

求1到n之间素数的个数_swustacmer_新浪博客

本题的 $n \leq 100000000$;对于本题最常规的一种做法就是打表，虽然时间做的出来，但是代码长度就不乐观了，但是我们可以适当的做些优化。比如首先我们把100000000分成1000份，那么每份的长度是100000。这样假设我们要求的是1到n之间的素数个数，那么我们可以把这些区间的素数个数加起来[1,100000],[100001,200000].....[.....,a]这样前面的区间的个数已经在打表中可得，只需求最后的区间就可以了。最后的那个区间我们就要素数筛选就可以了，这是其一种方法。

本题的另一种解法是D&+容斥原理，你可以这样想 $a = \sqrt{n}$ 最多不过10000。

首先我们都知道一个合数都可以用多个素数的乘积表示，比如： $30 = 2 * 3 * 5$ 。所以我们可以求出合数的个数，再从总数里减去就可以了。这样我们可以减去2的倍数(当然不包括2)，3的倍数.....。但是我们有发现6既是2的倍数也是3的倍数，这样就产生了容斥原理。下面是代码：

```
#include
#include
#include
int flag[10006];
int num[10000],total,now,n;
void solve(int index,int Mul,int K){
    int i,t;
    if(K==0){
        now+=n/Mul;
        return;
    }
    for(i=index;i
    Mul*=num[i];
    t=now;
    if(Mul<=n){
        solve(i+1,Mul,K-1);
    }
    if(t==now)//优化剪枝（1）
        return;
    Mul/=num[i];
}
}
int Judge(int x){
    int t,i;
    t=sqrt(x*1.0);
    for(i=2;i<=t;i++){
        if(n%i==0)
            return 0;
    }
    return 1;
}
int main(){
    int i,j,t,Count;
    memset(flag,0,sizeof(flag));
    for(i=2;i<=10000;i++){
        if(flag[i])
            continue;
        for(j=2;j*i<=10000;j++)
            flag[i*j]=1;
    }
    while(scanf("%d",&n)==1){
        Count=0;
        if(n<=10000){
```

```
for(i=2;i
if(!flag[i])
Count++;
}
else{
t=sqrt(1.0*n);
total=0;
for(i=2;i<=t;i++)
if(!flag[i])
num[total++]=i;
for(i=1;i<=total;i++){
now=0;
solve(0,1,i);
if(now==0)//优化(2)
break;
if(i&1)
Count+=now;
else
Count-=now;
}
Count-=total;
Count=n-Count-1;
if(Judge(n))
Count--;
}
printf("%d\n",Count);
}
return 0;
}
```

思想来自: <http://apps.hi.baidu.com/share/detail/16883367>