

北京邮电大学 数据结构课程设计	产品名称	产品版本	密级
	Covid-19 旅行模拟系统	V1.0	公开
	定稿日期： 2020 年 7 月 9 日		共 4 页



Covid-19 旅行模拟查询系统 V1.0

评价和改进意见

文档作者： 史嘉程

学 院： 计算机学院

专 业： 计算机类

班 级： 2018211318

指导教师： 杨 俊老师

1 系统评价

1.1 软件交互良好

① 本系统的用户交互界面设计的精致且功能完善，且操作简单，用户不需要复杂的说明即可上手操作。

② 本系统的特色即具有美观、简洁的旅行状态输出模块，即用户可以在地图上实时查询当前各个旅客的旅行状态，包括所在位置、交通工具、班次等，给用户较为良好的操作体验。

1.2 核心功能完善

本系统采用了分层思想进行模块化设计，整个系统分为以下两个层次：

- a) 用户交互层
- b) 应用服务层

这两个层次之间相互提供服务。其中应用服务层实现实时响应用户所提出的需求并负责主要的运算工作；用户交互层用于展示用户界面、接收用户提出的需求、实时输出计算结果，展示模拟旅行信息等功能，以图形化用户界面的方式实现。

在用户交互层中，用户可以选择旅客、出发地、目的地等关键信息，并且选择“最小风险策略”或者“限时最小风险策略”来获得旅行方案。输入任务将会传递给应用服务层来制定相应的旅行方案。

本系统能够图形化模拟旅行过程。通过在中国地图上展示旅客的关键信息，如出行方式，目的地，出发地等，并辅以文字的更加详细的旅行方案信息，直观了解所有旅客的实时状态。

此外，本系统支持在旅客旅行途中对旅行方案进行调整，即修改旅客出行计划。应用服务层将通过新的任务，根据旅客当前所在地即时间策划出新的旅行方案。该功能的实现能使本系统更具灵活性，也更加符合真实的用户应用场景。当然，旅客只有在抵达某个城市后才能进行旅行计划调整。

本系统还存有文本档格式的用户日志，用户可以通过用户日志来查看用户做出的关键操作以及旅行过程中的状态改变记录。

最后，本系统具有优秀的健壮性，即能够一些常见的异常，使系统更加健壮，

符合日常使用场景。如应对出发地、目的地相同，限定时间内无法到达目的地，用户在城市 A 提交了从城市 B 出发的任务等。

1.3 系统整体性能较高

虽然本人使用了 Python 进行编程，而 Python 语言在性能上与其它主流语言有一定的差距，但是本系统还是具有较高的性能。我在开发的时候格外重视软件性能和算法效率，并不断优化。

本系统只有两个线程，第一个是 GUI 的线程，第二个是时间的线程。在这两个线程中，都采用了一定的方式减少了他对系统的占用，使软件能够流畅，高效地运行。

在算法上，因为城市数量较少、交通工具的数量也受到了限定，因此我采用了 DFS 算法，并进行了一定程度的优化，如修枝等。在小规模的数据中，DFS 算法能在保证在短时间内得到最优解。

用户在实际使用时感受不到执行算法带来的卡顿。

2 改进意见

2.1 模块划分不够合理

本系统在模块划分方面上仍有提高的空间。因为本人是第一次开发一个完整的项目，在编写程序的习惯上仍保留了大一大二面向过程编程的问题，因此出现了模块之间的重叠，杂糅的情况。

2.2 算法效率仍需优化

众所周知，本系统采用的 DFS 是一种暴力的，类似穷举的算法。即时采用了修枝的优化方式，其性能仍然比其他算法低出不少。因为本系统的数据规模比较小，总共只有 12 个城市，每个城市的平均交通工具班次数量小于 20 个（考虑到大城市和小城市的区别以及地理位置产生的交通方式的单一化），因此采用 DFS 算法基本感受不到性能上的劣势。

但是真实情况的数据量远远大于本课程设计需要的数据量。在应对大数据量的情况下，DFS 不利于系统的扩展。因此应当采用效率更高的算法，或者把算法修改为一些启发式算法，比如可以把目光转向模拟退火算法(SA)、遗传算法(GA)、蚁群算法(ACO)、人工神经网络(ANN)等。

2.3 用户交互界面有较大提升空间

本系统的 GUI 主要由 Pyqt5 和 pyecharts 两个拓展库开发。由于本人能力和 pyqt5 的限制，用户界面的外观很难做到华丽美观，不能给用户带来赏心悦目的使用体验。

在旅行展示模块（地图）上，因为 pyecharts 的限制以及对前端可视化知识的缺乏，仅能可视化一些关键信息，如出发地、目的地、交通工具类型，所展示的信息较少，这一定程度上影响了用户交互的体验。