作业 4 中位数

1　理论分析

在 BFPRT 算法中，数组被分为 Q 个子块。算法分别求出每个子块的中位数，再求出“中位数的中位数”，将其作为枢轴，进一步递归处理。

每次划分后，一定比例的元素会被排除，比例与 Q 相关。在每个子块中，至少有 (Q-1)/2 个数小于子块的中位数，则每次划分后，至少可以排除 (Q-2)/2Q 比例的数。

设算法复杂度为 T(n)，则有

$$T(n) \leq T(n/Q) + T((Q+2)n/2Q) + c \cdot n$$

Q 增大时，每次可以排除更大比例的元素，但对应地，计算每组中位数的用时会增加，时间复杂度的常数因子会变大。

2　实验数据

数组大小为 n。在 n 分别为 100、1000、2000 的情况下，令 Q 分别等于 5、6、7、8、9、10、11，在无序和乱序两种情况下，测量找到 90% 分位数所需时间（单位为 ns），实验结果如表 1和 2。

n \ Q	5	6	7	8	9	10	11
100	159703	173130	159797	196163	161017	143385	173549
1000	1976262	1953071	1851205	1872897	1698337	1538145	1590003
2000	3979346	3775738	3361424	3483537	3953542	3737819	4016447

表 1: 顺序情况下，不同 n、Q 的算法性能

n \ Q	5	6	7	8	9	10	11
100	169992	153732	172851	202796	161319	133264	184107
1000	2156975	2242252	1996367	1986859	1910739	1704549	1736209
2000	4703768	4135997	3967101	3996790	4342464	4667219	4527882

表 2: 乱序（随机）情况下，不同 n、Q 的算法性能

顺序和乱序情况下，所用时间与 Q 的关系折线图如图 1和图 2。

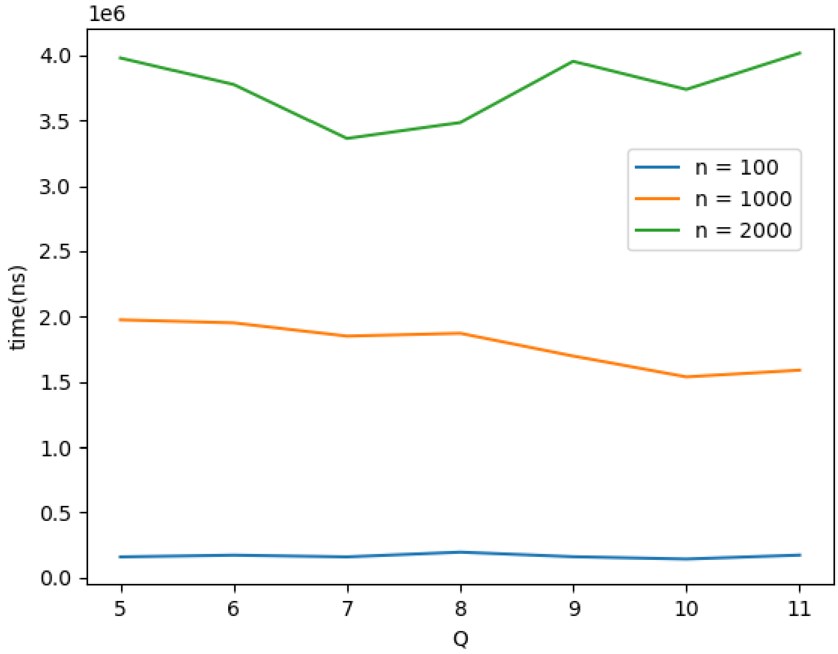

图 1: 顺序情况下，所用时间-Q 折线图

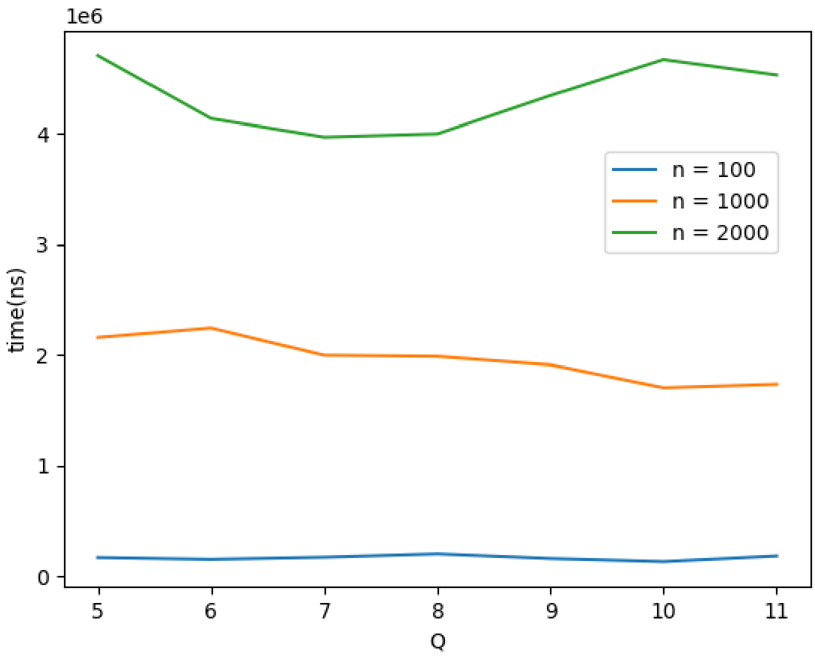

图 2: 乱序情况下，所用时间-Q 折线图

3 实验数据分析

在乱序情况下，算法用时高于顺序情况下的用时，但处于同一复杂度下。

在 n 为 100 时，Q 等于 10 时最优；n 为 1000 时，Q 等于 10 时最优；n 为 2000 时，Q 等于 7 时最优。但在任何情况下，在这几种 Q 取值中，所用时间都处在一个相对稳定的范围内。

在 n 增大时，常数因子增大会对复杂度有更显著的影响，所以 n 增大时，Q 有少量的相应减少。实际应用时，Q 值一般取 5。

Q 过大和过小都会使时间复杂度增加。实验中未表现出特别明显的趋势，可能与数据分布有关。