# 禁忌搜索算法解TSP问题

## 算法实现

1. 初始化最优解，禁忌表，藐视准则
2. 迭代
   1. 遍历邻域  
      在禁忌表中，更新藐视最优解  
      不在禁忌表中，更新迭代的最优解
   2. 判断藐视最优解是否满足条件
   3. 对迭代最优解进行转移，并更新禁忌表
3. 输出全局最优解

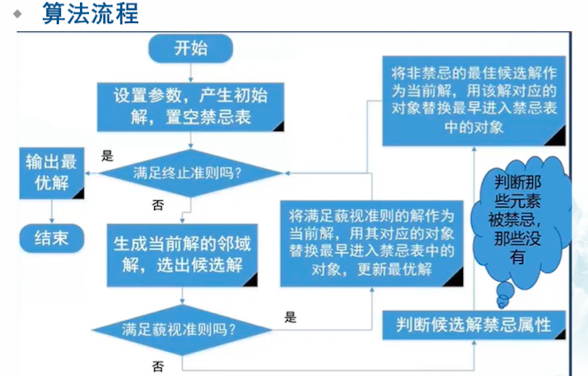
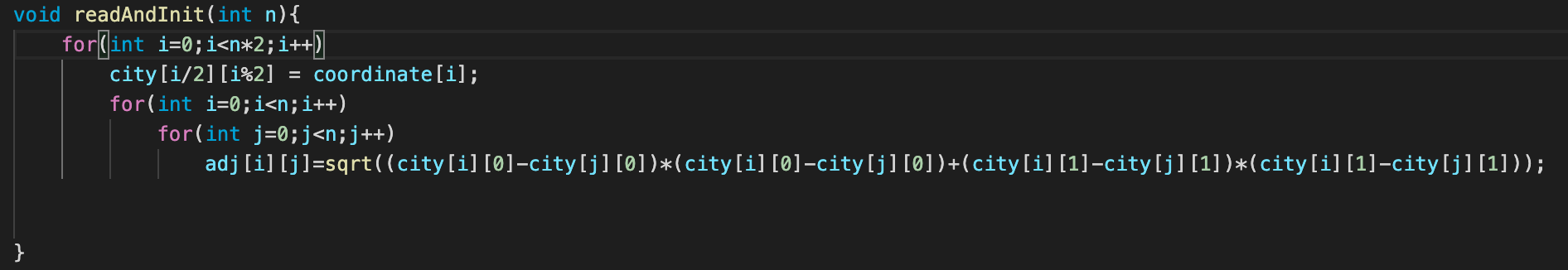


image-20220321173714733

## 关键代码

* 读入城市数据，并计算城市距离



* 初始化禁忌表

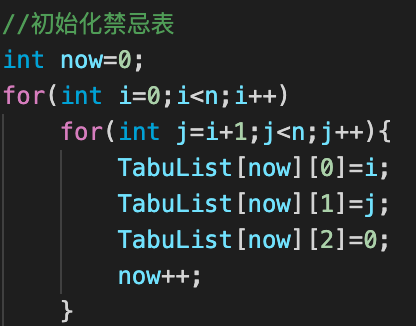


image-20220321164627897

* 转移最优解，对每次迭代后的最优解进行更新

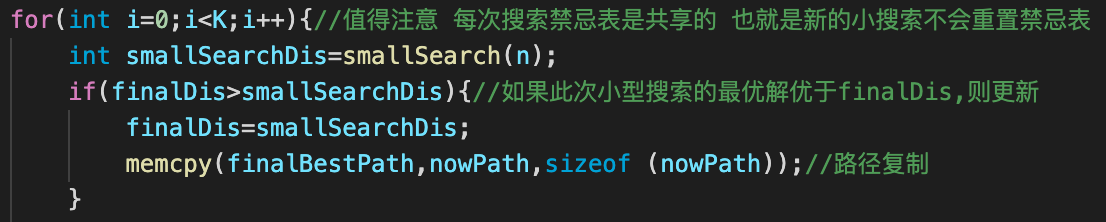
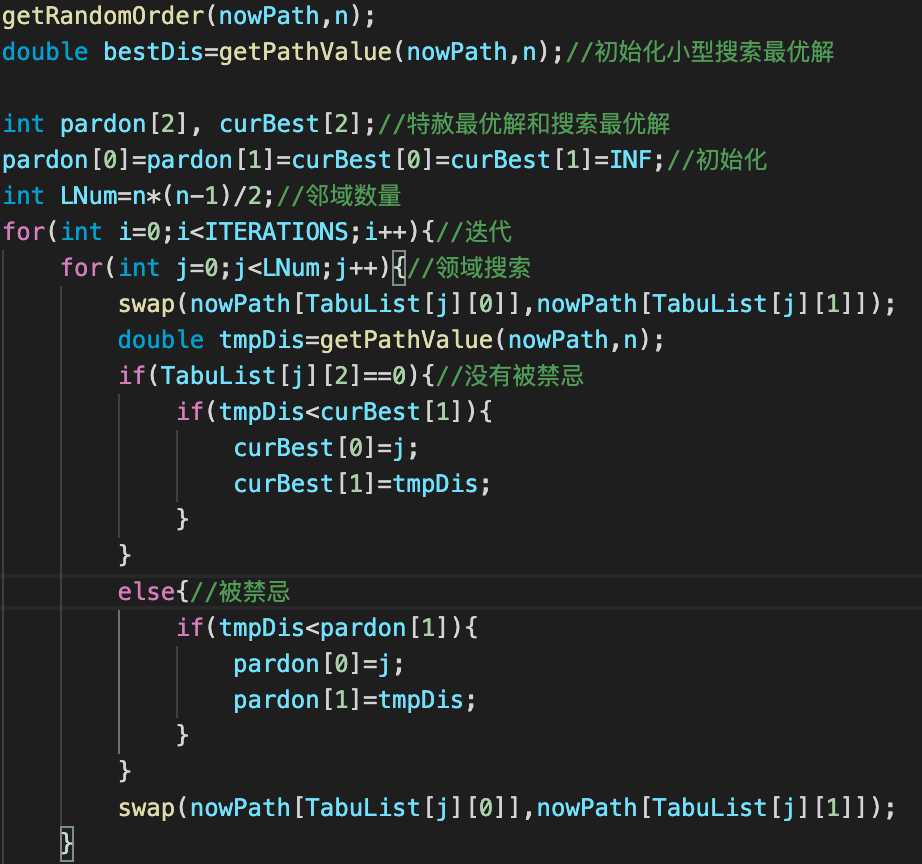


image-20220321164858741

* 邻域搜索，从初始解的邻域开始，进行局部搜索；在邻域搜索中，检查当前解是否在禁忌表中，若在表中，则检查是否可以被特赦；若不在表中，则更新迭代最优解
* 
* image-20220321165126763
* 更新禁忌表，将本次更新后的迭代最优解加入禁忌表中，并对已在表中的解进行更新

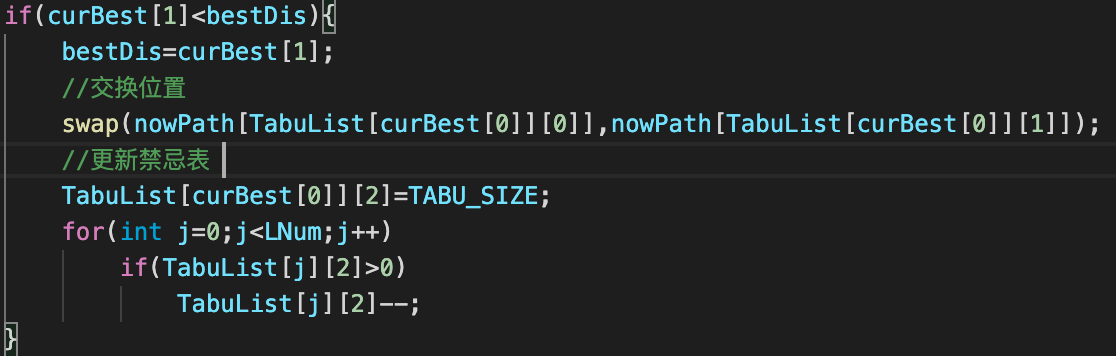


image-20220321165358917

## 运行结果

最优解如下图：

最短距离为：16784.6fig:

## 问题与收获

在学习该算法的过程中，对相关参数概念有点模糊不解，参考网上前辈的思路才清晰一点，对比不同的实现代码发现对最终的结果会有较大差异，不太清楚其中原因，但路线长度的范围大致在15000-20000之间。

禁忌搜索算法是一种模拟人记忆的启发式算法，在对邻域搜索时，对局部最优解进行标记即加入禁忌表，然后在后续搜索中尽量避免这些局部最优解从而尽量达到全局最优，在有限时间内给出最优解。