

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;
}
```

E. Colors and Intervals

题意

给定 $n \times k$ 的序列，第 i 个元素颜色为 c_i 其中 $1 \leq c_i \leq n$ 且每种颜色恰好出现 k 次。

要求选择 n 条线段，其中第 i 条线段左右两端点颜色均为 i 且序列中任意一点最多被 $\lfloor \frac{n}{k-1} \rfloor$ 条线段覆盖。

题解

首先设颜色 i 的出现位置为 $c(i,1), c(i,2) \dots c(i,k)$ 则不难发现最优方案一定是取 $[c(i,j), c(i,j+1)] (1 \leq j < k)$ 的线段。

考虑将所有颜色先按 $c(i,2)$ 排序从小到大排序，取走前 $\lfloor \frac{n}{k-1} \rfloor$ 个颜色的 $[c(i,1), c(i,2)]$ 然后删除该颜色，再依次考虑 $c(i,3) \dots c(i,k)$

这样操作轮数为 $k-1$ 最多可以取走 $(k-1) \lfloor \frac{n}{k-1} \rfloor \geq n$ 种颜色，因此可以保证每个颜色都被选取。

另外第 i 轮操作后考虑当前取走的颜色 a 和剩下的颜色 b 一定有 $c(a,i) < c(b,i)$

然后接下来第 j 轮中取颜色 b 的线段 $[c(b,j-1), c(b,j)]$ 时一定有 $c(a,i) < c(b,i) \leq c(b,j-1)$

所以 $[c(a,i-1), c(a,i)]$ 与 $[c(b,j-1), c(b,j)]$ 一定不相交。进一步，即任意两轮之中取的线段都不相交。

而同一轮的线段最多只有 $\lfloor \frac{n}{k-1} \rfloor$ 条，所以对任意一个点最多覆盖 $\lfloor \frac{n}{k-1} \rfloor$ 次。

于是上述构造满足题目约束，时间复杂度 $O(nk \log n)$

```
const int MAXN=105;
vector<int> c[MAXN];
bool vis[MAXN];
pair<int,int> ans[MAXN];
```

```
int main(){
    int n=read_int(),k=read_int(),bound=(n+k-2)/(k-1);
    _rep(i,1,n*k)
        c[read_int()].push_back(i);
    vector<int> vec;
    _for(i,1,k){
        vec.clear();
        _rep(j,1,n){
            if(!vis[j])
                vec.push_back(j);
        }
        sort(vec.begin(),vec.end(),[&](int a,int b){
            return c[a][i]<c[b][i];
        });
        _for(j,0,min((int)vec.size(),bound)){
            vis[vec[j]]=true;
            ans[vec[j]]=make_pair(c[vec[j]][i-1],c[vec[j]][i]);
        }
    }
    _rep(i,1,n)
        printf("%d %d\n",ans[i].first,ans[i].second);
    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

Last update: 2021/07/26 20:02

