

分类

- 最短路
- dp

题意

- 无向图，m个结点，单位长度花费为1
- 货物从1号点运到m号点，需要n天运完
- 某些节点在某些日子不能用，所以可能需要修改路线，但修改路线花费代价k
- 制定一个n天的计划，使得总花费最小，总花费=n天运输路线长度之和+k*改变运输路线的次数
- $1 \leq n \leq 100$
 $1 \leq m \leq 20$

思路

1. 预处理broken[c][i] 第c个结点第i天能不能用,不能用为1
2. 预处理cost[i][j]表示第i天到第j天走同一条路线（不改变路线）的花费 ——最短路*天数
标记出不能走的点 (x from i to j : flag[c] != broken[c][x]),做SPFA求cost[i][j]
3. f[x]表示第1天到第x天的总花费
初始状态 $f[x] = \text{cost}[1][x]$
状态转移 y from 1 to x-1 : $f[x] = f[y] + \text{cost}[y+1][x] + k$
最终答案为f[n]

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  typedef long long LL;
4  const int INF = 99999999;
5  int n,m,k,e;
6  int maps[25][25],broken[25][105],cost[105]
   [105],flag[25],vis[25],dis[25],f[105];
7  queue<int>q;
8  int spfa()
9  {
10     for(int i=2;i<=m;i++)dis[i] =INF;
11     dis[1] = 0;
```

```

12     memset(vis,0,sizeof(vis));
13     vis[1] = 1;
14     while(!q.empty()) q.pop();
15     q.push(1);
16     while(!q.empty())
17     {
18         int now = q.front();
19         q.pop();
20         vis[now] = 0;
21         for(int i=1;i<=m;i++)
22         {
23             if(i==now) continue;
24             if(flag[i]||maps[now][i]==0x7f) continue;
25             if(dis[i]>dis[now]+maps[now][i])
26             {
27                 dis[i]= dis[now]+maps[now][i];
28                 if(!vis[i])
29                 {
30                     vis[i] = 1;
31                     q.push(i);
32                 }
33             }
34         }
35     }
36     return dis[m];
37 }
38 int main()
39 {
40     ios::sync_with_stdio(false);
41     cin>>n>>m>>k>>e;
42     int x,y,c;
43     memset(maps,0x7f,sizeof(maps));
44     memset(broken,0,sizeof(broken));
45     for(int i=1;i<=m;i++) maps[i][i] = 0;
46     for(int i=1;i<=e;i++)
47     {
48         cin>>x>>y>>c;
49         maps[x][y] = c;
50         maps[y][x] = c;
51     }
52     int d;
53     cin>>d;
54     for(int i=1;i<=d;i++)
55     {
56         cin>>c>>x>>y;
57         for(int j=x;j<=y;j++)

```

```
58     {
59         broken[c][j] = 1;
60     }
61 }
62 for(int i=1;i<=n;i++)
63 {
64     for(int j=i;j<=n;j++)
65     {
66         memset(flag,0,sizeof(flag));
67         for(int k=1;k<=m;k++)
68             for(int l=i;l<=j;l++)
69             {
70                 flag[k] |= broken[k][l];
71             }
72         cost[i][j] = spfa();
73         if(cost[i][j]<INF)
74             cost[i][j] = cost[i][j]*(j-i+1);
75     }
76 }
77 for(int i = 1;i<=n;i++)
78 {
79     f[i] = cost[1][i];
80 }
81 for(int i=2;i<=n;i++)
82 {
83     for(int j=1;j<i;j++)
84     {
85         f[i] = min(f[i],f[j]+cost[j+1][i]+k);
86     }
87 }
88 cout<<f[n]<<endl;
89
90 }
```