

00 第九次作业指导书 V3（截止到周六中午 12:00）

V2 改动：

更正了诸多笔误=w=

详见批注

相对上次作业改动：

新添城市地图 6)~8)

新添出租车 7)~8)

重写“设计要求”一章。

V3 添加了一些要约，简化实验逻辑。

一、 主题

模拟的出租车呼叫应答系统（系统的基本时间单位 100ms）。

由设计者完成出租车模拟与应答系统。

由测试者借助设计者的接口完成乘客叫车模拟与作业测试。

二、 一些定义

1. 城市地图

- 1) 使用网格区域来模拟城市地图。所有的道路要么是水平方向，要么是垂直方向，如果两个点之间有道路，则这两个点之间存在一条连接。
- 2) 城市地图通过文件输入，文件格式为文本文件。文件内容为80行字符串，每行有80个字符，每个字符为0到3之间的整数，表示一个 80×80 的邻接矩阵 $A_{80 \times 80}$ 。其中第 i 行的第 j 个字符 $A_{i,j}$ 表示第 i 行和第 j 列的交叉点，记录的是地图中 (i,j) 到与右方交叉点和下方交叉点的连接情况，若
 - a) $A_{i,j}=0$ 表示 (i,j) 与 $(i,j+1)$ 和 $(i+1,j)$ 均无连接。
 - b) $A_{i,j}=1$ 表示 (i,j) 与 $(i,j+1)$ 有连接，但与 $(i+1,j)$ 无连接。
 - c) $A_{i,j}=2$ 表示 (i,j) 与 $(i,j+1)$ 无连接，但与 $(i+1,j)$ 有连接。
 - d) $A_{i,j}=3$ 表示 (i,j) 与 $(i,j+1)$ 和 $(i+1,j)$ 均有连接。

提示： (i,j) 与 $(i-1,j)$ 和 $(i,j-1)$ 的连接情况由 $A_{i-1,j}$ 和 $A_{i,j-1}$ 描述。

- 3) 文件需要由测试者确保地图上的所有的点都是连通的，即整个图是连通图，但不能存在点与图外的点有连接，例如 $A_{80,1}$ 不能是 2 或 3， $A_{80,80}$ 只能是 0。
- 4) 支持在运行过程中动态关闭或打开一些地图上已有的连接边，受影响边的总数由测试者确保不能超过 5，任意时刻需要由测试者确保地图上的所有的点都是连通的。
- 5) 对地图上每条打开的连接边，定义单位时间窗内从该边经过的出租车数为该边的流量，时间窗长度建议设为 50ms，但不得大于 100ms。

- 6) 为简化问题，不必实现 PPT 中关于“道路方向”以及“道路容量”的相关要求。
- 7) 地图新添“道路交叉方式”属性，即每个交叉结点现在可以是平面交叉或立体交叉。为保留原有地图的输入方法，程序应单独增加一文件输入，定义道路交叉情况。文件内容为 80 行字符串，每行有 80 个字符，每个字符为 0 或 1。0 表示立体交叉，1 表示平面交叉。理论上，两者的数量应该各占一半。该文件与地图文件一样，由测试者保证输入文件的有效性。交叉方式与结点的邻接关系无关，并不是十字路口才能设置立体交叉，直道也允许拥有立体交叉属性。
- 8) 地图新添“红绿灯”属性。红绿灯存在于每个平面交叉的丁字/十字路口，对于 L 型，直行的道路不设置红绿灯。换言之，当地图上的某个结点为平面交叉，且其相邻结点为 3 个或 4 个时，此路口才设置红绿灯。红绿灯的变化间隔在设置为 300ms，每经过变化间隔切换灯的颜色。需要注意的是，每个路口有两组灯，南北方向为一组，东西为另一组。同组灯的颜色始终相同，不同组之间颜色相异。各个路口红绿灯的初始化状态可由编程者自行决定，但是要求在说明文档中加以说明。为简化问题，忽略黄灯的情况。
- 编者注：对于立体交叉和平面交叉，可以近似地理解为，立体交叉不受灯的限制按照上次作业处理，平面交叉需要实现新增的红绿灯控制功能。

批注 [卢嵩1]: 补充说明交叉方式的输入

批注 [卢嵩2]: 更新了更加严格的设置红绿灯路口的定义

批注 [卢嵩3]: 新增红绿灯初始化状态的要求(其实就是没有要求)

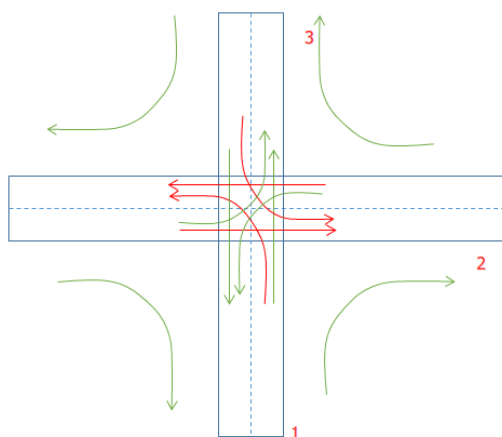
2. 出租车

- 出租车限定为 100 个，出租车起始位置由设计者通过随机数生成随机分布。
- 出租车行驶一条格子边的时间为 100ms。
- 每辆出租车的状态有四种：正在服务（表明在运行且车内有乘客）、即将服务（表明在运行且车内无乘客且系统分配了乘客）、等待服务（表明在运行且车内无乘客且系统未分配乘客）、停止运行（表明不在运行）。出租车初始状态为等待服务状态。
- 出租车的状态会有以下四种转换：
 - 出租车处于等待服务状态时系统为其分配了乘客，则进入即将服务状态。
 - 出租车处于等待服务状态时连续运行 20s 后，需要停止运行 1s（休息），然后再次运行。
 - 出租车处于即将服务状态时到达用户等待位置后，需要停止运行 1s（接客），然后再次运行。
 - 出租车处于正在服务状态时到达用户的目的地后，需要停止运行 1s（送客），然后再次运行。
- 停止运行仅出现于上述后三种情况，在即将服务、正在服务和停止运行状态下不能响应乘客请求。
- 出租车的行走方式有以下两种：
 - 在等待服务状态时，出租车如果遇到分支，选择流量最小的边行走，如果有多条流量最小的边，可随机选择一条分支边行走。
 - 在即将服务和正在服务状态时，要求出租车按照最短路径行走，如果最短路径对应多条可以行走的边，选择流量最小的边行走，如果仍有多条流量最小的边，可随机选择一条边行走。
- 出租车有信用积累，初始所有车信用为 0，每抢单一次会使其信用度加 1，

每成功服务顾客一次会使其信用度加 3。

8) 新增红绿灯对出租车行驶的影响:

在路口有红绿灯的情况下, 其对车辆的影响与真实情况相仿:



- 当出租车在路口进行右转时, 不受红绿灯限制。如图中从 1 到 2.
- 当出租车直行或左转(包括掉头)时, 遇到绿灯可以行驶。遇到红灯需等候到其变为绿灯为止。如图中, 若车辆从 1 驶向 3, 若南北向灯为红色则需要等待。
- 最短路径搜索时是按照静态策略, 即纯粹从距离出发, 忽略红绿灯可能带来的额外时间成本。
- 无论是最短路径还是随机, 车的路径选择与红绿灯无关。
- 灯不改变车的运动状态: 即使车被红绿灯阻塞, 但运动状态不变, 不会因此变成 stop。
- 车如果被红绿灯阻塞后, 下一次选择时, 可以重新选择路径, 也可以继续等待。

批注 [卢嵩4]: 补充了关于掉头的说明

3. 乘客请求响应

- 乘客在任意一点 C 向系统发出呼叫请求后, 系统只把呼叫请求发送给在以 C 为中心的 4×4 区域 (上下左右均延伸 2 个格子, 但不超出 80×80 的大范围) 里行驶的出租车。呼叫请求包括点 C 的坐标和目的地坐标信息, 如果目的地无效, 则系统拒绝响应该请求。
- 处于等待服务状态的出租车只要在抢单窗口内, 进入以 C 为中心的 4×4 区域里就会收到请求, 只要收到请求就会抢单。
- 系统以 3s 为一个抢单窗口, 在 3s 时间窗口关闭时进行选择。
- 如果 3s 内无出租车对系统应答则视为无车响应, 系统告知乘客无可用出租车。
- 如果有车响应, 系统为乘客从当前窗口中抢单的出租车中自动优先选择, 在抢单窗口关闭时刻选择处于等待服务中信用度最高的出租车。如果没有满足条件的出租车, 则系统告知乘客无可用出租车, 否则, 若有多辆信用度相同的出租车, 选择距离用户最近的一辆, 若还是有多辆二者均相同的出租车,

可以随机选择一辆。

- 6) 出租车一旦抢单，系统就会记录，即使出租车在抢单窗口关闭时已经离开了以 C 为中心的 4×4 区域也视为有效。
- 7) 一辆出租车可以在同一时刻抢不止一单，但每辆车一次服务只能够处理一个请求。
- 8) 一个乘客一旦发出请求，在接下来的 3s 抢单时间窗内，来自该地点发出的其他请求视为其他乘客发出的请求，与该乘客独立处理。

三、 设计要求

1. 对设计者的要求

- 1) 要求采用本讲介绍的方法进行分析和设计。
- 2) 针对上次作业中存在的代码问题和过程问题，修正 Bug，改进过程规格，并在 readme 中加以说明。
- 3) 要求对所有方法书写过程规格，Thread 类的 run 方法除外。对新增方法，应先写规格再实现代码；对已有方法，应确保规格和实现的一致性。
- 4) 要求使用多线程和线程安全设计。提供线程安全的乘客请求队列，供测试使用。注意请求队列容量不得小于 300 个。
- 5) 要求提供手段来让测试线程访问所有处于等待服务状态的出租车对象。
- 6) 要求提供查询出租车状态的方法作为测试接口（这里的测试接口不是 JAVA 里面的接口），至少能够在任意时刻给出出租车的状态信息，信息包括但不限于当前的时刻，出租车当前坐标。
- 7) 要求提供道路关闭或打开的功能，需要在 readme 中给出测试者使用该功能的方法，实现方法可自行定义。
- 8) 要求使用独立的线程类来模拟各个平面交叉路口的红绿灯控制。
- 9) 要求实现 repOK 方法，且必须与不变式逻辑一致。
- 10) 要求形成设计文档来阐述，涵盖Overview、过程规格、表示对象、抽象函数、不变式，且要求Overview中明确阐述类的抽象数据，且与抽象函数中的抽象数据一致；表示对象必须与代码中的类属性变量一致；不变式务必能够概括对象的有效性，且能够与过程规格相配合。

2. 对测试者的要求

1) 沿用上次的测试策略：通过测试线程输入、模拟乘客请求、模拟道路的关闭/打开，开展功能测试，并使用和之前一致的bug报告策略。

2) 要求针对每个类都进行不变式检查测试：每个方法被至少调用一次，在满足规格要求的前提下，测试不受限制，按照方法为单位，只要后置条件不满足或者不变式不满足，记为一个wrong类型的bug。

3) 测试者首先检查repOK是否满足其规格声明的不变式。如果满足，则测试中可直接使用该方法来进行不变式有效性检查；否则，报告一个repOK的wrong类型bug，测试者自己实现相应的repOK来开展测试（为避免不必要的申诉，尽量在实现者的repOK基础上补充或完善）。

4) 提交评价时，测试者必须提供测试使用的地图和测试部分的代码，以供

批注 [卢嵩5]: 新添之前遗漏的说明文档要求 Orz，感觉文档的工作量不比代码小，建议各位合理分配时间。

实现者复现问题。对所有测试者，测试repOK的代码也必须全部提交到课程网站，届时系统将会开放相应的功能。

5) 相应的功能点有扣分的限制，上次作业实现的功能最多扣10分，道路信息功能最多扣20分，道路行驶控制功能最多扣20分，设计要求未实现(红绿灯控制，道路交叉属性，repOK方法，类规格，设计文档不完善的)最多扣20分。过于短小的方法（15行以内）最多只报告一个wrong类型Bug。

批注 [卢嵩6]: 教主：我就是想看看你们有没有认真测啦~