

## 编程作业 2: 网络 IO 编程（20 分）

1. [13分] 设计学生成绩的文件内容：生成 $512 \times 32 \times 8$ （约13万）行的数据记录，每行记录包括5个字段，分别是：

- 学号（int 类型）：按步长1，从1开始递增至 $512 \times 32 \times 8$ ；
- 语文成绩（float小数点后一位），按照高斯分布生成0.0到100.0的分数；
- 数学成绩（float小数点后一位），按照高斯分布生成0.0到100.0的分数；
- 英语成绩（float小数点后一位），按照高斯分布生成0.0到100.0的分数；
- 综合成绩（float小数点后一位），按照高斯分布生成0.0到100.0的分数。

根据上述情况，完成如下文件的生成任务：

(1) [1分]按照学号从小到大的顺序，将每个学生信息中的5个字段按照逗号分隔，而最后一个字段加回车符，将每个学生信息转化为字符串，写入文本文件“学号-hw2.txt”，返回文件生成所需时间（单位：毫秒）；

(2) [2分] 按照学号从小到大的顺序，将每个学生信息（即5个字段）转化为字节数组byte[]，通过顺序写输出流的方式将该字节数组写入到数据文件“学号-hw2.dat1”，返回文件生成所需时间（单位：毫秒）；

(3) [3分]通过merge-sort算法对语文成绩按照从高到低的顺序对所有学生信息进行排序，若有相同成绩的学生信息，则按照学号从小到大排序，将排序后的学生信息类似的转化为字节数组byte[]，写到数据文件“学号-hw2.dat2”，返回文件生成所需时间（单位：毫秒）；

(4) [3分]构建一个二叉平衡树结构，对语文成绩的整数值（不考虑四舍五入，例如89.8对应为89）进行索引，在该二叉平衡树的叶节点，还需保存该成绩整数值对应的第一个学生信息所在文件“学号-hw2-q1.dat2”中的byte位置（即所谓的指针），从而可以根据语文成绩快速的随机访问到对应学生信息所在的byte位置，最后将该树结构，写到索引文件“学号-hw2.idx”，返回文件生成所需时间（单位：毫秒）；

根据上述生成的文件，完成如下的搜索查询任务：

(5) [2分]读取文件“学号-hw2.dat1”，根据标准输入(console)的学号信息，按照语文、数学、英语和综合分数的次序，返回该生的五门课程明细分数信息和所需查询时间（单位：毫秒）；

(6) [2分]读入文件索引文件“学号-hw2.idx”，根据标准输入(console)的两个语文成绩（例如89.5和92.1），查找包含最大小于该成绩的整数（即89）和最小大于该成绩的整数（即93）的叶节点，并找到该叶节点对应的学生信息所在byte位置，然后根据该byte位置，随机读取文件“学号-hw2.dat2”，返回在89.5与92.1区间（包含等于89.5和92.1的学生成绩）的所有学生数量和平均成绩和所需查询时间（单位：毫秒）；

2. [7分]完成一个client和server形式的文件读写的socket网络程序：

(1) [3分] 在客户端完成上述(1.5)和(1.6)的查询输入，并将输入的查询条件返回给服务器，服务端则分别文件的搜索，并将结果返回给客户端，并打印从查询条件输入到最终结果返回之间所需时间（单位：毫秒）；

(2) [4分] 在上述(2.1)功能基础之上进一步完成多线程处理用多个客户端查询请求的任务。

备注：在文件生成和查询正确的前提条件下，文件大小越少和查询时间越短，得分越高。

[参考资料]

- 1) 顺序访问: <https://mathbits.com/JavaBitsNotebook/Files/Use.html>
- 2) 随机访问: <https://www.digitalocean.com/community/tutorials/java-randomaccessfile-example>
- 3) 字节数组: <https://medium.com/@rrlinus5/stream-of-bytes-internal-concept-java-831c4d94ad4c>

**提交要求:**

- Deadline: 2025-11-09 23:59
- 每题包括3个文件: 源代码文件目录 (学号-hw2-q1/2-src), 包含生成数据文件的执行/运行文件目录 (学号-hw1-q1/2-run) 以及执行截屏文件(学号-hw1-q1/2-screen), 其中执行截屏文件要求截屏记录console输入脚本文件并输出结果。
- 通过 canvas 提交最终压缩文件包 (学号-hw2.zip)