数论 4

Min_25筛

算法简介

一种 $O\left(\frac {n^{\frac 34}}{\log n}\right)$ 计算积性函数 $F(x)$ 前缀和的算法。

算法思路

首先给定 Min_25 筛的适用条件：$F(p)$ 的值可以拆分为若干个完全积性函数 $f(p)$，且 $\sum f(x)$ 可以快速计算。

设 $\text{midv}(x)$ 表示 x 的最小素因子。将所有素数从小到大排列，记为 $p_1,p_2,p_3\cdots$

考虑构造完全积性函数 $f(x)$ 满足 $f(p)=F(p)$，设 $g(n,k)=\sum_{i=1}^n[\text{midv}(i)\gt p_k\text{ or } i\in \text{prime}]f(i)$。

考虑从 $g(n,k-1)$ 转移到 $g(n,k)$，等价于减去 $\text{midv}(i)=p_k$ 且 $i\not\in \text{prime}$ 的 $f(i)$ 的贡献。如果 $p_k^2\ge n$，则这样的数不存在。

否则将所有满足该条件的 i 提取出一个 p_k，记 $i'=\frac i{p_k}$，于是 $\text{midv}(i')\ge p_k,f(i')=\frac {f(i)}{f(p_k)}$。

考虑减去 $f(p_k)g(\lfloor \frac n{p_k}\rfloor,k-1)$，发现 $g(\lfloor \frac n{p_k}\rfloor,k-1)$ 多包含了 $\sum_{i=1}^{k-1}f(p_i)$，于是再补上。

于是有状态转移方程

$$ g(n,k) \begin{cases} g(n,k-1), &p_k^2\ge n\\ g(n,k-1)-f(p_k)(g(\lfloor \frac n{p_k}\rfloor,k-1)-\sum_{i=1}^{k-1}f(p_i)), &p_k^2\lt n \end{cases} $$

由于 $\lfloor \frac{\lfloor \frac na\rfloor}{b}\rfloor=\lfloor \frac n{ab}\rfloor$，于是 $g(a,b)$ 中的 a 只有 $O(\sqrt n)$ 个。

使用刷表法暴力转移上述方程直到 $p_{k+1}^2\gt n$，此时得到 $g(a,k)=\sum_{i=1}^a[i\in \text{prime}]f(i)=\sum_{i=1}^a[i\in \text{prime}]F(i),a\in \lfloor \frac nx\rfloor$。

不妨将此时的 $g(n,k)$ 记为 $g(n)$，由于玄学因素，该过程的时间复杂度为 $O\left(\frac {n^{\frac 34}}{\log n}\right)$。

接下来设 $S(n,k)=\sum_{i=1}^n[mdiv(i)\gt p_k]F(i)$，于是目标就是求 $S(n,0)+F(1)=S(n,0)+1$(积性函数必有 $F(1)=1$)。

将 $S(n,k)$ 的和分为 $\sum[i\in \text{prime},i\gt p_k]S(i)$ 和 $\sum[i\not\in \text{prime},\text{midv}(i)\gt p_k]S(i)$ 两部分。

前面部分有 $\sum[i\in \text{prime},i\gt p_k]S(i)=g(n)-\sum_{i=1}^k F(p_i)$，后面部分可以枚举最小素因子及其幂次，可以得状态转移方程

$$S(n,k)=g(n)-\sum_{i=1}^k F(p_i)+\sum_{i=k+1}^{p_i^2\le n}\sum_{p_i^j\le n}F(p_i^j)(S(\lfloor\frac n{p_i^j}\rfloor,i)+[j\neq 1])$$

最后补上 $F(p_i^j)[j\neq 1]$ 是因为 $F(p_i^j)S(\lfloor\frac n{p_i^j}\rfloor,i)$ 不包含 $F(p_i^j)$ 的贡献，但 $F(p_i)$ 的贡献已经计算。

由于 $p_i^2\gt n$ 时 $\sum_{p_i^j\le n}F(p_i^j)(S(\lfloor\frac n{p_i^j}\rfloor,i)+[j\neq 1])=0$□于是只需要枚举 $O\left(\frac {\sqrt n}{\log n}\right)$ 个素数即可。

由于玄学因素，该过程不需要记忆化且时间复杂度为 $O\left(\frac {n^{\frac 34}}{\log n}\right)$□

算法例题

例题一

洛谷p5325

题意

给定积性函数 $F(p^k)=p^k(p^k-1)$□计算 F 前 n 项和。

题解

构造完全积性函数 $f_1(x)=x^2,f_2(x)=x$□于是可以计算出 $g_1(n),g_2(n)$□

然后令 $g(n)=g_1(n)-g_2(n)=\sum_{i=1}^n[i\in \text{prime}]F(i)$□即可计算出 $S(n,0)$□

```
const int MAXN=1e5+5,Mod=1e9+7,inv2=500000004,inv6=166666668;
namespace Min_25{
    LL n,blk[MAXN<<1];
    int sqr,vis[MAXN],prime[MAXN],p_cnt;
    int sp1[MAXN],sp2[MAXN],sp[MAXN],g1[MAXN<<1],g2[MAXN<<1],g[MAXN<<1];
    int b_cnt,idx1[MAXN],idx2[MAXN];
    int S(LL a,int b){
        if(a<=prime[b])return 0;
        int pos=(a<=sqr)?idx1[a]:idx2[n/a];
        int ans=(g[pos]-sp[b])%Mod;
        for(int i=b+1;i<=p_cnt&&1LL*prime[i]*prime[i]<=a;i++){
            LL pow_p=prime[i];
            for(int j=1,mod_p;pow_p<=a;j++){
                mod_p=pow_p%Mod;
ans=(ans+1LL*mod_p*(mod_p-1)%Mod*(S(a/pow_p,i)+(j!=1)))%Mod;
                pow_p*=prime[i];
            }
        }
```

```
        return (ans+Mod)%Mod;
    }
    int cal(LL n){
        Min_25::n=n;
        sqr=sqrt(n+0.5);
        for(int i=2;i<=sqr;i++){
            if(!vis[i])prime[++p_cnt]=i;
            for(int j=1;j<=p_cnt&&i*prime[j]<=sqr;j++){
                vis[i*prime[j]]=1;
                if(i%prime[j]==0)break;
            }
        }
        _rep(i,1,p_cnt){
            sp1[i]=(sp1[i-1]+1LL*prime[i]*prime[i])%Mod;
            sp2[i]=(sp2[i-1]+prime[i])%Mod;
            sp[i]=(sp1[i]-sp2[i]+Mod)%Mod;
        }
        for(LL lef=1,rig=0;lef<=n;lef=rig+1){
            rig=n/(n/lef);
            blk[++b_cnt]=n/lef;
            int temp=blk[b_cnt]%Mod;
g1[b_cnt]=(1LL*temp*(temp+1)%Mod*(temp<<1|1)%Mod*inv6+Mod-1)%Mod;
            g2[b_cnt]=(1LL*temp*(temp+1)%Mod*inv2+Mod-1)%Mod;
            if(blk[b_cnt]<=sqr)
            idx1[blk[b_cnt]]=b_cnt;
            else
            idx2[rig]=b_cnt;
        }
        _rep(i,1,p_cnt){
            LL pow_p=1LL*prime[i]*prime[i];
            for(int j=1;j<=b_cnt&&blk[j]>=pow_p;j++){
                LL pos=blk[j]/prime[i];
                pos=(pos<=sqr)?idx1[pos]:idx2[n/pos];
                g1[j]=(g1[j]-1LL*prime[i]*prime[i]%Mod*(g1[pos]-
sp1[i-1]))%Mod;
                g1[j]=(g1[j]+Mod)%Mod;
                g2[j]=(g2[j]-1LL*prime[i]*(g2[pos]-sp2[i-1]))%Mod;
                g2[j]=(g2[j]+Mod)%Mod;
            }
        }
        _rep(i,1,b_cnt)
        g[i]=(g1[i]-g2[i]+Mod)%Mod;
        return (S(n,0)+1)%Mod;
    }
}
int main()
{
    LL n=read_LL();
    enter(Min_25::cal(n));
    return 0;
```

```
}
```

例题二

LOJ6053

题意

给定积性函数 $F(p^k)=p\oplus k$ 计算 F 前 n 项和。

题解

发现 $f(p)=p-1,p\gt 2$ 于是先求出 $g(n)=\sum_{i=1}^n[i\in \text{prime}]f(p)$ 然后修正一下 $F(2)$ 处的误差。最后套 Min_25 筛即可。

```
const int MAXN=1e5+5,Mod=1e9+7,inv2=500000004;
namespace Min_25{
    LL n,blk[MAXN<<1];
    int sqr,vis[MAXN],prime[MAXN],p_cnt;
    int sp[MAXN],g1[MAXN<<1],g2[MAXN<<1],g[MAXN<<1];
    int b_cnt,idx1[MAXN],idx2[MAXN];
    int S(LL a,int b){
        if(a<=prime[b])return 0;
        int pos=(a<=sqr)?idx1[a]:idx2[n/a];
        int ans=(g[pos]-sp[b])%Mod;
        for(int i=b+1;i<=p_cnt&&1LL*prime[i]*prime[i]<=a;i++){
            LL pow_p=prime[i];
            for(int j=1;pow_p<=a;j++){
                ans=(ans+1LL*(prime[i]^j)%Mod*(S(a/pow_p,i)+(j!=1)))%Mod;
                pow_p*=prime[i];
            }
        }
        return (ans+Mod)%Mod;
    }
    int cal(LL n){
        Min_25::n=n;
        sqr=sqrt(n+0.5);
        for(int i=2;i<=sqr;i++){
            if(!vis[i])prime[++p_cnt]=i;
            for(int j=1;j<=p_cnt&&i*prime[j]<=sqr;j++){
                vis[i*prime[j]]=1;
                if(i%prime[j]==0)break;
            }
        }
        _rep(i,1,p_cnt)
```

```
        sp[i]=(sp[i-1]+prime[i])%Mod;
        for(LL lef=1,rig=0;lef<=n;lef=rig+1){
            rig=n/(n/lef);
            blk[++b_cnt]=n/lef;
            int temp=blk[b_cnt]%Mod;
            g1[b_cnt]=(1LL*temp*(temp+1)%Mod*inv2+Mod-1)%Mod;
            g2[b_cnt]=(temp+Mod-1)%Mod;
            if(blk[b_cnt]<=sqr)
            idx1[blk[b_cnt]]=b_cnt;
            else
            idx2[rig]=b_cnt;
        }
        _rep(i,1,p_cnt){
            LL pow_p=1LL*prime[i]*prime[i];
            for(int j=1;j<=b_cnt&&blk[j]>=pow_p;j++){
                LL pos=blk[j]/prime[i];
                pos=(pos<=sqr)?idx1[pos]:idx2[n/pos];
                g1[j]=(g1[j]-1LL*prime[i]*(g1[pos]-sp[i-1]))%Mod;
                g1[j]=(g1[j]+Mod)%Mod;
                g2[j]=(g2[j]-(0LL+g2[pos]-(i-1)))%Mod;
                g2[j]=(g2[j]+Mod)%Mod;
            }
        }
        _rep(i,1,p_cnt)
        sp[i]=(sp[i]-i+2+Mod)%Mod;
        _for(i,1,b_cnt)
        g[i]=(g1[i]-g2[i]+2+Mod)%Mod;
        return (S(n,0)+1)%Mod;
    }
}
int main()
{
    LL n=read_LL();
    enter(Min_25::cal(n));
    return 0;
}
```

例题三

UOJ188

题意

定义 $F(x)$ 代表 x 的次大素因子，约定如果 x 不存在最小素因子则 $F(x)=0$计算 $\sum_{i=l}^rF(i)$

题解

设 $g(n)$ 表示 $1\sim n$ 中素数个数，$S(n,k)=\sum_{i=1}^n[mdiv(i)\gt p_k]F(i)$。

先将 $S(n,k)$ 分为素数贡献和合数贡献。显然素数贡献为 0，故只考虑合数贡献。

将合数分为最小素因子等于次大素因子的数和最小素因子不等于次大素因子的数，枚举合数的最小素因子及其幂次。

对于最小素因子不等于次大素因子的数，将最小素因子的全部幂次直接提取出来，显然这不会影响这些数的次大素因子，该类数的贡献为

$$\sum_{i=k+1}^{p_i^2\le n}\sum_{p_i^j\le n}^nS(\lfloor\frac n{p_i^j}\rfloor,i)$$

而对于最小素因子等于次大素因子的数，它一定可以写成 $p_i^jp_x(x\ge i)$ 的形式，于是该类数的贡献为

$$\sum_{i=k+1}^{p_i^2\le n}\sum_{p_i^{j+1}\le n}(g(\lfloor\frac n{p_i^j}\rfloor)-i+1)$$

于是可以得到式子

$$S(n,k)=\sum_{i=k+1}^{p_i^2\le n}\sum_{p_i^j\le n}^n\left(S(\lfloor\frac n{p_i^j}\rfloor,i)+\max(g(\lfloor\frac n{p_i^j}\rfloor)-i+1,0)\right)$$

```
const int MAXN=sqrt(1e11)+10;
namespace Min_25{
    LL n,blk[MAXN<<1],g[MAXN<<1];
    int sqr,vis[MAXN],prime[MAXN],p_cnt;
    int b_cnt,idx1[MAXN],idx2[MAXN];
    int id(LL x){return x<=sqr?idx1[x]:idx2[n/x];}
    LL S(LL a,int b){
        if(a<=prime[b])return 0;
        LL ans=0;
        for(int i=b+1;i<=p_cnt&&1LL*prime[i]*prime[i]<=a;i++){
            LL pow_p=prime[i];
            for(int j=1;pow_p<=a;j++){
                ans=ans+prime[i]*max(0LL,g[id(a/pow_p)]-i+1)+S(a/pow_p,i);
                pow_p*=prime[i];
            }
        }
        return ans;
    }
    LL cal(LL n){
        Min_25::n=n;
        sqr=sqrt(n+0.5);
        b_cnt=p_cnt=0;
        for(int i=2;i<=sqr;i++){
            if(!vis[i])prime[++p_cnt]=i;
            for(int j=1;j<=p_cnt&&i*prime[j]<=sqr;j++){
                vis[i*prime[j]]=1;
```

```
                if(i%prime[j]==0)break;
            }
        }
        for(LL lef=1,rig=0;lef<=n;lef=rig+1){
            rig=n/(n/lef);
            blk[++b_cnt]=n/lef;
            g[b_cnt]=blk[b_cnt]-1;
            if(blk[b_cnt]<=sqr)
            idx1[blk[b_cnt]]=b_cnt;
            else
            idx2[rig]=b_cnt;
        }
        _rep(i,1,p_cnt){
            LL pow_p=1LL*prime[i]*prime[i];
            for(int j=1;j<=b_cnt&&blk[j]>=pow_p;j++){
                int pos=id(blk[j]/prime[i]);
                g[j]=g[j]-(g[pos]-i+1);
            }
        }
        return S(n,0);
    }
}
int main()
{
    LL L=read_LL(),R=read_LL();
    enter(Min_25::cal(R)-Min_25::cal(L-1));
    return 0;
}
```

例题四

LOJ6682

题意

$$\sum_{i=1}^n\sum_{j=1}^n\sum_{k=1}^n[(j\mid i)\text{ and }((j+k)\mid i)]$$

题解

$$\sum_{i=1}^n\sum_{j=1}^n\sum_{k=1}^n[(j\mid i)\text{ and }((j+k)\mid i)]=\sum_{i=1}^n\sum_{j=1}^n\sum_{k=j+1}^n[(j\mid i)\text{ and }(k\mid i)]=\sum_{i=1}^n{\tau(i)\choose 2}$$

考虑 i 作为因子对 $\lfloor\frac ni\rfloor$ 个数产生了贡献，于是 $\sum_{i=1}^n\tau(i)=\sum_{i=1}^n\lfloor\frac ni\rfloor$可以整数分块解决。

记 $F(i)=\tau^2(i)$则 $f(p)=4$套 Min_25 筛即可。

```
const int MAXN=1e5+5,Mod=998244353,inv2=(Mod+1)/2;
namespace Min_25{
    LL n,blk[MAXN<<1];
    int sqr,vis[MAXN],prime[MAXN],p_cnt,g[MAXN<<1];
    int b_cnt,idx1[MAXN],idx2[MAXN];
    int S(LL a,int b){
        if(a<=prime[b])return 0;
        int pos=(a<=sqr)?idx1[a]:idx2[n/a];
        int ans=(g[pos]-4LL*b)%Mod;
        for(int i=b+1;i<=p_cnt&&1LL*prime[i]*prime[i]<=a;i++){
            LL pow_p=prime[i];
            for(int j=1;pow_p<=a;j++){
                ans=(ans+1LL*(j+1)*(j+1)%Mod*(S(a/pow_p,i)+(j!=1)))%Mod;
                pow_p*=prime[i];
            }
        }
        return (ans+Mod)%Mod;
    }
    int cal(LL n){
        Min_25::n=n;
        sqr=sqrt(n+0.5);
        for(int i=2;i<=sqr;i++){
            if(!vis[i])prime[++p_cnt]=i;
            for(int j=1;j<=p_cnt&&i*prime[j]<=sqr;j++){
                vis[i*prime[j]]=1;
                if(i%prime[j]==0)break;
            }
        }
        for(LL lef=1,rig=0;lef<=n;lef=rig+1){
            rig=n/(n/lef);
            blk[++b_cnt]=n/lef;
            g[b_cnt]=(blk[b_cnt]-1)%Mod;
            if(blk[b_cnt]<=sqr)
            idx1[blk[b_cnt]]=b_cnt;
            else
            idx2[rig]=b_cnt;
        }
        _rep(i,1,p_cnt){
            LL pow_p=1LL*prime[i]*prime[i];
            for(int j=1;j<=b_cnt&&blk[j]>=pow_p;j++){
                LL pos=blk[j]/prime[i];
                pos=(pos<=sqr)?idx1[pos]:idx2[n/pos];
                g[j]=(g[j]-(g[pos]-(i-1)))%Mod;
                g[j]=(g[j]+Mod)%Mod;
            }
        }
        _rep(i,1,b_cnt)
        g[i]=g[i]*4LL%Mod;
```

```
        return (S(n,0)+1)%Mod;
    }
}
int cal(LL n){
    LL lef=1,rig;
    int ans=0;
    while(lef<=n){
        rig=n/(n/lef);
        ans=(ans+1LL*(n/lef)*(rig-lef+1))%Mod;
        lef=rig+1;
    }
    return ans;
}
int main()
{
    LL n=read_LL();
    enter(1LL*(Min_25::cal(n)-cal(n)+Mod)*inv2%Mod);
    return 0;
}
```

例题五

题意

设 $n=p_1^{\alpha_1}p_2^{\alpha_2}\cdots p_k^{\alpha_k}$，则 $f(n)=\alpha_1+\alpha_2+\cdots \alpha_k$，求

$$\sum_{i=1}^nf(i!)$$

题解

发现 $f(ab)=f(a)+f(b)$，于是

$$\sum_{i=1}^nf(i!)=\sum_{i=1}^n\sum_{j=1}^if(j)=\sum_{i=1}^n(n-i+1)f(i)=(n+1)\sum_{i=1}^nf(i)-\sum_{i=1}^nif(i)$$

不妨将 p^k 对 $f(n)$ 的贡献分配给 $p,p^2\cdots p^k$，于是

$$(n+1)\sum_{i=1}^nf(i)-\sum_{i=1}^nif(i)=\sum_{i=1}^{p_i\le n}\sum_{k=1}^{p_i^k\le n}(n+1)\lfloor\frac n{p_i^k}\rfloor-\frac {\lfloor\frac n{p_i^k}\rfloor(\lfloor\frac n{p_i^k}\rfloor+1)}{2}p_i^k$$

对 $k\ge 2$ 的情况，暴力计算时间复杂度 $O(\sum_{k=2}^{2^k\le n}n^{\frac 1k})\approx O(\sqrt n)$。

对 $k=1$，则需计算

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

通过 Min_25 筛计算出 $g_1(n)=\sum_{i=1}^n[i\in \text{prime}],g_2(n)=\sum_{i=1}^n[i\in \text{prime}]i$

最后考虑整数分块，发现一个块的贡献为 $(g_1(r)-g_1(l-1))(n+1)\lfloor\frac n{p_i}\rfloor-(g_2(r)-g_2(l-1))\frac {\lfloor\frac n{p_i}\rfloor(\lfloor\frac n{p_i}\rfloor+1)}{2}p_i$

$g(r)-g(l-1)$ 恰好为 Min_25 筛的相邻块，可以直接计算。总时间复杂度 $O\left(\frac {n^{\frac 34}}{\log n}\right)$