

**《计算机算法课程实践》**

**实验报告**

**实验（ 一 ）**

**年 级： 2024**

**专 业： 计算机技术**

**姓 名： 刘兴宇**

**学 号： 2120240739**

# 1 实验内容

* 随机生成指定数量和指定坐标范围的点集数据。
* 设置穷举阈值，低于穷举阈值时采用穷举法。
* 根据数据利用分治法得到欧氏距离最近的一对点，并计算最小欧氏距离。
* 分析不同规模数据集对算法时空复杂度的影响。

# 2 设计思想

该实验的设计思想如下：

**1. 数据集制作：**

利用随机数生成点的坐标。

利用集合避免出现重复的坐标点。

**2. 穷举法：**

遍历vector中的每一个点对。

计算点对间的欧氏距离，并记录欧氏距离最小的点对坐标。

**3. 分治法：**

判断vector的size是否小于穷举阈值，若小于则调用穷举法寻找欧氏距离。（判断边界条件）

输入数据有两个vector。它们分别按照横纵坐标排序。

取Size/2下标的点，获得水平方向的中间点。将vector划分为左右两部分。

同时根据中间点坐标，将按照纵坐标排序的vector也划分为左右两部分。

递归调用分治法函数。

比较左右两部分的欧氏距离，得到最小值d。

由于最小值可能出现在跨越中线的情况。因此对这部分的点需要单独分析。

最小值只可能出现在以中线为中心的宽度为2d的纵向带状中的点。

利用鸽笼原理，距离最小的点只可能出现在该点周围5个点之内。

所以只用计算这5个点中，纵坐标距离小于d的点与该点的欧式距离。

从而得到跨越中线的欧式距离最小值。

将该值与最小距离d作比较，若小于d则更新最小距离点对，否则不变。

# 3 程序效果

程序的运行效果，输入及输出的相关要求和具体执行结果如下所示：

1. 得到不同规模的数据集。

2. 利用分治法和穷举法得到欧氏距离最小的点对。

## 3.1 输入

1. 数据集制作：需要生成的数据集规模、保存的文件名。

2. 查找部分：数据文件（txt格式）、穷举法阈值。

## 3.2 输出

1. 数据集制作部分：包含了点坐标的txt文件。

2. 查找部分：点的总数、穷举法阈值、欧式距离最近点对、最小欧氏距离、耗时。

# 4 算法分析

**时间复杂度分析**

**1. 穷举法部分**

双重循环遍历所有点对，时间复杂度为 O(n²)，其中 n 是点的数量。

**2. 带状区域处理**

外层循环 O(n)。

内层循环看起来是 O(n)，但实际上由于几何性质，每个点在y方向上只需要与有限个点比较，因此时间复杂度为 O(n)。

**3. 分治法部分**

递归树分析：T(n) = 2T(n/2) + O(n)。

分割点集：O(n)。

递归求解左右子问题：2T(n/2)。

合并结果（处理跨中线情况）：O(n)。

根据主定理，这个递归关系式的解为：T(n) = O(n log n)。

**4. 排序部分**

对点集按x和y坐标分别排序，时间复杂度为 O(n log n)。

**总体时间复杂度**

当穷举阈值较小时，算法总体时间复杂度为 O(n log n)。

当穷举阈值较大时，算法会更多地使用穷举法部分，可能接近 O(n²)。

**空间复杂度分析**

**1. 存储点集**

需要存储原始点集以及按x和y坐标排序的两个副本，空间复杂度为 O(n)。

**2. 递归调用栈**

递归深度为 O(log n)，每层递归需要存储左右子问题的点集，空间复杂度为 O(n log n)。

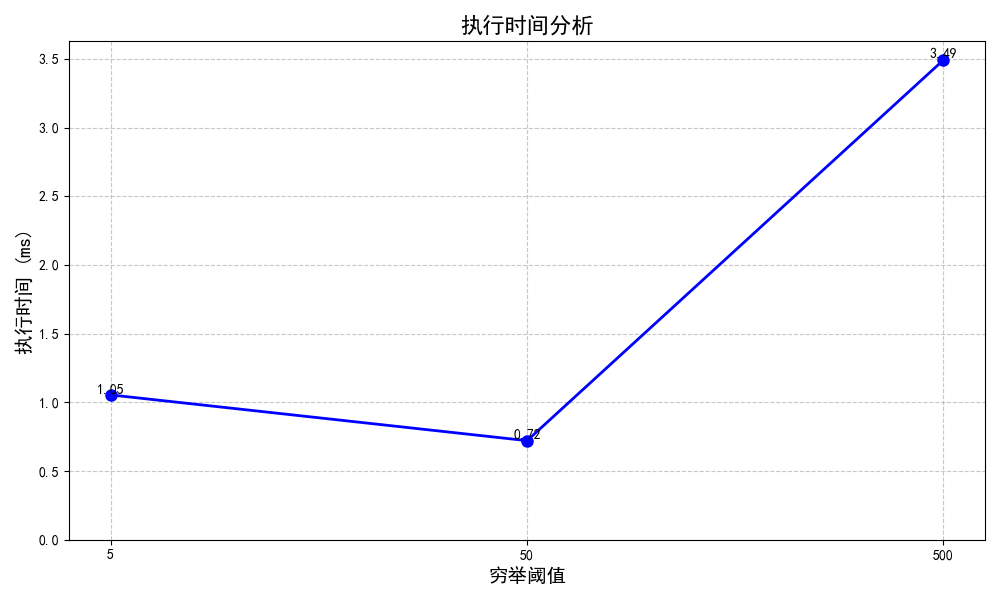
**3. 带状区域**

存储跨中线的点，最坏情况下为 O(n)。

**总体空间复杂度**

算法的总体空间复杂度为 O(n log n)。

对于n=1000，采用三个不同穷举阈值并分析其对于算法时空复杂度的影响；



阈值=5 (1.05ms)：

优点：穷举部分工作量小

缺点：递归层次深，函数调用开销大

阈值=50 (0.72ms)：

优点：平衡了递归开销与穷举计算量

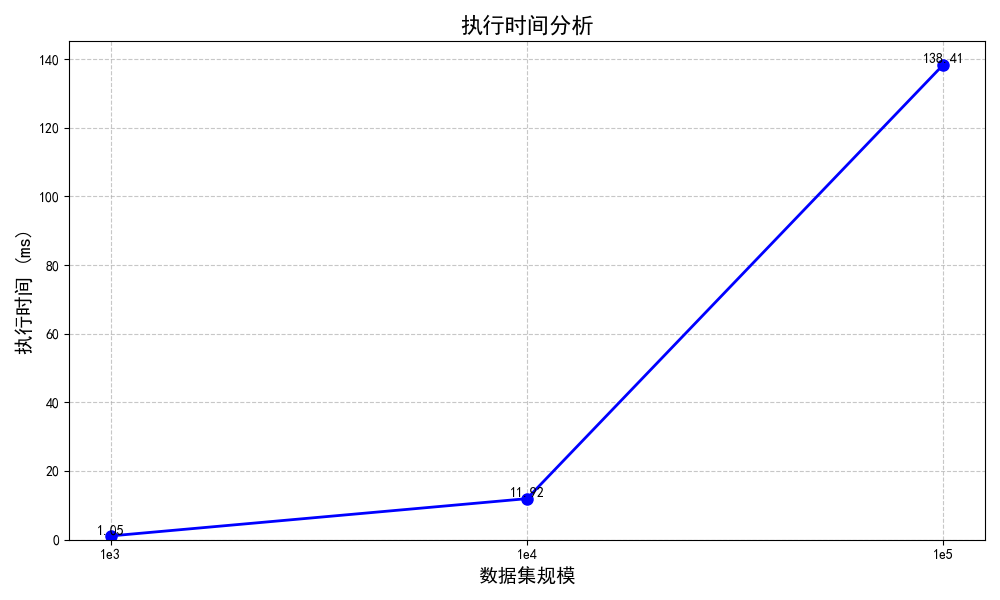
阈值=500 (3.49ms)：

优点：递归层次浅，函数调用开销小

缺点：穷举计算量过大

对空间复杂度几乎没有影响。

对于同一穷举阈值，分析三个不同规模的数据集（103、104、105）对于时空复杂度的影响。



穷举阈值固定（穷举阈值为5），不同规模数据集的执行时间：

n=10³（1.05ms）

n=10⁴（11.92ms）

n=10⁵（138.41ms）

时间复杂度分析

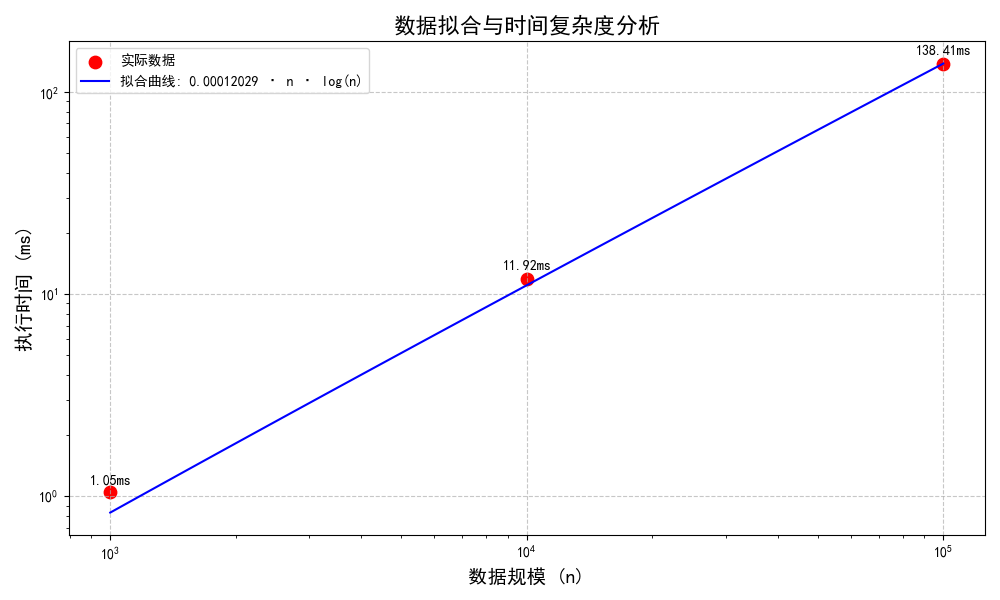
理论分析

最近点对算法的理论时间复杂度为O(n log n)。根据这一复杂度，当数据规模增长时，执行时间应该按照n log n的比例增长。

实验数据拟合

对实验数据进行拟合，得到近似函数：

0.00012029 · n · log(n)



实际数据与理论复杂度非常吻合，特别是在大规模数据集上。

对空间复杂度几乎没有影响。

# 5 总结

分治方法和穷举法之间有一个平衡，并不是穷举阈值越小耗时越少。

图像可以更清晰直观的展示出时间消耗和数据量的关系。