原 洲阁筛学习总结

洲阁筛学习总结

By Semiwaker

问题描述

给出一个积性函数$F(x)$，满足积性函数的基本性质：
F(1)=1
如果 a 和 b 互质，则$F(ab) = F(a)F(b)$
设$x = \prod p_i^{k_i}$，则有

$$F(x) = \prod F(p_i^{k_i})$$

特别的
p为质数，F(p) 是一个低阶多项式，$F(p^c)$可以快速算出。
现在要求

$$\sum_{i=1}^{n} F(i)$$

n的范围大概在10^{11}左右。

小范围问题的解决方法

接下来的部分是线性筛，可以跳过
假如n在10^7范围内，考虑怎么线性地求出每一个 F(x)。
分析线性筛的过程，大概分为几个要点：
1、质数的时候需要直接计算。
2、对于某个非质数 x，会分解为两部分：最小的质因数 p 和剩余的部分 y 。有$py = x$。

质数的部分就不必多说了，只能直接算。
非质数的部分分为3类。
1、p和y互质，那么显然$F(x) = F(p)F(y)$。
2、y是p的幂$y = p^c$，也就是 y 中只有 p 一个质因子，此时需要知道次数 c 然后直接计算。（特殊情况下可以从 F(y) 推过来）
3、y是p的倍数$p|y$，但是y中还有其他质因子，设$y = p^c z$，则$F(x) = F(z)F(p^{c+1})$。如果我们知道p^c，那么我们就可以这么计算：
$F(x) = F(\frac{y}{p^c})F(p^c \times p)$
我们分析需要知道什么：第二类需要知道每个数最小的质因数的次数 Cnt[x] ，第三类需要知道每个数中最小的质因子构成的部分 Fir[x]。
幸运的是，这两样都可以随着线性筛的过程得出。

具体来说，线性筛的过程如下：
1、由之前筛的结果判断 i 是否质数，质数的情况下，直接算出F(i)，Fir[i]=i，Cnt[i]=1。非质数的数据已经在之前算出了。
2、枚举已经求出的质数 P[j] 。
将$i \times P[j]$设为非质数。
3、判断 P[j] 是否是 i 的因数。
4、如果不是，$F(i \times P[j]) = F(i)F(P[j])$，
$Fir[i \times P[j]] = P[j]$，$Cnt[i \times P[j]] = 1$
5、如果是，分判断 i 是否等于 Fir[i] ，当 i=Fir[i] 时，i中只有P[j]一个质因数，此时$F(i \times P[j]) = F(P[j]^{Cnt[i]+1})$，按照定义直接 。
否则$F(i \times P[j]) = F(\frac{i}{Fir[i]})F(Fir[i] \times P[j])$，显然两项都小于i，已经计算过了。
然后无论如何$Fir[i \times P[j]] = Fir[i] \times P[j]$，$Cnt[i \times P[j]] = Cnt[i] + 1$。

前置技能

引理1

如果$x > a, b$
$\lfloor \frac{\lfloor \frac{x}{a} \rfloor}{b} \rfloor = \lfloor \frac{x}{ab} \rfloor$
证明：
设$y = \lfloor \frac{x}{a} \rfloor$。
设$k = \lfloor \frac{x}{ab} \rfloor$，则有$x = kab + c$，显然$c < ab$。
那么$y = kb + \lfloor \frac{c}{a} \rfloor$，因为$c < ab$，所以$\lfloor \frac{c}{a} \rfloor < b$，所以$\lfloor \frac{y}{b} \rfloor = k$。

引理2

$\lfloor \frac{n}{x} \rfloor$最多有$2\sqrt{n}$种取值。
证明
当$x \le \sqrt{n}$时，只有$\sqrt{n}$种取值。
当$x > \sqrt{n}$时，$\lfloor \frac{n}{x} \rfloor \le \sqrt{n}$，也只有$\sqrt{n}$种取值。

这个上限是比较精准的，大多数情况下是$2\sqrt{n} - 1$或者刚好$2\sqrt{n}$个。

定理：从n开始整除任意数任意次得到的不同结果只有$2\sqrt{n}$种。

根据引理2，n整除1次的结果只有$2\sqrt{n}$种。
根据引理1，n整除1次的结果包含整除2次的结果，以此类推，整除任意次的结果都属于n整除 1 次的结果。

洲阁筛核心思想

因为$n = 10^{11}$，所以我们需要在低于线性时间的复杂度内计算。
考虑线性筛的瓶颈在哪里？为什么线性筛必须是O(n)的？
因为**线性筛一定要把1~n的每一个质数都单独处理。**

n范围内的质数个数大概是$O(\frac{n}{lnn})$范围的，由于ln n只有几十，所以枚举所有质数需要的时间至少O(n)级别。

现在的关键点是：怎样才能不用枚举所有的质数？
核心思想：把质数分类
质数可以简单地分为两类：第一类是小于等于$\sqrt{n}$的质数，另外一类是大于$\sqrt{n}$的。

为什么要这样分类呢？
因为n以内的数，最多只有一个大于$\sqrt{n}$的质因数。
那么，如果我们把小于等于$\sqrt{n}$的质数筛掉，剩下的就全部都是质数。
所以，我们可以**通过分类来减少对质数的枚举。**

算法

总述

我们把 1 ~ n的每一个数分类，有大于$\sqrt{n}$的质数的，和没有的。
那么可以写成这样：

$$\sum_{i=1}^{n} F(i) = \sum_{i=1}^{\sqrt{n}} F(i) \Big(\sum_{p > \sqrt{n}}^{\lfloor \frac{n}{i} \rfloor} F(p)[p\text{为质数}] \Big) + \sum_{i=1}^{n} F(i)[i\text{无大于}\sqrt{n}\text{的质因子}]$$

也就是说，有大于$\sqrt{n}$的质因子的，我们枚举$\le \sqrt{n}$的质因子的部分，再枚举$> \sqrt{n}$的质因子。
没有的就直接枚举。

那么我们的算法现在可以分为三个部分。
第一个部分是枚举$\sqrt{n}$以内的每一个i，求F(i)，这部分我们可以线性筛。
第二个部分是对于每一个$\lfloor \frac{n}{i} \rfloor$，求

$$\sum_{p > \sqrt{n}}^{\lfloor \frac{n}{i} \rfloor} F(p)[p\text{为质数}]$$

第三个部分是求

$$\sum_{i=1}^{n} F(i)[i\text{无大于}\sqrt{n}\text{的质因子}]$$

第二部分

求

$$\sum_{p>\sqrt{n}}^{\lfloor \frac{n}{i} \rfloor} F(p)[p\text{为质数}]$$

我们只需要筛掉$\sqrt{n}$内的所有质数，那么剩下的就是我们要求的。
考虑怎样简化问题，因为 F(p) 是一个低阶的多项式，所以我们只需要对于每一项单独求和，第k项求出$\sum p^k$，然后合并即可。
即，现在的问题是，对于某个范围[1,m]内的所有大于$\sqrt{n}$的质数，求$\sum p^k$。

筛的过程可用递推实现。
设小于等于$\sqrt{n}$的质数有Pn个，从小到大排列为$P_{1..Pn}$。
设在 [1,j] 中，与$P_{1..i}$互质的所有数的k次方和为：g[i][j]。

显然边界为g[0][j]=j。

我们要求的为g[pn][j]。

考虑怎样从前 i-1 个质数推到前 i 个质数。
我们只需要算出与前i-1个质数互质的数，再减去与前i-1个质数互质，但是不与第i个质数互质的数即可。
设某个与前 i-1 个质数互质的数为x，那么$P[i]x$必然与第i个质数不互质。
$\sum(P[i]x)^k = P[i]^k \sum x$
又$P[i]x \le j$，那么有$x \le \lfloor \frac{j}{P[i]} \rfloor$
所以得到：

$$g[i][j] = g[i-1][j] - P_i^k g[i-1][\lfloor \frac{j}{P[i]} \rfloor]$$

根据之前提到的定理，j的取值有2$\sqrt{n}$种，i的取值有$O(\frac{\sqrt{n}}{ln\sqrt{n}})$种，乘起来还是O(n)级别。需要优化。

注意到，如果$P_i > j$，那么除了1以外的所有数都被筛去了。
显然只要j不为0，那么g[i][j]=1。

怎么利用这个性质呢？
如果$P_{i-1} > \lfloor \frac{j}{P[i]} \rfloor$，那么$g[i-1][\lfloor \frac{j}{P[i]} \rfloor] = 1.$
放宽一点$P_i > \lfloor \frac{j}{P[i]} \rfloor$，此时的判断条件转化为$P_i^2 > j$。
换句话说，
当$P_i^2 > j$时，有$g[i][j] = g[i-1][j] - P_i^k$
那么设$P_{L[j]}^2 > j \ge P_{L[j]-1}^2$，即L[j]是第一个满足条件的i。
对于$i < L[j]$，我们需要老老实实地去算。对于$i \ge L[j]$，我们只需要一个前缀和就可以了。

还有一点要注意的：
如果$P_i > j$，此时$\lfloor \frac{j}{P[i]} \rfloor = 0$，那么有

$$g[i][j] = g[i-1][j]$$

设小于等于j的质数有CntP[j]个，那么对于$i \ge CntP[j]$，是不需要去计算的。

综合起来，根据i的大小，可以分为3段。
第一段0到L[j]-1，直接算。
第二段L[j]到CntP[j]，预处理p^k的前缀和。
第三段CntP[j]到Pn，不需要算。

观察到递推式只和i-1有关，是可以**滚动数组**的，所以具体的实现方法如下：
当$i < L[j]$时，直接$g[j] = g[j] - P_i^k g[\lfloor \frac{j}{P[i]} \rfloor]$
当$i \ge L[j]$时，停止对$g[j]$的递推。
如果在某个i调用到了g[j]，那么我们再算上缺少的L[j]~min(i,CntP[j])的部分的前缀和即可。

设前缀和为 $SumP[i] = \sum_{t=0}^{i} p_t^k$，那么我们可以这样表示 g[i][j]:
$g[j] + max(0, SumP[min(i, CntP[j])] - SumP[L[j]])$

时间复杂度

由于每一个j对应的需要枚举的i是$\frac{\sqrt{j}}{lnj}$范围的，所以我们可以列式：

$$\sum_{i=1}^{\sqrt{n}} \frac{\sqrt{\frac{n}{i}}}{ln\frac{n}{i}} \approx O(\frac{n^{\frac{3}{4}}}{lnn})$$

细节

1、CntP[j]只需要保存$j \le \sqrt{n}$的。因为当$j > \sqrt{n}$时，所有i都满足$Pi \le j$。此时只要当CntP[j]=Pn即可。
2、关于j如何储存的问题。*肯定不是暴力哈希。*
由于j的范围是[1,n]，所有直接开数组是不可行的。
我们可以分类存放。
当$j \le \sqrt{n}$，我们直接开数组，存放在g0[j]。
当$j > \sqrt{n}$，设$\lfloor \frac{n}{x} \rfloor = j$，存放在g1[x]。

3、由于使用了滚动数组，所以 j 要从大到小枚举。而 L[j] 是随着 j 减小而减小的，而我们枚举的条件是$i < L[j]$，所以在$i \ge L[j]$直接停止枚举即可

第三部分

求

$$\sum_{i=1}^{n} F(i)[i\text{无大于}\sqrt{n}\text{的质因子}]$$

同样我们用递推去求。
这次为了方便，我们从大的质数往小的质数推。

没有大于$\sqrt{n}$的质因子，那么就意味着只有$\sqrt{n}$以内的质因子。
所以我们设：
x为在[1,j]的范围内，只有$P_{i..Pn}$这些质因子的数。（即i以后的质因子。）
$f[i][j] = \sum F(x)$

边界为$f[Pn+1][j] = 1$。

要求的为f[0][j]。

那么有：

$$f[i][j] = f[i+1][j] + \sum F(P_i^c) f[i+1][\lfloor \frac{j}{P_i^c} \rfloor]$$

简单来说，直接枚举当前质因子的幂，然后考虑乘起来不会超过j的，只包含$P_{i+1..Pn}$的数的和为多少。因为是积性函数，所以可直接乘起来。

同样，直接算这个递推式也是O(n)级别的，需要优化。

同样考虑$P_i^2 > j$，则$\lfloor \frac{j}{P_i} \rfloor < P_i$，此时$f[i+1][\lfloor \frac{j}{P_i} \rfloor] = 1$（因为只包含大于Pi的质数）。
而当$c \ge 2$，$\lfloor \frac{j}{P_i^c} \rfloor = 0$。
所以有

$$f[i][j] = f[i+1][j] + F(P_i)$$

同样当$P_i > j$时，$f[i][j] = f[i+1][j]$。

所以，同样根据i的大小分为3段。**(注意是从大到小的。)**
第一段 Pn 到 CntP[i]+1，这一段不用计算。
第二段 CntP[i] 到 L[i]，这一段我们要计算F(p)的前缀和。
第三段 L[i]-1 到 0，这一段老老实实递推。

同样地，可以用滚动数组。

设前缀和为$SumP[i] = \sum_{t=1}^{i} F(t)$
那么，在某个 i 时，f[j]的实际值为 f[j]+Max(0,SumP[L[i]-1]-SumP[Max(i,CntP[i])-1])

时间复杂度

同样为

$$O(\frac{n^{\frac{3}{4}}}{lnn})$$

细节

同样，当$j \le \sqrt{n}$时，记为 f0[j]，否则设$j=\lfloor \frac{n}{x} \rfloor$，记为f1[j]

计算答案

现在回到原本的式子：

$$\sum_{i=1}^{n}F(i)=\sum_{i=1}^{\sqrt{n}}F(i)\left(\sum_{p>\sqrt{n}}^{\lfloor\frac{n}{i}\rfloor}F(p)[p\text{为质数}]\right)+\sum_{i=1}^{n}F(i)[i\text{无大于}\sqrt{n}\text{的质因子}]$$

实际需要计算的是：
$Ans=\sum_{i=1}^{\sqrt{n}}F(i)g(\lfloor\frac{n}{i}\rfloor)+f(n)$
即
$Ans=\sum_{i=1}^{\sqrt{n}}F(i)g1(i)+f1(1)$

注意到 g0 和 f0 都没有用了。
那么我们在递推完之后，只需要把 g1 和 f1 没有加上去的前缀和补充完整即可。

例题 SPOJ DIVCNT3

$F(x)=\delta_0(x^3)$
那么有：
$F(p)=4$
$F(p^c)=3c+1$
因为F(p)是常数，所以只用算 0 次方和。

代码

```
#include <cstdio>
#include <cstring>
#include <cmath>
#include <cstdlib>
#include <algorithm>

using namespace std;

typedef long long LL;

const int MAXN=1000000,SQRTMAXN=1000100;
#define MMax(x,y) (((x)>(y))?(x):(y))
#define MMin(x,y) (((x)<(y))?(x):(y))

int Prime[MAXN/10],pn;
int D3[MAXN+10],Fir[MAXN+10];
bool vis[MAXN+10];

int CntP[SQRTMAXN],SumP1[SQRTMAXN];
int L0[SQRTMAXN],L1[SQRTMAXN];
LL g0[SQRTMAXN],g1[SQRTMAXN],f0[SQRTMAXN],f1[SQRTMAXN];

void GetPrime();
void GetG(int SqrtN,int PN,int n);
void GetF(int SqrtN,int PN,int n);
LL GetSum(LL n);

int main()
{
    freopen("divcnt3.in","r",stdin);
```

```
    freopen("divcnt3.out","w",stdout);

    GetPrime();
    int T;
    scanf("%d",&T);
    while (T--)
    {
        LL n;
        scanf("%lld",&n);
        printf("%lld\n",GetSum(n));
    }

    return 0;
}

void GetPrime()
{
    D3[1]=1;
    for (int i=2;i<=MAXN;++i)
    {
        if (!vis[i])
            Prime[pn++]=i,D3[i]=4,Fir[i]=i;
        for (int j=0;j<pn;++j)
        {
            if (i*Prime[j]>MAXN) break;
            vis[i*Prime[j]]=1;
            if (i%Prime[j]==0)
            {
                if (i/Fir[i]==1) D3[i*Prime[j]]=D3[i]+3;
                else D3[i*Prime[j]]=D3[i/Fir[i]]*D3[Prime[j]*Fir[i]];
                Fir[i*Prime[j]]=Fir[i]*Prime[j];break;
            }
                else D3[i*Prime[j]]=D3[i]*4,Fir[i*Prime[j]]=Prime[j];
        }
    }
}
void GetG(int SqrtN,int PN,LL n)
{
    for (int i=1;i<SqrtN;++i) g0[i]=i,g1[i]=n/i;
    for (int i=0;i<PN;++i)
    {
        for (int j=1;j<SqrtN && i<L1[j];++j)
        {
            LL y=n/j/Prime[i];
            g1[j]-=((y<SqrtN)?
                (g0[y]-MMax(0,MMin(i,CntP[y])-L0[y])):
                (g1[n/y]-MMax(0,i-L1[n/y])));
        }
        for (int j=SqrtN-1;j>=1 && i<L0[j];--j)
        {
            LL y=j/Prime[i];
            g0[j]-=(g0[y]-MMax(0,MMin(i,CntP[y])-L0[y]));
        }
    }
    for (int i=1;i<SqrtN;++i)
        g0[i]-=CntP[i]-L0[i]+1,
        g1[i]-=MMax(0,PN-L1[i])+1;
    for (int i=1;i<SqrtN;++i) g0[i]*=4,g1[i]*=4;
}
void GetF(int SqrtN,int PN,LL n)
{
    for (int i=1;i<SqrtN;++i) f0[i]=f1[i]=1;
    for (int i=PN-1;i>=0;--i)
    {
        for (int j=1;j<SqrtN && i<L1[j];++j)
        {
            LL y=n/j/Prime[i];
            for (int c=1;y;y /=Prime[i],++c)
                f1[j]+=(3*c+1)*((y<SqrtN)?
```

```
                    (f0[y]+4*MMax(0,CntP[y]-MMax(i+1,L0[y]))):
                    (f1[n/y]+4*MMax(0,PN-MMax(i+1,L1[n/y]))));
            }
            for (int j=SqrtN-1;j>=1 && i<L0[j];--j)
            {
                int y=j/Prime[i];
                for (int c=1;y;y/=Prime[i],++c)
                    f0[j]+=
                        (3*c+1)*
                        (f0[y]+4*MMax(0,CntP[y]-MMax(i+1,L0[y])));
            }
        }
        for (int i=1;i<SqrtN;++i)
            f0[i]+=4*MMax(0,CntP[i]-L0[i]),
            f1[i]+=4*(PN-L1[i]);
}
LL GetSum(LL n)
{
    int SqrtN=1,PN=0;
    for (;LL(SqrtN)*SqrtN<=n;++SqrtN);
    for (;LL(Prime[PN])*Prime[PN]<=n;++PN);
    L0[0]=0;
    for (int i=1;i<SqrtN;++i) for (L0[i]=L0[i-1];LL(Prime[L0[i]])*Prime[L0[i]]<=i;++L0[i]);
    L1[SqrtN]=0;
    for (int i=SqrtN-1;i>=1;--i)
    {
        LL x=n/i;
        for (L1[i]=L1[i+1];LL(Prime[L1[i]])*Prime[L1[i]]<=x;++L1[i]);
    }
    CntP[0]=0;
    for (int i=1;i<SqrtN;++i)
        for (CntP[i]=CntP[i-1];Prime[CntP[i]]<=i;++CntP[i]);

    GetG(SqrtN,PN,n);GetF(SqrtN,PN,n);
    LL Ans=f1[1];
    for (LL i=1;i<SqrtN;++i) Ans+=D3[i]*g1[i];
    return Ans;
}
```

By SemiWaker

ubuntu下PATH路径的设置——工作笔记 阅读数

在自己编写了一个shell小的脚本，而此脚本只在固定文件夹下可以执行，在其他的路径下，该脚本不能使用。所以就... 博文 来自： yuyantai1234的

LibreOJ #6053. 简单的函数 Min_25筛 阅读数

题意n<=1010n<=1010n 博文 来自： beginend

【数论】Min_25筛 阅读数

听说这玩意玩爆洲阁筛？？？ Min_25筛可以解决一类积性函数求和问题，筛质数假设我们现在要对n以内的质数求和... 博文 来自： 伊克赛艇旗舰店

BZOJ_P2152 聪聪可可(点分治) 阅读数

BZOJ传送门TimeLimit:3SecMemoryLimit:259MBSubmit:1542Solved:790[Submit][Status][Discuss]Description... 博文 来自： BeiYu

BZOJ 4241 历史研究 阅读数

BZOJ4214http://www.lydsy.com/JudgeOnline/problem.php?id=4241区间询问。询问元素大小与出现次数乘积... 博文 来自： ZLH_HHHH的博客

BZOJ 4484: [Jsoi2015]最小表示 拓扑排序 bitset 阅读数 246

4484:[Jsoi2015]最小表示TimeLimit: 20Sec MemoryLimit: 512MBSubmit: 120 Solved: 78[Submit][Status][Dis... 博文 来自： BlackJack

python3练习题--求质数（素数） 阅读数 1488

题目：求100以内的质数（素数）。代码：#!/usr/bin/python3importmathl=[]forainrange(1,100):forbinrange(2,... 博文 来自： PythonStory--记...

关于快速寻找素数的方法 阅读数 5142

利用素数筛选法进行素数的快速查找。原理很简单，素数一定是奇数，素数的倍数一定不是素数。思路如下：预定义... 博文 来自： 学渣的博客

Min25筛小结 阅读数 895

关于筛法，最近看到了很多，也尝试的学了一些。总的来说可以分为线性筛和亚线性筛。所谓线性筛，就是可以在线... 博文 来自： alpc_qleonardo

Loj 6053. 简单的函数（Min_25筛） 阅读数 39

Loj6053.简单的函数（Min_25筛）题目大意有一积性函数，当p为质数时有f(pc)=p XOR cf(p^c)=p\... 博文 来自： FLY_WHITE的博客

min_25筛（学习笔记） 阅读数 132

之前考试就学了一下min_25min_25min_25筛，现在理解得更多一些，补一波笔记先放题免得我忘了LOJ#6053.简... 博文 来自： 世界上最可爱的嘤...

[min25筛学习小记]LOJ6053 阅读数 659

min25筛解决一类积性函数求前缀和问题，主旨为模拟普通筛法过程。假设我们要求∑i=1..nF(i)∑i=1..nF(i)\sum_{i=... 博文 来自： ZLTJohn的博客

洲阁筛/Min25筛 阅读数 113

1：完整的Min25筛学习方案2：虽然不是但我也挂 博文 来自： Freopen的博客

若干式子的证明 阅读数 231

一些定理：1.设f,gf,gf,g为两个数论函数，ttt为完全积性函数，若：f(n)=∑i=1nt(i)g(⌊ni⌋)f(n)=∑i=1nt(i)g(⌊ni⌋)f(n)=\... 博文 来自： Coldfresh的博客

各种数论问题汇集 阅读数 1027

各种数论问题汇集SemiWaker数论函数、狄利克雷卷积、杜教筛、约数和、枚举优化、公式推导方法、[bool]条件框... 博文 来自： SemiWaker的博客

GDKOI2017总结 阅读数 577

GDKOI2017总结 博文 来自： SemiWaker的博客

LOJ 6235 洲阁筛 阅读数 199

尝试理解Min_25筛失败，绝望地用洲阁筛切了个板题。#include<bits/stdc++.h&am... 博文 来自： ACM, deep love

GDOI2017游记 阅读数

真的要盲了Day0：上午方的不行，看了一遍洲阁筛，感觉出了可能也不太会做==，于是跑去复SAM，做了一道模板... 博文 来自： L_0_Forever_L

C++写小型病毒 -_- 阅读数

1.建立线程运行其他可执行文件2.关闭任务管理器或者其他窗口3.ShellExecute的用法4.打开关闭显示器5.使鼠标乱跑... 博文 来自： stark_summer的

[51nod 1184]第N个质数 阅读数

题目大意找第n个质数，n 博文 来自： 不来也不去的

【LOJ6053】【Min_25筛模板题】简单的函数
阅读数
Descriptionhttps://loj.ac/problem/6053SolutionMin_25筛模板题，具体可以看2018年zzt的论文。注意一个地方... 博文 来自： ღ♂Hany01's B......

动态仙人掌 系列题解之一——3464: 动态仙人掌 I
阅读数
（重发下这篇原发于2014-03-18的博客，原博客被网易莫名禁掉了）现在好像各种题目出树已经出烦了，开始出仙... 博文 来自： VFleaKing的博客

Min_25筛代码
阅读数
Sn=∑i=1nfiSn=∑i=1nfiS_n=\sum_{i=1}^nf_iFn=∑i=1n[i为质数]fiFn=∑i=1n[i为质数]fiF_n=\sum_{i=1}^n[i为质... 博文 来自： zawedx的博客

51nod 1184 第N个质数（二分+大区间求素数模板）
阅读数
第1个质数是2，第3个质数是5，给出一个数N，求第N个质数。Input输入1个数N(1Output输出第N个质数。思路：... 博文 来自： CillyB的博客

51nod 1575 Gcd and Lcm
阅读数
51nod1575GcdandLcm链接：https://www.51nod.com/onlineJudge/questionCode.html#!problemId=1575... 博文 来自： ZLH_HHHH的博客

bzoj 4916: 神犇和蒟蒻 杜教筛
阅读数 412
题意求A=∑Ni=1μ(i2)和B=∑Ni=1φ(i2)A=\sum_{i=1}^N{\mu(i^2)}和B=\sum_{i=1}^N{\varphi(i^2)}n 博文 来自： beginend

求大区间内素数的个数（区间筛法）
阅读数 3884
闲来无事，补一下小知识。给定整数a和b，请问区间[a,b)内有多少个素数？a#include#include#include#include#i... 博文 来自： linlinsong—ACM...

埃氏筛法+线性筛法+杜教筛+min25筛总结
阅读数 670
埃氏筛法这个筛法是最朴素的筛法了，可以在O(nloglogn)O(nloglogn)O(nloglogn)的时间内（基本O(n)O(n)O(n)... 博文 来自： WAautomaton的...

Min_25筛学习小记
阅读数 1022
前言听说大家都会了Min_25Min_25Min_25或洲阁筛了，虚的一批的我马上学了一下。Min_25筛首先这种筛法可以... 博文 来自： XianHaoMing的博...

[杜教筛 约数和前缀和] 51Nod 1220 约数之和
阅读数 1731
吉丽博客传送门：http://jiruyi910387714.is-programmer.com/posts/195270.html套用公式后反演然后杜教筛求... 博文 来自： 雯舞

国家集训队2016论文集 高清完整.pdf版下载
02-09
国家集训队2016论文集
下载

min_25筛学习小记
阅读数 225
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... 博文 来自： Cold_Chair的博客

动态仙人掌（dinosaur）
阅读数 358
【分析】这道题在考场看到我是完全的蒙蔽的，惊人的妄想着能否用数据离散化后的dp来骗分。。。我果然很菜。。... 博文 来自： xyc1719的博客

动态仙人掌 系列题解之四——link-cut cactus
阅读数 580
（重发下这篇原发于2014-03-25的博客，原系列的其他三篇博客被网易莫名禁掉了。。。所以把那三篇连同最后这... 博文 来自： VFleaKing的博客

仙人掌相关问题的处理方法（未完待续）
阅读数 4094
仙人掌相关问题的处理方法如图所示：仙人掌图就是长得像仙人掌的图嘛（我真没看出哪里像了）定义：对一个无向... 博文 来自： Virgil's Blog

动态仙人掌 系列题解之二——3465: 动态仙人掌 II
阅读数 195
（重发下我这篇原发于2014-03-19的网易博客，原博客被网易莫名禁掉了。。被迫手动搬家，忧伤）动态仙人掌II能... 博文 来自： VFleaKing的博客

jzoj5987 【WC2019模拟2019.1.4】仙人掌毒题 （动态圆方树维护仙人掌）
阅读数
失智又被题目吓到了考虑树的情况，连通块数目就是总点数-存在的边。考虑仙人掌的情况，连通块数目就是总点数-... 博文 来自： 一位蒟蒻的小博客

Count prime hoj 区间求素数个数（大范围）
阅读数
/*区间筛素数简述：有的时候，我们需要知道某个特定区间的素数(区间大小较小，但数可能很大)。那么数组就开不... 博文 来自： ihe 111031041

【51Nod 1184】第N个质数
阅读数
Description第1个质数是2，第3个质数是5，给出一个数N，求第N个质数。Solution很明显这个要二分答案。现在... 博文 来自： Facico的博客