

[解题报告] ZOJ1587

[Source]

<http://acm.zju.edu.cn/onlinejudge/showProblem.do?problemCode=1857>

[Description]

无向图， $n(n \leq 500)$ 个顶点，每个顶点都有村庄，有些顶点上有消防站（一个顶点上可以有多个消防站），共有 $m(m \leq 100)$ 个消防站。图的特点是每个顶点最多与 20 条边相连。定义村庄的满意度为每个村庄到最近消防站的距离的最大值。现在要新建一个消防站，使得村庄的满意度最小，若有多组最优解，输出顶点编号最小者。

[Solution]

先求出每个顶点到最近消防站的距离，即要求每对顶点间的最短路，注意到每个顶点的度不超过 20，可以用邻接表形式的 floyd 或者 spfa。然后从小到大枚举每个顶点，找到建立新的消防站的最优顶点。这道题注意是多组数据，数据读入的方式不太好处理，另外需要注意顶点上可以重复建消防站，因此当所有点都有消防站时应输出 1。

[Code]

```
#include<cstdio>
#include<cstring>
#include<queue>
#include<algorithm>
#define N 600
#define M 400000
#define oo 0x3fffffff
using namespace std;

int map[N][N], num[N], w[N], tw[N];
int first[N], next[M], end[M], dis[M];
int cnt_edge;
char str[N], flag[N];
queue<int>que;

void addEdge(int u, int v, int w)
```

```

{
    end[cnt_edge]=v;
    dis[cnt_edge]=w;
    next[cnt_edge]=first[u];
    first[u]=cnt_edge++;
}

void spfa(int s,int n)
{
    int i,j,u,v,w;

    for (i=1;i<=n;i++)
        map[s][i]=oo;
    memset(flag,0,sizeof(flag));
    map[s][s]=0;
    flag[s]=1;
    que.push(s);

    while (!que.empty())
    {
        u=que.front();
        que.pop();
        flag[u]=0;

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

            if (map[s][v]>map[s][u]+w)
            {
                map[s][v]=map[s][u]+w;

                if (flag[v]==0)
                {
                    flag[v]=1;
                    que.push(v);
                }
            }
        }
    }
}

```

```

int main()
{
    int n,m,u,v,d,i,j,k,tmp,lon,ans;

    while (scanf("%d%d",&m,&n)!=EOF)
    {
        memset(first,0,sizeof(first));
        cnt_edge=1;

        for (i=0;i<m;i++)
            scanf("%d",&num[i]);

        getchar();
        while (gets(str)&&str[0])
        {
            sscanf(str,"%d%d%d",&u,&v,&d);
            addEdge(u,v,d);
            addEdge(v,u,d);
        }

        for (i=1;i<=n;i++)
            spfa(i,n);

        for (i=1;i<=n;i++)
            w[i]=oo;

        for (i=1;i<=n;i++)
            for (j=0;j<m;j++)
                w[i]=min(w[i],map[i][num[j]]);

        ans=0;
        lon=oo;
        for (i=1;i<=n;i++)
        {
            for (j=1;j<=n;j++)
                tw[j]=min(w[j],map[j][i]);

            tmp=0;
            for (j=1;j<=n;j++)
                tmp=max(tmp,tw[j]);
        }
    }
}

```

```
        if (lon>tmp)
        {
            ans=i;
            lon=tmp;
        }

    printf("%d\n",ans);
}

return 0;
}
```