利用GA算法解决TSP问题：

分析：

tsp问题即旅行商问题，在一个城市之中，商人形成的路径要将每一个城市访问一次，且仅一次且最后能够回到出发点（大概是这样，可能不是很准确），对于TSP问题一般有对称和非对称两种，也就是有向图和无向图的区别，与此对应的他们所形成的距离矩阵也不相同，一个上三角矩阵或者是只有对称线为0的矩阵。要想利用GA算法来解决TSP问题，首 先应该解决编码问题，交叉算子问题，变异问题。

1. 编码问题，对于正常的函数求值来说，我们一般采用的都是二进制编码，但是在TSP问题中我们每一个染色体（或者是个体）的表示应该是一条路径而不应该是一个二进制的数，因此对于TSP问题我们选择利用路径作为个体的基因值，这也就导致本来的GA算法的交叉和变异算法会产生不一样的点，因为在TSP问题当中我们必须要保证的在一条路径上一个点只能出现一次，而且每一条路径都是一个回路，而普通的GA算法的交叉与变异算子很明显并不能满足这两点要求
2. 交叉算子设定：对于TSP问题的GA算法对交叉算子的实现会有很多种方案，其实他们都保持着一个原则就是不允许在路径中有重复（一般只要没有重复，就不会有缺少的现象，而且不需要考虑回路问题，因为默认最后一个点是一个隐性点，不在基因数组显示，但是最后求距离的时候要算上的一个点。）在这里我选用的是次序交叉法
3. 变异算子的设定，其实这里的变异算子应该是一个组内交叉的算法，就是组内的基因进行互换，然后也就是完成了变异的操作，而且还确保不会发生重复

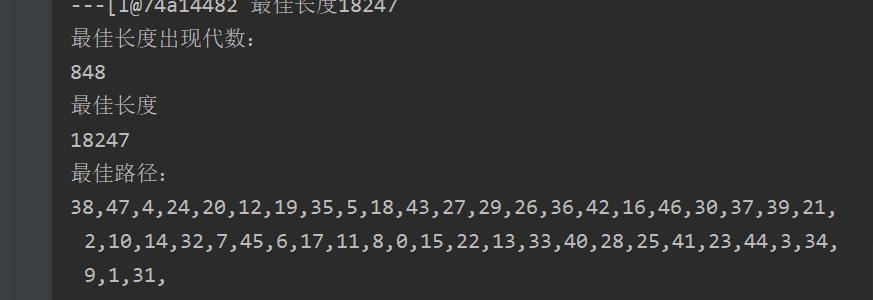
实现注意点：

1. 代码使用轮盘赌的方法
2. 交叉算子选择的是类OX交叉，就是一个个体选择某两个个位置见的基因再加上另一个个体去掉这些基因以后的所有基因所形成的的新基因就是子代种群。
3. 变异算子：这里的变异算子可以更复杂，比如多次变换来实现。
4. 对于种群的进化来说，我们使用了精英保留原则，即父代最优的保留在子代中，而且在交叉变异当中我们对这个基因不进行处理，以保证它的优越性可以一直传下去，不过这样可能产生的不好处就是收敛性可能不够理想。

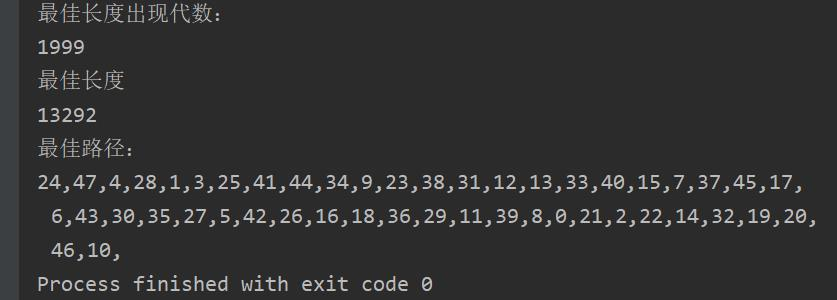
实验结果

Pc=0.4 pm=0.1

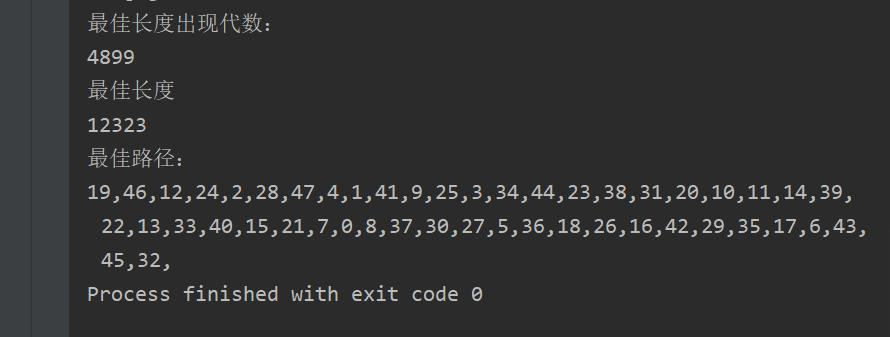
迭代1000



迭代2000



迭代5000



迭代10000

