数字取证技术(Digital Forensics Technology)习题课I

练习题

(♞ **思考**):在数字取证中,经常需要分析和识别大量日志数据中的模式和关联性,例如识别网络攻击过程中的事件关联或用户行为模式.假设有数万条由网络传感器生成的事件日志,需要建立一个事件关联矩阵,并用它与不同的用户行为模式矩阵相乘,以便快速地识别潜在的威胁和异常行为模式.由于数据量巨大,单线程计算矩阵乘法需要的时间会非常长,因此需要设计一种并行矩阵乘法算法来加速这一过程.

设计一个算法实现两个矩阵的并行乘法:输入为两个矩阵 A 和 B,矩阵 A 的维度为 $m \times n$,矩阵 B 的维度为 $n \times w$.输出为结果矩阵 C,其维度为 $m \times w$.

提示(Hint):思考矩阵乘法里哪些步骤是可以并行的;

(♞ **思考**):在数字取证领域,分析文本内容以发现潜在的模式和关键信息是一项常见的任务.假设你是一名取证专家,正在处理一项涉及大量电子邮件和聊天记录的案件.任务是从这些文本数据中,找出频率最高的100个单词二元组合及其频次,例如词组"客观-公平","物质-奖励"等.由于数据量非常庞大,需要设计一个高效的算法来处理这一任务,并确保能够快速准确地统计出最常见的单词二元组合.

提示(Hint):数据量非常庞大,这本质上是一个关于存储的算法问题;

复习(Review):有同学疑惑,为什么 HashTab 在不分治和分治的情形下差异很大;这主要是体现在时间复杂度方面:分治时是并行统计二元组频率,而合并其实所需的复杂度其实不大,而且也可以比较高效统计(这是因为我们还维护了一个词频的序,因此每个主机可以按这个内排序),而统计完后可以先过滤一遍,设定一个阈值τ(比如为5,那么小于5次的直接删去),这样又能减少很多空间复杂度,接下来再按照二元组频率排序(此时待排序的 HashTab 元素已经大大减少),就能快速获取前m位;

(♞ **思考**):(取证文件时间线排序)在数字取证过程中,通常需要将发现的证据(如文件、日志条目等)按照某种顺序排列以构建事件的时间线.在某次调查中,你获得了一份由不同来源(如文件系统元数据、网络传输记录等)收集到的时间戳数据.由于这些时间戳来自不同的系统,格式和精度也不同,你需要将它们按照真实的事件发生顺序进行排序,以辅助案件分析.

由于不同时间戳具有不同的格式和表示精度(如"YYYY-MM-DD HH:MM:SS","Unix时间戳","MM/DD/YYYY HH:MM"等),传统的按数字大小排序的方法不能直接适用.因此需要设计一个分治算法,该算法能够高效地处理此类复合格式的时间戳排序问题,并构建正确的事件时间线.

提示(Hint):回忆上一次课讲的分治排序方法;

复习(Review):可以先批处理一遍,比如令所有时间戳的格式都为UNIX/C格式,再进行排序;