求最小生成树中的最大边。而不是总长度

最小生成树裸题。

思想：从第1个顶点开始，添加与之有边的点；再从这些已添加的边中选择最小权值的继续添加。

用prim/kruskal模板改改就能A。

Prime

#include <cstdio>

#include <cstring>

#include<iostream>

#include<cmath>

#include<algorithm>

using namespace std;

const int INF=1000000007;

bool used[510];

int cost[510][510];

int n;

int mincost[510];//当前点到最小生成树集合中的点的最近距离

int prime()

{

for(int i=1;i<=n;i++)

{

mincost[i]=INF;

used[i]=0;

}

mincost[1]=0;

int ans=0;

while(1)

{

int v=-1;

for(int u=1;u<=n;u++)

{

if(used[u]==0 && (v==-1 || mincost[u]<mincost[v]))

v=u;

}

if(v==-1)break;

used[v]=1;

ans=max(ans,mincost[v]);

for(int u=2;u<=n;u++)

mincost[u]=min(mincost[u],cost[u][v]);

}

return ans;

}

int main()

{

//freopen("input.txt","r",stdin);

int T;

scanf("%d",&T);

while(T--)

{

scanf("%d",&n);

for(int i=1;i<=n;i++)

for(int j=1;j<=n;j++)

scanf("%d",&cost[i][j]);

printf("%d\n",prime());

}

return 0;

}

Kruskal

#include <cstdio>

#include <cstring>

#include<iostream>

#include<cmath>

#include<algorithm>

using namespace std;

const int INF=1000000007;

int n;

const int MAXN=510;//最大点数

const int MAXM=500010;//最大边数 int F[MAXN];//并查集使用

int F[MAXN];

int a[MAXN][MAXN];

struct Edge

{

int u,v,w;

} edge[MAXM]; //存储边的信息，包括起点/终点/权值

int tol(0);//边数，加边前赋值为0

void addedge(int u,int v,int w)

{

edge[tol].u=u;

edge[tol].v=v;

edge[tol++].w=w;

}

bool cmp(Edge a,Edge b)

{

//排序函数，讲边按照权值从小到大排序

return a.w<b.w;

}

int find(int x)

{

if(F[x]==-1)return x;

else return F[x]=find(F[x]);

}

int Kruskal(int n)//传入点数，返回最小生成树的权值，如果不连通返回-1

{

memset(F,-1,sizeof(F));

sort(edge,edge+tol,cmp);

int cnt=0;//计算加入的边数

int ans=0;

for(int i=0; i<tol; i++)

{

int u=edge[i].u;

int v=edge[i].v;

int w=edge[i].w;

int t1=find(u);

int t2=find(v);

if(t1!=t2)

{

ans=max(ans,w);

F[t1]=t2;

cnt++;

}

if(cnt==n-1)break;

}

if(cnt<n-1)return 0;//不连通

else return ans;

}

int main()

{

//freopen("input.txt","r",stdin);

int T;

scanf("%d",&T);

while(T--)

{

scanf("%d",&n);

for (int i=1; i<=n; i++)

for (int j=1; j<=n; j++)

scanf("%d",&a[i][j]);

int x;

tol=0;

for(int i=1;i<=n;i++)

for(int j=i+1;j<=n;j++)

addedge(i,j,a[i][j]);

for(int i=0;i<tol;i++)

F[i]=i;

printf("%d\n",Kruskal(n));

}

return 0;

}