

原创 [kuangbin带你飞]专题四 最短路练习

2019-07-27 18:26:54 _Y-_Y_ 阅读数 23 文章标签: 最短路 更多

版权声明: 本文为博主原创文章, 遵循 CC 4.0 BY-SA 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_44410512/article/details/97535271

```

1  /*
2  0 or 1
3  HDU - 4370
4  https://cn.vjudge.net/problem/HDU-4370#author=0
5  题意: 给你一个n*n矩阵, 让你新建一个只有0和1的n*n矩阵, 满足新建的这个矩阵与原来那个矩阵对应位置相乘最小,
6      新建的这个矩阵满足一下条件
7      1.X12+X13+...X1n=1
8      2..X1n+X2n+...Xn-1n=1
9      3.ΣXki =ΣXij
10 输入: n 和 n*n的矩阵
11 样例输入:
12      4
13      1 2 4 10
14      2 0 1 1
15      2 2 0 5
16      6 3 1 2
17 样例输出:
18      3
19 思路:
20 1.X12+X13+...X1n=1 于是1号节点的出度为1
21 2..X1n+X2n+...Xn-1n=1 于是n号节点的入度为1
22 3.ΣXki =ΣXij 于是2~n-1号节点的入度必须等于出度
23 所以建图, 跑最短路, 或者是最小环
24 */
25 #include <cstdio>
26 #include <iostream>
27 #include <cstring>
28 #include <algorithm>
29 #include <queue>
30 using namespace std;
31 #define maxn 310
32 #define INF 0x3f3f3f3f
33 int mp[maxn][maxn];
34 int dis[maxn];
35 int n;
36 void spfa(int st){
37     queue<int> q;
38     for(int i=1;i<=n;i++){
39         if(i==st){
40             dis[i]=INF;
41         }else{
42             dis[i]=mp[st][i];
43             q.push(i);
44         }
45     }
46     while(!q.empty()){
47         int u=q.front();
48         q.pop();
49         for(int i=1;i<=n;i++){
50             if(dis[i]>dis[u]+mp[u][i]){
51                 dis[i]=dis[u]+mp[u][i];
52                 q.push(i);
53             }
54         }
55     }
56 }
57 int main(){
58     while(~scanf("%d", &n)){
59         for(int i=1;i<=n;i++){

```

```

60     for(int j=1;j<=n;j++){
61         scanf("%d", &mp[i][j]);
62     }
63 }
64 spfa(1);
65 int ans=dis[n];
66 int c1=dis[1];
67 spfa(n);
68 int c2=dis[n];
69 printf("%d\n", min(ans,c1+c2));
70 }
71 return 0;
72 }
73

```

```

1  /*
2  Arbitrage
3  POJ - 2240
4  https://cn.vjudge.net/problem/POJ-2240#author=0
5  题意： 通过货币汇率的差异来赚钱
6  输入： 货币种类的数目n
7         下面n行是货币名称
8         货币交换的数目t
9         下面t行是 u -> v w
10        n为0时结束
11 样例输入：
12      3
13      USDoLLar
14      BritishPound
15      FrenchFranc
16      3
17      USDoLLar 0.5 BritishPound
18      BritishPound 10.0 FrenchFranc
19      FrenchFranc 0.21 USDoLLar
20
21      3
22      USDoLLar
23      BritishPound
24      FrenchFranc
25      6
26      USDoLLar 0.5 BritishPound
27      USDoLLar 4.9 FrenchFranc
28      BritishPound 10.0 FrenchFranc
29      BritishPound 1.99 USDoLLar
30      FrenchFranc 0.09 BritishPound
31      FrenchFranc 0.19 USDoLLar
32
33      0
34 样例输出：
35      Case 1: Yes
36      Case 2: N
37 解法：
38 map 映射 + spfa
39 */
40 #include <cstring>
41 #include <cstdio>
42 #include <algorithm>
43 #include <queue>
44 #include <map>
45 #include <iostream>
46 using namespace std;
47 #define maxn 100000
48 #define INF 0x3f3f3f
49 map<string,int> mp;
50 struct node{
51     int u,v;
52     double w;
53     int next;
54 }edge[maxn];
55 int head[maxn];

```

```

56 double dis[maxn];
57 string a,b;
58 int cnt,n,t;
59 int B=1;
60 void init(){
61     memset(dis,0,sizeof dis);
62     memset(head,-1,sizeof head);
63     cnt=0;
64 }
65 void add(int u,int v,double w){
66     edge[cnt].w=w;
67     edge[cnt].u=u;
68     edge[cnt].v=v;
69     edge[cnt].next=head[u];
70     head[u]=cnt++;
71 }
72 bool spfa(int B){
73     dis[B]=1;
74     queue<int> q;
75     q.push(B);
76     while(!q.empty()){
77         int u=q.front();
78         q.pop();
79         for(int i=head[u];~i;i=edge[i].next){
80             if(dis[edge[i].v]<dis[u]*edge[i].w){
81                 dis[edge[i].v]=dis[u]*edge[i].w;
82                 if(dis[B]>1) return true;
83                 q.push(edge[i].v);
84             }
85         }
86     }
87     return false;
88 }
89 int main(){
90     int T=1;
91     while(~scanf("%d", &n)&&n){
92         init();
93         for(int i=1;i<=n;i++){
94             cin>>a;
95             mp[a]=i;
96         }
97         scanf("%d", &t);
98         double w;
99         for(int i=0;i<t;i++){
100             cin>>a>>w>>b;
101             add(mp[a],mp[b],w);
102         }
103         printf("Case %d: ", T++);
104         if(spfa(B)) printf("Yes\n");
105         else printf("No\n");
106     }
107 }
108

```

```

1  /*
2  Candies
3  POJ - 3159
4  https://cn.vjudge.net/problem/POJ-3159#author=xym000
5  题意: 求最大的差异
6  输入: n个孩子, t个关系
7         a相信b永远不会比他自己多c个糖果
8  样例输入:
9         2 2
10        1 2 5
11        2 1 4
12  样例输出:
13        5
14  解法:
15  让第一个孩子拿到的糖果为0, 然后跑一边spfa, 保证所有的孩子能满足他们之间的关系, 然后输出dis[n]
16  注意:

```

```
17 后台数据卡queue但是不卡stack
18 */
19 #include <cstring>
20 #include <cstdio>
21 #include <iostream>
22 #include <algorithm>
23 #include <stack>
24 #include <queue>
25 using namespace std;
26 #define maxn 30010
27 #define maxm 150010
28 #define INF 0x3f3f3f
29 struct node{
30     int u,v,w;
31     int next;
32 }edge[maxm];
33 int head[maxn];
34 int dis[maxn];
35 bool color[maxn];
36 int cnt,n,t;
37 void init(){
38     memset(head,-1,sizeof head);
39     memset(dis,INF,sizeof dis);
40     cnt=0;
41 }
42 void add(int u,int v,int w){
43     edge[cnt].u=u;
44     edge[cnt].v=v;
45     edge[cnt].w=w;
46     edge[cnt].next=head[u];
47     head[u]=cnt++;
48 }
49 void spfa(){
50     memset(color,true,sizeof color);
51     stack<int> q;
52     dis[1]=0;
53     q.push(1);
54     color[1]=false;
55     while(!q.empty()){
56         int u=q.top();
57         q.pop();
58         color[u]=true;
59         for(int i=head[u];~i;i=edge[i].next){
60             if(dis[edge[i].v]>dis[u]+edge[i].w){
61                 dis[edge[i].v]=dis[u]+edge[i].w;
62                 if(color[edge[i].v]){
63                     q.push(edge[i].v);
64                     color[edge[i].v]=false;
65                 }
66             }
67         }
68     }
69     return;
70 }
71 int main(){
72     while(~scanf("%d %d", &n, &t)){
73         init();
74         int u,v,w;
75         for(int i=0;i<t;i++){
76             scanf("%d %d %d", &u, &v, &w);
77             add(u,v,w);
78         }
79         spfa();
80         printf("%d\n", dis[n]);
81     }
82     return 0;
83 }
84
```

```

1  /*
2  Cow Contest
3  POJ - 3660
4  https://cn.vjudge.net/problem/POJ-3660#author=freeLoop
5  题意：从n的同学参加比赛,有m条实力信息,问最后有多少名同学可以确定他的排名,保证合法
6  输入：n    m
7         A > B
8  样例输入：
9         5 5
10        4 3
11        4 2
12        3 2
13        1 2
14        2 5
15  样例输出：
16        2
17  解法：
18  (Floyed-Warshall) 传递闭包
19  */
20  #include <cstdio>
21  #include <cstring>
22  #include <iostream>
23  #include <algorithm>
24  #include <queue>
25  using namespace std;
26  #define maxn 110
27  #define INF 0x3f3f3f
28  int mp[maxn][maxn];
29  int n,t;
30  void init(){
31      memset(mp,0,sizeof mp);
32  }
33  int main(){
34      scanf("%d %d", &n, &t);
35      init();
36      int u,v;
37      for(int i=0;i<t;i++){
38          scanf("%d %d", &u, &v);
39          mp[u][v]=1;
40      }
41      for(int k=1;k<=n;k++){
42          for(int i=1;i<=n;i++){
43              for(int j=1;j<=n;j++){
44                  if(mp[i][k]&&mp[k][j]) mp[i][j]=1;
45              }
46          }
47      }
48      int ans=0;
49      for(int i=1;i<=n;i++){
50          int j;
51          for(j=1;j<=n;j++){
52              if(i==j) continue;
53              if(!mp[i][j]&&!mp[j][i]) break;
54          }
55          if(j>n) ans++;
56      }
57      printf("%d\n", ans);
58      return 0;
59  }
60

```

```

1  /*
2  Currency Exchange
3  POJ - 1860
4  https://cn.vjudge.net/problem/POJ-1860
5  题意：兑换货币
6         每个地方倒每个地方的兑换汇率和佣金都不一样
7         他要是回到出发点时自己的钱增加了,就输出YES 否NO
8  输入：
9         第一行是  : 点个数,边个数,出发点,开始的钱

```

10 下面m行是：从u->v u->v的汇率 u->v的佣金 v->u的汇率 v->u的佣金

11 样例输入：

12 3 2 1 20.0

13 1 2 1.00 1.00 1.00 1.00

14 2 3 1.10 1.00 1.10 1.00

15 样例输出：

16 YES

17 解法：

18 直接跑spfa要是发现开始的价值增加了，那就是YES 否则NO

19 */

20 #include<cstdio>

21 #include<cstring>

22 #include<algorithm>

23 #include<queue>

24 #include<iostream>

25 using namespace std;

26 #define INF 0x3f3f3f

27 #define maxn 10010

28 struct node{

29 double wa, a;

30 int e;

31 int next;

32 }edge[maxn];

33 int cnt;

34 int head[maxn];

35 int n,t,lei;

36 double num;

37 void init(){

38 memset(head,0,sizeof head);

39 cnt=1;

40 }

41 void add(int u,int v,double a, double wa){

42 edge[cnt].wa=wa;

43 edge[cnt].a=a;

44 edge[cnt].e=v;

45 edge[cnt].next=head[u];

46 head[u]=cnt++;

47 }

48 bool spfa(int sb){

49 queue<int> q;

50 bool color[maxn];

51 double d[maxn];

52 memset(d,0,sizeof d);

53 memset(color,true,sizeof color);

54 q.push(sb);

55 d[sb]=num;

56 color[sb]=false;

57 while(!q.empty()){

58 int st=q.front();

59 q.pop();

60 color[st]=true;

61 for(int i=head[st];i!=0;i=edge[i].next){

62 if((d[st]-edge[i].wa)*edge[i].a>d[edge[i].e]){

63 d[edge[i].e]=(d[st]-edge[i].wa)*edge[i].a;

64 if(d[sb]>num) return true;

65 if(color[edge[i].e]){

66 q.push(edge[i].e);

67 color[edge[i].e]=false;

68 }

69 }

70 }

71 }

72 return false;

73 }

74 int main(){

75 scanf("%d %d %d %lf", &n, &t, &lei, &num);

76 init();

77 int u,v;

78 double a,b,wa,wb;

79 for(int i=0;i<t;i++){

80 scanf("%d %d %lf %lf %lf %lf", &u, &v, &a, &wa, &b, &wb);

```

81     add(u, v, a, wa);
82     add(v, u, b, wb);
83 }
84 if(spfa(lei)) printf("YES\n");
85 else printf("NO\n");
86 return 0;
87 }
88

```

```

1  /*
2  Extended Traffic
3  LightOJ - 1074
4  https://cn.vjudge.net/problem/LightOJ-1074
5  题意: 每条街道的拥挤度P(x), 街a到街b的时间是(p(b)-p(a))^3, 找最短路, 若无法到达或结果小于3, 输出?
6  输入: T组样例
7      城市个数
8      1到n的繁荣度
9      城市中的道路数量
10     单向 u->v
11     q个查询
12     1到qi
13 样例输入:
14      2
15      5
16      6 7 8 9 10
17      6
18      1 2
19      2 3
20      3 4
21      1 5
22      5 4
23      4 5
24      2
25      4
26      5
27      2
28      10 10
29      1
30      1 2
31      1
32      2
33 样例输出:
34      Case 1:
35      3
36      4
37      Case 2:
38      ?
39 解法:
40 SPFA判负环
41 */
42
43 #include <cstring>
44 #include <cstdio>
45 #include <algorithm>
46 #include <queue>
47 #include <iostream>
48 #include <cmath>
49 using namespace std;
50 #define maxt 1000000
51 #define maxn 1000
52 #define INF 0x3f3f3f3f
53 struct node{
54     int u,v,w;
55     int next;
56 }edge[maxt];
57 int head[maxn];
58 int dis[maxn];
59 int d[maxn];
60 int color[maxn];
61 int cnt,n,t;

```

```

62 void init(){
63     memset(head,-1,sizeof head);
64     memset(dis,INF,sizeof dis);
65     memset(color,0,sizeof color);
66     memset(edge,0,sizeof edge);
67     memset(d,0,sizeof d);
68     cnt=0;
69 }
70 void add(int u,int v,int w){
71     edge[cnt].u=u;
72     edge[cnt].v=v;
73     edge[cnt].w=w;
74     edge[cnt].next=head[u];
75     head[u]=cnt++;
76 }
77 bool spfa(){
78     queue<int> q;
79     dis[1]=0;
80     q.push(1);
81     color[1]++;
82     while(!q.empty()){
83         int u=q.front();
84         q.pop();
85         for(int i=head[u];~i;i=edge[i].next){
86             if(dis[edge[i].v]>dis[u]+edge[i].w){
87                 dis[edge[i].v]=dis[u]+edge[i].w;
88                 q.push(edge[i].v);
89                 if(++color[edge[i].v]>=n) return false;
90             }
91         }
92     }
93     return true;
94 }
95 int main(){
96     int T;
97     int q,u,v,w;
98     scanf("%d",&T);
99     for(int s=1;s<=T;s++){
100         init();
101         scanf("%d",&n);
102         for(int i=1;i<=n;i++){
103             scanf("%d",&d[i]);
104         }
105         scanf("%d",&t);
106         for(int i=0;i<t;i++){
107             scanf("%d %d",&u,&v);
108             add(u,v,(d[v]-d[u])*(d[v]-d[u])*(d[v]-d[u]));
109         }
110         bool ll=spfa();
111         scanf("%d",&q);
112         printf("Case %d:\n",s);
113         while(q--){
114             scanf("%d",&w);
115             int ans=dis[w];
116             if(ans<3||ans==INF) printf("? \n");
117             else printf("%d \n",dis[w]);
118         }
119     }
120     return 0;
121 }
122

```

```

1  /*
2  Frogger
3  POJ - 2253
4  https://cn.vjudge.net/problem/POJ-2253#author=541607120101
5  题意：求一条1~2的路径 使得路径上的最大边权最小。
6  输入：石头数量 x y
7  样例输入：
8      2

```



```

9      0 0
10     3 4
11     3
12     17 4
13     19 4
14     18 5
15     0
16 样例输出:
17     Scenario #1
18     Frog Distance = 5.000
19
20     Scenario #2
21     Frog Distance = 1.414
22 思路:
23 dijkstra 计算边权最小即可
24 */
25 #include <iostream>
26 #include <cstdio>
27 #include <cstring>
28 #include <cstdlib>
29 #include <queue>
30 #include <algorithm>
31 #include <cmath>
32 #include <string>
33 #include <map>
34 #include <cmath>a
35 #define maxn 1005
36 #define INF 0x3f3f3f
37 using namespace std;
38 struct node{
39     int x,y;
40 }num[maxn];
41 double mp[maxn][maxn];
42 double calc(int i, int j) {
43     return (sqrt(pow((double)num[i].x-num[j].x, 2)+pow((double)num[i].y-num[j].y, 2)));
44 }
45 int n;
46 double dijkstra() {
47     bool vis[maxn];
48     double d[maxn];
49     memset(vis, false, sizeof(vis));
50     for(int i=0; i<n; i++) {
51         d[i] = mp[0][i];
52     }
53     d[0] = 0;
54     vis[0] = true;
55     for(int i=0; i<n-1; i++) {
56         double m = INF; int x;
57         for(int y=0; y<n; y++) if(!vis[y] && m >= d[y]) m = d[x=y];
58         vis[x] = true;
59         for(int y=0; y<n; y++){
60             if(!vis[y]) {
61                 double maxx = max(d[x], mp[x][y]);
62                 if(d[y] > maxx) d[y] = maxx;
63             }
64         }
65     }
66     return d[1];
67 }
68 int main(){
69     int m=0;
70     while(cin>>n,n){
71         m++;
72         for(int i=0;i<n;i++) cin>>num[i].x>>num[i].y;
73         for(int i=0;i<n;i++){
74             for(int j=0;j<n;j++){
75                 mp[i][j]=mp[j][i]=calc(i,j);
76             }
77             mp[i][i]=0;
78         }
79         printf("Scenario #%d\n", m);

```

```

80     printf("Frog Distance = %.3lf\n\n", dijkstra());
81 }
82 return 0;
83 }
84

```

```

1  /*
2  Heavy Transportation
3  POJ - 1797
4  https://cn.vjudge.net/problem/POJ-1797
5  题面 找出从1到n的路，使其称重最大
6  输入: T组样例 点个数和边个数
7         u -> v w
8  样例输入:
9         1
10        3 3
11        1 2 3
12        1 3 4
13        2 3 5
14  样例输出:
15        Scenario #1:
16  思路:
17  dijkstra
18  */
19 #include<cmath>
20 #include<cstdio>
21 #include<cstring>
22 #include<algorithm>
23 #define INF 0x3f3f3f3f
24 using namespace std;
25 int w[1010][1010];
26 int d[1010];
27 bool vis[1010];
28 int n,m;
29 void djs()
30 {
31     for(int i=1; i<=n; i++)
32     {
33         d[i]=w[1][i];
34         vis[i]=0;
35     }
36     for(int i=1; i<=n; i++)
37     {
38         int m=-1;
39         int x=-1;
40         for(int j=1; j<=n; j++)
41             if(!vis[j]&& d[j]>m)
42                 m=d[x=j];
43         if(x!=-1)
44         {
45             vis[x]=1;
46             for(int j=1; j<=n; j++)
47                 if(!vis[j]&& d[j]<min(d[x],w[x][j]))
48                     d[j]=min(d[x],w[x][j]); // 维护d数组
49         }
50     }
51 }
52 int main()
53 {
54     int T,t=1;
55     scanf("%d",&T);
56     while(T--)
57     {
58         scanf("%d%d",&n,&m);
59         for(int i=1; i<=n; i++)
60             for(int j=1; j<=n; j++)
61                 w[i][j]=i==j?0:-1;
62         for(int i=1; i<=m; i++)
63         {
64             int x,y,z;

```

```

65         scanf("%d%d%d",&x,&y,&z);
66         if(w[x][y]<z)
67             w[x][y]=w[y][x]=z;
68     }
69     djs();
70     printf("Scenario %d:\n%d\n\n",t++,d[n]);
71 }
72 }
73

```

```

1  /*
2  Invitation Cards
3  POJ - 1511
4  https://cn.vjudge.net/problem/POJ-1511#author=ty21
5  题意: 求最短路
6  输入: 点 边
7         u->v w
8  样例输入:
9         2
10        2 2
11        1 2 13
12        2 1 33
13        4 6
14        1 2 10
15        2 1 60
16        1 3 20
17        3 4 10
18        2 4 5
19        4 1 50
20  样例输出:
21        46
22        210
23  思路:
24  双向SPFA
25  */
26 #include <cstring>
27 #include <cstdio>
28 #include <algorithm>
29 #include <iostream>
30 #include <queue>
31 using namespace std;
32 #define maxn 1000100
33 #define INF 0x3f3f3f
34 struct node{
35     int u,v;
36     int w;
37     int next;
38 }edge[maxn],uedge[maxn];
39 int head[maxn],uhead[maxn];
40 int dis[maxn],udis[maxn];
41 int cnt,ucnt,n,t;
42 void init(){
43     memset(head,-1,sizeof head);
44     memset(uhead,-1,sizeof uhead);
45     memset(dis,INF,sizeof dis);
46     memset(udis,INF, sizeof udis);
47     cnt=0;
48     ucnt=0;
49 }
50 void add(int u,int v,int w,int &cnt, int *head, node *edge){
51     edge[cnt].w=w;
52     edge[cnt].u=u;
53     edge[cnt].v=v;
54     edge[cnt].next=head[u];
55     head[u]=cnt++;
56 }
57 void spfa(int st,int *dis,int *head, node *edge){
58     dis[st]=0;
59     queue<int> q;
60     q.push(st);

```

```

61 while(!q.empty()){
62     int u=q.front();
63     q.pop();
64     for(int i=head[u];~i;i=edge[i].next){
65         if(dis[edge[i].v]>dis[u]+edge[i].w){
66             dis[edge[i].v]=dis[u]+edge[i].w;
67             q.push(edge[i].v);
68         }
69     }
70 }
71 }
72 int main(){
73     int T,u,v,w;
74     scanf("%d", &T);
75     while(T--){
76         init();
77         scanf("%d %d", &n, &t);
78         for(int i=0;i<t;i++){
79             scanf("%d %d %d", &u, &v, &w);
80             add(u,v,w,cnt,head,edge);
81             add(v,u,w,ucnt,uhead,uedge);
82         }
83         spfa(1,dis,head,edge);
84         spfa(1,udis,uhead,uedge);
85         long long ans=0;
86         for(int i=1;i<=n;i++){
87             ans+=(long long)(udis[i]+dis[i]);
88         }
89         printf("%lld\n", ans);
90     }
91     return 0;
92 }
93

```

```

1  /*
2  Layout
3  POJ - 3169
4  https://cn.vjudge.net/problem/POJ-3169
5  题意: 关系好的奶牛想在一起, 关系不好的奶牛想远离对方, 问1的n的距离
6  输入: 奶牛数 关系好的个数t1 关系不好的个数t2
7          t1行 a b 之间最多 d
8          t2行 a b 之间最少 d
9  样例输入:
10         4 2 1
11         1 3 10
12         2 4 20
13         2 3 3
14  样例输出:
15         27
16  解法:
17  差分约束
18  关系好:   $dis[b]-dis[a]<d$   $a<b$   $add(a,b,d)$ 
19  关系坏:   $dis[a]-dis[b]<d$   $a>b$   $add(a,b,d)$ 
20  */
21 #include <cstring>
22 #include <cstdio>
23 #include <algorithm>
24 #include <queue>
25 #include <iostream>
26 #include <cmath>
27 using namespace std;
28 #define maxn 1000000
29 #define INF 0x3f3f3f3f
30 struct node{
31     int u,v,w;
32     int next;
33 }edge[maxn];
34 int head[maxn];
35 int dis[maxn];
36 int color[maxn];

```

```

37 int cnt,n,t1,t2;
38 void init(){
39     memset(head,-1,sizeof head);
40     memset(dis,INF,sizeof dis);
41     memset(color,0,sizeof color);
42     cnt=0;
43 }
44 void add(int u,int v,int w){
45     edge[cnt].u=u;
46     edge[cnt].v=v;
47     edge[cnt].w=w;
48     edge[cnt].next=head[u];
49     head[u]=cnt++;
50 }
51 bool spfa(){
52     queue<int> q;
53     dis[1]=0;
54     q.push(1);
55     color[1]++;
56     while(!q.empty()){
57         int u=q.front();
58         q.pop();
59         for(int i=head[u];~i;i=edge[i].next){
60             if(dis[edge[i].v]>dis[u]+edge[i].w){
61                 dis[edge[i].v]=dis[u]+edge[i].w;
62                 q.push(edge[i].v);
63                 if(++color[edge[i].v]>=n) return false;
64             }
65         }
66     }
67     return true;
68 }
69 int main(){
70     scanf("%d %d %d", &n, &t1, &t2);
71     init();
72     int u,v,w;
73     for(int i=0;i<t1;i++){
74         scanf("%d %d %d", &u, &v, &w);
75         if(u>v)swap(u,v);
76         add(u,v,w);
77     }
78     for(int i=0;i<t2;i++){
79         scanf("%d %d %d", &u, &v, &w);
80         if(u<v)swap(u,v);
81         add(u,v,-w);
82     }
83     if(spfa()){
84         if(dis[n]==INF){
85             printf("-2\n");
86         }else{
87             printf("%d\n", dis[n]);
88         }
89     }else{
90         printf("-1\n");
91     }
92     return 0;
93 }
94

```

```

1  /*
2  Marriage Match IV
3  HDU - 3416
4  https://cn.vjudge.net/problem/HDU-3416
5  题意: 找出所有最短路个数
6  输入 :
7      T组样例
8      点 边
9      u->v w
10     B E
11  样例输入:

```

```

12      2
13      7 8
14      1 2 1
15      1 3 1
16      2 4 1
17      3 4 1
18      4 5 1
19      4 6 1
20      5 7 1
21      6 7 1
22      1 7
23
24      2 2
25      1 2 1
26      1 2 2
27      1 2
28  样例输出:
29      2
30      1
31  解法:
32  双向SPFA枚举每条边找出最短路的连通图
33  最大流求最短路的个数
34  */
35  #include<cstdio>
36  #include<cstring>
37  #include<algorithm>
38  #include<queue>
39  #include<iostream>
40  using namespace std;
41  #define INF 0x3f3f3f
42  #define maxn 100010
43  struct node{
44      int u, v, w;
45      int next;
46  }edge[maxn], uedge[maxn], wedge[maxn*2];
47  int head[maxn], uhead[maxn], whead[maxn];
48  int cnt, ucnt, wcnt;
49  int dis[maxn], udis[maxn];
50  int d[maxn];
51  int t,n,B,E;
52  void init(){
53      memset(head, -1, sizeof head);
54      memset(uhead, -1, sizeof uhead);
55      memset(whead, -1, sizeof whead);
56      memset(dis, INF, sizeof dis);
57      memset(udis, INF, sizeof udis);
58      cnt = ucnt = wcnt = 0;
59  }
60  void add(int u, int v, int w, node edge[], int &cnt, int *head){
61      edge[cnt].w=w;
62      edge[cnt].u=u;
63      edge[cnt].v=v;
64      edge[cnt].next=head[u];
65      head[u]=cnt++;
66  }
67  void add_dinic(int u,int v,int w){
68      wedge[wcnt].w=w;
69      wedge[wcnt].u=u;
70      wedge[wcnt].v=v;
71      wedge[wcnt].next=whead[u];
72      whead[u]=wcnt++;
73
74      wedge[wcnt].w=0;
75      wedge[wcnt].u=v;
76      wedge[wcnt].v=u;
77      wedge[wcnt].next=whead[v];
78      whead[v]=wcnt++;
79  }
80  void spfa(int st, int *dis, int *head, node edge[]){
81      queue<int> q;
82      bool color[maxn];

```

```

83     memset(color,true,sizeof color);
84     q.push(st);
85     dis[st]=0;
86     color[st]=false;
87     while(!q.empty()){
88         int u=q.front();
89         q.pop();
90         color[u]=true;
91         for(int i=head[u];~i;i=edge[i].next){
92             if(dis[u]+edge[i].w<dis[edge[i].v]){
93                 dis[edge[i].v]=dis[u]+edge[i].w;
94                 if(color[edge[i].v]){
95                     q.push(edge[i].v);
96                     color[edge[i].v]=false;
97                 }
98             }
99         }
100     }
101 }
102 bool bfs(int B,int E){
103     memset(d,-1,sizeof d);
104     queue<int> q;
105     d[B]=0;
106     q.push(B);
107     while(!q.empty()){
108         int st=q.front();
109         q.pop();
110         for(int i=whead[st];~i;i=wedge[i].next){
111             if(d[wedge[i].v]==-1&&wedge[i].w>0){
112                 d[wedge[i].v]=d[st]+1;
113                 q.push(wedge[i].v);
114             }
115         }
116     }
117     return d[E]!=-1;
118 }
119 int dfs(int a,int b){
120     int r=0;
121     if(a==E) return b;
122     for(int i=whead[a];~i;i=wedge[i].next){
123         if(wedge[i].w>0&&d[wedge[i].v]==d[a]+1){
124             int x=min(wedge[i].w,b-r);
125             x=dfs(wedge[i].v,x);
126             r+=x;
127             wedge[i].w-=x;
128             wedge[i^1].w+=x;
129         }
130     }
131     if(!r) d[a]=-2;
132     return r;
133 }
134 int dinic(int B,int E){
135     int ans=0;
136     int t;
137     while(bfs(B,E)){
138         while(t=dfs(B,INF)) ans+=t;
139     }
140     return ans;
141 }
142 int main(){
143     int T;
144     scanf("%d",&T);
145     while(T--){
146         scanf("%d %d",&n,&t);
147         init();
148         for(int i=0;i<t;i++){
149             int u,v,w;
150             scanf("%d %d %d",&u,&v,&w);
151             add(u,v,w,edge,cnt,head);
152             add(v,u,w,uedge,ucnt,uhead);
153         }

```

```

154     scanf("%d %d", &B, &E);
155     spfa(B, dis, head, edge);
156     if(dis[E]==INF){
157         printf("0\n");
158         continue;
159     }
160     long long tmp = dis[E];
161     spfa(E, udis, uhead, uedge);
162     for(int i=0; i<t; i++){
163         int u=edge[i].u, v=edge[i].v;
164         if(edge[i].w&&dis[u]+udis[v]+edge[i].w==tmp){
165             add_dinic(u, v, 1);
166         }
167     }
168     printf("%d\n", dinic(B, E));
169 }
170 return 0;
171 }
172

```

```

1  /*
2  MPI Maelstrom
3  POJ - 1502
4  https://cn.vjudge.net/problem/POJ-1502#author=weissice
5  题意:
6      求从一台电脑发出的消息到最后一台电脑收到的最短时间
7  输入:
8      n台计算机
9      以矩阵形式输出相邻电脑通讯需要的时间, 只有矩阵的下三角
10  样例输入:
11      5
12      50
13      30 5
14      100 20 50
15      10 x x 10
16  样例输出:
17      35
18  注意:
19  stoi() 不是标准库里面的函数, 会 Compile Error
20  */
21 #include <cstring>
22 #include <cstdio>
23 #include <iostream>
24 #include <algorithm>
25 #include <queue>
26 using namespace std;
27 #define INF 0x3f3f3f
28 #define maxn 110
29 int mp[maxn][maxn];
30 int dis[maxn];
31 int n, B=1;
32 int return_int(char *a){
33     int ans=0;
34     int len=strlen(a);
35     for(int i=0; i<len; i++){
36         ans=a[i]-'0'+ans*10;
37     }
38     return ans;
39 }
40 void init(){
41     memset(mp, INF, sizeof mp);
42     memset(dis, INF, sizeof dis);
43 }
44 void spfa(int B){
45     dis[B]=0;
46     queue<int> q;
47     q.push(B);
48     while(!q.empty()){
49         int u=q.front();
50         q.pop();

```



```

51     for(int i=1;i<=n;i++){
52         if(dis[i]>dis[u]+mp[u][i]){
53             dis[i]=dis[u]+mp[u][i];
54             q.push(i);
55         }
56     }
57 }
58 return ;
59 }
60 int main(){
61     scanf("%d", &n);
62     init();
63     char a[10];
64     for(int i=1;i<=n;i++){
65         for(int j=1;j<=i;j++){
66             scanf("%s", a);
67             if(a[0]!='x') mp[i][j]=mp[j][i]=return_int(a);
68         }
69     }
70     spfa(B);
71     int ans=0;
72     for(int i=1;i<=n;i++){
73         if(ans<dis[i]) ans=dis[i];
74     }
75     printf("%d\n", ans);
76     return 0;
77 }
78

```

```

1  /*
2  Silver Cow Party
3  POJ - 3268
4  https://cn.vjudge.net/problem/POJ-3268#author=chenchonghan
5  题意:n个点m条边牛全要到x参加聚会然后在回家，问任意一台奶牛行走的最长时间
6  输入: 点 边 聚会地点
7      u -> v w
8  样例输入:
9      4 8 2
10     1 2 4
11     1 3 2
12     1 4 7
13     2 1 1
14     2 3 5
15     3 1 2
16     3 4 4
17     4 2 3
18 样例输出:
19     10
20 直接跑两边spfa即可
21 */
22 #include <cstring>
23 #include <cstdio>
24 #include <algorithm>
25 #include <iostream>
26 #include <queue>
27 using namespace std;
28 #define maxn 1000100
29 #define INF 0x3f3f3f
30 struct node{
31     int u,v;
32     int w;
33     int next;
34 }edge[maxn],uedge[maxn];
35 int head[maxn],uhead[maxn];
36 int dis[maxn],udis[maxn];
37 int cnt,ucnt,n,t,st;
38 void init(){
39     memset(head,-1,sizeof head);
40     memset(uhead,-1,sizeof uhead);
41     memset(dis,INF,sizeof dis);

```

```

42     memset(udis,INF, sizeof udis);
43     cnt=0;
44     ucnt=0;
45 }
46 void add(int u,int v,int w,int &cnt, int *head, node *edge){
47     edge[cnt].w=w;
48     edge[cnt].u=u;
49     edge[cnt].v=v;
50     edge[cnt].next=head[u];
51     head[u]=cnt++;
52 }
53 void spfa(int st,int *dis,int *head, node *edge){
54     dis[st]=0;
55     queue<int> q;
56     q.push(st);
57     while(!q.empty()){
58         int u=q.front();
59         q.pop();
60         for(int i=head[u];~i;i=edge[i].next){
61             if(dis[edge[i].v]>dis[u]+edge[i].w){
62                 dis[edge[i].v]=dis[u]+edge[i].w;
63                 q.push(edge[i].v);
64             }
65         }
66     }
67 }
68 int main(){
69     scanf("%d %d %d", &n, &t, &st);
70     int u,v,w;
71     init();
72     for(int i=0;i<t;i++){
73         scanf("%d %d %d", &u, &v, &w);
74         add(u,v,w,cnt,head,edge);
75         add(v,u,w,ucnt,uhead,uedge);
76     }
77     spfa(st,dis,head,edge);
78     spfa(st,udis,uhead,uedge);
79     int ans=0;
80     for(int i=1;i<=n;i++){
81         ans=max(ans,udis[i]+dis[i]);
82     }
83     printf("%d\n", ans);
84     return 0;
85 }
86

```

```

1  /*
2  Subway
3  POJ - 2502
4  https://cn.vjudge.net/problem/POJ-2502#author=0
5  题意： 从家到学校，可以选择坐（40km/s）地铁或（10km/s）步行，求最短时间
6  输入  第一行 起点坐标，终点坐标
7          下面是每个地铁经过的站 知道-1 -1结束
8  样例输入：
9          0 0 10000 1000
10         0 200 5000 200 7000 200 -1 -1
11         2000 600 5000 600 10000 600 -1 -1
12  样例输出：
13         21
14  解法：
15         建图，跑一遍Floyd-Warshall
16  */
17 #include <cstring>
18 #include <cstdio>
19 #include <algorithm>
20 #include <iostream>
21 #include <queue>
22 #include <cmath>
23 using namespace std;
24 #define maxn 1000

```

```

25 #define INF 0x3f3f3f
26 struct node{
27     double x,y;
28     int kid;
29 }poi[maxn];
30 double mp[maxn][maxn];
31 int cnt=0;
32 void build(double x,double y,int kid){
33     poi[cnt].x=x;
34     poi[cnt].y=y;
35     poi[cnt].kid=kid;
36     cnt++;
37 }
38 int main(){
39     double x,y;
40     int kid=0;
41     scanf("%lf %lf", &x, &y);
42     build(x,y,kid++);
43     scanf("%lf %lf", &x, &y);
44     build(x,y,kid++);
45     while(~scanf("%lf %lf", &x, &y)) {
46         if(x<0&&y<0){
47             kid++;
48             continue;
49         }
50         build(x,y,kid);
51     }
52     for(int i=0;i<cnt;i++){
53         for(int j=0;j<cnt;j++){
54             mp[i][j]=sqrt((poi[i].x-poi[j].x)*(poi[i].x-poi[j].x)+(poi[i].y-poi[j].y)*(poi[i].y-poi[j].y))/(10000.0/60);
55         }
56     }
57     for(int i=0;i<cnt-1;i++){
58         if(poi[i].kid==poi[i+1].kid){
59             mp[i][i+1]=mp[i+1][i]=sqrt((poi[i].x-poi[i+1].x)*(poi[i].x-poi[i+1].x)+(poi[i].y-poi[i+1].y)*(poi[i].y-poi[i+1].y))/(40000.0/60);
60         }
61     }
62     for(int k=0;k<cnt;k++){
63         for(int i=0;i<cnt;i++){
64             for(int j=0;j<cnt;j++){
65                 mp[i][j]=min(mp[i][k]+mp[k][j],mp[i][j]);
66             }
67         }
68     }
69     printf("%d\n", (int)(mp[0][1]+0.5));
70     return 0;
71 }
72

```

```

1  /*
2  The Shortest Path in Nya Graph
3  HDU - 4725
4  https://cn.vjudge.net/problem/HDU-4725#author=0
5  题意：求从1到n的最短路，有层数，点可能不在同一层
6  输入：节点数 额外边数 相邻层移动的成本
7  样例输入：
8      2
9      3 3 3
10     1 3 2
11     1 2 1
12     2 3 1
13     1 3 3
14
15     3 3 3
16     1 3 2
17     1 2 2
18     2 3 2
19     1 3 4
20 样例输出：
21     Case #1: 2

```

```

22     Case #2: 3
23  解法:
24  点与点建边, 层与层建边, 然后层上的点与相邻层和本层建边
25  spfa
26  */
27  #include <cstring>
28  #include <cstdio>
29  #include <algorithm>
30  #include <queue>
31  #include <iostream>
32  #include <cmath>
33  using namespace std;
34  #define maxn 1000000
35  #define INF 0x3f3f3f3f
36  struct node{
37     int u,v,w;
38     int next;
39 }edge[maxn*3];
40 int head[maxn];
41 int dis[maxn];
42 int vv[maxn], lay[maxn];
43 int cnt,n,t,c;
44 void init(){
45     memset(head,-1,sizeof head);
46     memset(vv,0,sizeof(vv));
47     memset(dis,INF,sizeof dis);
48     cnt=0;
49 }
50 void add(int u,int v,int w){
51     edge[cnt].u=u;
52     edge[cnt].v=v;
53     edge[cnt].w=w;
54     edge[cnt].next=head[u];
55     head[u]=cnt++;
56 }
57 void spfa(){
58     queue<int> q;
59     dis[1]=0;
60     q.push(1);
61     while(!q.empty()){
62         int u=q.front();
63         q.pop();
64         for(int i=head[u];~i;i=edge[i].next){
65             if(dis[edge[i].v]>dis[u]+edge[i].w){
66                 dis[edge[i].v]=dis[u]+edge[i].w;
67                 q.push(edge[i].v);
68             }
69         }
70     }
71     return ;
72 }
73 int main(){
74     int T;
75     int q,u,v,w;
76     scanf("%d", &T);
77     for(int s=1;s<=T;s++){
78         scanf("%d %d %d", &n, &t, &c);
79         init();
80         for(int i=1;i<=n;i++){
81             scanf("%d", &q);
82             lay[i]=q;
83             vv[q]=1;
84         }
85         for(int i = 1; i < n; i++){
86             if(vv[i]&&vv[i+1]){
87                 add(n+i,n+i+1,c);
88                 add(n+i+1,n+i,c);
89             }
90         }
91         for(int i = 1; i <= n; i++) {
92             add(n+lay[i],i,0);

```

```

93         if(lay[i] > 1)
94             add(i,n+lay[i]-1,c);
95         if(lay[i] < n)
96             add(i,n+lay[i]+1,c);
97     }
98     for(int i=0;i<t;i++){
99         scanf("%d %d %d", &u, &v, &w);
100        add(u,v,w);
101        add(v,u,w);
102    }
103    spfa();
104    if(dis[n]==INF) dis[n]=-1;
105    printf("Case #d: %d\n", s, dis[n]);
106 }
107 return 0;
108 }
109

```

```

1  /*
2  Til the Cows Come Home
3  POJ - 2387
4  https://cn.vjudge.net/problem/POJ-2387
5  题意: 最短路裸题
6  输入: 点 边
7          u -> v w
8  样例输入:
9          5 5
10         1 2 20
11         2 3 30
12         3 4 20
13         4 5 20
14         1 5 100
15  样例输出:
16         90
17  解法:
18         SPFA
19  */
20 #include<cstdio>
21 #include<cstring>
22 #include<algorithm>
23 #include<queue>
24 #include<iostream>
25 using namespace std;
26 #define INF 0x3f3f3f
27 #define maxn 10010
28 struct node{
29     int u;
30     int v;
31     int w;
32     int next;
33 }edge[maxn],uedge[maxn];
34 int cnt,ucnt;
35 int t,n;
36 int head[maxn],uhead[maxn];
37 int dis[maxn], udis[maxn];
38 void init(){
39     memset(head,0,sizeof head);
40     memset(dis,INF,sizeof dis);
41     cnt=1;
42 }
43 void add(int u, int v, int w, node edge[], int &cnt,int *head){
44     edge[cnt].w=w;
45     edge[cnt].u=u;
46     edge[cnt].v=v;
47     edge[cnt].next=head[u];
48     head[u]=cnt++;
49 }
50 void spfa(int st,int dis[],int head[],node edge[]) {
51     dis[st] = 0;
52     queue<int>q;

```

```

53     q.push(st);
54     while (!q.empty()) {
55         int u = q.front();
56         q.pop();
57         for (int i = head[u]; i!=0; i = edge[i].next) {
58             int v = edge[i].v;
59             if (dis[v] > dis[u] + edge[i].w) {
60                 dis[v] = dis[u] + edge[i].w;
61                 q.push(v);
62             }
63         }
64     }
65     return;
66 }
67 int main(){
68     while(~scanf("%d %d", &t, &n)){
69         init();
70         int u,v,w;
71         for(int i=0;i<t;i++){
72             scanf("%d %d %d", &u, &v, &w);
73             add(u, v, w, edge, cnt, head);
74             add(v, u, w, edge, cnt, head);
75             //add(v, u, w, uedge, ucnt, uhead);
76         }
77         spfa(1, dis, head, edge);
78         cout<<dis[n]<<endl;
79     }
80     return 0;
81 }
82

```

```

1  /*
2  Tram
3  POJ - 1847
4  https://cn.vjudge.net/problem/POJ-1847
5  题意：从A到B要选择开关的次数，开关默认指向第一个交叉点
6  输入：N个交叉点数，从A到B
7         第i行第一个数表示第i个交叉节点可以连其他节点的个数，Ki表示连接的节点
8  样例输入：
9         3 2 1
10        2 2 3
11        2 3 1
12        2 1 2
13  样例输出：
14        0
15  解法
16  建图，直接跑Floyd—warshall
17  */
18  #include <cstdio>
19  #include <algorithm>
20  #include <cstring>
21  using namespace std;
22  #define maxn 110
23  #define INF 0x3f3f3f3f
24  int mp[maxn][maxn];
25  int N,A,B;
26  int main(){
27     while(~ scanf("%d %d %d", &N, &A, &B)){
28         memset(mp,INF,sizeof mp);
29         for(int i=1;i<=N;i++){
30             int n,m;
31             scanf("%d", &n);
32             if(n==0) continue;
33             scanf("%d", &m);
34             mp[i][m]=0;
35             n--;
36             while(n--){
37                 scanf("%d", &m);
38                 mp[i][m]=min(1,mp[i][m]);
39                 mp[m][i]=min(1,mp[m][i]);

```

```

40     }
41 }
42 for(int k=1;k<=N;k++){
43     for(int i=1;i<=N;i++){
44         for(int j=1;j<=N;j++){
45             if(mp[i][j]>mp[i][k]+mp[k][j])
46                 mp[i][j]=mp[i][k]+mp[k][j];
47         }
48     }
49 }
50 if(mp[A][B] == INF) printf("-1\n");
51 else printf("%d\n", mp[A][B]);
52 }
53 return 0;
54 }
55

```

```

1  /*
2  Wormholes
3  POJ - 3259
4  https://cn.vjudge.net/problem/POJ-3259#author=SCU2018
5  题意: 通过虫洞回到这个原点之前的时间
6  虫洞路径分为两种, 双向的时间是正方向, 单向的时间是反方向
7  输入:
8      T组样例
9      点, 双向路径, 单向路径
10     u v w
11  样例输入:
12     2
13     3 3 1
14     1 2 2
15     1 3 4
16     2 3 1
17     3 1 3
18
19     3 2 1
20     1 2 3
21     2 3 4
22     3 1 8
23  样例输出:
24     NO
25     YES
26  思路:
27  SPFA直接跑, 要是发现原点时间减小了, 就输出YES 否NO
28  */
29 #include <cstring>
30 #include <cstdio>
31 #include <algorithm>
32 #include <queue>
33 using namespace std;
34 #define maxn 100010
35 #define INF 0x3f3f3f
36 struct node{
37     int u,v,w;
38     int next;
39 }edge[maxn];
40 int head[maxn];
41 int dis[maxn];
42 int n,m1,m2;
43 int cnt,B=1,E=n;
44 void init(){
45     memset(head,-1,sizeof head);
46     memset(dis,INF,sizeof dis);
47     cnt=0;
48 }
49 void add(int u,int v,int w){
50     edge[cnt].u=u;
51     edge[cnt].v=v;
52     edge[cnt].w=w;
53     edge[cnt].next=head[u];

```

```

54     head[u]=cnt++;
55 }
56 bool spfa(int B){
57     dis[B]=0;
58     queue<int> q;
59     q.push(B);
60     while(!q.empty()){
61         int u=q.front();
62         q.pop();
63         for(int i=head[u];~i;i=edge[i].next){
64             if(dis[edge[i].v]>dis[u]+edge[i].w){
65                 dis[edge[i].v]=dis[u]+edge[i].w;
66                 if(dis[B]<0) return true;
67                 q.push(edge[i].v);
68             }
69         }
70     }
71     return false;
72 }
73 int main(){
74     int T;
75     scanf("%d", &T);
76     while(T--){
77         scanf("%d %d %d", &n, &m1, &m2);
78         init();
79         int u,v,w;
80         for(int i=0;i<m1;i++){
81             scanf("%d %d %d", &u, &v, &w);
82             add(u,v,w);
83             add(v,u,w);
84         }
85         for(int i=0;i<m2;i++){
86             scanf("%d %d %d", &u, &v, &w);
87             add(u,v,-w);
88         }
89         if(spfa(B)) printf("YES\n");
90         else printf("NO\n");
91     }
92     return 0;
93 }
94

```

```

1  /*
2  昂贵的聘礼
3  POJ - 1062
4  https://cn.vjudge.net/problem/POJ-1062
5  题意: 他和酋长买东西, 他可以通过和别人交换来找减少自己最少要花费的金币, 但只能在等级差内交易
6  输入  等级差距 N个物品的总数
7         物品价格 主人地位 代替品总数
8         代替品的编号, 优惠价格
9  样例输入:
10      1 4
11      10000 3 2
12      2 8000
13      3 5000
14      1000 2 1
15      4 200
16      3000 2 1
17      4 200
18      50 2 0
19  样例输出:
20      5250
21  */
22  #include <cstdio>
23  #include <cstring>
24  #include <algorithm>
25  #include <iostream>
26  #include <queue>
27  using namespace std;
28  #define maxn 100000

```



```
29 #define INF 0x3f3f3f
30 struct node{
31     int u,v,w;
32     int next;
33 }edge[maxn];
34 int head[maxn];
35 int dis[maxn];
36 int cnt,m,n;
37 bool color[maxn];
38 int l[maxn];
39 void init(){
40     memset(head,-1,sizeof head);
41     memset(dis,INF,sizeof dis);
42     memset(color,true,sizeof color);
43     cnt=0;
44 }
45 void add(int u,int v,int w){
46     edge[cnt].u=u;
47     edge[cnt].v=v;
48     edge[cnt].w=w;
49     edge[cnt].next=head[u];
50     head[u]=cnt++;
51 }
52 void spfa(){
53     queue<int> q;
54     q.push(0);
55     dis[0]=0;
56     while(!q.empty()){
57         int st=q.front();
58         q.pop();
59         for(int i=head[st];~i;i=edge[i].next){
60             if(dis[st]+edge[i].w<dis[edge[i].v]&&color[edge[i].v]){
61                 dis[edge[i].v]=dis[st]+edge[i].w;
62                 q.push(edge[i].v);
63             }
64         }
65     }
66 }
67 int solve(){
68     int ans=INF;
69     for(int i=1;i<=n;i++){
70         memset(dis,INF,sizeof dis);
71         memset(color,true,sizeof color);
72         int minle=l[i];
73         for(int j=1;j<=n;j++){
74             if(l[j]<minle||l[j]-minle>m){
75                 color[j]=false;
76             }
77         }
78         spfa();
79         ans=min(ans,dis[1]);
80     }
81     return ans;
82 }
83 int main(){
84     init();
85     scanf("%d %d", &m, &n);
86     int p,x;
87     int t,v;
88     for(int i=1;i<=n;i++){
89         scanf("%d %d %d", &p, &l[i], &x);
90         add(0,i,p);
91         while(x--){
92             scanf("%d %d",&t, &v);
93             add(t,i,v);
94         }
95     }
96     printf("%d\n", solve());
97     return 0;
98 }
99
```

有 0 个人打赏

文章最后发布于: 2019-07-2

©2019 CSDN 皮肤主题: 大白 设计师: CSDN官方博客