

# Codeforces Global Round 15

[比赛链接](#)

## C. Maximize the Intersections

### 题意

给定一个圆，圆上有  $2n$  个点，问作  $n$  条弦(所有弦的端点都不相同)最多能有多少个交点。其中有  $k$  弦已经给出，且不考虑三线共点的情况。

### 题解

比赛时候直接蒙了结论，居然对了，补一下证明：

首先，通过手画例子分析，不难发现任意两条弦  $(a,b),(c,d)$  如果不相交，则取  $(a,c)(b,d)$  一定更优。

另外，对于剩下  $2n-2k$  个点，不妨顺时针排序记为  $a_1, a_2 \dots a_{2n-2k}$

易知作  $(a_i, a_{i+n-k})(i=1 \sim n-k)$  是唯一使得  $n-k$  条弦两两相交的方法。

因为弦  $(a_i, a_j)$  如果  $a_j \neq i+n-k$  则  $a_i, a_j$  间的点少于  $n-k-1$  于是  $(a_i, a_j)$  一定不能与其他  $n-k-1$  条弦都相交。

```
const int MAXN=205;
bool vis[MAXN];
vector<pair<int,int> >vec;
vector<int> node;
int main()
{
    int T=read_int();
    while(T--){
        int n=read_int(),k=read_int();
        _for(i,0,n*2)vis[i]=false;
        vec.clear();
        _for(i,0,k){
            int u=read_int()-1,v=read_int()-1;
            vis[u]=vis[v]=true;
            if(u>v)swap(u,v);
            vec.push_back(make_pair(u,v));
        }
        node.clear();
        _for(i,0,n*2)if(!vis[i])
            node.push_back(i);
        _for(i,0,n-k)
            vec.push_back(make_pair(node[i],node[i+n-k]));
    }
}
```

```
    sort(vec.begin(),vec.end());
    int ans=0;
    _for(i,0,n)
    _for(j,i+1,n){
        if(vec[i].second>vec[j].first&&vec[i].second<vec[j].second)
            ans++;
    }
    enter(ans);
}
return 0;
}
```

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

Permanent link: [https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal\\_string:jxm2001:contest:cf\\_global\\_15&rev=1627265118](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:contest:cf_global_15&rev=1627265118) 

Last update: **2021/07/26 10:05**