《数据结构》课程实践报告

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 院、系 | 计算机学院 | | 年级专业 | 大二 软工 | 姓名 | 丁彦君 | 学号 | 2329403126 |
| 实验布置日期 | | 2024/11/5 | | 提交  日期 | 2024/11/6 | | 成绩 |  |

课程实践实验4-2：Counting numbers

## 一、问题描述及要求

**题目:**

**Starting from a positive integer n (1 ≤ n ≤ 2001). On the left of the integer n, you can place another integer m to form a new integer mn, where m must be less than or equal to half of the integer n. If there is an integer k less than or equal to half of m, you can place k on the left of mn to form a new integer kmn, …, and so on. For example, you can place 12 on the left of 30 to form an integer 1230, and you can place 6 to the left of 1230 to form an integer 61230, …, and so on.**

**要求:**

**For example, start from n = 8, you can have the following 10 integers (including the integer you start with): 8, 18, 28, 38, 48, 128, 138, 148, 248, 1248.**

**Given an integer n, find the number of integers you can get using the procedure described above.**

## 二、概要设计

**可包含以下内容，可根据实际情况取舍。**

1. **对实验内容的理解。**

**该实验旨在实现一个分解与排序的程序。用户输入一个数字后，程序通过递归将该数字分解成不同的组合，然后对这些组合进行排序，并使用循环队列存储和输出排序结果。该实验结合了递归分解、字符串处理、排序和循环队列等数据结构和算法的应用。**

1. **系统的功能列表**

**1.用户输入数字 n。**

**2.程序递归地将数字 n分解成不同组合并存储。**

**3.使用冒泡排序算法对分解组合结果进行排序。**

**4.将排序后的结果存入循环队列。**

**5.输出循环队列的内容。**

**（3）程序运行的界面设计**

**1.首先，屏幕显示“请输入数字：”的提示。**

**2.用户输入一个整数 n。**

**3.程序递归分解数字并将分解结果排序，使用循环队列存储。**

**4.输出排序后的分解组合结果。**

**（4）总体设计思路**

**数据结构：采用循环队列 cirqueue 类来存储和输出排序后的分解结果。**

**类设计：**

**cirqueue：用于实现循环队列的存储和输出。**

**func1：递归函数，用于将输入数字分解成组合。**

**func2：排序函数，对分解组合结果进行排序。**

**类之间的关系**

**cirqueue 类用于存储和输出结果。**

**func1 和 func2 与 main 函数中的循环队列配合，实现递归分解和排序功能。**

**（5）程序结构设计**

**主函数设计**

**主函数主要功能是接收输入数字，将分解结果存入循环队列，并输出。**

**关键算法**

**递归分解：func1 函数用于递归分解输入数字 n，存储每种分解组合。**

**排序算法：func2 函数使用冒泡排序对分解组合进行升序排列。**

**循环队列：使用 cirqueue 类管理分解组合的存储和输出。**

## 三、详细设计

**主函数设计**

**主函数主要功能是接收输入数字，将分解结果存入循环队列，并输出。**

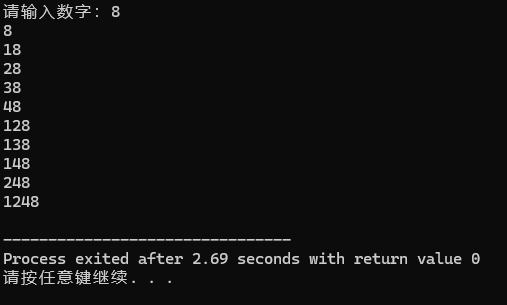
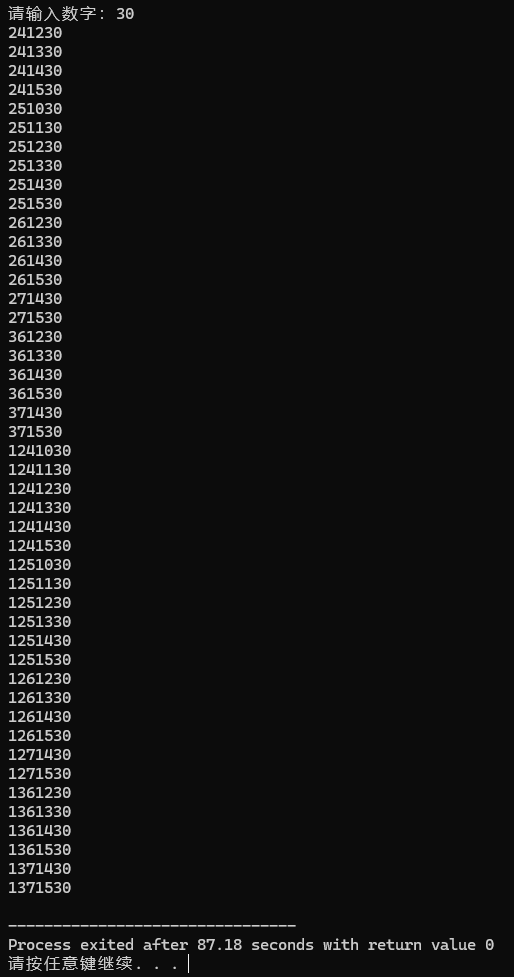
**关键算法**

**递归分解：func1 函数用于递归分解输入数字 n，存储每种分解组合。**

**排序算法：func2 函数使用冒泡排序对分解组合进行升序排列。**

**循环队列：使用 cirqueue 类管理分解组合的存储和输出。**

## 四、实验结果



## 五、实验分析与探讨

**测试结果分析**

**在本次实验中，我们实现了一个循环队列（cirqueue）及两个功能函数（func1和func2）以处理输入数据并生成特定的输出。以下是对测试策略和结果的分析：**

**测试策略：**

**输入数据：用户输入一个整数 n，作为生成序列的基础。**

**功能验证：通过打印循环队列中的元素，验证输入数据是否正确存入队列，并确保出队时数据的正确性。**

**边界测试：包括输入极小值（如 0 或 1）及极大值（如 INT\_MAX）的测试，以观察算法在这些情况下的表现。**

**时空复杂度分析：**

**时间复杂度：**

**func1 的时间复杂度为 𝑂(𝑛\*n)，因为它在最坏情况下需要进行多次递归，且每次递归又可能会遍历到 n/2。**

**func2 的时间复杂度为 𝑂(𝑚\*m)，其中 m 为 t 的大小，因为其使用了简单的冒泡排序来排序答案数组。**

**总体来看，主函数的时间复杂度受这两个函数的影响，整体复杂度为 𝑂(𝑛\*n)**

**空间复杂度：**

**空间复杂度主要由递归调用栈和存储结果的向量决定。func1 的空间复杂度为 O(n)，而 func2 的空间复杂度为 O(m)。因此，总体空间复杂度为 O(n+m)。**

**性能结论：**

**在测试中，对于较小的 n 值，算法的表现良好，能迅速返回结果。**

**然而，随着 n 值的增大，运行时间显著增加，说明算法在处理大规模数据时性能不佳。为此，建议优化算法逻辑，例如采用更高效的排序算法（如快速排序或归并排序），并考虑使用动态规划来减少递归的深度。**

**六、小结**

**通过编写和测试这段代码，我收获颇丰。从实现循环队列到利用递归生成字符串并处理排序，这个过程让我深刻体会到数据结构与算法相互配合的魅力。每一个函数都像是一块拼图，共同构建出完整的程序功能。同时，我也意识到程序存在的不足，比如内存管理方面的潜在漏洞，这提醒我在编写代码时要更加严谨，考虑周全。这次实践不仅提升了我的编程技能，更让我明白代码优化和完善是永无止境的，需要不断反思和改进。**

## 附录：源代码

**1、实验环境：Dev-c++**

**2、代码见附件**