

[解题报告] POJ3207

[Source] <http://poj.org/problem?id=3207>

[Description]

圆上有 n 个点，编号 $0 \sim n-1$ ($n \leq 1000$)，给出 m ($m \leq 500$) 条边，并且满足每个点的度不超过 1。要求每条边只能画在圆外或圆内，问是否存在使每两条边都不会相交的方案。

[Solution]

2-SAT 判定问题。建图方法：每条边看做两个点 v, v' （一个点表示圆内，另一个表示圆外），如果两条边在圆的同侧时会相交，说明这两个状态之间是矛盾的，因此连边 $v_1 \rightarrow v_2', v_2' \rightarrow v_1, v_1' \rightarrow v_2, v_2 \rightarrow v_1'$ 。然后就是求强连通分量，若存在表示一条边的两个点 v, v' 在同一个强连通分量中，则不存在不相交的方案（因为同一条边即在圆外又在圆内的状态是矛盾的）；否则存在不相交的方案。

[Code]

```
#include<cstdio>
#include<cstring>
#include<algorithm>
#define N 2020
#define M 600000
using namespace std;

struct Edge
{
    int u, v;
};

Edge edge[M];

int
first[N], next[M], end[M], dfn[N], ins[N], low[N], stack[N], color[N];
int top, c, idx, cnt_edge;

void addEdge(int u, int v)
{
    end[cnt_edge] = v;
```

```

        next[cnt_edge]=first[u];
        first[u]=cnt_edge++;
    }

void tarjan(int u)
{
    int i,v;

    stack[++top]=u;
    ins[u]=1;
    dfn[u]=low[u]=++idx;

    for (i=first[u];i;i=next[i])
    {
        v=end[i];

        if (dfn[v]==0)
        {
            tarjan(v);
            if (low[u]>low[v])
                low[u]=low[v];
        }
        else if (ins[v]==1&&low[u]>dfn[v])
            low[u]=dfn[v];
    }

    if (low[u]==dfn[u])
    {
        c++;
        do
        {
            v=stack[top--];
            color[v]=c;
            ins[v]=0;
        }
        while (u!=v);
    }
}

int main()
{
    int n,m,i,j,ans;

    scanf("%d%d",&n,&m);

```

```

memset(first,0,sizeof(first));
memset(dfn,0,sizeof(dfn));
memset(ins,0,sizeof(ins));
    cnt_edge=1;
    top=0;
    c=0;
    idx=0;

    for (i=1;i<=m;i++)
    {
        scanf("%d%d",&edge[i].u,&edge[i].v);
        if (edge[i].u>edge[i].v)
            swap(edge[i].u,edge[i].v);
    }

    for (i=1;i<=m;i++)
        for (j=i+1;j<=m;j++)
            if
(edge[j].u>edge[i].u&&edge[j].u<edge[i].v&&edge[j].v>edge
[i].v||
edge[j].v>edge[i].u&&edge[j].v<edge[i].v&&edge[j].u<edge[
i].u)
            {
                addEdge(i,j+m);
                addEdge(j+m,i);
                addEdge(i+m,j);
                addEdge(j,i+m);
            }

    for (i=1;i<=m+m;i++)
        if (dfn[i]==0)
            tarjan(i);

ans=1;
for (i=1;i<=m;i++)
    if (color[i]==color[i+m])
    {
        ans=0;
        break;
    }

if (ans)

```

```
        printf("panda is telling the truth...\n");
else
    printf("the evil panda is lying again\n");

return 0;
}
```