

PAT (Advanced Level) Practise 1003 解题报告

dijsktra DFS

GitHub

- [问题描述](#)
- [解题思路](#)
- [代码](#)
- [提交记录](#)

问题描述

Emergency (25)

时间限制 400 ms

内存限制 65536 kB

代码长度限制 16000 B

判题程序 Standard

作者 CHEN, Yue

As an emergency rescue team leader of a city, you are given a special map of your country. The map shows several scattered cities connected by some roads. Amount of rescue teams in each city and the length of each road between any pair of cities are marked on the map. When there is an emergency call to you from some other city, your job is to lead your men to the place as quickly as possible, and at the mean time, call up as many hands on the way as possible.

Input

Each input file contains one test case. For each test case, the first line contains 4 positive integers: N (≤ 500) - the number of cities (and the cities are numbered from 0 to $N-1$), M - the number of roads, C_1 and C_2 - the cities that you are currently in and that you must save, respectively. The next line contains N integers, where the i -th integer is the number of rescue teams in the i -th city.

Then M lines follow, each describes a road with three integers c1, c2 and L, which are the pair of cities connected by a road and the length of that road, respectively. It is guaranteed that there exists at least one path from C1 to C2.

Output

For each test case, print in one line two numbers: the number of different shortest paths between C1 and C2, and the maximum amount of rescue teams you can possibly gather.

All the numbers in a line must be separated by exactly one space, and there is no extra space allowed at the end of a line.

Sample Input

```
5 6 0 2
```

```
1 2 1 5 3
```

```
0 1 1
```

```
0 2 2
```

```
0 3 1
```

```
1 2 1
```

```
2 4 1
```

```
3 4 1
```

Sample Output

```
2 4
```

大意是：

n个城市m条路，每个城市都有救援小组。已知各边权值，给定起点和重点，求从起点到重点的最短路径条数以及最短路径上的救援小组数目之和。

$n \leq 500$.

解题思路

最短路径，可以用dijkstra或深搜（数据不大）。dijkstra多记录一下路径数和小组总数。

代码

dijsktra

```
1.  #include<iostream>
2.  #include<cstdio>
3.  #include<cstring>
4.  using namespace std;
5.  int main()
6.  {
7.      //freopen("in.txt","r",stdin);
8.      int i,j,k,n,m,s,t,c1,c2,a[501][501],b[501],c[501],d[501],e[501],f[501]={0};
9.      memset(a,0x3f,sizeof(a));
10.     memset(d,0x3f,sizeof(d));
11.     cin>>n>>m>>c1>>c2;
12.     for (i=0;i<n;i++) cin>>b[i];
13.     for (i=0;i<m;i++)
14.     {
15.         cin>>j>>k>>t;
16.         a[j][k]=a[k][j]=t;
17.     }
18.     c[c1]=b[c1];
19.     d[c1]=0;
20.     e[c1]=1;
21.     for (i=0;i<n;i++)
22.     {
23.         s=0x3f3f3f3f;
24.         t=-1;
25.         for (j=0;j<n;j++)
26.             if ((f[j]==0) && (d[j]<s))
27.             {
28.                 t=j;
29.                 s=d[j];
30.             }
31.             if (t== -1) break;
32.             f[t]=1;
33.             for (j=0;j<n;j++)
34.                 if (f[j]==0)
35.                     if (d[j]>d[t]+a[t][j])
36.                     {
37.                         d[j]=d[t]+a[t][j];
38.                         c[j]=c[t]+b[j];
```

```

39.         e[j]=e[t];
40.     }
41.     else
42.     if (d[j]==d[t]+a[t][j])
43.     {
44.         e[j]+=e[t];
45.         if (c[j]<c[t]+b[j]) c[j]=c[t]+b[j];
46.     }
47. }
48. cout<<e[c2]<<' '<<c[c2];
49. return 0;
50. }

```

DFS

```

1.  #include<iostream>
2.  #include<cstdio>
3.  #include<cstring>
4.  using namespace std;
5.  int n,m,s,t,maxt,mind,c1,c2,d,a[501][501],b[501]={0},c[501]={0};
6.  void dfs(int k)
7.  {
8.      int i,j;
9.      if (k==c2)
10.     {
11.         if (d<mind)
12.         {
13.             mind=d;
14.             maxt=t;
15.             s=1;
16.         }
17.         else if (d==mind)
18.         {
19.             s++;
20.             if (t>maxt) maxt=t;
21.         }
22.     }
23.     c[k]=1;
24.     for (i=0;i<n;i++)
25.     if ((c[i]==0)&&(a[k][i]>0))
26.     {
27.         d+=a[k][i];
28.         t+=b[i];
29.         dfs(i);

```

```

30.         d-=a[k][i];
31.         t-=b[i];
32.     }
33.     c[k]=0;
34. }
35. int main()
36. {
37.     //freopen("in.txt","r",stdin);
38.     int i,j,k;
39.     memset(a,-1,sizeof(a));
40.     cin>>n>>m>>c1>>c2;
41.     for (i=0;i<n;i++) cin>>b[i];
42.     for (i=0;i<m;i++)
43.     {
44.         cin>>j>>k>>t;
45.         a[j][k]=a[k][j]=t;
46.     }
47.     maxt=0;
48.     mind=0x7FFFFFFF;
49.     t=b[c1];
50.     d=0;
51.     dfs(c1);
52.     cout<<s<<' '<<maxt;
53.     return 0;
54. }

```

提交记录

dijsktra

测试点	结果	用时(ms)	内存(kB)	得分/满分
0	答案正确	2	1408	10/10
1	答案正确	4	1408	3/3
2	答案正确	6	1408	3/3
3	答案正确	6	1408	3/3
4	答案正确	5	1388	3/3
5	答案正确	8	1408	3/3

DFS

测试点	结果	用时(ms)	内存(kB)	得分/满分
0	答案正确	3	1280	10/10
1	答案正确	5	1408	3/3
2	答案正确	5	1404	3/3
3	答案正确	7	1408	3/3
4	答案正确	8	1396	3/3
5	答案正确	140	1408	3/3