i+j)^kf((i,j))(i,j)=\sum_{T=1}^nT^k\sum_{i=1}^{\lfloor\frac{n}{T}\rfloor}\sum_{j=1}^{\lfloor\frac{n}{T}\rfloor}(i+j)^k\sum_{t|T}tf(t)\mu(\frac{T}{t})$

    我们令 $S(n)=\sum_{i=1}^n\sum_{j=1}^n(i+j)^k$，$F(n)=\sum_{i=1}^ni^k$，$G(n)=\sum_{i=1}^nF(i)$，容易得到 $S(n)=G(2n)-2G(n)$

    后面那个东西显然是 $\sum_{d|n}d\mu(d)\mu(\frac{n}{d})$，对于 $n=p^k$，当 $k>2$ 时为 0，这个东西可以线筛

    另外自然数幂和也可以线筛，因为他是完全积性函数，可以做到 $O(n)$ 预处理，单词询问 $O(\sqrt{n})$

    [Luogu P6156 简单题]

12. 简要题意：求 $\prod_{i_1=1}^n\prod_{i_2=1}^n\cdots\prod_{i_k=1}^n[i_1,i_2,\cdots,i_k]\bmod 998244353$

    $n\le 10^6,k\le 10^{100},T=1000$

    简要题意：我们考虑单独算每个质数 p 的贡献

    我们注意到 $lcm=p^t$ 太难计算，考虑换成 $p^t|lcm$ 的方式来计算

    对于一个 $lcm=p^t$ 的 k 元组，我们考虑分别算 t 次贡献，每次贡献为 p，那么我们可以这样计算贡献，令 $f_i(p)$ 表示 $p^i|[i_1,i_2,\cdots,i_k]$ 的 k 元组个数，这样一个 p 的贡献就是 $p^{\sum_{i=1}^{\log_pn}f_i(p)}$

    容易得到 $f_i(p)=n^k-(n-\lfloor\frac{n}{p^i}\rfloor)^k$， 这个整除启示我们进行数论分块

    我们可以预处理出前缀 i 的 p^t 的乘积，然后就可以直接数论分块，另外对于 k 我们直接对 $\varphi(\varphi(998244353))$ 取模即可，预处理 $O(n)$，单次询问 $O(\sqrt{n}\log n)$

    [Luogu P7360 「JZOI-1」红包]

13. 简要题意：给定一个长度为 n 的排列 p，求 $\sum_{i=1}^n\sum_{j=1}^n(i,j)(a_i,a_j)$

    $n\le 10^5$

    简要题解：首先我们划一下式子，这里扔掉 (i,j)，我们采用 φ 而不是 μ，能够得到 $\sum_{d=1}^n\varphi(d)\sum_{d|i}\sum_{d|j}(a_i,a_j)$

    我们考虑对于一个 d，不妨设有 m 个 a_i，那么贡献就是 $\sum_{i=1}^m\sum_{j=1}^m(a_i,a_j)$，用类似的式子化简可得

$\sum_{d=1}^{n}\varphi(d)(\sum_{i=1}^{m}[d|a_i])^2$

我们考虑一个暴力，暴力枚举 d，然后暴力枚举这 m 个 a_i，然后对于这些 a_i，暴力枚举约数

容易得到时间复杂度为 $O(\sum_{i=1}^{n}\tau(i)\tau(a_i))$，根据排序不等式，这个东西就是 $\sum_{i=1}^{n}\tau^2(i)\approx\frac{1}{\pi^2}n\log^3 n+o(n\log^2 n)$

Loj 6539. 奇妙数论题

14. 简要题意：求 $\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{p=1}^{\lfloor\frac{n}{j}\rfloor}\sum_{q=1}^{\lfloor\frac{n}{j}\rfloor}[(i,j)=1][(p,q)=1]$

    求 $n\le 2\times 10^9$

    简要题意：注意到后面的式子像是在枚举 gcd 为 j 的二元组 (p,q)，我们按照这个思路化简式子

    $\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{p=1}^{n}\sum_{q=1}^{n}[(i,j)=1][(p,q)=j]\to\sum_{i=1}^{n}\sum_{p=1}^{n}\sum_{q=1}^{n}[(i,p,q)=1]$，这个式子我们就很熟了

    容易得到 $\sum_{d=1}^{n}\mu(d)\lfloor\frac{n}{d}\rfloor^3$，这个东西可以直接数论分块加杜教筛，复杂度仍然是 $O(n^{\frac{2}{3}})$

    Luogu P6055 [RC-02] GCD

15. 简要题意：给定 n 和 m，有 m 次操作，每次操作要么给出 x,y,z，对于所有 $(i,x)=y$ 的 a_i 加上 z，要么给定 x，查询 $\sum_{i=1}^{x}a_i$

    $n,m\le 5\times 10^4$

    简要题解：我们考虑对于位置 k，我们的加上 $z[(k,x)=y]$，按照莫反的套路得到 $z\sum_{d|\frac{x}{y},dy|x}\mu(d)$

    我们考虑枚举 $\frac{x}{y}$ 的约数，然后对于所有 dy 的倍数都加上 $z\mu(d)$，我们可以将倍数增加，改成单点加，这样我们在查询的时候需要查询约数和，这里的复杂度为 $O(\sqrt{n}\log n)$

    那么前缀和的形式变成了 $\sum_{i=1}^{x}\lfloor\frac{x}{i}\rfloor a_i$，我们数论分块，复杂度仍然是 $O(\sqrt{n}\log n)$

    总的时间复杂度 $O(n\sqrt{n}\log n)$，似乎有 $O(n\sqrt{n\log n})$ 的做法

    hdu 4947 GCD Array

16. 简要题意：求 $\prod_{i=1}^{n}\prod_{j=1}^{n}\frac{[i,j]}{(i,j)}\bmod 104857601$

    $n\le 10^6$

    简要题解：$\prod_{i=1}^{n}\prod_{j=1}^{n}\frac{[i,j]}{(i,j)}=(n!)^{2n}\prod_{t=1}^{n}(t^{2\sum_{i=1}^{\lfloor\frac{n}{t}\rfloor}\varphi(i)-1)})^{-2}$

$\prod$ 转指数上 $\sum$ 是常见套路

可以做到预处理 $O(n)$，单次 $O(\sqrt{n}\log n)$

Luogu P5221 Product

17. 简要题意：给定 m，每次随机选择一个 1 到 m 的整数，与手上的数取 gcd，求期望多少次手上的数变成 1

$m \le 10^5$

简要题解：令 f_n 表示 n 变成 1 的期望次数，容易得到 $f_n = 1 + \frac{1}{m}\sum_{d|n} f(d)\sum_{i=1}^{m}[(i,n)=d]$

这个式子拿莫反变一下能够得到 $f_n = 1 + \frac{1}{m}\sum_{T|n}\lfloor\frac{m}{T}\rfloor\sum_{t|T} f_t\mu(\frac{T}{t})$，后面的那个东西我们令其为 g_n

然后我们在求 f_n 的时候只需要枚举约数即可，需要注意要把 f_n 提出来，算完 f_n 再更新 g_n 即可时间复杂度 $O(n\log n)$
CF 1139D Steps to One

18. 简要题意：给定 n，求 $\prod_{x=1}^{n}\prod_{d|x}\frac{d^{\tau(d)}}{\prod_{t|d}(t+1)^2}$

$m \le 2.5\times 10^9$

简要题解：我们首先对 $d^{\tau(d)}$ 下手，注意到该式等价于 $\prod_{t|d} d = \prod_{t|d} t\frac{d}{t} = (\prod_{t|d} t)^2$，这个变换再次展示了 $\prod$ 和 $\sum$ 的巨大区别 = =

那么我们现在的式子就是 $\prod_{x=1}^{n}\prod_{d|x}\prod_{t|d}\frac{t^2}{(t+1)^2}$，我们变换一下次序，容易得到 $\prod_{d=1}^{n}(\frac{d^2}{(d+1)^2})^{\sum_{i=1}^{\lfloor\frac{n}{d}\rfloor}\lfloor\frac{n}{di}\rfloor}$，指数上那个东西我们可以看成函数 $f(n)=\sum_{i=1}^{n}\lfloor\frac{n}{i}\rfloor=\sum_{i=1}^{n}\tau(i)$，那么答案就是这个东西 $\prod_{d=1}^{n}(\frac{d^2}{(d+1)^2})^{f(\lfloor\frac{n}{d}\rfloor)}$，暴力算的话数论分块套数论分块的复杂度是 $O(n^{\frac{3}{4}})$ 的，不能通过此题，但我们预处理 f 的前 $n^{\frac{2}{3}}$ 项就能做到 $O(n^{\frac{2}{3}})$

总的时间复杂度为 $O(\sqrt{n}\log n + n^{\frac{2}{3}})$

Luogu 「EZEC-3」四月樱花

19. 简要题意：给定 n,m,p,q，对于一个长度为 n 的序列 a_i，a_i 的每一位都是 $[1,m]$，如果整个序列的 gcd 小于等于 q，且整个序列的 lcm 大于等于 p，那么这个序列是合法的，每个合法序列的价值是 $\prod_{i=1}^{n} a_i$，求所有合法序列的价值的和

$n \le 998244351, p,q \le m \le 2\times 10^5$

简要题解：注意到一个序列的 gcd 一定是 lcm 的约数，所以我们考虑直接求 g_k 表示 gcd 为 1，lcm 为

k 的合法序列的价值的和，这样我们再搞一个调和级数的枚举就能求出所有合法的 $g_{x,y}$

对于 g_k，显然我们不太好直接操作，我们考虑求 $f_k=\sum_{d|k} g_d$，即求 gcd 为 1，且 lcm 是 k 的约数的合法序列的价值的和，我们尝试列一下 f_k 的式子

$$
\begin{aligned}
f_k &= \sum_{i_1=1}^{m}\sum_{i_2=1}^{m}\cdots\sum_{i_n=1}^{m}[(i_1,i_2,\cdots,i_n)=1][[i_1,i_2,\cdots,i_n]|k]\prod_{j=1}^{n} i_j \\
&= \sum_{d|k}\mu(d)(\sum_{d|i,i|k} i)^n \\
&= \sum_{d|k}\mu(d)(d\sigma(\frac{k}{d}))^n \\
&= \sum_{d|k}\mu(d)d^n\sigma^n(\frac{k}{d})
\end{aligned}
$$

显然这个东西我们可以用调和级数的做法算出 f_1 到 f_m，然后用作经典的莫比乌斯反演，我们能算出 g_k，但是现在我们注意到我们只计算了 lcm 小于等于 m 的情况，但题目实际要求 lcm 大于等于 p 的情况，正难则反，我们求出只有 gcd 的要求下的所有答案，简单 lcm 小于 p 的即可，所有东西的 n 次方都可以预处理，时间复杂度 $O(n\log n)$

The 2021 CCPC Guangzhou Onsite K Magus Night

20. 简要题意：给定 n，求 $\frac{1}{n}\sum_{i=1}^{n} f(i)$，其中 $f(x)$ 是积性函数，且 $f(p^k)=\frac{p^k}{k}$，答案对质数 4179340454199820289 取模

    $n\leq 10^{12}$

    简要题解：看到 $f(p^k)$ 的式子，容易想到 $min25$，但是 10^{12} 加上需要用龟速乘导致 $min25$ 等一类亚线性筛跑不动，我们考虑 PN 筛，构造函数 $g(x)=x$，可以求得 $h(p^k)=\frac{p^k}{k\times(k-1)}$，$g(x)$ 的前缀和可以 $O(1)$ 求，那么我们的时间复杂度为 $O(\sqrt{n})$

    2022杭电多校6 B Jo loves counting

21. 简要题意：现在有 T 次询问，每次询问给定 n,m 求 $\sum_{i=1}^{n}\sum_{j=1}^{m}\varphi(ij)$

    $T\leq 10^4,n,m\leq 10^5$

    简要题解：我们根据 $\varphi(ab)=\varphi(a)\varphi(b)\frac{(a,b)}{\varphi((a,b))}$，根据这个化简我们可以得到 $\sum_{T=1}^{n}\sum_{t|T}\frac{t}{\varphi(t)}\mu(\frac{T}{t})f(T,\lfloor\frac{n}{T}\rfloor)f(T,\lfloor\frac{m}{T}\rfloor)$，其中 $f(x,y)=\sum_{i=1}^{y}\varphi(ix)$，注意到所有合法的 $f(x,y)$ 都满足 $xy\leq n$，那么总共只有 $O(n\log n)$ 个 $f(x,y)$，且可以 $O(n\log n)$ 预处理

    但是只预处理 $f(x,y)$ 我们依次询问也只能做到 $O(n)$，不能通过此题

我们考虑预处理 $f(x,y)f(x,z)$ 的前缀和，我们只预处理 $f(x,y)f(x,z),y,z \le B$ 的前缀和，这一部分的时间复杂度大概是 $O(nB\log n)$ 的，那么在询问的时候我们考虑对于 $\lfloor \frac{n}{i} \rfloor > B$ 的 i 暴力算，这样的 i 只有 $\lfloor \frac{n}{B} \rfloor$ 个，求一次的时间复杂度为 $O(1)$，那么总的时间复杂度为 $O(nB\log n + T(\lfloor \frac{n}{B} \rfloor + \sqrt{n})$，我们取 B 为 50 左右可以通过此题

Luogu P4240 毒瘤之神的考验

22. 简要题意：给定 n,p，求 $\sum_{i=1}^{n}\sum_{j=1}^{n}(i,j)^{(i,j)}$

$n \le 1.5 \times 10^6$

简要题解：经过简单的式子化简，我们可以得到 $ans = \sum_{T=1}^{n}\sum_{t|T}\mu(\frac{T}{t})f^2(t^T,\lfloor \frac{n}{T} \rfloor)$，其中 $f(x,y)=\sum_{i=1}^{y}x^i$

我们考虑直接枚举 T,t 这个东西的时间复杂度是 $O(n\log n)$，$f(x,y)$ 显然是一个等比数列求和东西，我们直接套用等比数列求和的公式的话的时间复杂度为 $O(\log p)$，总的时间复杂度为 $O(n\log n\log p)$

但是我们注意到 t^T 的前缀和最多才到 $\lfloor \frac{n}{T} \rfloor$ 项，如果我们计算这部分的时间复杂度是 $O(\log\lfloor \frac{n}{T} \rfloor)$ 应该能显著减少常数，同时我想起来等比数列前缀和(n项)有 $O(\log n)$ 的做法，然后我们把这个做法套上去，发现这样就过了

但其实 $O(\sum_{i=1}^{n}\sum_{j=1}^{\lfloor \frac{n}{i} \rfloor}\log(\lfloor \frac{n}{ij} \rfloor)=O(n\log n)$，感觉非常神奇

Luogu P6825 「EZEC-4」求和

23. 简要题意：给定一个长度为 n 的序列 a_i，求 $max_{1\le i<j\le n}lcm(a_i,a_j)$

$n,a_i \le 10^5$

简要题解：注意到 a_i 的值域很小，我们考虑一个针对值域的做法，首先我们把所有 a_i 的约数都求出来，令其为集合 S，那么最终的答案一定为从 S 中选出两个互质的数的最大 lcm，我们考虑二分 lcm，那么如果 $\sum_{x\in S}\sum_{y\in S}[(x,y)=1][xy\ge lcm]$，按照莫比乌斯反演的套路，我们得到 $\sum_{d\in S}\mu(d)\sum_{d|x}\sum_{d|y}[xy\ge lcm]$，我们枚举 d，后面枚举 x 之后是一个双指针，总时间复杂度为 $O(n\log^2 n)$

其实这题还有一个 $O(n\log n)$ 的做法，我们还是考虑对 S 求 lcm，我们维护一个栈，从大到小加入数 x，如果栈中存在一个 y 与 x 互质，那么对于 $x<z<y$，z 一定没用了，因为我们现在的答案至少是 xy，而后面加入的数 w 与 z 的答案至多是 $wz<xy$，所以我们可以一直弹栈顶直到栈中不存在与 x 互质的数，那么我们如何判断是否存在于 x 互质的数呢？答案还是莫反，我们有 $\sum_{y\in S}[(x,y)=1]=\sum_{d|x}\mu_d cnt_d$，其中 cnt_d 表示有多少数是 d 的倍数，这个东西可以在入栈和出栈时顺便维护，时间复杂度 $O(n\log n)$

CF 1285F Classical?

24. 简要题意：给定两个长度为 n 的序列 a_i, b_i，求长度为 n 的序列 c_i，其中 $c_k = \max_{gcd(i,j)=k} |a_i - b_j|$

    $n \le 10^5, a_i, b_i \le 10^9$

    简要题解：我们考虑只统计 $a_i \le b_j$ 的答案，对于 $a_i > b_j$ 的答案我们只需要交换 a 和 b 重新求一次即可，另外我们只考虑 $gcd(i,j) = 1$ 的情况，对于 $gcd(i,j) = d$ 的情况，我们只需要将 i 和 j 都除掉 d 即可

    首先我们将 a 按升序排序，b 按降序排序，那么我们考虑维护一个当前有效 b 的集合 S，我们当前枚举到 a_i，我们考察集合 S 中是否有与 a_i 互质的 b_j，如果存在这么一个 b_j，那么集合内所有小于 b_j 的数都没有用了，原因我们可以这样考虑对于我们接下来枚举到的 $a_{i'} > a_i$，其与 $b_{j'} < b_j$ 所组成的答案一定小于 a_i 和 b_j 组成的答案，那么我们的做法就是每次删除 S 中与 a_i 互质的数 b_j 以及小于 b_j 的数，注意到我们在一开始就完成了所有 b_j 的插入，所以我们可以维护一个栈来代替集合，至于如何判断是否存在与 a_i 互质的数，通过莫反，我们发现只需要 cnt_d，cnt_d 表示 S 中是 d 的倍数的个数

    时间复杂度 $O(n \log^2 n)$，似乎还有比较好想的 $O(n \log^3 n)$ 的二分做法

    2018-2019 Summer Petrozavodsk Camp, Oleksandr Kulkov Contest 2 B Yet Another Convolution

25. 简要题意：给定 n，求 $\sum_{i=1}^{n} \sum_{j=1}^{n} gcd(fib(i), fib(j))$

    $n \le 10^{10}$

    简要题解：我们知道 $gcd(fib(i), fib(j)) = fib(gcd(i,j))$，那么我们容易化简得到 $ans = \sum_{t=1}^{n} fib(t) \sum_{i=1}^{\lfloor \frac{n}{t} \rfloor} \sum_{j=1}^{\lfloor \frac{n}{t} \rfloor} [(i,j)=1]$，后面那两个 $\sum$，如果我们套用传统的 μ 去化简的话是不容易处理的，但我们注意到 $\sum$ 的上指标相同，能够想起一个经常被遗忘的式子 $\sum_{i=1}^{n} \sum_{j=1}^{n} [(i,j)=1] = 2\sum_{i=1}^{n} \varphi(i) - 1$,

    我们令 $S(n)$ 表示 φ 的前缀和，那么原式就是 $\sum_{i=1}^{n} fib(i) S(\lfloor \frac{n}{i} \rfloor)$，相当于是一个数论分块套杜教筛，时间复杂度为 $O(n^{\frac{2}{3}})$

    The Hangzhou Normal U Qualification Trials for ZJPSC 2021 B Baom and Fibonacci

26. 简要题意：给定一个长度为 n 的序列 a_i 和两个整数 X 和 Y，求有多少数对 (i,j) 满足存在一个整数 v，使得 $(v, a_i) = X$，同时 $[v, a_j] = Y$

    $n \le 2 \times 10^5, a_i, X, Y \le 10^{18}$

    简要题解：我们考虑如果存在一个数对，那么一定满足 $X|Y, X|a_i, a_j|Y$，我们令 S 表示 Y 的素数集合，然后我们对于每个素数单独考虑，对于一个素数 $p \in S$，我们令 c_x 表示 X 的次数，c_y 表示 y 的次

数，c_i 表示 a_i 的次数，c_j 表示 a_j 的次数，那么对于 $c_x = c_y$ 的情况我们一定可以选出一个 v，对于 $c_x < c_y$，通过讨论我们发现，$c_i = c_x$ 和 $c_j = c_y$ 两者必须至少满足一个我们才能找到合法的 v

另外我们发现 10^{18} 范围内的数最多只有 15 个质因子，我们状压一下，相当于求 $\sum_{S\oplus T=U} fsg_T$，这个东西拿高维前缀和求一下即可，时间复杂度 $O(V^{\frac{1}{4}}+2^{15}15)$

CF 1016G Appropriate Team

27. 简要题意：给定一个长度为 n 的序列 a_i，求 $\sum_{i=1}^n\sum_{j=1}^n \varphi(gcd(a_i,a_j^3))$

    $n\leq 3\times 10^5, a_i\in[1,n]$

    简要题解：我们直接化简

$$\begin{aligned}\sum_{i=1}^n\sum_{j=1}^n\varphi(gcd(a_i,a_j^3)) &= \sum_{t=1}^n\varphi(t)\sum_{i=1}^n\sum_{j=1}^n[(a_i,a_j^3)=t]\\ &=\sum_{t=1}^n\varphi(t)\sum_{i=1}^n\sum_{j=1}^n[(\frac{a_i}{t},\frac{a_j^3}{t})=1]\\ &=\sum_{t=1}^n\varphi(t)\sum_{i=1}^n\sum_{j=1}^n\sum_{d|(\frac{a_i}{t},\frac{a_j^3}{t})}\mu(d)\\ &=\sum_{t=1}^n\varphi(t)\sum_{d=1}^{\lfloor\frac{n}{t}\rfloor}\mu(d)(\sum_{d|a_i})(\sum_{d|a_j^3})\end{aligned}$$

    注意到我们只需要预处理 $\sum_{d|a_i}$ 和 $\sum_{d|a_i^3}$ 即可，后面那个东西本质和前面的一样，时间复杂度 $O(n\log n)$

    牛客 contest 43058E 炫酷反演魔术

------------- 本文结束 感谢阅读 -------------

Tech # 积性函数