移动服务

一. 考察内容：

动态规划

二. 题目分析：

[题目大意]

3个人执行m个命令，每个命令有指向一个位置，三个人要有一个人从原来的位置移动到命令指向的位置响应命令，每个移动有不同的花费，求完成所有命令的最小花费。

[写题思路]

设状态f[i][j][k]表示第i个指令之后，第一个人在j，第二个人在k，（当然，另一个人执行了i指令，在c[i]）。

考虑转移：对于任意一个状态，我们都可以让三个人中的任意一个人响应下一个状态，如果第一个人响应下一个命令，那么f[i+1][k][c[i]]=f[i][j][k]+v[j][c[i+1]]；如果第二个人响应下一个命令，那么f[i+1][j][c[i]]=f[i][j][k]+v[k][c[i+1]]；如果让第三个人响应下一个命令，那么f[i+1][j][c[i]]=f[i][j][k]+v[c[i]][c[i+1]]，将这三种情况取最小，即是处理前i个命令的最少花费。

由于题目内存限制，考虑将第一维用滚动数组实现。

三. 代码实现：

```
#define _CRT_SECURE_NO_DEPRECATE
/************************
*创建时间：2018 08 20
*文件类型：源代码文件
*题目来源：
*当前状态：
*备忘录：
*作者：HtBest
************************/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
// #include <sys/wait.h>
// #include <sys/types.h>
// #include <unistd.h>
using namespace std;
#define MAXN 201
#define MAXM 1001
int n,m,a[MAXN][MAXN],c[MAXM],f[2][MAXN][MAXN];
/* Variable explain:

*/
void read()
{
    freopen("service.in","r",stdin);
    freopen("service.out","w",stdout);
    scanf("%d%d",&n,&m);
    for(int i=1;i<=n;++i)for(int j=1;j<=n;++j)scanf("%d",&a[i][j]);
    for(int i=1;i<=m;++i)scanf("%d",&c[i]);
```

```
    return;
}
void dp()
{
    int T=0;
    //设状态f[i][j][k]为前i个操作后另外两个人在j和k时的最小花费(第3个人在b[i])
    for(int i=1;i<=n;++i)for(int j=i+1;j<=n;++j)f[0][i][j]=1e9;
    c[0]=3;
    f[0][1][2]=0;
    for(int i=0;i<m;++i)
    {
        int l=c[i];//第三个人在l
        if(l==c[i+1])continue;//不变
        for(int j=1;j<=n;++j)for(int k=j+1;k<=n;++k)f[T^1][j][k]=1e9;
        for(int j=1;j<=n;++j)
        {
            if(j==l)continue;
            for(int k=j+1;k<=n;++k)//第二个人要比第一个人靠后
            {
                if(k==l)continue;
                if(c[i+1]!=k&&c[i+1]!=l)f[T^1][min(k,l)][max(k,l)]=min(f[T^1][min(k,l)]
[max(k,l)],f[T][j][k]+a[j][c[i+1]]);//让第一个人去响应i+1请求
                if(c[i+1]!=j&&c[i+1]!=l)f[T^1][min(j,l)][max(j,l)]=min(f[T^1][min(j,l)]
[max(j,l)],f[T][j][k]+a[k][c[i+1]]);//让第二个人去响应i+1请求
                if(c[i+1]!=j&&c[i+1]!=k)f[T^1][j][k]=min(f[T^1][j][k],f[T][j][k]+a[l]
[c[i+1]]);//让第三个人去响应i+1请求
            }
        }
        T^=1;
    }
    int ans=1e9;
    for(int i=1;i<=n;++i)for(int j=i+1;j<=n;++j)ans=min(ans,f[T][i][j]);
    printf("%d\n",ans);
}
int main()
{
    read();
    dp();
    return 0;
}
```

<题目跳转> <查看代码>