**迪杰斯特拉派初赛**

|  |  |
| --- | --- |
| 团队名称： | 魂斗罗 |
| 团队成员： | 洪树斌-445221199401131213-上海交通大学 |
| 王文琪-440-上海交通大学 |

1. **赛题解析**

题目内容为蚂蚁取得玉米与水果的过程，要求：1、经过路径花费最少；2、经过玉米间与水果间；3、经过两条必经路径；4、要求经过储藏间的个数小于9个；5、不能经过两条指定路径。总的来说，可以归结为带限制条件的最优路径问题。

因此，可以将问题简化为两个条件：1、经过并经点与边；2、经过的点数需要小于限制的点数。之后将不能经过的路径从一开始的数据输入时去除，再进行最优路径的寻路即可。

1. **算法原理**

**2.1 核心算法**

我们小组采用的核心算法是利用“状态图层拆分法”，即依据必须经过的点与边对图层进行拓展拆分为多个图层，将一张图拓展为包含多个图层的大图。

**2.1.1 必经点**

以两个必经点为例子。如图1所示，从点0到点4，要求必经过点2，3，则显然路径为0->2->3->4。



图1 简单示意图

此时可以采用状态图层拆分法，依据点2和点3进行拆分为四个状态图层，如表1所示：

表1 拆分后图层状态表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 图层 | 00 | 01 | 10 | 11 |
| 状态 | 未经过点2，3 | 经过点2 | 经过点3 | 经过点2和3 |

其中图层00可以到达01和10，01和10之间不联通，01和10均可与图层11连通。

对应的可以将图1拆分复制为四个图层，如图2所示。



图2 经过必经点拆分后的总图

如图2所示，则题目可演变为从点0到点4(11)的最优路径，在图层00（即第一层）到图层01之间的唯一通路是2->2(01),设置该路径花费为0，而图层00到10之间的唯一通路是3->3(10)，图层01到图层11之间的通路为3(01)->3(11)，图层10到图层11之间的通路为2(10)->2(11)。这样经过拆分后可以确保从点0出发到点4(11)，至少为经过2和3（或者他们对应的拆分点）。

**2.1.2 必经边**

必经边与必经点的情况类似，唯一的区别在于连通图层之间的边加入了权值，对于图1，以必经过边23为例子。

此时根据边23进行状态图层拆分，对于N个限制条件，需要拆分为个状态图层。只需要经过一条必经边时，则拆分为2个状态。



图3 经过必经边拆分后的总图

如图3所示，则原题目变为从0到4(1)的最优路径，连通图层0和图层1有两条路径，分别是23(1)，和32(1)，它们的边长度与23相同，则无论经过23(1)还是32(1)，都等价于经过23，此时则可以确保最优路径包含边23。

**2.2 实现说明**

在本题目中，本小组采用“状态图层拆分法”，题目如图4所示。

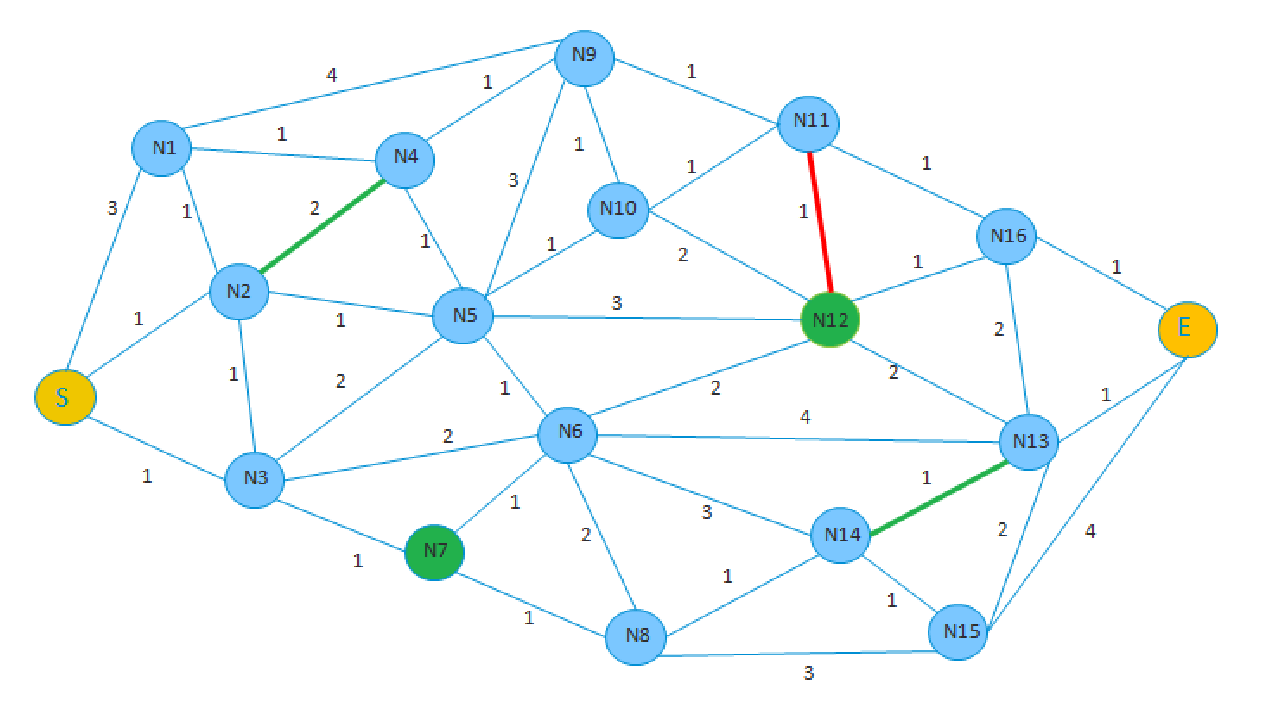


图4 题目路径图

其中包含两个必经点，两条必经边，对于不能经过的边，在则认为该边不联通即可。则可将图层进行拓展为个图层。

本小组首先实现方法readData(String)录入数据，读取目录test/data.txt中的内容，采用邻接矩阵来保存图的信息。

进而利用方法graphTrans(int)对邻接矩阵进行拓展。图中共18个点，进行拓展后，则邻接矩阵规模变为。之后利用迪杰斯特拉法dijkstra()寻找最优路径，并记录经过的点的个数，来判断找到的路径是否符合条件，当无法满足条件时，输出次优解。

1. **程序说明**

**3.1 输入数据说明**

本小组采用文件录入数据的方法，将测试数据填入test/data.txt中，其中内容为：

点个数 边个数

边起点 边终点 边权值

···

边起点 边终点 边权值

允许经过的限制点数 必经过点个数 必经过边个数

必经过的点

···

必经过的点

必经过的边

···

必经过的边

其中0默认为起点，最大的点默认为终点。

**3.2 运行环境与方法**

本小组的编程环境为：windows+jdk1.8。

运行方法：将压缩包中的test文件夹与zte.jar拷贝到机器环境下，修改test/data.txt的录入数据，运行“java –jar zte.jar”，则可得到最优路径的输出结果。

**3.3 输出说明**

当设置经过点数为9时，会出现无结果，此时输出两个次优路径，如图5所示。

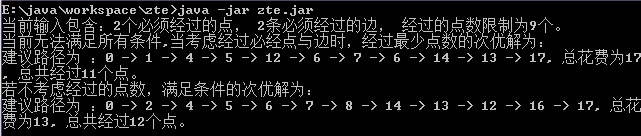


图 5 运行输出结果

其中得到两个次优解。其中由于限制点数为9，则考虑必经点和必经边，得到经过最少点为11的次优解；以及不考虑经过点数的次优解。

当修改限制点数为13时，可得到最优解，如图6所示。

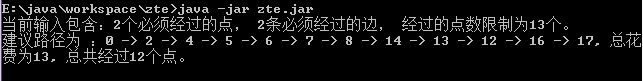


图6 修改限制点数后的最优解

1. **致谢**