

比赛链接

补题情况

题目	蒋贤蒙	王赵安	王智彪
A	2	0	0
D	0	0	0
E	0	0	0
F	0	0	0
J	0	0	0
K	0	0	0

题解

A. Bags of Candies

题意

多组数据。每组数据给定 $n \leq 10^{11}$ 定义两个数能匹配当且仅当两个数的最大公因数不唯一，求 $1 \sim n$ 的最大匹配数。

打表法

设 $D(n)$ 表示大于 $\lfloor \frac{n}{2} \rfloor$ 的质数个数。首先证明答案为 $\lfloor \frac{D(n)-1}{2} \rfloor$

首先 1 和任意大于 $\lfloor \frac{n}{2} \rfloor$ 的质数显然都不能配对，所以这是答案上界。

对其他数，将它放入它的最大素因子所在的桶中。于是对每个素因子 P 桶中至少有 $P, 2P$ 如果桶中有偶数个数，直接两两配对。

否则将 $2P$ 放入 2 的桶中，其他两两配对。易知最后至多只剩下一个数，证毕。

接下来考虑如何计算 $D(n)$ 考虑分段打表，每个段大小为 10^7 ，统计该范围内的素数个数。

然后查询时预处理完整段的前缀和，处理最后一个不完整的段的答案即可。

关于查询区间 $[L, R]$ 答案，可以用埃氏筛 $O(\sqrt{n})$ 预处理后 $O((r-l)\log\log(r-l))$ 查询。

打表复杂度 $O(n\log\log n)$ 每组数据复杂度 $O(10^7\log\log n)$ 打表时间比较长而且打表数据需要压缩。

```
const LL MAXN=1e11;
const int SqrN=sqrt(MAXN)+5,MAXL=1e7;
namespace Prime{
    int prime[SqrN],pcnt;
```

```
bool pvis[SqrN],vis[MAXL];
void init(){
    _for(i,2,SqrN){
        if(!pvis[i])
            prime[pcnt++]=i;
        for(int j=0;j<pcnt&&prime[j]*i<SqrN;j++){
            pvis[prime[j]*i]=true;
            if(i%prime[j]==0)
                break;
        }
    }
}
int count(LL L,LL R){
    int len=R-L+1,cnt=0;
    _for(i,0,len)vis[i]=false;
    if(L==1)vis[0]=true;
    _for(i,0,pcnt){
        for(LL
j=1LL*max((LL)prime[i],(L+prime[i]-1)/prime[i])*prime[i];j<=R;j+=prime[i])
            vis[j-L]=true;
        }
        _for(i,0,len)
            cnt+=!vis[i];
        return cnt;
    }
}
const int MAXB=1e4+5;
char buf[]="";//打表结果
LL a[MAXB];
void decode(){
    int n=strlen(buf);
    for(int i=0;i<n;i+=4){
        int s=0;
        _for(j,i,i+4){
            s*=62;
            if(buf[j]>='0'&&buf[j]<='9')
                s+=buf[j]-'0';
            else if(buf[j]>='a'&&buf[j]<='z')
                s+=buf[j]-'a'+10;
            else
                s+=buf[j]-'A'+36;
        }
        a[i/4+1]=s;
    }
}
LL calc(LL n){
    if(n==0)return 0;
    LL b=(n-1)/MAXL;
    return a[b]+Prime::count(b*MAXL+1,n);
}
```

```
void solve(){
    LL n=read_LL();
    enter(n-(n-calc(n)+calc(n/2)-1)/2);
}
int main(){
    decode();
    Prime::init();
    _for(i,1,MAXB)
        a[i]+=a[i-1];
    int T=read_int();
    while(T--){
        solve();
    }
    return 0;
}
```

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:%E7%BB%84%E9%98%9F%E8%AE%AD%E7%BB%83%E6%AF%94%E8%B5%9B%E8%AE%B0%E5%BD%95:contest19&rev=1631367880

Last update: 2021/09/11 21:44