较深的数论知识选讲

COLD_CHAIR

目录

- 类欧几里得算法
- 群论及burnside引理
- Mobius反演及杜教筛、min_25筛

- 由于下午的讲课是省选难度，所以选的题都比较简单，基本上是例题难度，难题放到了晚上。

类欧几里得算法

- 先看一道经典题：

$$f(a,b,c,n)=\sum_{i=0}^{n}\lfloor\frac{ai+b}{c}\rfloor$$

- a,b,c,n都是正整数。
- n<=10^18

当a>=c或b>=c时

设$a' = a \ mod \ c, b' = b \ mod \ c$

$$f(a,b,c,n) = \sum_{i=0}^{n} \lfloor \frac{a'i+b'}{c} \rfloor + \lfloor \frac{a}{c} \rfloor * i + \lfloor \frac{b}{c} \rfloor$$
$$= f(a',b',c,n) + n(n+1)/2 * \lfloor \frac{a}{c} \rfloor + (n+1) * \lfloor \frac{b}{c} \rfloor$$

推式子

设$m = \lfloor \frac{an+b}{c} \rfloor$。

$$
\begin{aligned}
& f(a,b,c,n) \\
&= \sum_{i=0}^{n} \lfloor \frac{ai+b}{c} \rfloor \\
&= \sum_{j=1}^{m} \sum_{i=0}^{n} [\lfloor \frac{ai+b}{c} \rfloor >= j] \\
&= \sum_{j=0}^{m-1} \sum_{i=0}^{n} [\lfloor \frac{ai+b}{c} \rfloor >= j+1] \\
&= \sum_{j=0}^{m-1} \sum_{i=0}^{n} [ai >= jc+c-b] \\
&= \sum_{j=0}^{m-1} \sum_{i=0}^{n} [ai > jc+c-b-1] \\
&= \sum_{j=0}^{m-1} \sum_{i=0}^{n} [i > \lfloor \frac{jc+c-b-1}{a} \rfloor] \\
&= \sum_{j=0}^{m-1} n - \lfloor \frac{jc+c-b-1}{a} \rfloor \\
&= nm - \sum_{j=0}^{m-1} \lfloor \frac{jc+c-b-1}{a} \rfloor \\
&= nm - f(c, c-b-1, a, m-1)
\end{aligned}
$$

发现每次变化后：
f(a, *,c , *)就变成了f(c % a, *, a, *)

这和gcd有着惊人的相似之处，
那么复杂度也是O(log n)的。

例题 JZOJ3492 数数

- 记count（x）表示x二进制下1的个数。
- 求$\sum_{i=1}^{n} count\ (a+\mathrm{bi})$
- a<=10^12, b<=10^4, n<=10^12

一句话题解

- $Ans = \sum_{k=0}^{63} \sum_{i=1}^{n} \left\lfloor \frac{a+bi}{2^k} \right\rfloor - \left\lfloor \frac{a+bi}{2^{k+1}} \right\rfloor * 2$

51nod 1187 寻找分数

- 给出a/b和c/d
- 求p/q，满足a/b < p/q < c/d，且q最小。
- 1<=a,b,c,d<=10^9

solution

- 设$f(a,b,c,d)$表示答案
- 当$a >= b$时，$f(a,b,c,d) = f\left(a\%b, b, c - d * \left\lfloor\frac{a}{b}\right\rfloor, d\right) + \left\lfloor\frac{a}{b}\right\rfloor$
- 当$a < b$且$c >= d$时，最优答案显然是$p = 1, q = 1$
- 那么只剩a<b且c<d的情况了，此时$f(a,b,c,d) = \frac{1}{f(b,a,d,c)}$
- 复杂度显然。

总结

- 类欧几里得算法就是这样的每次取模、交换的函数化归。

群论及burnside引理

- 参考资料（很多定义copy自这里）：
- 《群论与Burnside's 引理在信息学竞赛中的应用》 胡渊鸣
- 由于群论的许多证明比较冗长，所以在这里不详细展开，只做大概的讲述。
- 有兴趣的同学可自行查阅。

群的简介

- 群是一种代数结构，由一个集合+一个二元运算组成。
- 设集合为S，运算为*，当S和*满足以下四个性质时，可称之为群。
- 群公理：
  - 封闭性：$\forall a, b \in S, \mathrm{a} * \mathrm{b} \in S$
  - 结合律：$\forall a, b, \mathrm{c} \in S, (a * b) * c = a * (b * c)$
  - 单位元：$\exists e \in S, \forall a \in S, a * e = e * a = a$
  - 逆元：$\forall a \in S, \exists b \in S, a * b = b * a = e$

群的例子

- $[0, p)$的mod p加法。
- $[1, p)$的mod p乘法。
- 整数加法。

群的性质

- 单位元唯一。
- 逆元唯一。
- 消去定理。

置换

- 考虑集合M，函数f:M->M，如果f是双射（一一对应），就称f是M上的一个置换。
- 对于两个置换f,g，对于m∈M，定义*运算为$(f * g)(m) = f\big(g(m)\big)$，即f*g为f,g的复合函数。

置换群

- 顾名思义，置换群就是一个群，群的集合的元素是一些置换，运算就是之前说的*（复合）。
- 例子：
  - 项链的旋转
  - 矩阵的行列交换。

Burnside引理

- 为了方便，置换f作用于集合M，我们称M中元素m为染色。
- 你可以想象有一串项链，你给每个珠子染色，设旋转为置换，那么对于一个染色，通过旋转可以得到其它的染色（可能不变）。
- 不动点：
  - 如果一个染色m，在置换f作用下不变，则称染色m是在置换f下的不动点。
- 本质相同的染色：
  - 如果染色m1能够通过置换得到染色m2，则称m1和m2是本质相同的染色。

- Burnside引理：

$$\text{本质不同的染色数} = \left(\sum_{\text{置换}f} \text{置换}f\text{下的不动点数}\right)/\text{置换数}$$

例题

- 给一串有n个珠子的项链，给每个珠子染上k种颜色的一种，若两种染色方案能通过旋转而相同，则视为相同。
- 求本质不同染色数。
- $n \leq 10^6$

- 直接套burnside引理。
- 枚举旋转了$i \in [0,n)$步，如果是不动点，那么有$\forall x \in [0,n), c[x] = c[(x+i) \ mod \ n]$。
- 假设从0开始跳，每次跳x步，跳了y次第一次回到原点，显然有
- y=lcm(x,n)/x=x*n/gcd(x,n)/x=n/gcd(x,n)
- 一个环的长度是n/gcd(x,n)，一共有gcd(x,n)个环，每个环的颜色一样。
- 因此不动点数是$k^{\gcd(i,n)}$。
- 复杂度是$O(n \log n)$

- 刚才那题，$n \leq 10^{18}$

- 实际上并不用真的枚举旋转了i步。
- 只用枚举d | n，那么符合gcd(i,n)=d的i的个数是phi(n/d)。
- 复杂度：$O(n^{\frac{1}{4}} + \mathrm{n的约数个数} * \log n)$

- 还是刚才那题，同样的范围。
- 不过这次可以翻折。

结论

- 任意无限翻折+无限旋转都可以通过一次翻折（或者不翻）+一次旋转得到。
- 证明：
  1.旋转部分显然。
- 2.翻折：
- 一开始在x上，第一次翻折到2k-x上，第二次翻折到2k'-(2k-x)=2(k-k')+x上
- 由此看出两次翻折后就相当于普通的旋转，所以只用考虑翻了或者没翻。

- 并且这里的翻折只能是沿一条固定的轴来翻折。
- 假设分别以k1、k2为中心翻折，那么沿k1翻折可以看作沿k2翻折后再平移。
- 所以就直接假设以0为中心来翻折。

- 若旋转i步，则有c[x]=c[x+gcd(I,n)]
- 设d=gcd(I,n)
- 那么有c[x]=c[x+d],c[x]=c[-x+d]
- 第x%d个环要和第(-x+d)%d个环合并起来。
- 原来的环的个数是d，那么新环的个数是(d+1)/2。

[HNOI2009]图的同构记数

- 如果两个图交换顶点标号若干次相同，则视为同构。
- 求n个点不同构的图的数量。
- 加强版：$N<=5000$

- 考虑置换就是所有的排列。
- 如果枚举所有的排列，大概就是要把i和p[i]视作同一个点。
- i有的出边p[i]都要有。
- 假设最后缩点完后还有x个点，那么再让每条边选不选就是$2^{x(x-1)/2}$的方案。
- 问题n很大。

- 发现并不需要知道具体的排列。
- 只需要的环个数，其实因为分配标号还需要知道每个环的大小。
- 设n=a1+a2+...+ax(ai<=ai+1)
- 那么分配标号方案是$\frac{n!}{\prod ai!} * \prod(ai-1)! = \frac{n!}{\prod ai}$
- 然后做n次无限背包就好了。

莫比乌斯反演

- 前置函数：
- 设$n = \prod_{i=1}^{p0} pi^{qi}$
- $\mu(n) = (\prod[qi \le 1]) * (-1)^{p0}$
- $\phi(n) = n * \prod \frac{pi-1}{pi}$
- $\sum_{d|n} \mu(d) = [n = 1]$，证明留作思考。
- $\sum_{d|n} \phi(d) = n$，证明留作思考。

- 若有$f(x)=\sum_{d|x} g(d)$，则有$g(x)=\sum_{d|x} f(d)*\mu(x/d)$
- 证明：将$f=\cdots$代入$g=\cdots$，可得：
- $g(x)=\sum_{d|x} \mu\left(\frac{x}{d}\right)*\sum_{d'|d} g(d')$
- $g(x)=\sum_{d|x} g(d)*\sum_{d'|\frac{x}{d}} \mu(d')=\sum_{d|x} g(d)*\left[\frac{x}{d}=1\right]$
- 另一形式：
- 若有$f(x)=\sum_{x|d} g(d)$，则有$g(x)=\sum_{x|d} f(d)*\mu(d/x)$
- 证明类似不展开。

例题

- 求$\sum_{i=1}^{n}\sum_{j=1}^{n}\gcd(i,j)$
- $N <= 10^6$

- $=\sum_{d=1}^{n}\sum_{i=1}^{n}\sum_{j=1}^{n}[\gcd(i,j)=d]*d$
- $=\sum_{d=1}^{n}\sum_{i=1}^{n/d}\sum_{j=1}^{n/d}[\gcd(i,j)=1]*d$
- $=\sum_{d=1}^{n}d*\sum_{i=1}^{n/d}\sum_{j=1}^{n/d}\sum_{d'|\gcd(i,j)}\mu(d')$
- $=\sum_{d=1}^{n}d*\sum_{i=1}^{n/d}\sum_{j=1}^{n/d}\sum_{d'|\mathrm{i}\ \&\&\ d'|j}\mu(d')$
- $=\sum_{d=1}^{n}d*\sum_{d'=1}^{n/d}\mu(d')*\sum_{i=1}^{\frac{n}{d*d'}}\sum_{j=1}^{\frac{n}{d*d'}}1$
- $=\sum_{d=1}^{n}d*\sum_{d'=1}^{\frac{n}{d}}\mu(d')*\left(\frac{n}{d*d'}\right)^{2}$

- 设$T = d * d'$
- 则 $= \sum_{T=1}^{n} \left(\frac{n}{T}\right)^2 * \sum_{d|T} \mu(T/d) * d$
- 后面这个函数可以调和级数$O(n \log n)$的复杂度预处理。
- 发现我们只是利用$\sum_{d|n} \mu(d) = [n = 1]$这一性质去做简单变换。
- 事实上同样可以用$\sum_{d|n} \phi(d) = n$来变换。

- $= \sum_{d=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} [\gcd(i,j) = d] * d$
- $= \sum_{d=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} d * \sum_{d'|i\ \&\&\ d'|j} \phi(d')$
- ……
- $= \sum_{T=1}^{n} \left(\frac{n}{T}\right)^2 * \phi(T)$
- 这个式子显然更加简洁，且可以做到O(n)。
- 比较两个式子，你会发现有：

$$\phi(n) = \sum_{d|n} \mu\left(\frac{n}{d}\right) * d$$

- 怎么证明呢？

狄利克雷卷积

- 对于两个积性函数f、g，定义他们的狄利克雷卷积：
- $h(n) = (f * g)(n) = \sum_{d|n} f(d) * g(n/d)$
- 显然h也是一个积性函数，这是一个非常有用的结论。

- 回到之前，我们想把f(n) = $\sum_{d|n}\mu\left(\frac{n}{d}\right)*d$化简。
- 因为$f=\mu*Id$,所以f是一个积性函数，因此只考虑f = p^q时的答案，最后再组合起来。
- 显然有$f(p^q)=p^q-p^{q-1}=p^{q-1}*(p-1)=\phi(p^q)$
- 那么$f=\phi$就出来了。

- 我们再来一题感受一下。

51nod 1190 最小公倍数之和V2

- 给出2个数a, b，求LCM(a,b) + LCM(a+1,b) + .. + LCM(b,b)。
- 1<=a<=b<=10^9
  T组数据,1<=T<=50000
- 且数据为随机，没有构造的卡人数据。

$Ans = \sum_{i=a}^{b} lcm(i)$

$= b * \sum_{d|b} \sum_{i=\lfloor \frac{b}{d} \rfloor}^{\lceil \frac{a}{d} \rceil} i * (gcd(i, \frac{b}{d}) = 1)$

$= b * \sum_{d|b} \sum_{i=\lfloor \frac{b}{d} \rfloor}^{\lceil \frac{a}{d} \rceil} i * \sum_{d'|gcd(i, \frac{b}{d})} \mu(d')$

$= b * \sum_{d|b} \sum_{d'|\frac{b}{d}} \mu(d') * d' * \sum_{i=\lfloor \frac{b}{d} \rfloor}^{\lceil \frac{a}{d} \rceil} i * (d'|gcd(i, \frac{b}{d}))$

$= b * \sum_{d|b} \sum_{d'|\frac{b}{d}} \mu(d') * d' * \sum_{i=\lfloor \frac{b}{d*d'} \rfloor}^{\lceil \frac{a}{d*d'} \rceil} i$

$= b * \sum_{d|b} \sum_{d'|\frac{b}{d}} \mu(d') * d' * (\lfloor \frac{b}{d*d'} \rfloor - \lceil \frac{a}{d*d'} \rceil + 1) * (\lfloor \frac{b}{d*d'} \rfloor + \lceil \frac{a}{d*d'} \rceil)/2$

设T = d * d'

$= b * \sum_{T|b} (\lfloor \frac{b}{T} \rfloor - \lceil \frac{a}{T} \rceil + 1) * (\lfloor \frac{b}{T} \rfloor + \lceil \frac{a}{T} \rceil)/2 * \sum_{d|T} \mu(d) * d$

我们观察一下$\sum_{d|T} \mu(d) * d$

狄利克雷卷积做了这么多，轻松可得:

若$T = \prod p_i^{q_i}$,那么

$\sum_{d|T} \mu(d) * d = \prod 1 - p_i$

在递归枚举约数的时候维护一下即可。

- 那么一开始讲的函数反演有什么用呢？
- 一般我们不用函数反演做数论式子化简，那样太复杂了，而是用于直接算不好算，间接算好算的地方，其实就是一个容斥，μ是一个容斥系数。

51nod 1355 斐波那契的最小公倍数

- 斐波那契数列定义如下：
- F(0) = 0 F(1) = 1
  F(n) = F(n-1) + F(n-2)
- 给出n个正整数a1, a2,...... an，求对应的斐波那契数的最小公倍数，由于数字很大，输出Mod 1000000007的结果即可。
- $N <= 50000$，$ai <= 10^6$

- $\gcd(f[i], f[j]) = f[\gcd(i, j)]$
- 归纳可得，这里不展开。

- 多个数的lcm也不好求，可以通过min-max容斥转换成gcd。
- $Lcm(A) = \prod_{B \in A} \gcd(B) * (-1)^{|B|+1}$

- 设f[n]表示gcd恰好为n的指数和，直接算显然不好算。
- 设g[n]表示gcd为n的倍数的指数和，$g(n) = \sum_{n|d} f(d)$
- 若有y个a[i]是n的倍数，则$g[n] = 2^y - 1$
- 反演回来得：$f(n) = \sum_{n|d} g(d) * \mu(d/n)$

- 求$\sum_{i=1}^{n}\sum_{j=1}^{n}\gcd(i,j)$
- 询问次数 $\leq 10^4, n \leq 10^6$
- $= \sum_{T=1}^{n}\left(\frac{n}{T}\right)^2 * \phi(T)$

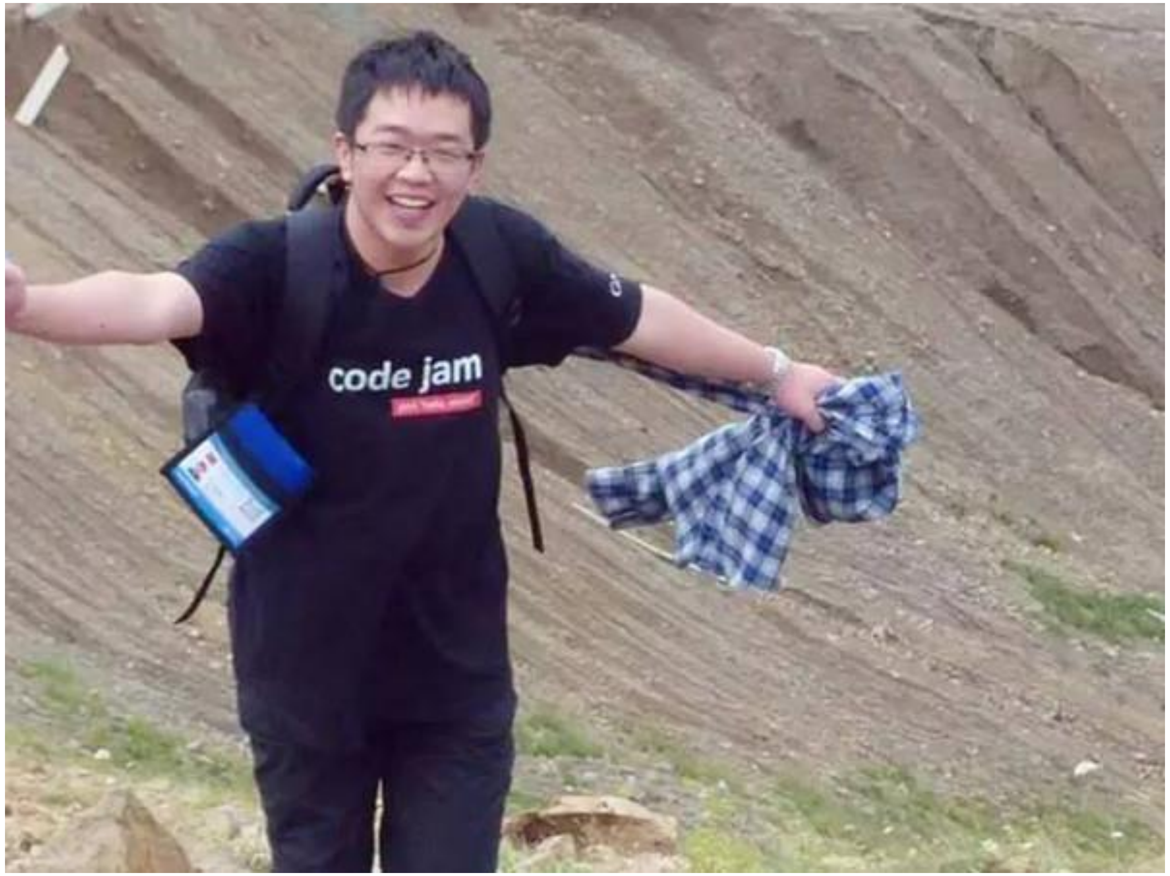
- $= \sum_{T=1}^{n} \left(\frac{n}{T}\right)^2 * \phi(T)$
- 预处理ϕ的前缀和，然后n/T只有$O(\sqrt{n})$种取值，所以一次查询复杂度可以做到$O(\sqrt{n})$。
- 写法如下：

```
for(int i = 1, j; i <= n; i = j + 1) {
    j = n / (n / i);
    ans += (long long) (s[j] - s[i - 1]) * ((n / i) * (n / i));
}
```

- 求$\sum_{i=1}^{n}\sum_{j=1}^{n}\gcd(i,j)$
- $n \le 10^{11}$

- 需求是生产的主要推动力。

杜教(xudyh)



杜教筛

- $\sum_{i=1}^{n} \mu(i)$
- $\sum_{i=1}^{n} \phi(i)$
- $\sum_{i=1}^{n} \phi(i) * i^k$
- n$\leq 10^{11}$

- 设$s(n)=\sum_{i=1}^{n}\mu(i)$
- $\sum_{i=1}^{n}\sum_{j|i}\mu(j)=1$
- $\sum_{i=1}^{n}\sum_{j=1}^{n/i}\mu(j)=1$
- $\sum_{i=1}^{n}s(n/i)=1$
- $s(n)=1-\sum_{i=2}^{n}s\left(\frac{n}{i}\right)$
- 直接递归+分块+hash表，复杂度$O(n^{\frac{3}{4}})$。
- 预处理前$O(n^{\frac{2}{3}})$的答案，总复杂度$O(n^{\frac{2}{3}})$。

- $\sum_{i=1}^{n}\sum_{j|i}\phi(j) = n*(n+1)/2$
- $s(n) = n*(n+1)/2 - \sum_{i=2}^{n} s\left(\frac{n}{i}\right)$

- 注意在求出$s(n)$的时候同时求出了所有的$s(n/i)$，因此之前那题可以$n \leq 10^{11}$。

- $\sum_{i=1}^{n} \mathrm{i}^k \sum_{j|i} \phi(j) = \sum_{i=1}^{n} i^{k+1}$
- $\sum_{i=1}^{n} i^k * \sum_{j=1}^{n/i} j^k * \phi(j) = \sum_{i=1}^{n} i^{k+1}$
- $s(n) = \sum_{i=1}^{n} i^{k+1} - \sum_{i=2}^{n} i^k * s(n/i)$
- 所以可能还需要套个自然数幂和。

Min_25筛

min_25筛是洲阁筛的简化版，虽然我并不会洲阁筛。

min_25筛可以筛一些特殊积性函数的前缀和，有些不是积性函数也可以筛，比如说最大真因子。

同杜教筛一样，同时筛出了所有$\lfloor \frac{n}{i} \rfloor$的前缀和。

至于min_25能筛的积性函数有哪些要求，在博客后面会讨论

所有时间复杂度证明见朱大佬2018国家预备队论文。

筛的本质：

$1-n$内的所有数的最小质因子都会不大于$\sqrt{n}$，同欧拉筛法，我们就是在枚举到一个数的最小质因子的时去统计到它。

Step 1:

洲阁筛的第一步就是求出n以内所有质数p的$\sum f(p)$

$f(p)$一般是和p有关的多项式，如果$f(p)$能够拆成若干个完全积性函数的式子，那么这个f就是可以筛的。

这个可能有点难理解，可以看下面的式子来得到。

以筛质数个数为例，显然有$f(p)=1$。

预处理$\sqrt{n}$以内的质数，第j大的质数设为p_j

$g(n,j)$表示$2-n$的数中，要么是质数，要么最小质因子大于p_j,把合法的数当作质数算的答案。

解释一下何为当作质数：
对于一个质数p，$f(p)$一般是和p有关的式子，对于一个x，我们不考虑它是不是质数，而是直接像质数那样算出和x有关的值。

$$g(n,j)=\begin{cases} g(n,j-1) & p_j^2>n \\ g(n,j-1)-f(p_j)[g(\frac{n}{p_j},j-1)-\sum_{i=1}^{j-1}f(p_i)] & p_j^2<=n \end{cases}$$

第二条式子中，因为g的定义含有质数，而我们的目的是使p_j成为最小质因子，所以要减掉它们。

由于用到的n一定是$\lfloor \frac{n}{i} \rfloor$，所以只用记录$\sqrt{n}$个值。
初值:
在求质数个数中,$g(n,0)=n-1$，其它题中，可能是和n有关的奇奇怪怪的东西。

- 题目：LOJ#6235. 区间素数个数

```
for(ll i = 1, j; i <= n; i = j + 1) {
    j = n / (n / i); w[++ m] = n / i;
    if(w[m] <= sqr) id1[w[m]] = m; else id2[n / w[m]] = m;
    g[m] = w[m] - 1;
}
fo(j, 1, p[0]) for(int i = 1; i <= m && p[j] * p[j] <= w[i]; i ++) {
    int k = (w[i] / p[j] <= sqr) ? id1[w[i] / p[j]] : id2[n / (w[i] / p[j])];
    g[i] += - g[k] + j - 1;
}
```

Step 2:

递归版:

知道了g就可以求答案了。

设$s(n,j)$表示2-n中的所有数，最小质因子大于等于p_j的所有数x的$\sum f(x)$

$$s(n,j)=g(n)-\sum_{i=1}^{j-1} f(p_i)+\sum_{k=j}^{p0}\sum_{e|e>=1\text{和}p_k^{e+1}<=n} f(p_k^e)*s(n/p_k^e,k+1)+f(p_k^{e+1})$$

不用hash，直接强行递归复杂度为$O(\frac{n^{\frac{3}{4}}}{log\ n})$

递推版：

更改s的定义，类似于之前的g
$s(n,j)$表示2-n的数中，要么是质数，要么最小质因子大于等于p_j的数x的$\sum f(x)$
$s(n,j)=s(n,j+1)+\sum_{e|e>=1和p_j^{e+1}<=n} f(p_j^e)*s(n/p_j^e)+f(p_j^{e+1})$
初值$s(n,p0+1)=g(n)$,写法类似于g，就是先枚举j，然后再转移。

对比：

根据LL实测，递归版在求一个的时候会比递推版快，因为有些状态走不到。

但是如果对于所有的$\lfloor \frac{n}{i} \rfloor$都要求前缀和，就只能用递推版了。

LOJ #6053. 简单的函数

某一天，你发现了一个神奇的函数$f(x)$，它满足很多神奇的性质：

1. $f(1)=1$。
2. $f(p^c)=p\oplus c$（p 为质数，$\oplus$ 表示异或）。
3. $f(ab)=f(a)f(b)$（a 与 b 互质）。

你看到这个函数之后十分高兴，于是就想要求出 $\sum\limits_{i=1}^{n}f(i)$。

由于这个数比较大，你只需要输出 $\sum\limits_{i=1}^{n}f(i) \bmod (10^9+7)$。

$n \le 10^{10}$

- 按照套路，我们先筛质数的答案。
- 质数的指数都是1，且除了2的质数都是奇质数，那么异或1就相当于-1。
- 所以要筛质数个数和质数和。
- 之后的部分就直接套式子递归就好了。

JZOJ **5594.** 最大真因数

- 一个合数的真因数是指这个数不包括其本身的所有因数，例如6 的正因数有
  1; 2; 3; 6，其中真因数有1; 2; 3。一个合数的最大真因数则是这个数的所有真因数
  中最大
  的一个，例如6 的最大真因数为3。
  给定正整数l 和r，请你求出l 和r 之间（包括l 和r）所有合数的最大真因数之和。

- 1<= l $\leq r <= 5 * 10^9$

- 这东西看上去不是积性函数筛不了，但是注意到在min_25筛的第一个dp里。
- 会有枚举到每个数的最小质因子的操作，那么剩下的部分就是最大真因数了。
- 因此用min_25筛一个质数和，顺便求出答案就好了。