实验报告

实现求平面上最近点对的$\Theta(nlogn)$的算法，并分析比较在不同输入规模下$\Theta(nlogn)$和$\Theta(n^2)$算法的实际运行时间

实验环境

CPU：Intel Core i5-7200U

内存：8GB

操作系统：Windows 10 64位教育版

编程语言：C++

编译环境：MSVC 2017 32-bit

算法分析

Brute-force

通过遍历平面上所有点来暴力求解，其时间复杂度$T(n)=\Theta(n^2)$

Divide-and-conquer

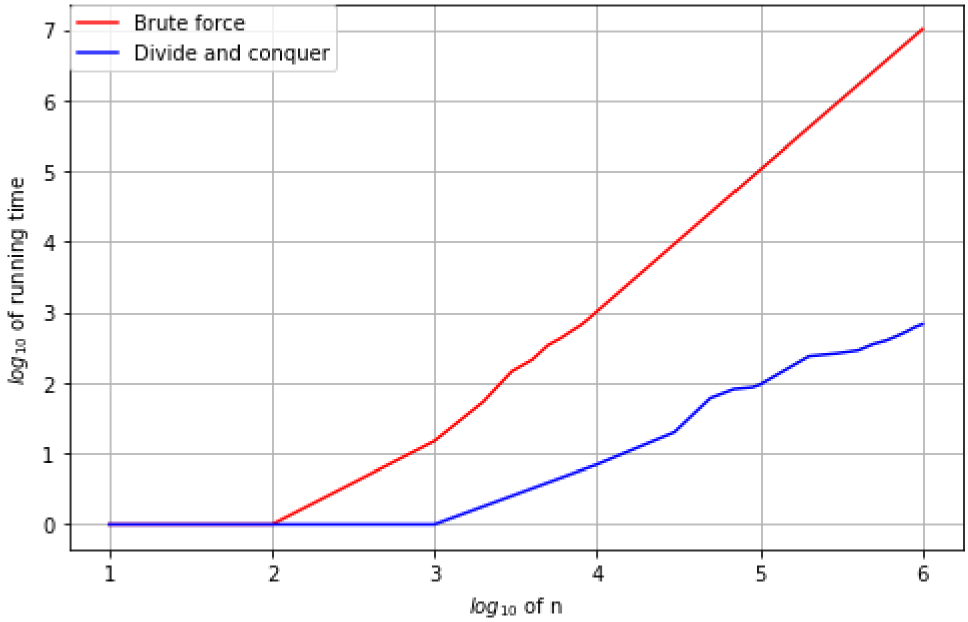
采用分治和递归思想，其时间复杂度$T(n)=\Theta(nlgn)$

结果分析

取$n=10^i$，分别使用2种算法进行实验，每组实验均进行5次取运行时间的平均值，并记录其运行时间和计算结果，运行时间如下表（单位为ms）：

n	**Brute-force**	**Divide-and-conquer**
10	1	1
100	1	1
1000	15	1
10000	1029	7
100000	104675	95
1000000	10230243	681

不难看出，$\Theta(nlgn)$算法在运行速度上明显优于$\Theta(n^2)$算法。为了更加明确不同输入规模下两种算法的差异，进一步的实验加密了n的间隔，所得运行时间如下图：



分析上图可得，当$n < 10^2$时，由于数据规模较小，两种算法用时均很少，但当$n > 10^3$时，$\Theta(nlogn)$算法运行速度迅速超过$\Theta(n^2)$算法，且两者差距越来越大，到$n = 10^6$时已经达到了约10^4倍。在空间占用上，$\Theta(nlogn)$算法由于递归层数和临时数组会占用一定的空间，但在本实验环境下并不是影响性能的主要因素。

因此，两种算法中$\Theta(nlogn)$算法在运行时间上有很大优势，特别是在较大数据规模和对时间有较高要求时，应该优先选择$\Theta(nlogn)$算法。