**实验报告**

**实验名称**  分治算法

**班 级**

**学 号**

**姓 名**

**成 绩**

|  |
| --- |
| **实验概述：** |
| **【实验目的及要求】**  通过编程熟悉并掌握分治算法，了解最大子数组问题和矩阵相乘的 𝑆𝑡𝑟𝑎𝑠𝑠𝑒𝑛 算法。 |
| **实验内容：** |
| **【实验代码1】见附件“求连续子数组的最大和.py”**  **【实验过程1】**  编写程序，分别通过遍历求解和分治算法求解问题。  分治算法的思想是设𝑀=n/2，从𝑀处将原数组分成两个数组，先在两个数组中分别搜索最大子数组，再在跨越分界线 𝑀 的数组中寻找最大子数组。整个数组的最大子数组一定是在左半个数组的最大子数组，右半个数组的最大子数组和跨越分界线𝑀的最大子数组这三种情况中选取和的最大值。上述划分过程还可继续进行，直到不可再分。  在编写的程序中：第一次输入，输入1求解常规数组问题，输入2求解100位随机数组问题，输入3求解1000位随机数组问题，输入0退出程序；第二次输入，输入0使用遍历算法，输入1使用分治算法。  **【实验结论1】**  **常规数组：**    **100位随机数组：**    **1000位随机数组：**    分析：在对遍历算法和分治算法的时间复杂度分析中，遍历算法的时间复杂度约为O()，而分治算法的时间复杂度约为O(nlogn)。在实际运行过程中也证实了这一点，分治算法的效率要比遍历算法高得多，这也得益于其“分而治之”的思想，利用递归原理，将大问题分解成小问题。  **------------------------------------------------------------------**  **【实验代码2】见附件“矩阵乘法的 𝑺𝒕𝒓𝒂𝒔𝒔𝒆𝒏 算法.py”**  **【实验过程2】**  编写程序，分别通过朴素矩阵算法和𝑆𝑡𝑟𝑎𝑠𝑠𝑒𝑛算法求解矩阵乘法。  𝑆𝑡𝑟𝑎𝑠𝑠𝑒𝑛算法的思想核心还是分治算法，将大矩阵分解成小矩阵问题，分别计算求解，最后由局部解合并成最终解。其中为了提高算法效率，用更多的加减法替代乘法。  在编写的程序中：第一次输入，输入0退出程序，输入1不重新生成列表，输入大于1的整数n生成n×n数组代替n维矩阵；第二次输入，输入1使用朴素矩阵算法，输入2使用𝑆𝑡𝑟𝑎𝑠𝑠𝑒𝑛算法。  **【实验结论2】**  **50维矩阵：**    **100维矩阵：**    **200维矩阵：**    **500维矩阵：**      **1000维矩阵：**      分析：在对朴素矩阵算法和𝑆𝑡𝑟𝑎𝑠𝑠𝑒𝑛算法的时间复杂度分析中，𝑆𝑡𝑟𝑎𝑠𝑠𝑒𝑛算法降低时间复杂度的关键在于减少乘法次数，但是在整体复杂度较低时效果不明显，只有在整体复杂度较高时效果很明显。在实际运行过程中也证实了这一点，50、100、200维矩阵运算时差别不明显，500、1000维矩阵运算时𝑡𝑟𝑎𝑠𝑠𝑒𝑛算法效率明显更高，运行时间更短。 |