任务一

问题考虑，本质为求图中任意两点间、任意一点和边上任意点间最短距离。任意两点间距离可由Dijkstra算法求解（无向正权图）。对每个点使用Dijkstra算法可以求得该点到其他点的最短距离，因此获得任意两点的最短距离的复杂度为O(mn)+O(n^3)。

而点和边上某点最短距离则略复杂，但略微思考不难将边上某点的到结点的距离得到如下公式：

记边ek=(vi,vj)上一点为P，vi、vj到某结点vt的最短距离为p(it)、p(jt)，ek的长度为w，(p到vi的距离)/w=l（分比系数，范围0-1），则p到vt的最短距离为min{lw+p(it),(1-l)w+p(jt)}，记为f\_kt(l)。这是因为，边上一点到某处，可以转化为先到边上某个端点，再到该处，从而有这个公式。

同时显然，|p(it)-p(jt)|<=w，并且f(l)是一个先增后减的折线，最大值在lm=(1+(p(jt)-p(it))/w)/2时取得，根据限制条件可知0<=lm<=1此时f\_ktmax=(p(it)+p(jt)+w)/2

因此，在求得任意两点间的最短距离后，还可求任意一边到任意一点的最短距离（定义边到点的最短距离为边上的点到该点最大的最短距离）。复杂度为O(mn)。

显然任意ek=(vi,vj)，f\_kt>=max{p(it),p(jt)}，因此求点到每边距离的最大值作为该点的最大距离（复杂度为O(mn)），再求最大距离的最小值即可（复杂度为O(n)）。

总结任务一思路：利用Dijkstra算法求得任意两点间最短距离，利用f\_kt=(p(it)+p(jt)+w)/2求得边ek到点vt的距离，进而得到每点到图的所有其他位置的最长距离，取最小的最长距离作为答案，算法复杂度为O(mn)+O(n^3)。

细节实现：

1、如何储存输入的数据？

由于Dijkstra算法需要多次搜索某结点的后继结点，因此使用正向表或邻接图的结构较为合适。这时每个边被储存了两边，因此求边到点的最短距离只需要选择i<j的(vi,vj)即可（注意不存在自环）。或者为了在求点边距离时方便增加m\*3的矩阵记录边数据，这样更容易阅读、方便实现，但是会增加录入的工作量，两者各有优劣。

2、Dijkstra算法最初的P(k)的无穷取多少？

显然大于max\_N\*max\_M\*max\_t即可，在本问题中取5000001即可

任务二

与任务一思路类似，只不过这次需要求边上任意一点ei=(vi1,vi2)到另一条边ej=(vj1,vj2)上任意一点的最短距离。

设长为w1的ei上一点p1距离vi1分比为（即w(vi1,p1)/w(vi1,vi2)=l1），长为w2的ej上一点p2距离vj1分比为。由任务一可求得vi1、vi2、vj1、vj2的最短距离。

类比任务一的转化思路可知，p1到p2的最短距离。求得取ej上一点p2到ei上任意一点的最小距离的最大值（即假定在p2点建医院，ei上的所有点到p2的最大距离）（记作）为的时候取得，记，，，，由计算和化简得到，，计算这步计算的复杂度是O(1)，计算出所有的复杂度是O(m^2)，注意。

接下来我们需要求以及，其中表示在ej上一点（分比为）作为医院时，任意位置到医院的最长时间，answer[j]表示ej上最优点作为医院的话，任意位置到医院的最长时间。

观察所知，对于每一个相同的j和不同的i，的每一段的斜率都是的-1、0、1倍，并且（纵）截距均为0.5的整数倍。因此在比较的时候，新产生的分界点以及原有的分界点（上文的c、d）均为的整数倍，且处于[0,1]内。因此这些分界点有不超过种情况，最多由条折线构成，每条折线的长度均为的整数倍。因此求每个的时候，可用一个Fj[4\*w2]的矩阵记录这条折线的斜率（记-1、0、1表示-w2、0、w2），再记录下f(0)即可。不难想到，，求answer[j]的复杂度为O(m)，求answer[]的复杂度为O(m^2)，求的复杂度为O(m)，求所有的复杂度为O(m∑w)。answer[]中的最小值即为所求答案。

此算法的总复杂度为O(m^2)+O(m∑w)+O(n^3)，其中∑w表示所有边的权值之和。或者可以写成O(m^2w)+O(n^3)，其中w表示边的平均权值。