蚁群算法

1. 实验目的

使用蚁群算法来解决TSP实际问题。

1. 实验内容

用遗传算法求解31个城市的TSP问题（从第一个城市出发最后再回来）

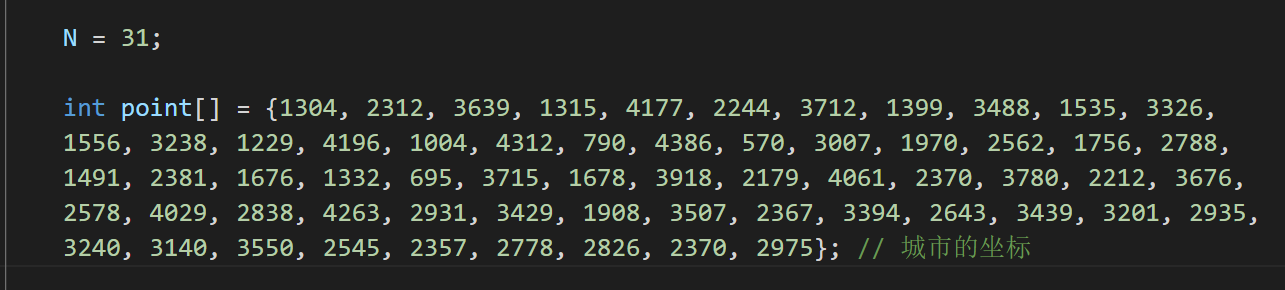
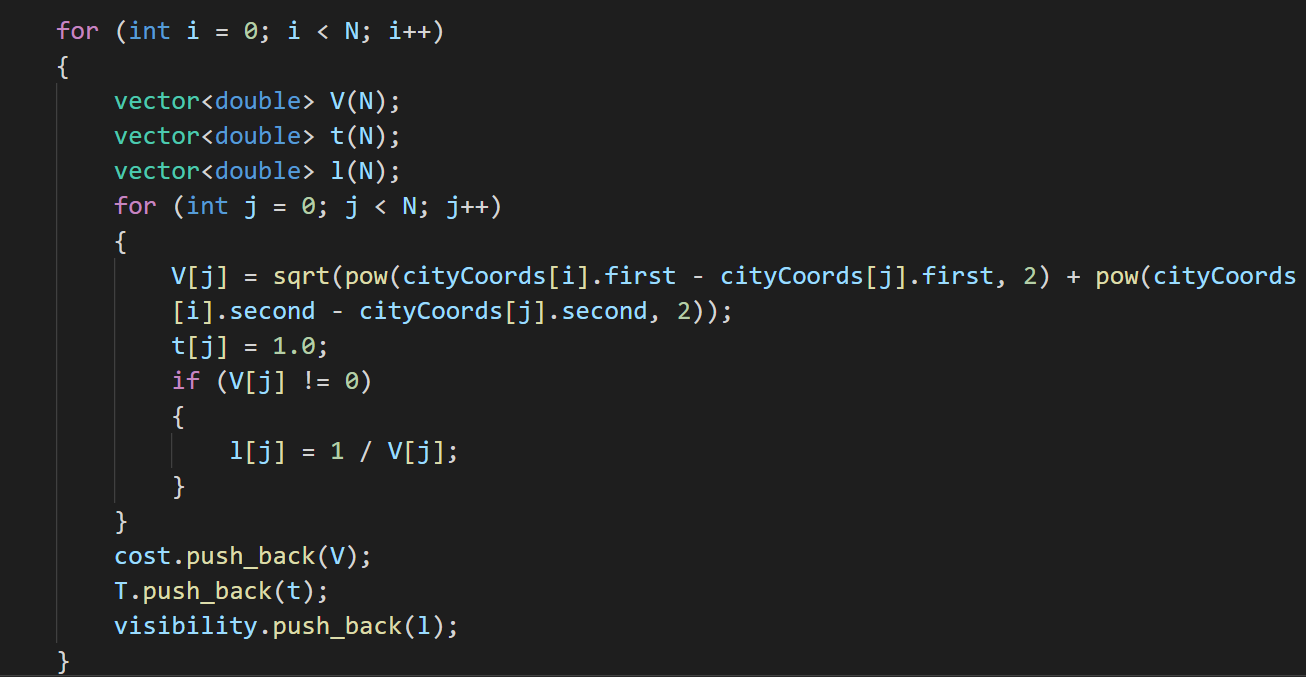
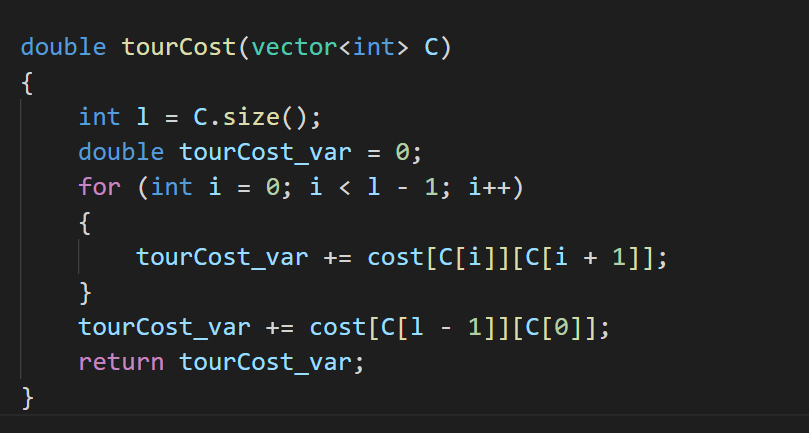
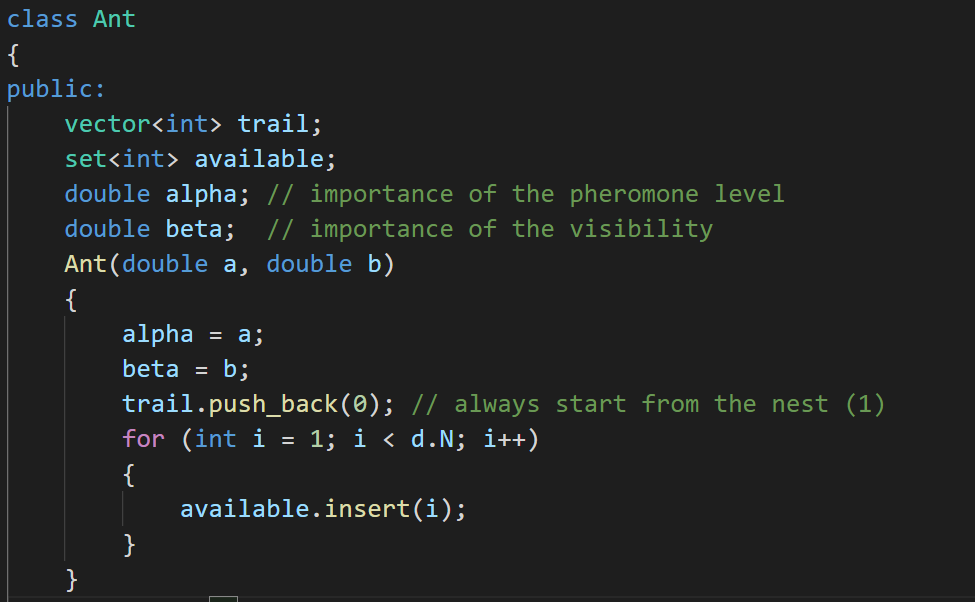
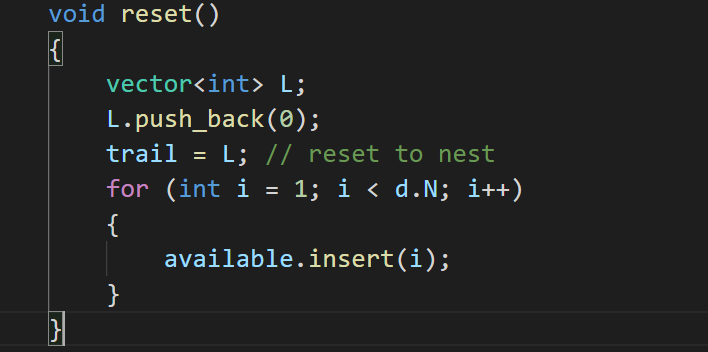
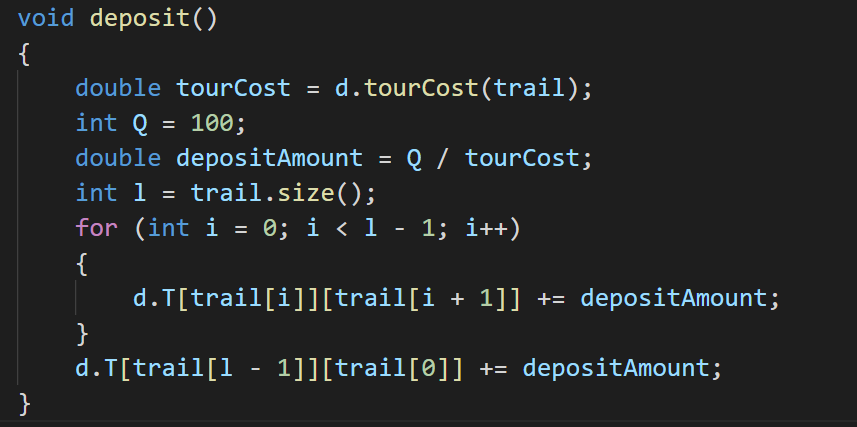
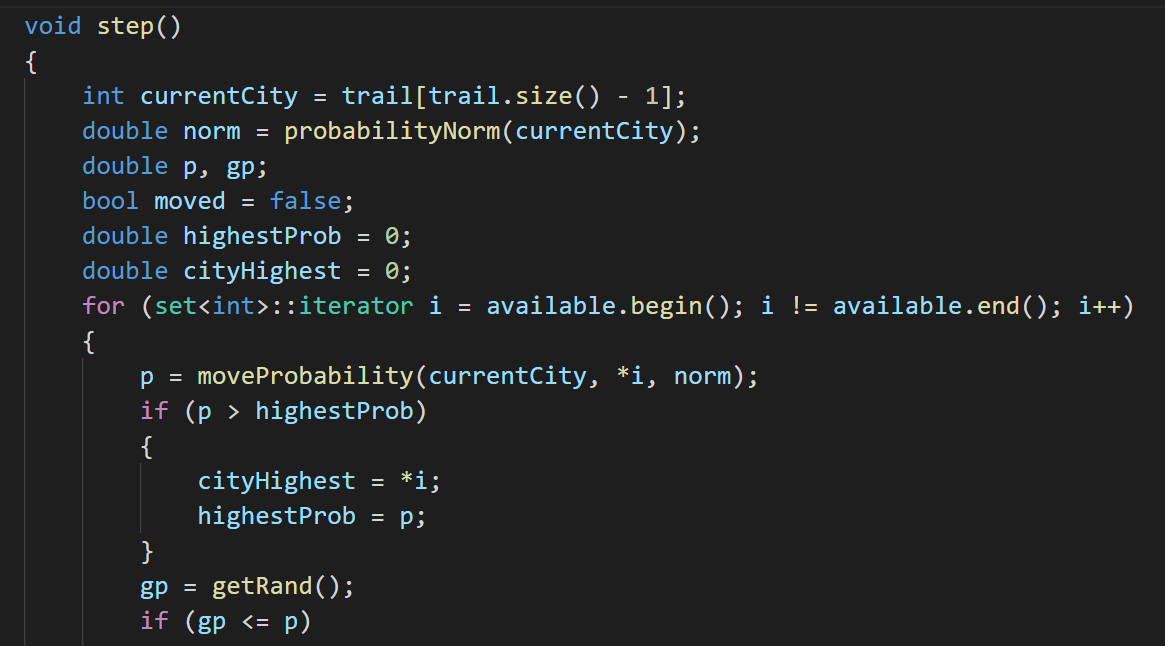
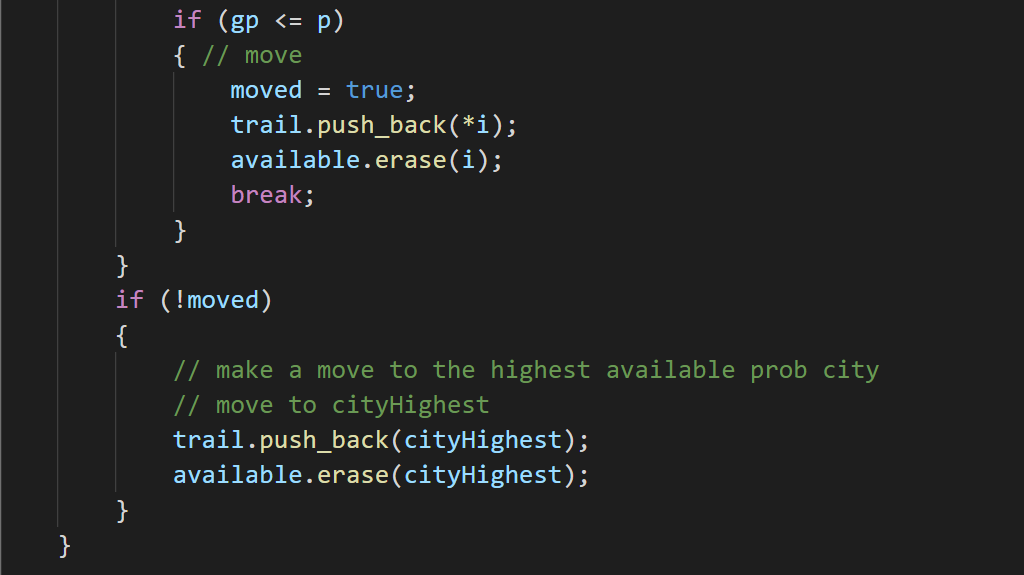
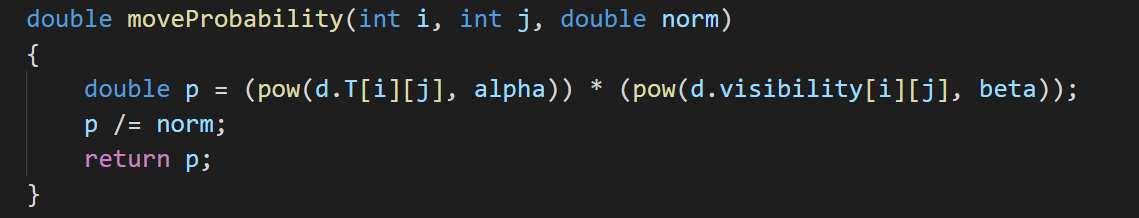
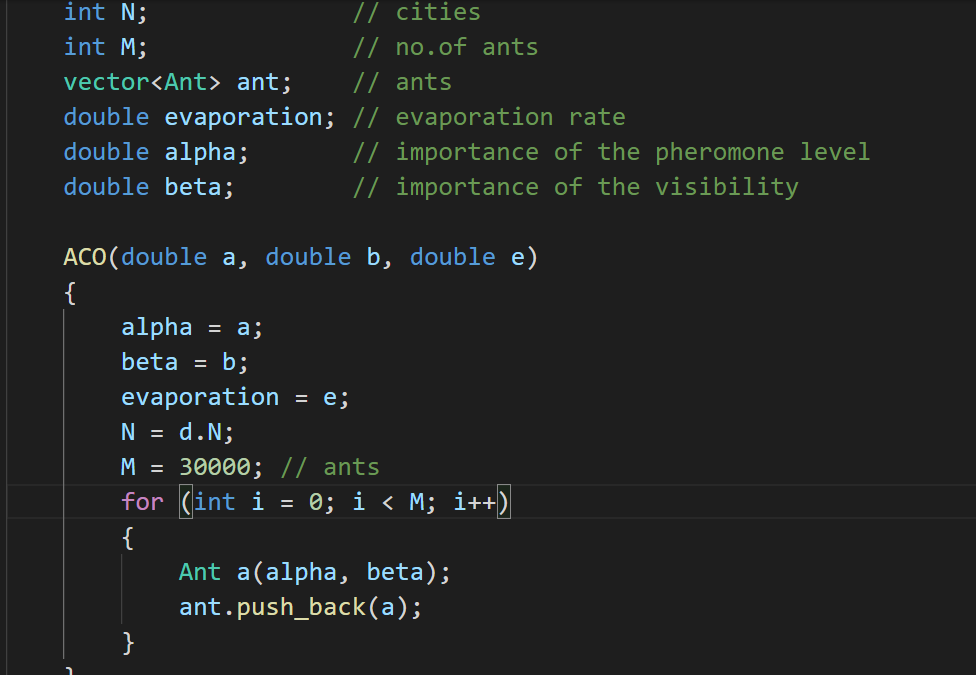
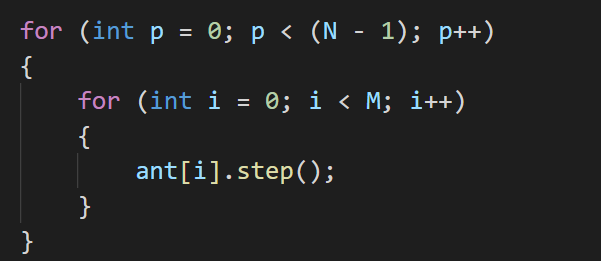
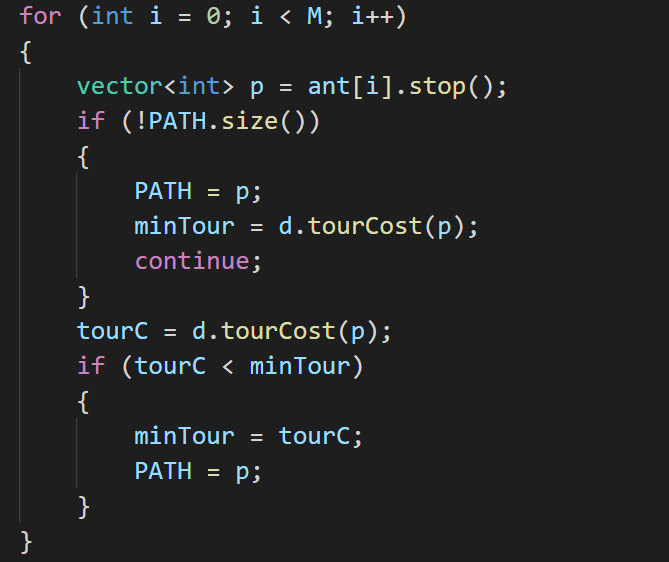
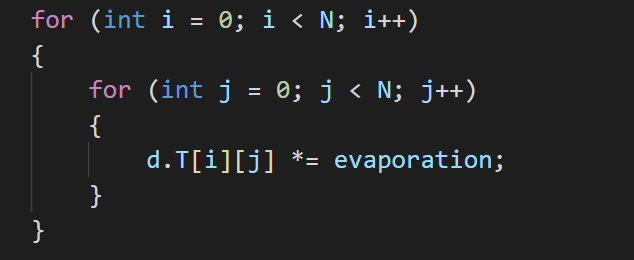
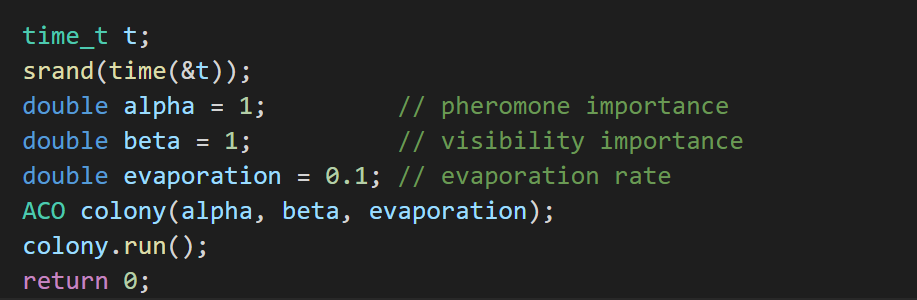
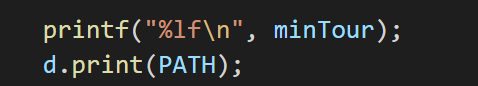
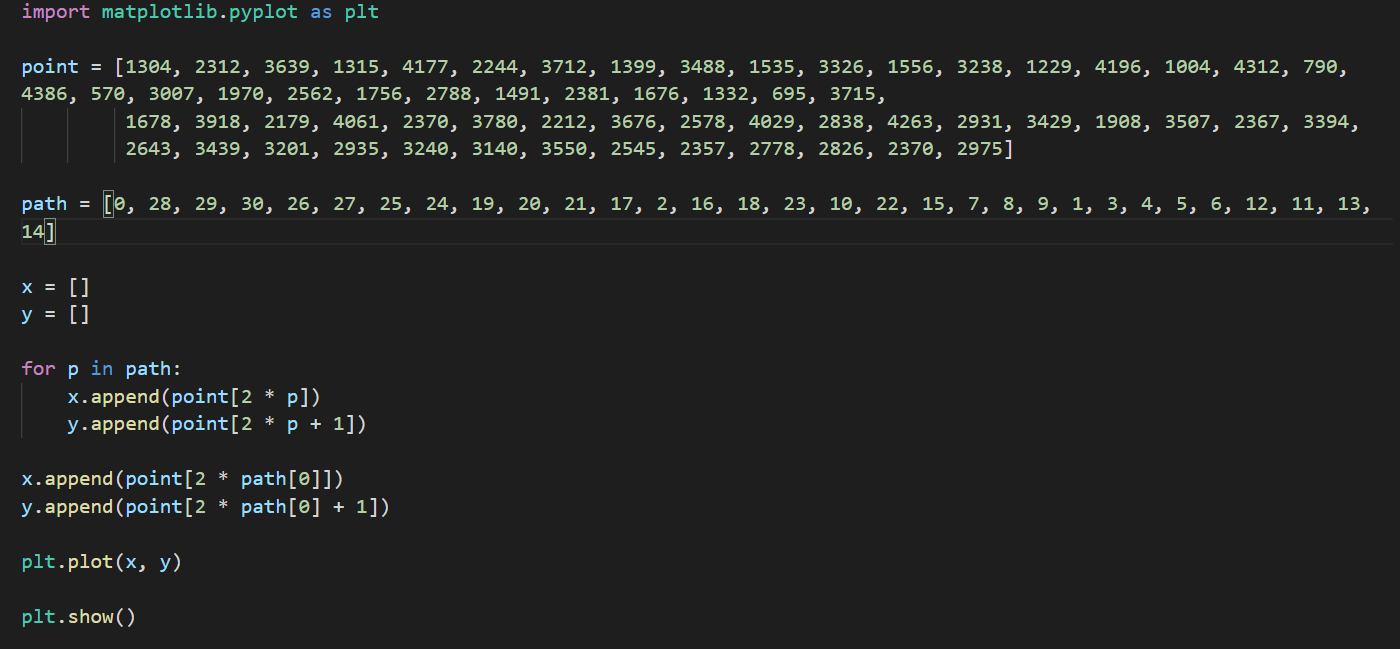
全国31个省会城市的坐标为[1304 2312; 3639 1315; 4177 2244; 3712 1399; 3488 1535; 3326 1556; 3238 1229; 4196 1004; 4312 790; 4386 570; 3007 1970; 2562 1756; 2788 1491; 2381 1676; 1332 695; 3715 1678; 3918 2179; 4061 2370; 3780 2212; 3676 2578; 4029 2838; 4263 2931; 3429 1908; 3507 2367; 3394 2643; 3439 3201; 2935 3240; 3140 3550; 2545 2357; 2778 2826; 2370 2975]

1. 实验环境

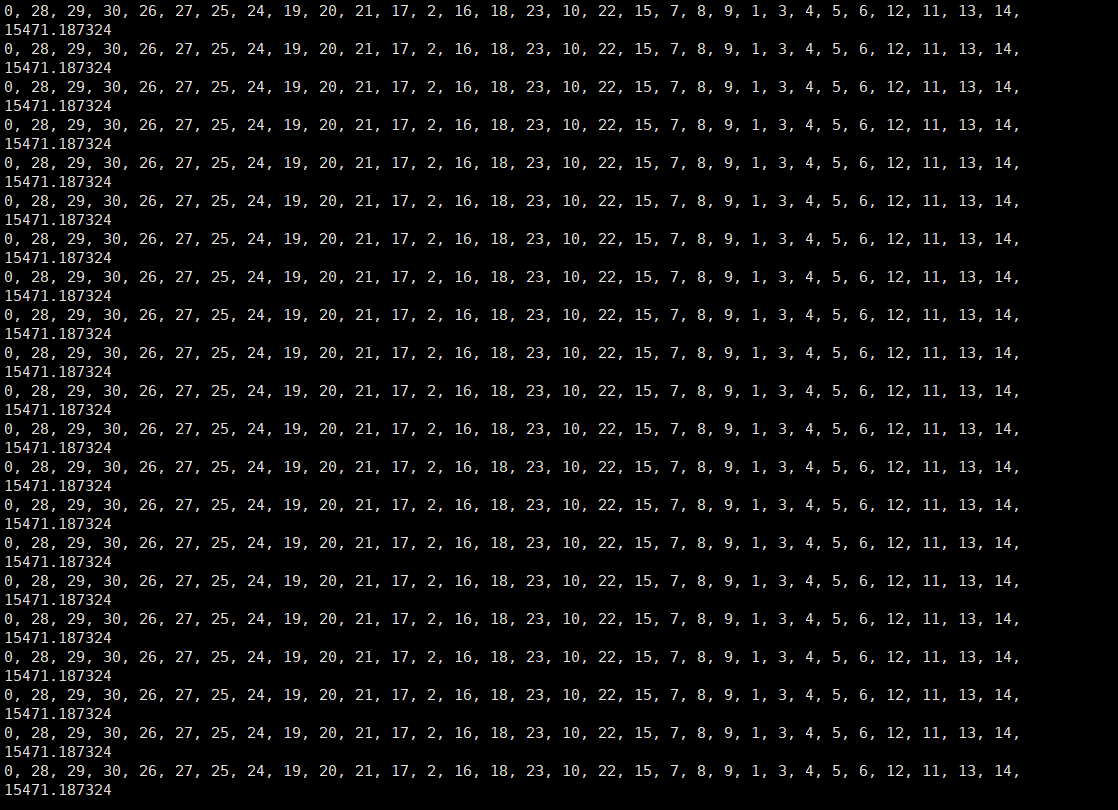
Visual Studio code

本地进行测试，服务器调节数据跑实验

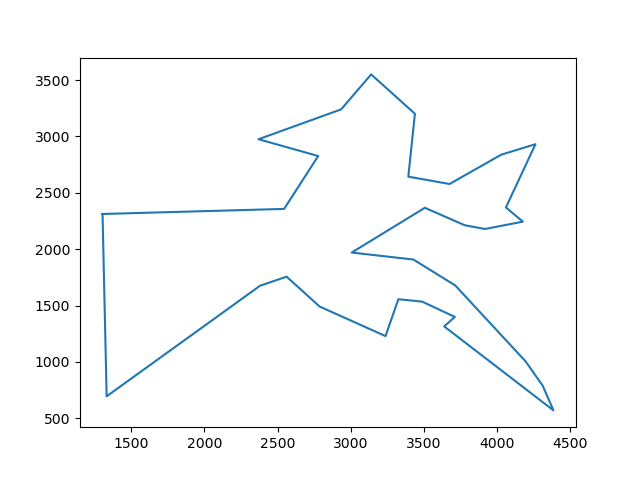
C++语言进行开发，python进行画图

1. 实验步骤
2. 首先初始化每个城市的坐标。
3. 使用欧氏距离来计算两个点之间的距离并创建一个记录每个点之间的距离的矩阵，并计算出相应的值放入矩阵。
4. 通过该函数计算一条路径的总长度。
5. 每个Ant的初始化函数和一些初始化变量。
6. 每个Ant的参数重置。
7. 信息素增加更新。
8. Ant移动一步。 
9. 计算移动可能性。
10. 整个种群的初始化参数和初始化函数。
11. 将所有Ant走完整条路径。
12. 记录并保存最好路径。
13. 信息素挥发。
14. 停止条件。自定义每个Ant走多少次。
15. 调用之前的算法，进行整个算法的流程。计算出最短路径并保存。
16. 最终输出答案。 
17. 结果可视化。
18. 实验结果

我选择了如下的参数：停止条件设置为30，种群数量设置为30000。

下面是使用该参数得到的结果。

下面是将该路程可视化后得到的结果。



可以大致看出没有交叉路线，通过多次运行，多次能收敛到一个较好的结果，说明该程序的鲁棒性和计算速度都很好，容易陷入局部最优解。

1. 实验总结

蚁群算法是一种用来寻找优化路径的概率算法。其灵感来源于蚂蚁在寻找食物过程中发现路径的行为。这种算法具有分布计算、信息正反馈和启发式搜索的特征，本质上是进化算法中的一种启发式全局优化算法。

将蚁群算法应用于解决优化问题的基本思路为：用蚂蚁的行走路径表示待优化问题的可行解，整个蚂蚁群体的所有路径构成待优化问题的解空间。路径较短的蚂蚁释放的信息素量较多，随着时间的推进，较短的路径上累积的信息素浓度逐渐增高，选择该路径的蚂蚁个数也愈来愈多。最终，整个蚂蚁会在正反馈的作用下集中到最佳的路径上，此时对应的便是待优化问题的最优解。

蚂蚁找到最短路径要归功于信息素和环境，假设有两条路可从蚁窝通向食物，开始时两条路上的蚂蚁数量差不多：当蚂蚁到达终点之后会立即返回，距离短的路上的蚂蚁往返一次时间短，重复频率快，在单位时间里往返蚂蚁的数目就多，留下的信息素也多，会吸引更多蚂蚁过来，会留下更多信息素。而距离长的路正相反，因此越来越多的蚂蚁聚集到最短路径上来。

蚂蚁具有的智能行为得益于其简单行为规则，该规则让其具有多样性和正反馈。在觅食时，多样性使蚂蚁不会走进死胡同而无限循环，是一种创新能力；正反馈使优良信息保存下来，是一种学习强化能力。两者的巧妙结合使智能行为涌现，如果多样性过剩，系统过于活跃，会导致过多的随机运动，陷入混沌状态；如果多样性不够，正反馈过强，会导致僵化，当环境变化时蚁群不能相应调整。