2020牛客暑期多校训练营（第三场）

比赛网址

训练结果

- 时间:2020-07-18
- rank:75/1175
- 完成情况：8/8/12

题解

A.Clam and Fish

题意

有n个位置，每个位置有4种状态：（有鱼有诱饵），（有鱼无诱饵），（无鱼有诱饵），（无鱼无诱饵），对于每一个位置，你可以最多拿走当前的一件物品或者用当前手里的一个诱饵凭空刷出一条鱼来收入囊中，请问你从1走到n最多能拿多少鱼？

题解

solved by fyh hxm

当前位置有鱼那必拿，剩下只能有无诱饵的决策问题了。当前位置什么都没有肯定拿诱饵置换，有诱饵的话就需要考虑是拿诱饵还是换鱼，二分位置pos令在这之前全拿诱饵，显然是满足单调性。

```
#include<bits/stdc++.h>
using namespace std;
#define mem(a,b) memset(a,b,sizeof(a))
typedef long long LL;
typedef pair<int,int> PII;
#define X first
#define Y second
inline int read()
{
    int x=0,f=1;char c=getchar();
    while(!isdigit(c)){if(c=='-')f=-1;c=getchar();}
    while(isdigit(c)){x=x*10+c-'0';c=getchar();}
    return x*f;
}
const int maxn=2000010;
int T,n,a[maxn],m,ans1,ans2;
char s[maxn];
```

```
bool judge(int index)
{
    int have=0,res=0;
    for(int i=1;i<=m;i++)
    {
        if(i<=index && a[i]==1)have++;
        else if(have)have--,res++;
    }
    if(have==0){ans2=max(ans2,res);return 1;}
    return 0;
}
int main()
{
    T=read();
    while(T--)
    {
        n=read();
        m=ans1=ans2=0;
        scanf("%s",s);
        for(int i=0;i<n;i++)
        {
            if(s[i]=='2' || s[i]=='3')ans1++;
            else a[++m]=s[i]-'0';
        }
        int L=1,R=m;
        while(R-L>1)
        {
            int mid=L+R>>1;
            if(judge(mid))L=mid;
            else R=mid;
        }
        judge(L);judge(R);
        printf("%d\n",ans1+ans2);
    }
    return 0;
}
```

B.Classical String Problem

题意

给了一个字符串，实现两个操作

1. 把最左/右边的 x 个字符移到最右/左边
2. 查询第 x 个字符

题解

发现无论怎么移动，这个字符串一定成环状，只是起始位置改变，我们直接维护起点的位置即可

```
#include<iostream>
#include<cstdio>
#include<algorithm>
#include<cstring>
#define ll long long
using namespace std;
int read()
{
    int k=0,f=1;char c=getchar();
    for(;!isdigit(c);c=getchar()) if(c=='-') f=-1;
    for(;isdigit(c);c=getchar()) k=k*10+c-'0';return k*f;
}
int n,m,k;
char s[2000005],str[5];
int main()
{
    scanf("%s",s+1);
    n=strlen(s+1);
    m=read();
    k=1;
    for(int i=1;i<=m;i++)
    {
//      cout<<k<<" ";
        int x;
        scanf("%s%d",str,&x);
        if(str[0]=='A')
        {
            printf("%c\n",s[(k+x-2)%n+1]);
        }
        else
        {
            if(x>0)
                k=(k+x-1)%n+1;
            else
                k=(k+n+x-1)%n+1;
        }
    }
    return 0;
}
```

C.Operation Love

题意

给你一个手的坐标，让你判断是左手还是右手。

题解

solved by fyh

找到和手腕连接的部分，用叉积判断大拇指或者小拇指在左边还是右边。

```
#include<bits/stdc++.h>
using namespace std;
#define mem(a,b) memset(a,b,sizeof(a))
typedef long long LL;
typedef pair<double,double> PDD;
#define X first
#define Y second
inline int read()
{
    int x=0,f=1;char c=getchar();
    while(!isdigit(c)){if(c=='-')f=-1;c=getchar();}
    while(isdigit(c)){x=x*10+c-'0';c=getchar();}
    return x*f;
}
const double eps=1e-10;
struct Point{
    double x,y;
    Point() {}
    Point(double _1,double _2):x(_1),y(_2) {}
    Point operator - (const Point&s) const {return Point(x-s.x,y-s.y);}
    double operator * (const Point&s) const {return x*s.y-y*s.x;}
    double len() {return x*x+y*y;}
}a[50];
typedef Point Vec;
double dis(Point a,Point b){
    return (a-b).len();
}
int T;
int main()
{
    T=read();
    while(T--)
    {
        int left=0;
        for(int i=1;i<=20;i++)scanf("%lf%lf",&a[i].x,&a[i].y),a[i+20]=a[i];
        for(int i=1;i<39;i++)
        {
            if(fabs(dis(a[i],a[i+1])-81.0)<eps)
```

```
            {
//              printf("here:%lf %lf\n",a[i].x,a[i].y);
                if(fabs(dis(a[i+1],a[i+2])-36.0)<eps)
                {
                    if((a[i]-a[i+1])*(a[i+2]-a[i+1])>0) left=0;
                    else left=1;
                }
                else if(fabs(dis(a[i+1],a[i+2])-64.0)<eps)
                {
                    if((a[i]-a[i+1])*(a[i+2]-a[i+1])>0) left=1;
                    else left=0;
                }
                break;
            }
        }
        if(left)puts("left");
        else puts("right");
    }
    return 0;
}
```

D.Points Construction Problem

题意

让你在平面上把 n 个点涂黑，要求曼哈顿距离为1的且颜色不同的点对为 m 个

题解

发现按正方形涂是点对最小的做法，我们先按正方形涂，发现剩下的可以单独涂就单独涂

```
#include<iostream>
#include<cstdio>
#include<algorithm>
#include<cstring>
#define ll long long
using namespace std;
int read()
{
    int k=0,f=1;char c=getchar();
    for(;!isdigit(c);c=getchar()) if(c=='-') f=-1;
    for(;isdigit(c);c=getchar()) k=k*10+c-'0';return k*f;
}
int T,n,m,ans[105];
void pr(int x)
{
    int sum=1;
```

```
    printf("1 1\n");
    if(sum==x) return;
    for(int i=2;i<=8;i++)
    {
        for(int k=1;k<=i-1;k++)
        {
            sum++;
            printf("%d %d\n",i,k);
            if(sum==x) return;
        }
        for(int k=1;k<=i-1;k++)
        {
            sum++;
            printf("%d %d\n",k,i);
            if(sum==x) return;
        }
        sum++;printf("%d %d\n",i,i);
        if(sum==x) return;
    }
}
int main()
{
    int sum=1;ans[1]=4;
    for(int i=2;i<=8;i++)
    {
        for(int k=1;k<=i-1;k++)
        {
            sum++;
            ans[sum]=ans[sum-1]+2*(k==1);
        }
        for(int k=1;k<=i-1;k++)
        {
            sum++;
            ans[sum]=ans[sum-1]+2*(k==1);
        }
        sum++;ans[sum]=ans[sum-1];
    }
    for(T=read();T;T--)
//  for(n=1;n<=50;n++)
//      for(m=1;m<=200;m++)
    {
        n=read();m=read();
        if(n==1)
        {
            if(m==4) printf("Yes\n1 1\n");
            else printf("No\n");
            continue;
        }
        else
        {
```

```
            if(m&1)
            {
                puts("No");continue;
            }
            if(m<ans[n])
            {
                puts("No");continue;
            }
            if(m>n*4)
            {
                puts("No");continue;
            }
            if(n*4==m)
            {
                puts("Yes");
                for(int i=1;i<=n;i++)
                    printf("%d %d\n",i,i);
                continue;
            }
            int fl=1;
            for(int i=1;i<=n;i++)
            {
                if(ans[i]+(n-i)*4==m)
                {
                    puts("Yes");
                    pr(i);
                    for(int j=1;j<=n-i;j++)
                    {
                        printf("%d %d\n",-j,-j);
                    }
                    fl=0;
                }
                else if(ans[i]+(n-i)*4-2==m)
                {
                    puts("Yes");
                    pr(i);
                    for(int j=1;j<=n-i-1;j++)
                        printf("%d %d\n",-j,-j);
                    puts("0 1");
                    fl=0;
                }
                if(!fl) break;
            }
        }
    }
    return 0;
}
```

E.Two Matchings

题意

对于一个长为n的序列，进行两次完全不同的两两配对，使得配对后每一对的差值和最小

题解

solved by wxg hxm

最优的一定是相邻的配对

次优的在避免相邻的基础上还需要尽可能的进，容易发现要使配对尽可能的近，只有通过4个一组配对和6个一组配对，在此基础上进行dp即可

```
#include<algorithm>
#include<iostream>
#include<cstdlib>
#include<cstring>
#include<cstdio>
#include<vector>
#include<queue>
#include<cmath>
#include<map>
#include<set>
#define LL long long int
#define REP(i,n) for (int i = 1; i <= (n); i++)
#define Redge(u) for (int k = h[u],to; k; k = ed[k].nxt)
#define cls(s,v) memset(s,v,sizeof(s))
#define mp(a,b) make_pair<int,int>(a,b)
#define cp pair<int,int>
using namespace std;
const int maxn = 200005,maxm = 100005;
LL INF = 1000000000000000ll;
inline int read(){
    int out = 0,flag = 1; char c = getchar();
    while (c < 48 || c > 57){if (c == '-') flag = 0; c = getchar();}
    while (c >= 48 && c <= 57){out = (out << 1) + (out << 3) + c - 48; c =
getchar();}
    return flag ? out : -out;
}
LL A[maxn],n;
LL ans = 0,f[maxn];
int main(){
    int T = read();
    while (T--){
        n = read();
```

```
        REP(i,n) A[i] = read();
        sort(A + 1,A + 1 + n); ans = 0;
        for (int i = 1; i < n; i += 2) ans += A[i + 1] - A[i];
        int m = n / 2;
        f[1] = INF;
        f[2] = A[4] + A[3] - A[1] - A[2];
        if (n > 4) f[3] = A[6] + A[5] + A[3] - A[4] - A[1] - A[2];
        for (int i = 4; i <= m; i++){
            int t = i * 2;
            f[i] = min(f[i - 2] + A[t] + A[t - 1] - A[t - 2] - A[t - 3],f[i
- 3] + A[t] + A[t - 1] + A[t - 3] - A[t - 2] - A[t - 4] - A[t - 5]);
        }
        printf("%lld\n",ans + f[m]);
    }
    return 0;
}
```

F.Fraction Construction Problem

题意

给你a,b,让你找到c□d□e□f□使得$\frac{c}{d}-\frac{e}{f}=\frac{a}{b}$□

另外满足□$d,f<b$

题解

solved by fyh

移项再通分□$\frac{c}{d}=\frac{e*b+a*f}{b*f}$□这个分数必须约掉一个比$b$还大的约数才能保证$d<b$□

开始分类讨论：假如b是质数，且a不是b的倍数，那么不可能找到一个比b还大的约数，显然无解。

假如b是质数，且a是b的倍数，那就把那e,f都取1就好了

假如b是合数，为了方便后续约分，令$b=f*k$□其中f□k互质□$\frac{e*b+a*f}{b*f}=\frac{ef+a}{b}$.这个分数需要再来一个约分，即$(ek+a)\equiv0\pmod{b}$□解完模方程组，我们发现这里$k$就是$d$了。

假如b是合数，但是不存在$b=f*k$□其中f□k互质，即b是某个质数的整数次幂，则需要对模方程组进行同时约分。

```
#include<bits/stdc++.h>
using namespace std;
#define mem(a,b) memset(a,b,sizeof(a))
typedef long long LL;
```

```
typedef pair<LL,LL> PII;
#define X first
#define Y second
inline int read()
{
    int x=0,f=1;char c=getchar();
    while(!isdigit(c)){if(c=='-')f=-1;c=getchar();}
    while(isdigit(c)){x=x*10+c-'0';c=getchar();}
    return x*f;
}
const int maxN=2000000;
int phi[maxN+10],prime[maxN+10],isn[maxN+10],F[maxN+10],len,A;
LL pre[maxN+10];
bool vis[maxN+10];
void init()
{
    phi[1]=1;
    for(int i=2;i<=maxN;i++)
    {
        if(!vis[i])prime[++len]=i;
        for(int j=1;j<=len;j++)
        {
            if(i*prime[j]>maxN)break;
            vis[i*prime[j]]=1;
            isn[i*prime[j]]=prime[j];
            if(i%prime[j]==0){
                break;
            }
        }
    }
    for(int i=2;i<=maxN;i++)
        if(vis[i])
        {
            int tmp=i;
            F[i]=1;
            while(tmp%isn[i]==0)F[i]*=isn[i],tmp/=isn[i];
        }
//  for(int i=2;i<=100;i++) printf("%d %d\n",i,F[i]);
}
LL gcd(LL a,LL b){return b==0 ? a : gcd(b,a%b);}
PII exgcd(LL a,LL b)//解出ax+by=gcd(a,b)
{
    if(!b){return make_pair(1,0);}
    PII tmp=exgcd(b,a%b);
    return make_pair(-tmp.Y,-tmp.X-(a/b)*tmp.Y);
}
LL modeq(LL a,LL b)//求解ax=1(mod b)
{
    LL x=exgcd(a,b).X;
    return (x%b+b)%b;
```

```
}

int main()
{
    init();
    int T=read();
    while(T--)
    {
        LL a=read(),b=read();
        if(b==1 || (!vis[b] && a%b!=0)){printf("-1 -1 -1 -1\n");continue;}
        else if(!vis[b])
        {
            printf("%lld 1 1 1\n",a/b+1,1,1,1);
            continue;
        }
        else
        {
            LL k1=F[b],f=b/F[b];
            if(f==1)k1=isn[b],f=b/k1;
            LL GCD=gcd(k1,f);
//          printf("%lld %lld\n",k1,f);
            if(a%GCD!=0)
            {
                printf("-1 -1 -1 -1\n");
                continue;
            }
            else
            {
                a/=GCD;k1/=GCD;f/=GCD;
                LL e=modeq(k1,f);
                e=((e*(f-a))%f+f)%f;
//              printf("a:%lld k1:%lld f:%lld %lld\n",a,k1,f,e);
                if(!e)e=f;
                printf("%lld %lld %lld %lld\n",(a+e*k1)/f,k1*GCD,e,f*GCD);
            }
        }
    }
    return 0;
}
```

G.Operating on a Graph

题意

给了一个图，每个点最开始属于自己的编号的组，每次指定一个组，把组里所有点所连向的其他组全部改为指定的组，最后输出每个点属于哪个组。

题解

直接用并查集维护组□set记录组里的边，每次合并用启发式合并就可以过掉了

```
#include<iostream>
#include<cstdio>
#include<algorithm>
#include<cstring>
#include<set>
#define ll long long
using namespace std;
int read()
{
    int k=0,f=1;char c=getchar();
    for(;!isdigit(c);c=getchar()) if(c=='-') f=-1;
    for(;isdigit(c);c=getchar()) k=k*10+c-'0';return k*f;
}
const int N=2000055;
typedef pair<int,int> P;
int T,n,m,q,head[N],nextt[N],to[N],w[N],num[N];
int fa[N],size[N];
set<P> s[N];
P ad[N],now[N];
int find(int x) {return fa[x]==x?x:fa[x]=find(fa[x]);}
int main()
{
    for(T=read();T;T--)
    {
        n=read();m=read();
        for(int i=0;i<n;i++)
            fa[i]=i,size[i]=1,num[i]=i;
        for(int i=1;i<=m;i++)
        {
            int a,b;
            a=read();b=read();
            if(a>b) swap(a,b);
            s[a].insert(P(a,b));
            s[b].insert(P(a,b));
        }
        q=read();
        for(int i=1;i<=q;i++)
        {
            int x=read();
            if(x!=find(x)) continue;
            set<P>::iterator it,itt;
            int k=num[x];
            int cnt=0,tot=0;
            for(it=s[k].begin();it!=s[k].end();it++)
                now[++tot]=*it;
```

```
            for(int j=1;j<=tot;j++)
            {
                k=num[x];
                int y;
                y=find(now[j].first);
                if(y==x) y=find(now[j].second);
                if(y==x) continue;
                int kk=num[y];
                fa[y]=x;
                int cnt=0;
                if(s[k].size()>s[kk].size())
                {
                    for(itt=s[kk].begin();itt!=s[kk].end();itt++)
                    {
                        if(find(itt->first)!=x||find(itt->second)!=x)
                            ad[++cnt]=*itt;
                    }
                }
                else
                {
                    num[x]=kk;
                    for(itt=s[k].begin();itt!=s[k].end();itt++)
                    {
                        if(find(itt->first)!=x||find(itt->second)!=x)
                            ad[++cnt]=*itt;
                    }
                }
                for(int K=1;K<=cnt;K++)
                    s[num[x]].insert(ad[K]);
            }
        }
        for(int i=0;i<n;i++)
            s[i].clear();
        for(int i=0;i<n;i++)
            printf("%d ",find(i));
        puts("");
    }
    return 0;
}
```

L.Problem L is the Only Lovely Problem

签到题

训练实况

12：00 发现**L**签到题

12:05 wxg过**L**，发现**B**也是水题，

12:26 wxg过**B**，fyh，hxm想**A**

12:43 fyh二分条件判错，1wa 过**A**，fyh开写**C**，wxg，hxm讨论**E**

13:12 fyhwa**C**，不知道原因 fyh开想**F**

13:30wxghxm讨论出**E**的错误贪心方法，1wa，开始调**C**

13：55 发现没有问题 只能是eps 调大eps 过**C**

14:20 E继续错误贪心 wa，fyh和hxm分类讨论**F**，之后wxg写**D**

15:28 wxg过**D**

15，35 fyh过**F**，wxg想G，fyh和hxm继续想**E**

16:15 发现之前sb了突然想到**E**可以DP，

16:21 hxm过**E**

16:55 wxg绝杀**G**

训练总结

wxg:这场比赛题目写的多，但是速度还是不行，G和D很早就想出来怎么写，但因为觉得复杂迟迟没有开工，要锻炼一下写细节多的题。

hxm:这场比赛在$$E$$题上拖延了很久,还好wxg及时得出dp的算法，$$F$$题思考上表现出对数论的熟练度还是比较低，同时需要准备充足的模板

fyh:虽然过题数一样，但是相同题数罚时倒数第二，导致排名感人。我在写手性那题的时候交了一发wa就不知道哪里错了，开始看别的题，最后还是队友发现应该是eps开太小了，这既是审题的问题（题目说了六位有效数字）也是对eps理解的问题，应该好好研究一下eps的使用。还有A题二分条件不等号写错了，这些都导致了罚时的无意义增加（至少增加了1:30）