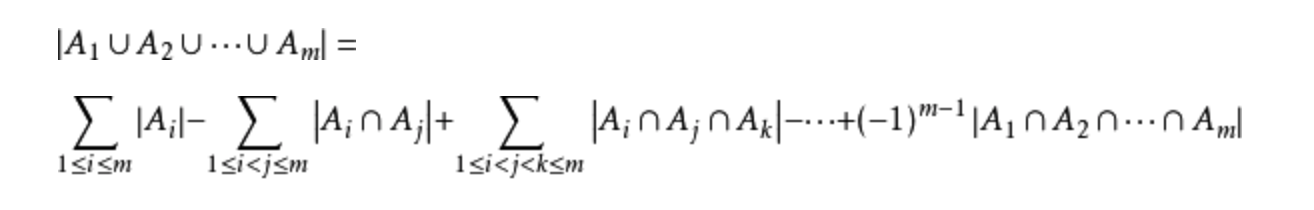
容斥原理—基础计数工具

例1:

给出一个数n，求1到n中，共有多少个数同时不是2 5 11 13的倍数。

Source：nowcoder

思路：

容斥原理定义：

代码：

#include"bits/stdc++.h"

using namespace std;

typedef long long ll;

**int** main()

{

ll n;

**while**(scanf(**"%lld"**,&n)!=**EOF**)

{

ll cnt;

cnt=n-(n/2)-(n/5)-(n/11)-(n/13);

cnt=cnt+(n/10)+(n/22)+(n/26)+(n/55)+(n/65)+(n/143);

cnt=cnt-(n/110)-(n/130)-(n/715)-(n/286);

cnt=cnt+(n/1430);

printf(**"%lld\n"**,cnt);

}

return 0;

}

例**2:**

给定n个数，求从中任意选数的所有方案gcd的总和。n个数字都<=m(给定)。n<=10^5,m<=10^6。

Source：SRM20 P2.

思路:对于每个数字x(1<=x<=m)处理出n个数中有多少个是它的倍数，记为b，那么有2^b-1种gcd=k\*x的方案，再容斥掉其倍数（ans[j]）得到ans[i]。

使用的仍是自带容斥的技巧，就是直接容斥掉已经计算过的答案ans，这些答案ans已经自带上一层容斥了。

复杂度分析：1枚举n次，2枚举n/2次，所以总共枚举n\*(1+1/2+1/3+1/4+…+1/n)，后面的数列是常见的近似ln(n)，所以总复杂度O(n ln n)。

#include"bits/stdc++.h"

using namespace std;

const int maxn=500010;

const int N=10000010;

const int M=1e9+7;

int n,x,b[N],fx[maxn],f[N],ans=0,mx=0;

int main(){

scanf("%d",&n);

for(int i=1;i<=n;i++)scanf("%d",&x),mx=max(mx,x),b[x]++;

fx[0]=1;for(int i=1;i<=n;i++)fx[i]=(fx[i-1]<<1)%M;

for(int g=mx;g>=2;g--){

x=b[g];

for(int i=g+g;i<=mx;i+=g){

x+=b[i];

f[g]=(f[g]-f[i]+M)%M;

}

f[g]=(f[g]+1ll\*(fx[x]-1)\*(n-x)%M)%M;

ans=(ans+f[g])%M;

}

printf("%d",ans);

return 0;

}

课后习题：

【题目1】CF585[E. Present for Vitalik the Philatelist](http://codeforces.com/contest/585/problem/E)

【题意】给定n个数字，定义一种合法方案为选择一个数字Aa，选择另外一些数字Abi，令g=gcd(Ab1...Abx)，要求满足g≠1且gcd(Aa,g)=1，求方案数取模1e9+7。

【题目2】“玲珑杯”线上赛 Round #17

【题意】有n个硬币，初始全部正面朝上，现在有m次操作，每次把编号是x的倍数的硬币翻面，最后问多少个硬币正面朝上。