

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

A - Clam and Fish

依次经过 n 个格子，每个格子上可能有 fish 或 clam

每个格子上可以下面的选一个操作：

如果有 clam 可以制作一个鱼饵

如果有 fish 可以得到一条鱼

消耗一个鱼饵获得一条鱼

什么事都不做

显然有鱼的位置选操作三。剩下的有能制作鱼饵就制作鱼饵，否则就用鱼饵捕鱼。最后加上剩下鱼饵的二分之一。

```
#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<<" "; cout<<#b<<" = "<<b<<endl
using namespace std;
const ll INF = 1LL<<60;
const int inf = 1<<30;
```

```
const int maxn = 2e6+5;
inline void fastio() {ios::sync_with_stdio(false);cin.tie(0);cout.tie(0);}

char s[maxn];
int main()
{
    fastio();
    int _; char x;
    for(cin>>_;;_--){
        int n; cin>>n;
        cin>>s+1;
        int ans = 0,cnt = 0;
        rep(i,1,n){
            if(s[i]=='0'){
                if(cnt) ans++,cnt--;
            }else if(s[i]=='1'){
                cnt++;
            }
            else ans++;
        }
        ans += cnt/2;
        cout<<ans<<endl;
    }
    return 0;
}
```

B - Classical String Problem

题意：两种操作，把字符串最后x个接到前面，把字符串最前面x个接到后面。最后询问一次当前字符串。

题解：相当于把字符串搞成一个环，有一个指针指着开头，每次操作相当于动指针，就每次加减移动字符数然后模长度就好了。

```
#include <bits/stdc++.h>
using namespace std;
const int N = 2e6+5;
char s[N];
int main()
{
    int Q;
    scanf("%s",s);
    scanf("%d",&Q);
    char op[3];
    int n = strlen(s);
    int st = 0,x;
    for (int i = 1;i<= Q;i++) {
```

```

scanf("%s%d",op,&x);
if (op[0] == 'A') {
    x--;
    int pos = ((st+x)%n+n)%n;
    printf("%c\n",s[pos]);
} else {
    st = ((st+x)%n+n)%n;
}
}
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 $9s$ 和 $6s$ 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;
}
```

D - Points Construction Problem

题意:给一个坐标轴，在整数点上染色，一开始所有的都是白色，要求染 n 个，使得有 m 对相邻的点颜色不同。

题解:构造，首先可以发现只有 m 为偶数的时候有方案，然后对于一个新染色点，可能多4对，2对或0对，然后枚举多4对的有几个点，2对的有几个点，算出0对的有几个点，然后先一行空一格染一个，染4对的，然后再最后一个4对的上侧和右侧增加2对的，看剩下的0对的能不能被完全放进这个L形里。可以就可以，不行就没了。

```
#include <bits/stdc++.h>
using namespace std;
const int N = 55;
int ans[N][2];
int n, m;
int main()
{
    int cas;
    scanf("%d", &cas);
    while (cas--) {
        int n, m;
        scanf("%d%d", &n, &m);
        int minans = 4 * n;
        for (int i = 1; i <= n; i++) {
            int x = i; int y = n / x;
            if (n > x * y)
                minans = min(minans, 2 * (x + y + 1));
        }
    }
}
```

```

        else minans = min(minans, 2*(x+y));
    }
    bool find = false;
    for (int cnt4 = 1; cnt4 <= n && !find; cnt4++) {
        for (int cnt2 = 0; cnt2 + cnt4 <= n && !find; cnt2++) {
            if (cnt2 * 2 + cnt4 * 4 != m) continue;
            int cnt0 = n - cnt4 - cnt2;
            int x = cnt2 / 2; int y = cnt2 - x;
            if (cnt0 <= x * y) {
                find = true;
                int tmp = 0;
                for (int i = 1; i <= cnt4; i++)
                    ans[++tmp][0] = i < 1;
                for (int i = 1; i <= x; i++) {
                    tmp++;
                    ans[tmp][0] = ans[cnt4][0] + i;
                    ans[tmp][1] = ans[cnt4][1];
                }
                for (int i = 1; i <= y; i++) {
                    tmp++;
                    ans[tmp][0] = ans[cnt4][0];
                    ans[tmp][1] = ans[cnt4][1] + i;
                }
                for (int i = 0; i < cnt0; i++) {
                    int tx = i % x + 1, ty = i / x + 1;
                    tmp++;
                    ans[tmp][0] = ans[cnt4][0] + tx;
                    ans[tmp][1] = ans[cnt4][1] + ty;
                }
            }
        }
    }

    if (find) {
        printf("Yes\n");
        for (int i = 1; i <= n; i++)
            printf("%d %d\n", ans[i][0], ans[i][1]);
    } else {
        printf("No\n");
    }
}
return 0;
}

```

F - Fraction Construction Problem

给 $1 \leq a, b \leq 2e6$ 问是否存在 $1 \leq c, d, e, f \leq 4e12$ 且 $d, f \nmid b$ 使得 $\frac{cd}{ab} - \frac{ef}{ab} = \frac{cd}{ab}$

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",&_);_--){
        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□ sq 个操作，每次将所有与 so_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-->0)
    {
        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],sl[505]="lovely";
int main()
{
    cin>>s;
    for(int i=0;sl[i];i++)
    {
        if(s[i]>='A'&&s[i]<='Z')s[i]=s[i]-'A'+'a';
        if(sl[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

Last update: 2020/07/23 20:44