最近点对实验报告

姓名：覃果　学号：2020012379　班级：软件02

2022 年 3 月 11 日

摘要

通过暴力计算，分治法分别计算出二维平面上最近的一对点，比较两个算法的运行速度，并且实现用户图形界面。

1　实验环境

CPU: AMD Ryzen 5 4600U with Radeon Graphics 2.10 GHz
OS: Windows 10 家庭中文版 20H2
Python interpreter: Python 3.8.7

2　算法分析

2.1　暴力计算

2.1.1　算法描述

遍历计算出每两个点之间的距离，不断更新最近的点对

```
Naive_Closest_Points:
if len(points) == 0:
    raise ValueError("the point array must not be empty")
if len(points) == 1:
    return ValueError("the point array must not be just one element")
min_val = distance(points[0], points[1])
min_points = [0, 1]
for i in range(len(points)):
    for j in range(i + 1, len(points)):
        tmp = distance(points[i], points[j])
```

```
            if tmp ¡ min_val:
                min_val = tmp
                min_points[0] = i
                min_points[1] = j
    return (min_val, points[min_points[0]], points[min_points[1]])
```

2.1.2 算法分析

时间复杂度：
由算法过程，进行了两层循环，故 $T(n) = O(n^2)$
空间复杂度：
算法过程中使用了一个长为n的数组以及一些常数大小的变量，故为 $O(n)$
算法评价：算法过程简单，实现容易，但复杂度有可优化的空间

2.2 分治算法

2.2.1 算法描述

先将点按照x方向的坐标大小排序，之后利用分治的思想划分成两个子问题。
同时，当划分到一个比较小的规模时（即递归基），对其按y方向的坐标大小排序，
分治处理完子问题时假设两个子问题的部分分别都是对y有序的
之后合并子问题时，对y使用归并。这样原问题也将对y有序。
由于递归基时对y排了序，故可知假设成立
之后从序列中提出在划分线左右d范围内的点构成一个新的序列。d是两个子问题中较小的那个距离。

这个序列对y也有序，之后对每一个元素，计算它与其后5个元素的距离。取出所有距离中的最小者d_0。

d与d_0中的较小者，以及这个距离对应的两个点便是原问题的结果。

2.2.2 算法分析

时间复杂度：

由算法过程，将问题转化成了两个规模减半的子问题，合并的复杂度为归并的复杂度。

故$T(n) = 2T(\frac{n}{2}) + O(n)$，由主定理知，$T(n) = O(nlgn)$
空间复杂度：由算法过程，对于空间复杂度$S(n)$，同样有$S(n) = 2S(\frac{n}{2} + O(n))$
同理$S(n) = O(nlgn)$

算法评价：利用分治的思想，大大降低了算法的时间复杂度，但空间复杂度有所上升。

3 实验设计思路

对于不同大小的n，分别计算出两个算法的运行时间，并进行比较。

同时，对于用户图形界面。选择了PyQt进行实现。

设计了一个鼠标行为：在窗口中点击，变会在对应的地方产生一个黑色的点

两个按钮，calculate按钮：点击后会计算出当前所有点中最近的点对，并将它们标红并连接起来。

clear按钮：清除当前所有已有的点。

4 结果分析

运行时间表格如下：

n	暴力/s	分治/s
50	0.003	0.001
100	0.007	0.004
200	0.02	0.007
500	0.147	0.011
1000	0.628	0.027
2000	2.459	0.037
5000	10.293	0.11
10000	40.694	0.236
20000	164.746	0.565
50000	1024.543	1.778
100000	\	3.777
200000	\	8.172
500000	\	20.35
1000000	\	44.959
10000000	\	631.626

作图如下：

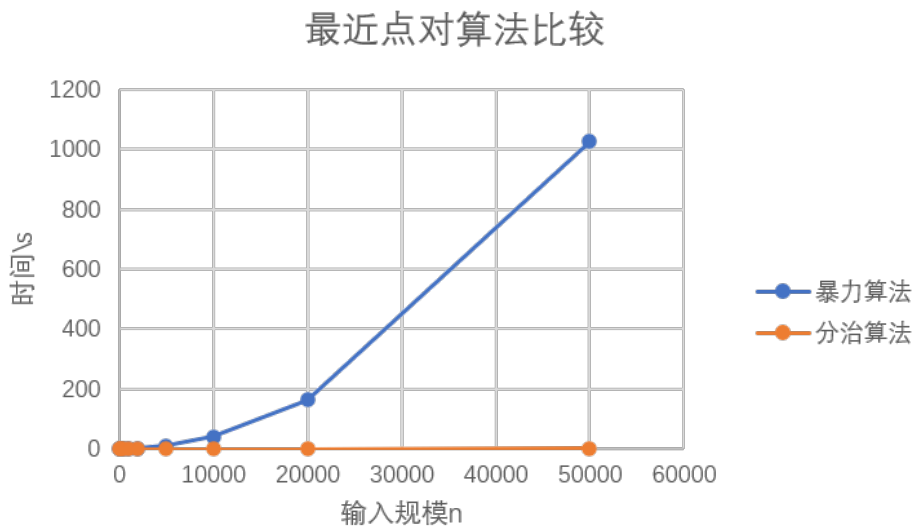

图 1: 结果分析

可见，分治算法的速度比暴力算法显著地提升了。这是复杂度上巨大的改进!

分治算法对1百万规模的输入与暴力算法1万规模的输入用时相当。甚至对千万规模的输入比暴力算法5万规模的输入快了将近一半!

这进一步显示出$\mathrm{O(nlgn)}$的算法与$O(n^2)$的算法运行速度的巨大不同。