

原 CCF 201409-4 最优配餐

2018年08月24日 22:24:29 Tobo_Obito 阅读数：112

问题

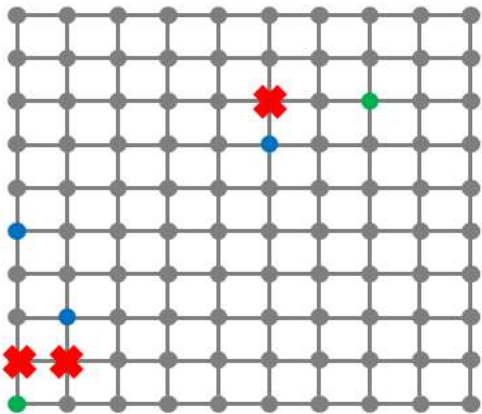
问题描述

栋栋最近开了一家餐饮连锁店，提供外卖服务。随着连锁店越来越多，怎么合理的给客户送餐成为了一个急需解决的问题。

栋栋的连锁店所在的区域可以看成是一个 $n \times n$ 的方格图（如下图所示），方格的格点上的位置上可能包含栋栋的分店（绿色标注）

些格点是不能经过的（红色标注）。

方格图中的线表示可以行走的道路，相邻两个格点的距离为1。栋栋要送餐必须走可以行走的道路，而且不能经过红色标注的点。



送餐的主要成本体现在路上所花的时间，每一份餐每走一个单位的距离需要花费1块钱。每个客户的需求都可以由栋栋的任意分店配送，每个分店

的限制。

现在你得到了栋栋的客户的需求，请问在最优的送餐方式下，送这些餐需要花费多大的成本。

输入格式

输入的第一行包含四个整数 n, m, k, d ，分别表示方格图的大小、栋栋的分店数量、客户的数量，以及不能经过的点的数量。

接下来 m 行，每行两个整数 x_i, y_i ，表示栋栋的一个分店在方格图中的横坐标和纵坐标。

接下来 k 行，每行三个整数 x_i, y_i, c_i ，分别表示每个客户在方格图中的横坐标、纵坐标和订餐的量。（注意，可能有多个客户在方格图中的同一个位

接下来 d 行，每行两个整数，分别表示每个不能经过的点的横坐标和纵坐标。

输出格式

输出一个整数，表示最优送餐方式下所需要花费的成本。

样例输入

10 2 3 3
1 1
8 8
1 5 1
2 3 3
6 7 2
1 2
2 2
6 8

样例输出

29

评测用例规模与约定

前30%的评测用例满足： $1 \leq n \leq 20$ 。

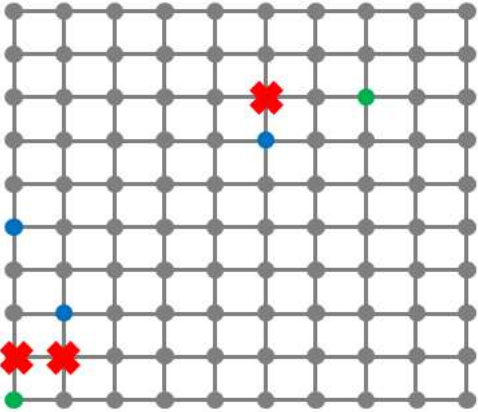
所有评测用例都满足：1<=n<=1000，1<=m, k, d<=n^2。可能有多个客户在同一个格点上。每个客户的订餐量不超过1000，每个客户所需要的餐量不超过1000，每个客户所需要的餐都能被送到。

问题分析

一看这道题是个地图题，想到了DFS和BFS，然后发现是求最优问题，果断尝试BFS，但觉得好像哪里不对劲。其实这道题就是求最短路径。然后就是正确的BFS = 。

首先明确一点：不论一个位置有多少个顾客，这个位置所有顾客的需求总量一定是由“最近”的一家分店承担的（可以用反证法证明）。然后就是正确的BFS = 。

不过，还有个20分的坑估计很多人会跳进去。注意评测用例规模：“1<=n<=1000，1<=m, k, d<=n^2。可能有多个客户在同一个格点上。每个客户的订餐量不超过1000，每个客户所需要的餐都能被送到。”，考虑一种容易考虑的情况（如下图）



若左下角(1,1)有唯一一家分店，右上角(1000,1000)有1000,000个顾客，每个顾客的订餐量为1000，无禁止通过的点。那么，从分店(1,1)走到(1000,1000)需要2000步左右；然后1000,000个顾客，每人订餐量1000，总订餐量10^9，总共花费2000 * 10^9 = 2 * 10^12，显然这超出了int的范围，使用long long类型，使用时注意与计算相关的变量也使用long long类型避免在默认类型转换时发生错误。如果程序其他部分编写没问题，把这个问题80分改到100分了。

代码

```
1 #include<stdio>
2 #include<queue>
3 using namespace std;
4
5 const int MAX_SIZE = 1000;
6
7 struct point_info{
8     int x,y;
9     long long step;
10 };
11
12 bool Legal(int x,int y,int n){
13     if(x<1||x>n||y<1||y>n)
14         return false;
15     return true;
16 }
17
18 long long customer_need[MAX_SIZE+1][MAX_SIZE+1] = {0LL}; //记录一个点所有顾客的总需求
19 int customers[MAX_SIZE+1][MAX_SIZE+1] = {0}; //记录同一点有几个顾客
20
21 int main(){
22     int n,m,k,d,x,y;
23     long long ans = 0,value;
24     queue<point_info> q;
25     bool ban[MAX_SIZE+1][MAX_SIZE+1] = {false};
26     point_info front, t;
```

登录


注册


×


https://blog.csdn.net/Tobi_Obito/article/details/82025911


2/9


```
29 | scanf("%d %d %d %d",&n,&m,&k,&d); 30 |
31 | while(m--){
32 |     point_info start;
33 |     start.step = 0LL;
34 |     scanf("%d %d",&start.x,&start.y);
35 |     ban[start.x][start.y] = true;
36 |     q.push(start);
37 | }
38 |
39 | for(int i=0;i<k;i++){
40 |     scanf("%d %d %lld",&x,&y,&value);
41 |     customer_need[x][y] += value;
42 |     customers[x][y]++;
43 | }
44 |
45 | while(d--){
46 |     scanf("%d %d",&x,&y);
47 |     ban[x][y] = true;
48 | }
49 |
50 | while(!q.empty()){
51 |     front = q.front();
52 |     q.pop();
53 |
54 |     if(customer_need[front.x][front.y]!=0LL){
55 |         k -= customers[front.x][front.y];
56 |         ans += front.step * customer_need[front.x][front.y];
57 |         if(k==0)
58 |             break;
59 |     }
60 |
61 |     for(int i=0;i<4;i++){
62 |         x = front.x+move[i][0];
63 |         y = front.y+move[i][1];
64 |         if(Legal(x,y,n)&&!ban[x][y]){
65 |             t.x = x;
66 |             t.y = y;
67 |             t.step = front.step + 1;
68 |             ban[x][y] = true;
69 |             q.push(t);
70 |         }
71 |     }
72 | }
73 |
74 | printf("%lld\n",ans);
75 |
76 | return 0;
77 | }
```


0
















想对作者说点什么

CCF-CSP—2014.09.—4 最优配餐 (BFS) 题解

阅读数 156

题目来源: [点击查看链接](#)[题目描述](#) 栋栋最近开了一家餐饮连锁店, 提... [博文](#) 来自: [摇摇乐的...](#)

CCF 201409-4 参考别人 100

阅读数 226

201409-4试题名称: 最优配餐时间限制: 1.0s内存限制: 256.0MB问题描述: [问题描述](#)... [博文](#) 来自: [java](#)

cfc 201409-4 最优配餐

阅读数 673

[问题描述](#) 栋栋最近开了一家餐饮连锁店, 提供外卖服务。随着连锁店越来越多, ... [博文](#) 来自: [人生如梦...](#)