

矩阵乘法比较实验报告

洪方舟

Student ID: 2016013259

Email: hongfz16@163.com

March 10, 2018

1. 实验目的

- a. 编写程序实现查找点集中最近点对
- b. 对比 $O(n^2)$ 算法与 $O(n\lg n)$ 算法的实际运行时间
- c. 通过对递归深度的限制优化实现 $O(n\lg n)$ 算法的运行效率最大化

2. 实验环境

操作系统: macOS High Sierra (Version 10.13.1)

处理器: 1.6GHz Intel Core i5

编程语言: C++

编译器: g++

3. 实验方法

- a. 随机生成不同数量级的点集，分别使用两种算法运算最近点对，并测出运行时间
- b. 通过将两种方法的结果对比来确保算法的正确性

4. 实验结果

Table 1: 分治方法中，设置递归在点集大小为1时停止

点集大小	分治法耗时(ms)	常规方法耗时(ms)
10	0.014	0
100	0.134	0.025
1000	1.466	2.331
10000	18.698	236.863
100000	191.908	19319.7
1000000	1796.85	X

Table 2: 分治方法中，设置递归在点集大小为100时停止

点集大小	分治法耗时(ms)	常规方法耗时(ms)
10	0.004	0.001
100	0.038	0.023
1000	0.532	2.288
10000	6.554	159.721
100000	69.454	19588.7
1000000	769.546	X

5. 分析与总结

- a. 在“最近点对”的实验中可以看到，分治方法 $O(n\lg n)$ 复杂度相对于常规方法是一个很大的提升
- b. 由于常数并不是非常的大，所以递归深度的影响不是很严重
- c. 通过反复试验，将递归设置在点集大小为100时停止，得到的效率是最高的，优化后时间能够减少一半

6. 源代码及可执行文件说明

a. 控制台程序

1. 在 `/src/TClosestPair` 目录下存放着控制台程序的源代码，使用 `makefile` 来组织项目，在命令中使用 `make` 命令，将会在同目录下产生名为 `main` 的可执行文件；
2. 在 `/bin/terminal` 目录下分别存放着可以在 `Windows` 和 `macOS` 环境下运行的可执行文件。

b. GUI

1. 可以在 `Windows` 环境下运行的 `GUI` 程序存放在 `/bin/GUI` 目录下；
2. 该程序支持两种输入点的方式，一种是输入数字，随机自动生成相应数量的点；另一种是点击屏幕输入点，若使用该方式输入点，请先点击 `StartClickInput` 按钮，在屏幕上点选过后，再点击 `EndClick` 按钮；
3. 程序提供两种求最近点对的方法，对应按钮为 `$O(n\log n)$ GetClosestPair` 和 `$O(n^2)$ GetClosestPair`，输入点对完成之后可以点击两个按钮之中的一个来计算最近点对，注意如果是点击屏幕输入点对的方式，请一定在输入完成之后点击 `EndClickInput` 再继续下面的计算操作；
4. `Cancel` 按钮负责清除计算结果，`ClearPoints` 将会把屏幕上的点清空；
5. 计算点对完成之后，将会在屏幕中连接两点，标出那两点的坐标，并且在屏幕左上方给出最近距离；
6. 源代码保存于 `/src/QtClosestPair` 目录中；