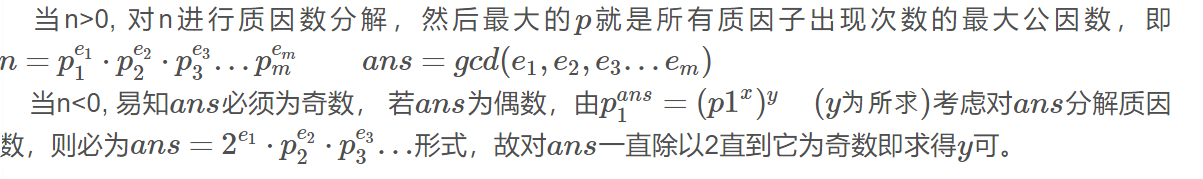
题目大意：输入T组数据，每组一个整数n(可能为负数),求满足a^p =n的最大的p



注意n要开long long，不能信他的鬼范围开到int就行…..

卡了好久好久

#include<bits/stdc++.h>

#include<vector>

using namespace std;

typedef long long ll;

const int maxn=1000100;

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

int prime[maxn]; //保存素数

int k;

void Euler(int n)

{

int i, j;

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);

ios::sync\_with\_stdio(false);

cin.tie(0);

Euler(maxn);

int T;

cin>>T;

ll n;

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

{

bool flag=0;

cin>>n;

if(n<0)

{

n=-n;

flag=1;

}

int ans(0);

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

{

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

{

int x=0;

while(n%prime[i]==0)

{

x++;

n/=prime[i];

}

if(ans==0)ans=x;

else ans=\_\_gcd(ans,x);

}

if(n==1)break;

}

if(n>1)ans=1;

if(flag)

while(ans%2==0)ans/=2;

printf("Case %d: %d\n",cas,ans);

}

return 0;

}