牛客多校第二场

这场彻底跪了。下文贴了很多莫名WA掉的代码们。如果读者能找到到底是哪里WA掉了，麻烦您联系我一下。

D

唯一一道水题。

```
#include<stdio.h>
#include<math.h>

int h1,h2,m1,m2,s1,s2;
int ans=0;

void solve()
{
    scanf("%d:%d:%d",&h1,&m1,&s1);
    scanf("%d:%d:%d",&h2,&m2,&s2);
    ans=abs((h1-h2)*3600+(m1-m2)*60+s1-s2);
}

int main()
{
    solve();
    printf("%d",ans);
    return 0;
}
```

F

这个题TLE掉了。原因是单调队列不熟练，还需要一个记忆化的小技巧。

记忆化技巧代码：

```
int i;
for(i=1;i<=n;i++)
{
    int j;
    for(j=1;j<=m;j++)
    {
        if(!Gcd[i][j])
        {
            int k;
```

```
            for(k=1;k*i<=n&&k*j<=m;k++)
            {
                Gcd[k*i][k*j]=k;
                A[k*i][k*j]=i*j*k;
            }
        }
    }
}
```

单调队列：

```
#include<stdio.h>

int a[5010][5010],b[5010][5010];
int head,tail,q[50100];

int gcd(int a,int b)
{
    return b?gcd(b,a%b):a;
}

int lcm(int a,int b)
{
    return a*b/gcd(a,b);
}

int main()
{
    int n,m,k;
    scanf("%d%d%d",&n,&m,&k);
    long long res=0;
    int i;
    for(i=1;i<=n;i++)
    {
        head=1,tail=0;
        int j;
        for(j=1;j<=m;j++)
        {
            a[i][j]=lcm(i,j);
            while(tail>=head&&j-q[head]>=k)
            {
                head++;
            }
            while(tail>=head&&a[i][j]>a[i][q[tail]])
            {
                tail--;
            }
            q[++tail]=j;
            if(j>=k)
```

```
            {
                b[i][j-k+1]=a[i][q[head]];
            }
        }
    }
    int j;
    for(j=1;j<=m-k+1;j++)
    {
        head=1,tail=0;
        for(i=1;i<=n;i++)
        {
            while(tail>=head&&i-q[head]>=k)
            {
                head++;
            }
            while(tail>=head&&b[i][j]>b[q[tail]][j])
            {
                tail--;
            }
            q[++tail]=i;
            if(i>=k)
            {
                res+=b[q[head]][j];
            }
        }
    }
    printf("%lld\n",res);
    return 0;
}
```

如果能把两者结合起来就好了。

C

我觉得从奇顶点开始DFS一定没错。但是评测机不这么认为……

```
#include<stdio.h>
#include<vector>

using namespace std;

int degree[200010];

vector<int> Adj[200010];
bool vis[200010] = {false};

int flag;
```

```
void DFS(int u)
{
    vis[u]=true;
    if(degree[u]%2==1)
    {
        printf("%d",u);
        (flag==0)?putchar(' '):putchar('\n');
        flag^=1;
    }
    int i;
    for(i=0;i<Adj[u].size();i++)
    {
        int v=Adj[u][i];
        if(vis[v]==false)
        {
            DFS(v);
        }
    }
}

int main()
{
    int n;
    scanf("%d",&n);
    if(n==1)
    {
        printf("1\n1 1\n");
        return 0;
    }
    int m=n-1;
    int x,y;
    int sum=0;
    while(m--)
    {
        scanf("%d%d",&x,&y);
        degree[x]++;
        degree[y]++;
        Adj[x].push_back(y);
        Adj[y].push_back(x);
        if(degree[x]%2==1&&degree[y]%2==1)
        {
            sum++;
        }
        else if(degree[x]%2==0&&degree[y]%2==0)
        {
            sum--;
        }
    }
    printf("%d\n",sum);
    flag=0;
```

```
    int u;
    for(u=0;u<n;u++)
    {
        if(vis[u]==false&&degree[u]%2==1)
        {
            DFS(u);
            break;
        }
    }
}
```

后记：我知道错到哪里了！我一直以为是不重复的覆盖……

其实重复也是可以的，也就是说我硬生生拔高了原题的难度……

唉，怪不得。那么代码就更简单了low的不谈

上面这个代码是不重复覆盖的优秀代码

B

储存斜率用了两种做法。开longlong那个TLE掉了double那个WA了，实在无语……

```
#include<iostream>
#include<vector>
#include<map>

using namespace std;

#define x first
#define y second

vector<pair<double,double> > points;

int maxPoints()
{
    int res = 0;
    for (int i = 0; i < points.size(); ++i)
    {
        map<double,int> m;
        int duplicate = 1;
        for (int j = i + 1; j < points.size(); ++j)
        {
            if (points[i].x== points[j].x&& points[i].y== points[j].y)
            {
                ++duplicate;
                continue;
```

```
            }
            if(points[j].x*points[i].y==points[i].x*points[j].y)
            {
                continue;
            }
            double dx = points[j].x- points[i].x;
            double dy = points[j].y- points[i].y;
            ++m[{dx/dy}];
        }
        res = max(res, duplicate);
        map<double,int>::iterator it;
        for (it = m.begin(); it != m.end(); ++it)
        {
            res = max(res, it->second + duplicate);
        }
    }
    return res;
}

int main()
{
    int n;
    scanf("%d",&n);
    double x,y;
    while(n--)
    {
        cin >> x >> y;
        double temp=x*x+y*y;
        points.push_back(make_pair(x/temp,y/temp));
    }
    cout << maxPoints() << endl;
    return 0 ;
}
```

```
#include<iostream>
#include<vector>
#include<map>

using namespace std;

vector<pair<long long,long long> > points;

long long gcd(long long a,long long b)
{
    return (b == 0) ? a : gcd(b, a % b);
}

int maxPoints()
```

```
{
    int res = 0;
    for (int i = 0; i < points.size(); ++i)
    {
        map<pair<int, int>, int> m;
        int duplicate = 1;
        for (int j = i + 1; j < points.size(); ++j)
        {
            if (points[i].first== points[j].first&& points[i].second==
points[j].second)
            {
                ++duplicate;
                continue;
            }
            long long x1=points[i].first;
            long long x2=points[j].first;
            long long y1=points[i].second;
            long long y2=points[j].second;
            if(x1*y2==y1*x2)
            {
                continue;
            }
            long long dx =x2*(x1*x1+y1*y1)-x1*(x2*x2+y2*y2);
            long long dy =y2*(x1*x1+y1*y1)-y1*(x2*x2+y2*y2);
            long long d = gcd(dx, dy);
            ++m[{dx / d, dy / d}];
        }
        res = max(res, duplicate);
        map<pair<int,int>,int>::iterator it;
        for(it = m.begin(); it != m.end(); ++it)
        {
            res = max(res, it->second + duplicate);
        }
    }
    return res;
}

int main()
{
    int n;
    scanf("%d",&n);
    long long x,y;
    while(n--)
    {
        cin >> x >> y;
        points.push_back(make_pair(x,y));
    }
    cout << maxPoints() << endl;
    return 0 ;
}
```

后记：这种采用了反演的写法要考虑eps——

事实证明，没有通过就是eps的问题。唉，太悲伤了