实验二：用SA算法求解旅行商问题

1. 问题描述：

旅行商问题（Travelling Salesman Problem, 简记TSP，亦称货郎担问题)：设有n个城市和距离矩阵D=[dij]，其中dij表示城市i到城市j的距离，i，j=1，2 … n，则问题是要找出遍访每个城市恰好一次的一条回路并使其路径长度为最短。

Q1：每个城市都是连通的吗？

A：n个城市之间均有道路，但距离不等

Q2：如何找出所有城市路径的排列？

A：不是穷举出城市之间的所有排列，N个城市的穷举太费时，所以是由一个初始的排列，然后根据生成规则找出下一个。

Q3：SA算法在此问题中有什么样的优势？

A：可以跳出局部最优解，所以在求解过程中，解不是一直下降的。

Q4：SA算法是用于问题解是连续的还是离散的？

A：离散和连续都可以，但是要有一个规则生成下一个解，用SA算法判断是否接受下一个解

Q5：用什么样的数据结构？

A：用一个N\*N的矩阵存储各个城市之间的距离，然后每个解用一个一维数组存储N个数的排列。

Q6：产生新的遍历路径的方法会生成重复的路径吗？

A：

模拟退火算法参数：

初始温度 T0

终止温度 T\_min

退火速率 r，一般为指数式下降，其中r是小于1的正数，一般取值为0.8到0.99

状态的评价函数D，在TSP问题中是城市之间的距离

初始状态（随机生成）s0

当前状态 s

产生的新的状态 s\_next

接受概率 exp(-ΔD/D)

Mapkob链长L：定长20000