

算法基础LAB3实验报告

PB19071535徐昊天

一.实验内容

1. 给定任意一个序列 a_1, \dots, a_n ，其中没有重复元素。计算满足 $i < j < k$ 且 $a_j > a_i > a_k$ 的三元组 (i, j, k) 数。
2. 给定任意一个序列 a_1, \dots, a_n 和参数 K ，计算满足 $\sum_{i=L+1}^R a_i \geq K$ 的最短区间长度 $R - L$ 。
3. 给定 n 种硬币，它们面额分别为 a_1, \dots, a_n 。用这些硬币进行找零，金额为 V ，计算找零组合。
4. 给定一个01矩阵 $A \in \{0, 1\}^{n \times n}$ ， A 中会存在很多元素全1且连续的子方阵 $S \in \{1\}^{k \times k}$ 。计算最大子方阵 S 的边长 k 。

二.实验目的

1. 利用OJ编程，强化编程能力，培养编程细节，提升编程素养。
2. 熟悉书本和课堂上涉及的算法，进一步巩固算法知识。
3. 通过动态规划和贪心算法处理问题，提升解决问题的效率。

三.实验内容

I.领奖台

参考作业中设计过得逆序对算法和三元逆序对算法，对归并排序进行修改：

1. 设计数据结构从而简化运算

设计结构体，其中包括序列元素的值、序列中以该结点在前的逆序对数、序列中以该结点在后的逆序对数、该结点在数组中的原位置。

2. 分情况讨论

由于在分治法的基础上作修改，故应分为两种情况： i, j 在左边数组中， k 在右边数组中； i 在左边数组中， j, k 在右边数组中。

第一种情况：只需对 i 分别讨论，找到左侧数组 i 右侧不和其构成逆序对的元素数量与右侧数组和其构成逆序对的元素数量相乘即可得到结果。

第二种情况：首先计算 $a_j > a_k, a_j > a_i$ 的数量，可结合作业第二问进行统计；其次计算 $a_j > a_k > a_i$ 的数量，该问题可作为第一种情况的对称情况；最后将以上两者相减即可得到结果。

将以上两种情况数量相加，即为所求三元组数。

II.最短达标区间

首先用数组储存前 i 位的序列元素之和，通过遍历序列并合理调用双端队列即可。

1. 若队列非空且从队头到 i 的范围满足达标区间的条件，则比较此区间长度与已有的最短区间长度，取其最小值，并将队头出列。
2. 若队列非空且前 i 位序列元素之和小于从0到队尾元素对应下标的元素之和，则将队尾元素出队。
3. 在每一轮迭代结束后将下标入队尾。

如此即可满足最后返回的区间大小达标且最短。

III.找零

运用动态规划的方法，开一个数组 dp 用于储存不同金额时的找零方案数，并通过二层循环在 $dp[j]$ 上加上所有 $dp[j-a[i-1]]$ ，即去除一定金额时的找零方案数，并在计算的同时对 1000000007 取余，循环结束后即可得到最终结果。

IV.最大连续子方阵

通过二层循环对方阵中为1的元素进行遍历，并通过查询元素左侧元素、上侧元素、左上侧元素并取其最小值，然后将该元素设为得到的最小值加一，即为该元素为右下角的最大连续子方阵边长。在遍历的同时更新方阵的最大连续子方阵边长，遍历结束后即可得到最大连续子方阵的边长。

四.实验收获与感想

1. 利用OJ编程，养成了更好的编程习惯，促进自身追求更优美的代码风格，提升了编程素养。
2. 提升了编程能力，提前适应上机编程的难度，为将来刷算法题或机试打下了基础。
3. 增强了对代码细节的把控能力，对程序的追求不止于结果正确，而是追求更低的时间空间消耗、数据的完整性和精确性等。
4. 增强了对经典算法的理解，通过复现算法更加深入了解了算法的原理和过程。
5. 通过设计贪心算法和动态规划解决问题，提升了编程的思维能力。