## 素筛法

### 埃氏筛法

### 欧拉筛法：

if(i%prime[j]==0)break;

保证每个合数：

理解为：只会被它的最小素因子筛掉，

或者理解为：只被它最大的因数筛掉（非素数一定存在1和他本身以外的因数）

把复杂度降到了O(N)，一亿大约1秒多一点点

时间复杂度分析：

之前说了1000万次1秒，这个O(N)具体来说是1亿多次循环了，也只要1秒。

为什么？，

1000万次1秒，算上了输入输出时间，是最保守的估计

1000万次1秒,指的对数组赋值，浮点数运算，这种相对比较慢的操作，以及很多个操作混合一起。

而i++，比较，还有乘法运算，这类是很快的操作，单个来说1亿次半秒不到

for(){

赋值语句

}

1亿次大约要0.5秒

耗时都体现在了内层循环，所以也显出break的重要

for(j=0;i\*prime[j]<n;j++){

vis[i\*prime[j]]=false;

if(i%prime[j]==0){

break;

}

}

## 素数计数

### 素筛法：

最常规思路是：求x以内素数个数，就筛选x以内素数，然后得到的素数个数就是π(x)，可以加一些技巧，遍历素筛的vis数组，用一个数组计算π(i)<这样可以求出：小于x所有的数i的π(i);

无论如何,筛法时间复杂度空间复杂度都是o(x);

### Meisell-Lehmer算法

π(x)表示：小于等于x的数中素数个数.

给定x求解π(x) 如何求？

求π(x)的Meisell-Lehmer算法，通过了数学推导，使得时间复杂度低于线性。

关于这个算法网上的资料较少，谷歌英文资料除了我个人英语水平外，实现原理上也确实没有明确清晰的讲解，知识列举了一些公式，没有额外什么描述。

但之后翻到了一个人个人网站讲的是目前发现最清晰的。

<http://dengtesla.com/2018/02/22/%E5%85%B3%E4%BA%8EMeissel-Lehmer%E7%AE%97%E6%B3%95%E8%AE%A1%E6%95%B0%E7%B4%A0%E6%95%B0/>

基维百科：

<https://en.wikipedia.org/wiki/Meissel%E2%80%93Lehmer_algorithm>

#### 描述算法思路：

把x以内所有素数分成两部分，前a个素数 和其余素数,a是一个人为任意取的正整数。取什么后面说，取得好不好影响算法效率

设其余素数有t个，那么π(x)=a+t;

设函数P(k,x,a)代表这样的数的个数，这些数满足

* 恰好有k个质因子
* 全都小于等于x
* 每个数的每个质因子大于pa，pa代表第a个质数

这个函数k>=0, k==0时0个质因子代表了1这个数

因此P(0,x,a)=1;

由于P(1,x,a)代表了小于等于x,且只有一个质因数，且这个数大于a

的数的个数

所以有π(x)=a+ P(1,x,a);

设ϕ(x,a)=P(0,x,a)+ P(1,x,a)+ P(2,x,a)+…+ P(n,x,a)

n代表了：小于等于x的数中，质因子全都大于a的数中，质因子最多的数，的质因子的个数

ϕ(x,a)代表了：小于等于x，且质因数不含小于pa 的数 的个数

有以下公式:

π(x)=a+ P(1,x,a)=a+ ϕ(x,a)-1- ( P(2,x,a)+ P(3,x,a) + …+ P(n,x,a) )

到这里好像不能化简了，该轮到a的作用了。

P(2,x,a)+ P(3,x,a) + …+ P(n,x,a)这些项共n-2项，

N代表：小于等于x的数中，质因子全都大于a的数中，质因子最多的数，的质因子的个数

设x=1000亿，在这里想一想，怎么取a使得n比较小

a如果取得小，n就比较大

a如果取得大，n就比较小，

要使n=1,那么a要取到π(x^(1/2))

要使n=2,那么a要取到π(x^(1/3))

要使n=3,那么a要取到π(x^(1/4))

.。。。可以想象为什么，比如a=π(x^(1/2))，表明了ϕ(x,a)里每个数的质因数都大于pa,也就是必定大于x^(1/2)，两个或者更多大于x^(1/2)相乘必定大于x;因此可以证明a=π(x^(1/2))时，n==1

**求解ϕ(x,a)：**

ϕ(x,a)= ϕ(x,a-1)- ϕ(x/pa,a-1)

用动态规划的思路，这个方程正确性介绍一下。ϕ(x,a)就是小于等于x且质因数全都大于pa的数的个数。ϕ(x,a)和ϕ(x,a-1)差了什么。

ϕ(x,a-1)- ϕ(x,a)代表了：小于等于x且 每个数：质因数存在至少一个等于pa，其余大于pa .这样的数的个数。

由于每个数包含因数pa，把x的范围里每个数除以pa,这样再计数得到数量不变，ϕ(x,a-1)- ϕ(x,a)等价于：小于x/ pa，其余大于p[a-1].这样的数的个数。

得到ϕ(x,a-1)- ϕ(x,a)= ϕ(x/pa,a-1)

有了这个转移方程如果程序硬算ϕ(x,a)，复杂度随着a指数增长。

还要考虑优化

## 素数检测

给一个数n，检测n是不是素数

### 素筛:

对于小于10^9的数，只要用素筛就够了，判断是不是质数直接查询vis数组；

**性能：**

需要初始化数组和进行一遍素筛，时间复杂度，空间复杂度都是o(MAX);MAX是n的上限。但是每次检测的速度是o(1)

### 试除法：

判断一个数n是不是素数，逐个试验区间[2,sqrtn]的每个数，发现存在一个属于[2,sqrtn]的i使得n%i==0，就断定n不是素数。如果不存在这样的i，就表明n是素数。

**性能：**

单次判断素数时间是sqrt(n);

### Miller-Rabin检测法：

解决这个问题已知最快的方法，对于特别巨大的数字，比如10^18这种，用素筛从时间和时间空间都行不通。用试除法也要10^9的计算时间，还是太慢。

数太大就要考虑借助数学的方法了。

Miller-Rabin借助了几个定理，使单次判断素数时间变成logn