

C5 A 小水獭和最长最短路

21373078 李正扬

题意

- 给定 n 个点 m 条边的无向联通图，求所有节点之间最短路的 \max 值，输出所有两点间最短路等于 \max 值的点对。
- 时间限制：2000ms
- 数据范围： $1 \leq t \leq 10$ $1 \leq n \leq 300$ $n-1 \leq m \leq n*(n-1)/2$
 - 边数最多情况为完全图（稠密图）

思路

- 全源最短路问题
- Floyd模板题 $O(n^3)$ 451ms
- 核心代码:

```
for(int k=1; k<=n; k++)
{
    for(int i=1; i<=n; i++)
    {
        for(int j=1; j<=n; j++)
        {
            if(dist[i][k]!=INF&&dist[k][j]!=INF)
                dist[i][j]=min(dist[i][j],dist[i][k]+dist[k][j]);
        }
    }
}
```

- 穷举*i,j*间的所有断点并更新两点距离

思路

- 同样可以使用 n 次Dijkstra求全源最短路
用 $n \times n$ 二维数组保存距离 时间复杂度 $O(n^3)$ 1796ms
- 但不可使用堆优化 稠密图中时间复杂度 $O(n^3 \log n)$ TLE
- 求出全源最短路后 三次 n^2 循环
 - 1 找出最长距离 maxdist
 - 2 统计路径数 num
 - 3 输出对应点对 print(i,j)

代码

```
#include <bits/stdc++.h>
const int INF = INT_MAX;
using namespace std;
const int maxn=510,maxm=90010;
int n,m,g[maxn][maxn];
int dist[maxn][maxn];
bool visd[maxn][maxn];
void floyd()
{
    for(int i=1; i<=n; i++)
        for(int j=1; j<=n; j++)
            dist[i][j]=g[i][j];
    for(int k=1; k<=n; k++)
    {
        for(int i=1; i<=n; i++)
        {
            for(int j=1; j<=n; j++)
            {
                if(dist[i][k]!=INF&&dist[k][j]!=INF)
                    dist[i][j]=min(dist[i][j],dist[i][k]+dist[k][j]);
            }
        }
    }
}
```

```
int main ()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        scanf("%d%d",&n,&m);
        for(int i=1; i<=n; i++)
            for(int j=1; j<=n; j++)
            {
                if (i==j) g[i][j] = 0;
                else g[i][j]=INF;
            }
        for(int i=1; i<=m; i++)
        {
            int u,v,x;
            scanf("%d%d%d",&u,&v,&x);
            g[u][v] = min(g[u][v], x);
            g[v][u] = min(g[v][u], x);
        }
        floyd();
        int maxdist = -0x3f3f3f3f;
        int num=0;
        for(int i=1;i<=n;i++)
        {
            for(int j=1;j<=n;j++)
            {
                if(dist[i][j]<INF&&maxdist<dist[i][j]&&dist[i][j]!=0)
                {
                    maxdist = dist[i][j];
                }
            }
        }
        for(int i=1;i<=n;i++)
        {
            for(int j=1;j<=n;j++)
            {
                if(maxdist==dist[i][j])
                {
                    num++;
                }
            }
        }
        printf("%d %d\n",maxdist,num);
        for(int i=1;i<=n;i++)
        {
            for(int j=1;j<=n;j++)
            {
                if(maxdist==dist[i][j])
                {
                    printf("%d %d\n",i,j);
                }
            }
        }
    }
    return 0;
}
```