

acm之城市平乱

闲聊

由于我们专业数据结构没有学完所以我的图这一部分的知识很少,由于要比赛了选择了图论这一部分没有办法只有看了.(虽然我队友会)今天第一次去看Dijkstra算法.

题目信息

城市平乱

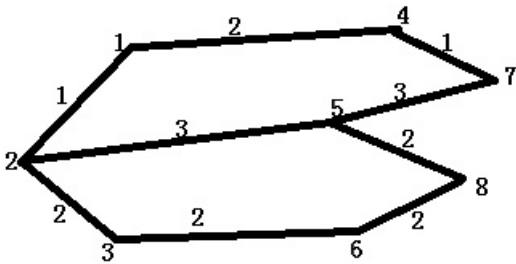
描述

南将军统领着N个部队，这N个部队分别驻扎在N个不同的城市。

他在用这N个部队维护着M个城市的治安，这M个城市分别编号从1到M。

现在，小工军师告诉南将军，第K号城市发生了暴乱，南将军从各个部队都派遣了一个分队沿最近路去往暴乱城市平乱。

现在已知在任意两个城市之间的路行军所需的时间，你作为南将军麾下最厉害的程序员，请你编写一个程序来告诉南将军第一个分队到达叛乱城市所需的时间。



- 注意，两个城市之间可能不只一条路。

输入

第一行输入一个整数T，表示测试数据的组数。(T<20)

每组测试数据的第一行是四个整数N,M,P,Q(1<=N<=100,N<=M<=1000,M-1<=P<=100000)其中N表示部队数，M表示城市数，P表示城市之间的路的条数,Q表示发生暴乱的城市编号。

随后的一行是N个整数，表示部队所在城市的编号。

再之后的P行，每行有三个正整数，a,b,t(1<=a,b<=M,1<=t<=100)，表示a,b之间的路如果行军需要用时为t

数据保证暴乱的城市是可达的。

输出

对于每组测试数据，输出第一支部队到达叛乱城市时的时间。每组输出占一行

样例输入

```
1
3 8 9 8
1 2 3
1 2 1
2 3 2
1 4 2
2 5 3
3 6 2
```

```
4 7 1
5 7 3
5 8 2
6 8 2
```

样例输出

```
4
```

AC代码

```
#include <iostream>
#include <string.h>

#define INF 0x3f3f3f3f
using namespace std;

int shib[105];
int chens[1005][1005];
int dist[1005];
int N,M,P,Q;

bool pan(int a)
{
    for(int i=0;i<N;i++)
    {
        if(shib[i]==a)
            return true;
    }
    return false;
}

void Dijkstra()
{
    int s[10005];
    int mindis,u;
    for(int i=0;i<M;i++)
    {
        dist[i]=chens[Q][i];
        s[i]=0;
    }
    s[Q]=1;
    for(int i=0;i<M;i++)
    {
        mindis = INF;
        for(int j=0;j<M;j++)
        {
            if(s[j]==0&&dist[j]<mindis)
            {
                u=j;
                mindis = dist[j];
            }
        }
        s[u]=1;
        for(int j=0;j<M;j++)
        {
            if(s[j]==0)
            {
```

```

        if(chens[u][j]<INF&&dist[u]+chens[u][j]<dist[j])
        {
            dist[j]=dist[u]+chens[u][j];
        }
    }
}
}
int main()
{
    int T;
    cin>>T;
    while(T--)
    {
        memset(chens,0x3f3f3f3f,sizeof(chens));
        memset(shib,-1,sizeof(shib));
        cin>>N>>M>>P>>Q;
        for(int i=0;i<N;i++)
        {
            cin>>shib[i];
        }
        for(int i=0;i<P;i++)
        {
            int a,b,t;
            cin>>a>>b>>t;

            chens[a][b]=min(t,chens[a][b]);
            chens[b][a]=min(t,chens[b][a]);
        }
        if(pan(Q))
            cout<<0<<endl;
        else
        {
            Dijkstra();
        }
        int mm=INF;
        for(int i=0;i<M;i++)
            if(pan(i))
            {
                mm=min(mm,dist[i]);
            }
            cout<<mm<<endl;
        }
        return 0;
    }
}

```

作者语

这个题是我套模板的