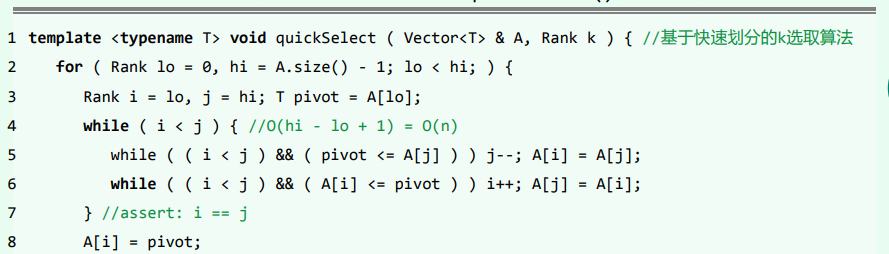
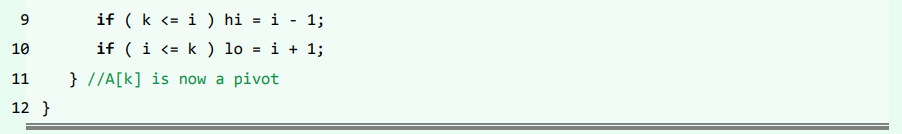
HW7 中位数

赵楷越 522031910803

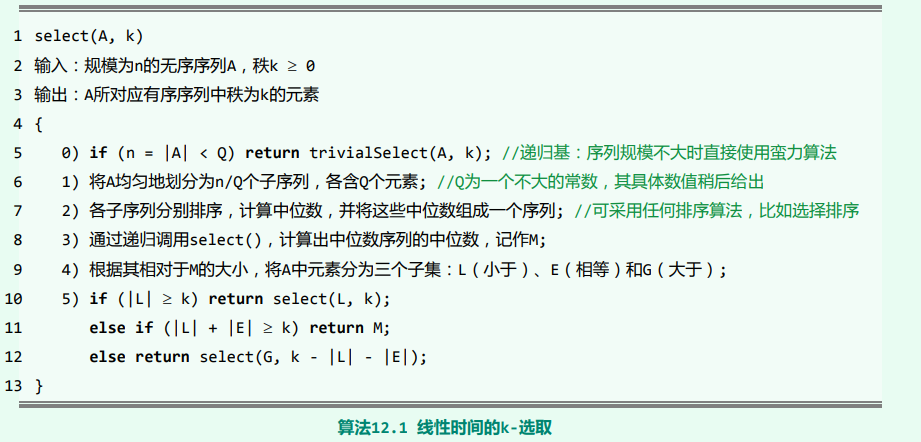
1两种选择算法的实现选择

对于quickSelect方法和linearSelect方法，我均选择了参考书上的算法和伪代码进行了相应代码的编写。基于减而治之、逐步逼近的思路，可实现quickSelect算法。



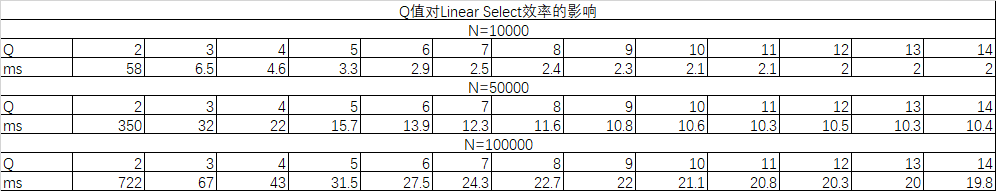


延续quickSelect算法的思路，以下k-选取算法在最坏情况下运行时间依然为O(n)。

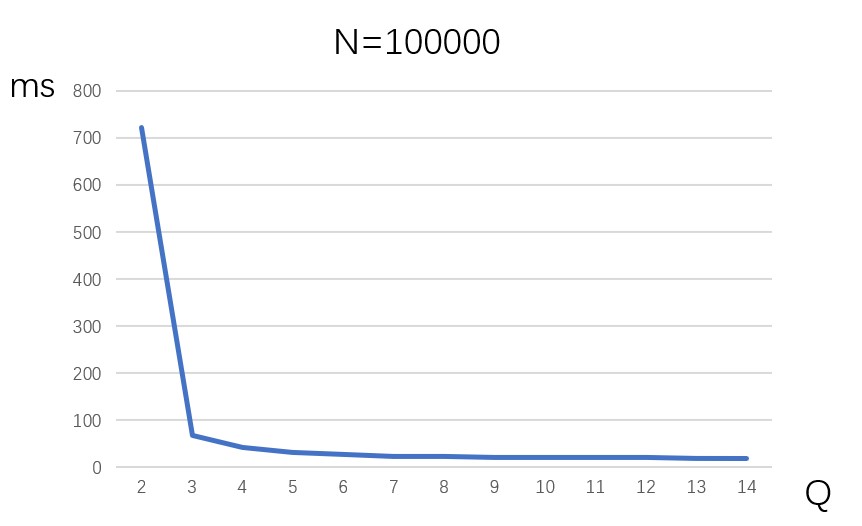


2实验及对应结果分析

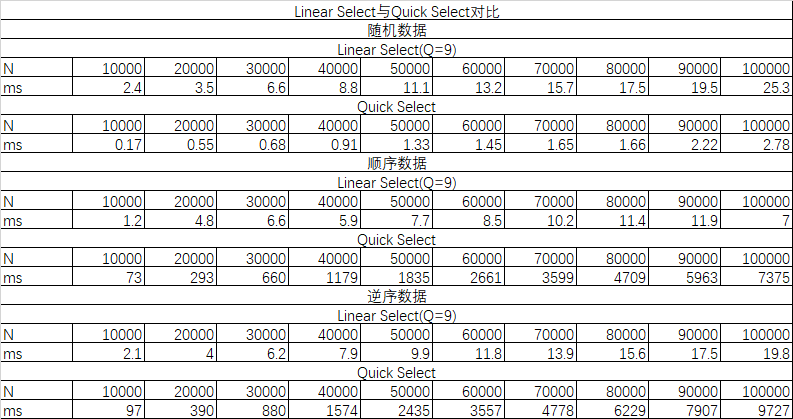
为了研究Q值对linearSelect效率的影响，我选择测试在数组大小N分别为10000，50000，100000的规模情况下，使数组为随机生成的数字，改变Q使Q取2-14中的一值时，linearSelect算法的效率。对应实验结果表格如下所示。



将其中N=100000时的情况绘制成折线图。我们发现当Q=9时，linearSelect算法便可以达到一个最优效率。而算法的效率随着Q的增大而提高。



为了将linearSelect与quickSelect算法进行对⽐，我们分别在随机数据、顺序数据、逆序数据的三种情况下，使数据规模N从10000递增至100000，研究其算法效率与数据规模和数据特性（顺序或乱序的情况下）的关系。因为当Q为9时，linearSelect的性能已经最优平稳，因此在实验中我们选择Q=9。实验数据表格如下所示。



我们发现，在数组数据有序的情况下，quickSelect算法的性能极差，因为它每次选取了最低位上的数据作为pivot，但数据又是有序的，因此导致算法性能崩溃。而linearSelect在不同规模，不同类型的数据情况下表现均良好。表明了其线性性能特征。下表是对在逆序数据情况下，linearSelect和quickSelect算法的性能随数据规模的对比。

