

专题2

目录

Problem ID	Title
1001	Arbitrage
1002	Free DIY Tour
1003	Minimum Transport Cost
1004	六度分离
1005	畅通工程续
1006	一个人的旅行
1007	HDU Today
1008	最短路
1009	Choose the best route
1010	Here We Go(relians) Again
1011	Einbahnstrasse
1012	In Action
1013	最短路径问题

1001 Arbitrage

AC代码

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int n, m, i, j, k, t;
4  double d[35][35];
5  string x[35], x1, x2;
6  int main()
7  {
8      while (scanf("%d",&n)!=EOF)
9      {
10         if (n == 0)break;
11         memset(d,0,sizeof(d));/*因为是求最大值，清零即可*/
12         map<string,int> p;/*使每个货币名称对应1到n的序号，便于后续给d数组赋值*/
13         for (i = 1; i <= n; i++)
14         {
15             cin >> x[i];
16             p[x[i]] = i;
```

```

17     }
18     scanf("%d",&m);
19     double c; /*汇率*/
20     for (i = 1; i <= m; i++)
21     {
22         cin >> x1 >> c >> x2;
23         d[p[x1]][p[x2]] = c;
24     }
25     /*数据存储结束，化成d[i][j]的形式*/
26     for (k = 1; k <= n; k++) /*Floyd算法*/
27         for (i = 1; i <= n; i++)
28             for (j = 1; j <= n; j++)
29                 if (d[i][k] * d[k][j] > d[i][j])
30                     d[i][j] = d[i][k] * d[k][j];
31     printf("Case %d: ",++t);
32     if (d[1][1] > 1) printf("Yes\n");
33     else printf("No\n");
34 }
35 return 0;
36 }

```

1002 Free DIY Tour

AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  int t;
4  int n,m;
5  int d[105][105];
6  int path[105][105];
7  int fun[105];
8  int a, b;
9  int i, j, k;
10 int cnt = 0;
11
12 int main()
13 {
14     scanf("%d", &t);
15     while (t--) {
16         scanf("%d", &n);
17         for (i = 1; i <= n; i++)
18             scanf("%d", &fun[i]);
19         fun[n + 1] = 0;
20         for (i = 1; i <= n + 1; i++)
21             for (j = 1; j <= n + 1; j++)
22                 d[i][j] = -1;
23         memset(path, 0, sizeof(path));
24         scanf("%d", &m);
25         while (m--) {
26             scanf("%d%d", &a, &b);
27             d[a][b] = fun[b];
28             path[a][b] = b;
29         }

```

```

30     for (k = 1; k <= n + 1; k++)
31         for (i = 1; i <= n + 1; i++)
32             for (j = 1; j <= n + 1; j++)
33                 if (d[i][k] != -1 && d[k][j] != -1)
34                     {
35                         if (d[i][k] + d[k][j] > d[i][j]) {
36                             d[i][j] = d[i][k] + d[k][j];
37                             path[i][j] = path[i][k];
38                         }
39                     }
40     int u = 1;
41     int e = n + 1;
42     printf("CASE %d#\n", ++cnt);
43     printf("points : %d\n", d[u][e]);
44     printf("circuit : ");
45     while (u != e)
46     {
47         printf("%d->", u);
48         u = path[u][e];
49     }
50     printf("%d\n", 1);
51     if (t)printf("\n");
52 }
53 return 0;
54 }

```

1003 Minimum Transport Cost

AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  #define inf 99999999
4  int d[1010][1010];
5  int path[1010][1010];
6  int n;
7  int a, b;
8  int fun[1010];
9  int main()
10 {
11     while (scanf("%d", &n), n) {
12         for(int i=1;i<=n;i++)
13             for (int j = 1; j <= n; j++) {
14                 scanf("%d", &d[i][j]);
15                 if (d[i][j] == -1)d[i][j] = inf;
16                 path[i][j] = 0;
17             }
18         for (int i = 1; i <= n; i++)
19             scanf("%d", &fun[i]);
20         for(int i=1;i<=n;i++)
21             for (int j = 1; j <= n; j++) {
22                 if (d[i][j] != inf ) {
23                     d[i][j] += fun[j];
24                     path[i][j] = j;

```



```

16         while (m--) {
17             scanf("%d%d", &a, &b);
18             d[a][b] = 1;
19             d[b][a] = 1;
20         }
21         for(int k=0;k<n;k++)
22             for(int i=0;i<n;i++)
23                 for (int j = 0; j < n; j++) {
24                     /*if (d[i][k] != inf && d[k][j] != inf) {*/
25                         if (d[i][j] > d[i][k] + d[k][j])
26                             d[i][j] = d[i][k] + d[k][j];
27                     /* }*/
28                 }
29         for(int i=0;i<n;i++)
30             for (int j = 0; j < n; j++) {
31                 if (d[i][j] == inf || d[i][j] > 7) {
32                     flag = 0;
33                     break;
34                 }
35             }
36         if (flag)printf("Yes\n");
37         else printf("No\n");
38     }
39     return 0;
40 }

```

1005 畅通工程续

AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  #define inf 0x7FFFFFFF
4  int Map[210][210];
5  int n,m; /*城镇数目，道路数目*/
6  int s, e, x;
7  int a, b;
8  int Min;
9  int Dist[210];
10 int vis[210];
11 int nex;
12 int main()
13 {
14     while (scanf("%d%d", &n, &m) != EOF) {
15         for (int i = 0; i < n; i++) {
16             vis[i] = 0;
17             Dist[i] = inf; /*注意需要初始化为无穷大，不能默认为0，因为初始是无边的，0
18 则是无边且距离为0*/
19             for (int j = 0; j < n; j++)
20                 Map[i][j] = inf;
21         }
22         while (m--) {
23             scanf("%d%d%d", &a, &b, &x);
24             Map[a][b] = min(Map[a][b], x); /*防止两个点之间有多条不同长度的路*/

```

```

24     Map[b][a] = Map[a][b];/*无向图*/
25 }
26 scanf("%d%d", &s, &e);
27 Dist[s] = 0;
28 vis[s] = 1;
29 while (s != e) {
30     Min = inf;
31     for (int i = 0; i < n; i++) {
32         if (Map[s][i] != inf) {
33             Dist[i] = min(Dist[i], Map[s][i] + Dist[s]);/*松弛操作，关
键*/
34         }
35         if (!vis[i] && Dist[i] < Min) {/*判断vis的作用是防止往回找，例如
第一次是防止找自己*/
36             nex = i;/*寻找最短路*/
37             Min = Dist[i];/*更新最小值*/
38         }
39     }
40     if (Min == inf)break;/*别忘了这句，不然如果是-1就死循环了*/
41     s = nex;/*在此之前已经保存了相应的Dist，而这里需要更新比较的目标，即本次的
最短路结果，来与下次的比较*/
42     vis[nex] = 1;/*表示该点的最短路已经求出，无需改变*/
43 }
44 if (Dist[e] == inf)printf("-1\n");
45 else printf("%d\n", Dist[e]);
46 }
47 return 0;
48 }

```

1006 一个人的旅行

AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  #define inf 0x7fffffff
4  int Map[1010][1010];
5  int vis[1010];
6  int Dist[1010];
7  int main()
8  {
9      int t, s, d;/*路，起始点数目，终点数目*/
10     int a, b, time;
11     int Min;
12     int start, end;/*设0点为起点，1001点为终点*/
13     int nex;
14     while (scanf("%d%d%d", &t, &s, &d) != EOF) {
15         for (int i = 0; i <= 1001; i++) {/*初始化*/
16             Dist[i] = inf;
17             vis[i] = 0;
18             for (int j = 0; j <= 1001; j++)
19                 Map[i][j] = inf;
20         }
21         while (t--) {

```

```

22         scanf("%d%d%d", &a, &b, &time);
23         Map[a][b] = min(Map[a][b], time);
24         Map[b][a] = Map[a][b];
25     }
26     while (s--) {
27         int o;
28         scanf("%d", &o);
29         Map[0][o] = 0;
30         Map[o][0] = 0; /*让0点与s个初始点相连*/
31     }
32     while (d--) {
33         int p;
34         scanf("%d", &p);
35         Map[1001][p] = 0; /*让1001点和d个目标点相连*/
36         Map[p][1001] = 0;
37     }
38     start = 0;
39     end = 1001;
40     Dist[start] = 0;
41     vis[start] = 1;
42     while (start != end) {
43         Min = inf;
44         for (int i = 1; i <= 1001; i++) {
45             if (Map[start][i] != inf) {
46                 Dist[i] = min(Dist[i], Dist[start] + Map[start][i]);
47             }
48             if (Dist[i] < Min && !vis[i]) {
49                 nex = i;
50                 Min = Dist[i];
51             }
52         }
53         if (Min == inf) break;
54         start = nex;
55         vis[start] = 1;
56     }
57     if (Dist[end] == inf) printf("-1\n");
58     else printf("%d\n", Dist[end]);
59 }
60 return 0;
61 }

```

1007 HDU Today

AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  #define inf 10000000
4  int n;
5  int d[160][160];
6  char s[35], e[35]; /*起点终点名称*/
7  map<string, int> stt; /*公交车站名称和序号*/
8  int t;
9  char s1[35], s2[35];

```

```

10  int i, j, k;
11
12  int main()
13  {
14      while (scanf("%d", &n)) {
15          int flag = 0;
16          for(i=1;i<=155;i++)
17              for (j = 1; j <= 155; j++) {
18                  if (i == j)d[i][j] = 0;
19                  else d[i][j] = inf;
20              }
21          stt.clear();
22          if (n == -1)break;
23          scanf("%s%s", s, e);/*输入起点终点名称*/
24          int cnt = 0;
25          stt[s] = ++cnt;
26          if(stt[e]==0)stt[e] = ++cnt;
27          else /*起点终点重合*/
28              flag=1;
29      }
30      while (n--) {
31          scanf("%s%s%d", s1, s2, &t);
32          if(stt[s1]==0)stt[s1] = ++cnt;
33          if(stt[s2]==0)stt[s2] = ++cnt;
34          d[stt[s1]][stt[s2]] = min(d[stt[s1]][stt[s2]], t);/*存入数据*/
35          d[stt[s2]][stt[s1]] = d[stt[s1]][stt[s2]];/*无向图*/
36      }
37      if (flag) {
38          puts("0");
39          continue;
40      }
41      for (k = 1; k <= cnt; k++)/*Floyd*/
42          for (i = 1; i <= cnt; i++)
43              for (j = 1; j <= cnt; j++)
44                  d[i][j] = min(d[i][j], d[i][k] + d[k][j]);
45      if (d[1][2] == inf)printf("-1\n");
46      else printf("%d\n", d[1][2]);
47  }
48  return 0;
49  }

```

1008 最短路

AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  #define inf 0x7fffffff
4  int n, m;
5  int Dist[110];
6  int vis[110];
7  int a, b, c;
8  int Map[110][110];
9  int Min;

```



```

10 int nex;
11 int main()
12 {
13     while (scanf("%d%d", &n, &m)) {
14         if (!n && !m) break;
15         for (int i = 1; i <= n; i++) {
16             Dist[i] = inf;
17             vis[i] = 0;
18             for (int j = 1; j <= n; j++)
19                 Map[i][j] = inf;
20         }
21         while (m--) {
22             scanf("%d%d%d", &a, &b, &c);
23             Map[a][b] = min(Map[a][b], c);
24             Map[b][a] = Map[a][b];
25         }
26         int s = 1;
27         int e = n;
28         Dist[s] = 0; /*开始时忘了*/
29         vis[s] = 1; /*开始时忘了*/
30         while (s != e) {
31             Min = inf;
32             for (int i = 1; i <= n; i++) {
33                 if (Map[s][i] != inf && !vis[i]) {
34                     Dist[i] = min(Dist[i], Dist[s] + Map[s][i]);
35                 }
36                 if (Dist[i] < Min && !vis[i]) {
37                     nex = i;
38                     Min = Dist[i];
39                 }
40             }
41             /*if (Dist[nex] == inf) break;*/
42             s = nex;
43             vis[nex] = 1;
44         }
45         printf("%d\n", Dist[e]);
46     }
47     return 0;
48 }

```

1009 Choose the best route

AC代码

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  #define inf 0x7fffffff
4  int Map[1010][1010];
5  int vis[1010];
6  int Dist[1010];
7  int n, m;
8  int s, e;
9  int p, q, t;
10 int w;

```

```

11  int i, j;
12  int Min;
13  int nex;
14
15  int main()
16  {
17      while (scanf("%d%d%d", &n, &m, &e)!=EOF) {
18          for (i = 0; i <= n; i++) {
19              Dist[i] = inf;
20              vis[i] = 0;
21              for (j = 1; j <= n; j++)
22                  Map[i][j] = inf;
23          }
24          while (m--) {
25              scanf("%d%d%d", &p, &q, &t);
26              Map[p][q] = min(Map[p][q], t);
27          }
28          s = 0;
29          scanf("%d", &w);
30          while (w--) {
31              int temp;
32              scanf("%d", &temp);
33              Map[s][temp] = 0;
34          }
35          vis[s] = 1;
36          Dist[s] = 0;
37          while (s != e) {
38              Min = inf;
39              for (i = 1; i <= n; i++) {
40                  if (Map[s][i] != inf)
41                      Dist[i] = min(Dist[i], Dist[s] + Map[s][i]);
42                  if (!vis[i] && Dist[i] < Min) {
43                      nex = i;
44                      Min = Dist[nex];
45                  }
46              }
47              if (Min == inf)break;
48              s = nex;
49              vis[nex] = 1;
50          }
51          if (Dist[e] == inf)printf("-1\n");
52          else printf("%d\n", Dist[e]);
53      }
54      return 0;
55  }

```

1010 Here We Go(relians) Again

未AC

1011 Einbahnstrasse

未AC

1012 In Action

未AC

1013 最短路径问题

未AC
