

2020牛客暑期多校训练营（第三场）

比赛情况

题号	A	B	C	D	E	F	G	H	I	J	K	L
状态	0	0	0	0	0	0	0	-	-	-	-	0

0 在比赛中通过 0 赛后通过! 尝试了但是失败了- 没有尝试

比赛时间

2020-07-18 12:00-17:00

题解

B - Basic Gcd Problem

题意：当 $x > 1$ 时， $f_c(x) = \max_{i=1 \dots x-1} f_c(\gcd(i, x))$

找到长度 9 和 6 的边，叉集判断顺逆时针关系即可。

```
#include<bits/stdc++.h>
#define pii pair<int,int>
#define ll long long
using namespace std;
#define eps 1e-4
struct Node
{
    double x,y;
    Node(double x=0,double y=0):x(x),y(y){}
    Node operator -(Node a){return Node(x-a.x,y-a.y);}
}p[22];
int dcmp(double x)
{
    if(x<-eps)return -1;
    return x>eps;
}
double sqr(double x){return x*x;}
double dissqr(Node a,Node b){return sqr(a.x-b.x)+sqr(a.y-b.y);}
double Cross(Node a,Node b){return a.x*b.y-a.y*b.x;}
int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        for(int i=0;i<20;i++)scanf("%lf%lf",&p[i].x,&p[i].y);
```

```
int j=0;
for(int i=0;i<20;i++)
{
    if(dcmp(dissqr(p[i],p[(i+1)%20]) - 81)==0)
        {j=i;break;}
}
if(dcmp(dissqr(p[(j+2)%20],p[(j+1)%20]) - 36)==0)
{
    if(Cross(p[(j+2)%20]-p[(j+1)%20],p[j]-
p[(j+1)%20])<0)puts("right");
    else puts("left");
}
else
{
    if(Cross(p[(j-1+20)%20]-p[j],p[(j+1)%20]-p[j])<0)puts("right");
    else puts("left");
}
}
return 0;
}
```

C - Operation Love

题面中给出右手图形，左手与之对称。之后询问给定图形是右手还是左手（可以平移旋转，可能顺/逆时针给出点）。

找到长度 9 和 6 的边，叉集判断顺逆时针关系即可。

```
#include<bits/stdc++.h>
#define pii pair<int,int>
#define ll long long
using namespace std;
#define eps 1e-4
struct Node
{
    double x,y;
    Node(double x=0,double y=0):x(x),y(y){}
    Node operator -(Node a){return Node(x-a.x,y-a.y);}
}p[22];
int dcmp(double x)
{
    if(x<-eps)return -1;
    return x>eps;
}
double sqr(double x){return x*x;}
double dissqr(Node a,Node b){return sqr(a.x-b.x)+sqr(a.y-b.y);}
double Cross(Node a,Node b){return a.x*b.y-a.y*b.x;}
```

```

int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        for(int i=0;i<20;i++)scanf("%lf%lf",&p[i].x,&p[i].y);
        int j=0;
        for(int i=0;i<20;i++)
        {
            if(dcmp(dissqr(p[i],p[(i+1)%20]) - 81) == 0)
                {j=i;break;}
        }
        if(dcmp(dissqr(p[(j+2)%20],p[(j+1)%20]) - 36) == 0)
        {
            if(Cross(p[(j+2)%20]-p[(j+1)%20],p[j]-
p[(j+1)%20])<0)puts("right");
            else puts("left");
        }
        else
        {
            if(Cross(p[(j-1+20)%20]-p[j],p[(j+1)%20]-p[j])<0)puts("right");
            else puts("left");
        }
    }
    return 0;
}

```

F - Fraction Construction Problem

给 $a, b \in [1, 2 \times 10^6]$ 问是否存在 $c, d, e, f \in [1, 4 \times 10^{12}]$ 且 $d, f \leq b$ 使得 $\frac{cd}{ab} - \frac{ef}{ab}$

分类讨论一下，如果 a 和 b 不互质可以很容易构造出来；如果互质，分解 b 如果 b 只有一种质因子则不存在，否则令 d 和 f 为 b 的两个互质的因数，然后通分，分子就是个拓欧。

```

#pragma GCC optimize(2)
#pragma GCC optimize(3,"Ofast","inline")
#include<bits/stdc++.h>
#define ALL(x) (x).begin(),(x).end()
#define ll long long
#define ull unsigned long long
#define pii_ pair<int,int>
#define mp_ make_pair
#define pb push_back
#define fi first
#define se second

```

```
#define rep(i,a,b) for(int i=(a);i<=(b);i++)
#define per(i,a,b) for(int i=(a);i>=(b);i--)
#define show1(a) cout<<#a<<" = "<<a<<endl
#define show2(a,b) cout<<#a<<" = "<<a<<" ; "<<#b<<" = "<<b<<endl
using namespace std;
const ll INF = 1LL<<60;
const int inf = 1<<30;
const int maxn = 2e6+5;
const ll B = 4e12;
inline void fastio() {ios::sync_with_stdio(false);cin.tie(0);cout.tie(0);}

bool vis[maxn];
vector<int> prime;
inline ll gcd(ll a,ll b) {return b==0?a:gcd(b,a%b);}
inline ll exgcd(ll a,ll b,ll &x,ll &y)
{
    ll d;
    if(!b) d=a,x=1,y=0;
    else {d=exgcd(b,a%b,y,x);y-=a/b*x;}
    return d;
}
void init()
{
    int n = 2e6;
    rep(i,2,n){
        if(!vis[i]) prime.pb(i);
        for(ll j=(ll)i*i;j<=n;j+=i) vis[j] = 1;
    }
}
int main()
{
    int _; init();
    ll a,b,c,d,e,f;
    for(scanf("%d",&_);_>0;_--){
        scanf("%lld%lld",&a,&b);
        if(b==1){
            printf("-1 -1 -1 -1\n");
        }else{
            ll k = gcd(a,b);
            if(k > 1){
                a/=k,b/=k;
                d = b,f = b;
                c = a+1,e = 1;
                printf("%lld %lld %lld %lld\n",c,d,e,f); continue;
            }else{
                if(!vis[b]){ // ab互质且b为质数
                    printf("-1 -1 -1 -1\n"); continue;
                }
                for(int x:prime){
                    if(b%x==0){

```

```

        d = 1;
        while(b%x==0) d*=x,b/=x;
        break;
    }
}
if(b==1){ // 只有一种质因子
    printf("-1 -1 -1 -1\n"); continue;
}
f = b;
exgcd(f,d,c,e);
if(c<0){
    ll k = abs(c)/d+1;
    c += k*d,e -= k*f;
}
printf("%lld %lld %lld %lld\n",c*a,d,-e*a,f);
}
}
}
return 0;
}

```

G - Operating on a Graph

每个点初始自己一组， q 个操作，每次将所有与 o_i 组有连边的组并入该组，问最后每个点各属于哪个组。

并查集搞一搞即可，每次将准备并入的组连出去的边与 $G[o_i]$ 合并，注意vector合并时小的插入大的。

```

#include<bits/stdc++.h>
#define pii pair<int,int>
#define ll long long
#define pb push_back
using namespace std;
const int N=8e5+10;
vector<int>G[N],s;
int read()
{
    int x=0,f=1;char c=getchar();
    while(c<'0' || c>'9'){if(c=='-')f=-1;c=getchar();}
    while(c>='0'&&c<='9'){x=x*10+c-'0';c=getchar();}
    return x*f;
}
int father[N];
int find(int x){return x==father[x]?x:father[x]=find(father[x]);}
int main()
{
    int t=read();

```

```
while(t--)  
{  
    int n=read(),m=read();  
    for(int i=0;i<n;i++)vector<int>().swap(G[i]),father[i]=i;  
    for(int i=0;i<m;i++)  
    {  
        int x=read(),y=read();  
        G[x].pb(y),G[y].pb(x);  
    }  
    int q=read();  
    for(int i=1;i<=q;i++)  
    {  
        int u=read();  
        if(find(u)!=u)continue;  
        for(int v:G[u])  
        {  
            int fv=find(v);  
            if(fv!=u)  
            {  
                father[fv]=u;  
                if(s.size()<G[fv].size())swap(s,G[fv]);  
                s.insert(s.end(),G[fv].begin(),G[fv].end());  
                vector<int>().swap(G[fv]);  
            }  
        }  
        if(s.size()>G[u].size())swap(s,G[u]);  
        G[u].insert(G[u].end(),s.begin(),s.end());  
        vector<int>().swap(s);  
    }  
    for(int i=0;i<n;i++)printf("%d ",find(i));  
    puts("");  
}  
return 0;  
}
```

L - Problem L is the Only Lovely Problem

签到水题。

```
#include<bits/stdc++.h>  
using namespace std;  
char s[505],s1[505]="lovely";  
int main()  
{  
    cin>>s;  
    for(int i=0;s1[i];i++)
```

```
{
    if(s[i]>='A'&& s[i]<='Z')s[i]=s[i]-'A'+ 'a';
    if(s1[i]!=s[i]){puts("ugly");return 0;}
}
puts("lovely");
return 0;
}
```

比赛总结与反思

From:

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

Permanent link:

https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:wangzai_milk:20200718%E6%AF%94%E8%B5%9B%E8%AE%B0%E5%BD%95&rev=1595424215

Last update: 2020/07/22 21:23