题意：

给出n个整数，每个整数ai找出最小的xi使得Φ(xi)>=ai,求x1+x2+...+xn

解法1：

打欧拉函数表，二分查找

但要注意，因为欧拉函数表并不是递增的，就是会有phi[i]>phi[i+1]的情况，但二分查找只能查找有序的

我们分析题目发现，只要找到第一个大于等于a[i]的phi[x]就行了，这样我们可以处理一下phi数组，phi[i]=max(phi[i],phi[i-1])，就是当前phi[i]表示的是1-i中最大的phi值，这样就是非递减的了，就可以用二分查找了

因为欧拉函数的定义是不超过n的整数中与n互质的数的个数，所以如果这样的话phi[1]应该是0，但在计算的时候规定phi[1]=1，所以在这个题中，要避免这个情况，也就是说，题目中说找phi[x]>=a[i]的最小x，而如果a[i]=1,那么我们应该算的x=2，而不是应该让他是1(因为phi[1=1是规定的一个特殊情况)

所以二分搜的时候是从2开始

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const int maxn=1e6+10;

int phi[maxn];

int a[10010];

void init(int n)

{

for(int i=2; i<=n; i++)

phi[i]=0;

phi[1]=1;

for(int i=2; i<=n; i++)

if(!phi[i])

for(int j=i; j<=n; j+=i)

{

if(!phi[j])

phi[j]=j;

phi[j]=phi[j]/i\*(i-1);

}

for(int i=2;i<=n;i++)

phi[i]=max(phi[i],phi[i-1]);

}

int main()

{

freopen("input.txt","r",stdin);

init(maxn);

int T,n;

cin>>T;

for(int o=1;o<=T;o++)

{

ll ans(0);

cin>>n;

int x;

for(int i=0;i<n;i++)

{

cin>>x;

ans+=lower\_bound(phi+1,phi+maxn,x)-phi;

cout<<lower\_bound(phi+1,phi+maxn,x)-phi<<"\n";

}

cout<<"Case "<<o<<": "<<ans<<" Xukha\n";

}

return 0;

}

解法二：打欧拉函数表，从小到大一依次找

注意这个找法写的比较巧

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const int maxn=1e6+10;

int phi[maxn];

int a[10010];

void init(int n)

{

for(int i=2; i<=n; i++)

phi[i]=0;

phi[1]=1;

for(int i=2; i<=n; i++)

if(!phi[i])

for(int j=i; j<=n; j+=i)

{

if(!phi[j])

phi[j]=j;

phi[j]=phi[j]/i\*(i-1);

}

}

int main()

{

// freopen("input.txt","r",stdin);

init(maxn);

int T,n;

cin>>T;

for(int o=1;o<=T;o++)

{

ll ans(0);

cin>>n;

int x;

for(int i=0;i<n;i++)

cin>>a[i];

sort(a,a+n);

int j=0;

for(int i=2; i<maxn && j<n;i++)

while(phi[i]>=a[j] && j<n)

ans+=i,j++;

cout<<"Case "<<o<<": "<<ans<<" Xukha\n";

}

return 0;

}

解法三：

我们知道一根长度为p的竹竿它的score其实就是欧拉函数值φ(p)。又因为一个素数p的φ(p)=p-1，所以我们只需要从x+1(x是幸运数字)开始找第一个出现的素数，那就是最小花费。

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const int maxn=1e6+10;

bool is\_prime[maxn]; //初始化，=false表示是素数

int prime[maxn]; //保存素数

void Euler(int n)

{

int i, j, k;

k = 0;

for(i = 2; i < n; i++)

{

if(is\_prime[i] == false)

prime[k++] = i;

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

{

is\_prime[ i\*prime[j] ] = true;

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

break;

}

}

}

int main()

{

// freopen("input.txt","r",stdin);

int T,n;

cin>>T;

Euler(maxn);

for(int o=1;o<=T;o++)

{

ll ans(0);

cin>>n;

int x;

for(int i=0;i<n;i++)

{

cin>>x;

for(int j=x+1;;j++)

if(is\_prime[j]==0)

{

ans+=j;

break;

}

}

cout<<"Case "<<o<<": "<<ans<<" Xukha\n";

}

return 0;

}