

研究分析报告书

对 100 万条数据进行操作时，从文件读数据过程耗时很长，达到了 16 分钟左右（可能时同时生成了太多辅助结构的原因）。但是对数据操作的速度不是很慢，执行按学号、姓名、电话号码查询、删除时，耗时 10 秒内，但比起小规模的数据，这个耗时可以说很长了。可以说，当数据到达百万时，整个系统的执行速度变得很慢，针对这个问题，以下分析原因、以及提出解决方法。

效率分析：

本次实验中实现查找、删除时，用到了辅助结构二叉搜索树，总的时间复杂度为 $O(\log_2(n))$ ，速度相对较快。导致系统读取速度变慢的原因应该是在读取的同时建立了四棵树（一棵二叉树和三棵用于辅助的树），导致计算量加大，效率变慢。

在输出年龄最小的学生时，用的时层次序遍历的方式，复杂度为 $O(n)$ ，当数据达到百万时，效率会大大降低。可以考虑建立最小堆的方式，但是每次读最小年龄学生都需要建立小根堆，其实效率的提升不是很大，毕竟小根堆的建立也是比较耗时的。

算法改进：

① 在读取文件时，考虑只按照层次序遍历生成一棵二叉树。其他三棵用于辅助的树可以等到使用到相应功能的时候在生成，这样可以提高读取速度。

② 如果比较常查找某个关键字，也可以建一个哈希表，如果是以姓名为关键字可以使用字符串哈希 KMP 等算法。这样复杂度会降为 $O(1)$ ；

可以建立以年龄为关键字的小根堆来实现查找年龄最小的学生，这样时间复杂度可以直接降到 $O(1)$ 。