

作业一：基于主定理与递归树进行时间复杂度的分析

题目一

请用主定理方法和递归树方法分别分析以下递归表达式的时间复杂度：

$$T(n) = 8T\left(\frac{n}{2}\right) + 100n^3$$

主定理方法分析：

递归树方法分析：

题目二

标准的矩阵乘法算法需要 $O(n^3)$ 的时间。Strassen 矩阵乘法是一种更高效的分治算法。它将一个 $n \times n$ 的大矩阵分解为若干个 $n/2 \times n/2$ 的子矩阵，巧妙地通过 7 次子矩阵的递归乘法和一系列的矩阵加减法（时间复杂度为 $O(n^2)$ ）来完成计算，减少了递归调用的次数。

Strassen 矩阵乘法的时间复杂度可以用以下递归表达式描述，请用主定理方法和递归树方法分别分析其时间复杂度。

$$T(n) = 7T\left(\frac{n}{2}\right) + O(n^2)$$

主定理方法分析：

递归树方法分析：

作业二：算法设计与应用

题目描述：

- 本次作业共包含 5 个问题，每个问题要求用 Python 编程实现
- 重点考察算法设计与应用，特别是数组相关的基础算法
- 请根据题目要求实现相应的算法

问题 1-1：最大子数组和

给定一个整数数组 ‘nums’，请找到和最大的连续子数组，并返回其和。

示例：

- 输入：‘nums = [-2,1,-3,4,-1,2,1,-5,4]’ → 输出：‘6’ → 连续子数组 ‘[4,-1,2,1]’ 的和最大，为 6。
- 输入：‘nums = [1]’ → 输出：‘1’ → 连续子数组 ‘[1]’ 的和为 1。
- 输入：‘nums = [5,4,-1,7,8]’ → 输出：‘23’ → 连续子数组 ‘[5,4,-1,7,8]’ 的和为 23。

约束：

- $1 \leq \text{nums.length} \leq 10^5$
- $-10^4 \leq \text{nums}[i] \leq 10^4$

进阶：在掌握 $O(n)$ 解法（Kadane 算法）后，尝试用分治思想再实现一次。

问题 1-2：寻找峰值

给定一个下标从 0 开始的整数数组 ‘nums’，找到任意一个峰值元素并返回其下标。峰值元素严格大于其相邻元素。可将 $\text{nums}[-1] = \text{nums}[n] = -\infty$ 。

示例：

- 输入：‘nums = [1,2,3,1]’ → 输出：‘2’ → ‘nums[2] = 3’ 为峰值。
- 输入：‘nums = [1,2,1,3,5,6,4]’ → 输出：‘5’ → ‘nums[5] = 6’ 为峰值（下标 ‘1’ 也可）。

约束：

- $1 \leq \text{nums.length} \leq 1000$
- $-2^{31} \leq \text{nums}[i] \leq 2^{31} - 1$
- 对所有有效 i , 均有 $\text{nums}[i] \neq \text{nums}[i + 1]$ 。

问题 1-3: 多数元素

给定大小为 ' n ' 的数组 ' nums '，返回出现次数超过 $\lfloor \frac{n}{2} \rfloor$ 的元素。可假设多数元素必定存在。

示例:

- 输入: ' $\text{nums} = [3, 2, 3]$ ' \rightarrow 输出: '3'
- 输入: ' $\text{nums} = [2, 2, 1, 1, 1, 2, 2]$ ' \rightarrow 输出: '2'

约束:

- $n = \text{nums.length}$
- $1 \leq n \leq 5 \times 10^4$
- $-10^9 \leq \text{nums}[i] \leq 10^9$

进阶: 能否在线性时间、 $O(1)$ 额外空间内完成? (Boyer-Moore 投票法)

问题 1-4: 寻找重复数

给定一个长度为 ' $n + 1$ ' 的整数数组 ' nums '，其中每个整数都在区间 ' $[1, n]$ ' 内，数组中仅有一个整数重复出现至少两次。请在不修改数组、且仅使用常数级额外空间的前提下，返回这个重复数。

示例:

- 输入: ' $\text{nums} = [1, 3, 4, 2, 2]$ ' \rightarrow 输出: '2'
- 输入: ' $\text{nums} = [3, 1, 3, 4, 2]$ ' \rightarrow 输出: '3'
- 输入: ' $\text{nums} = [3, 3, 3, 3, 3]$ ' \rightarrow 输出: '3'

约束:

- $1 \leq n \leq 10^5$
- $\text{nums.length} = n + 1$
- $1 \leq \text{nums}[i] \leq n$

- 除恰有一个整数出现至少两次外，其余整数均只出现一次。

进阶：

- 为什么一定存在重复数？（抽屉原理）
- 能否在 $O(n)$ 时间内求解？（Floyd 龟兔赛跑）

问题 2-1：数组中的第 K 大元素

给定整数数组 ‘nums’ 和整数 ‘k’，返回数组中第 ‘k’ 大的元素（非第 ‘k’ 个不同元素）。你能在不整体排序的情况下完成吗？

示例：

- 输入：‘nums = [3,2,1,5,6,4], k = 2’ → 输出：‘5’
- 输入：‘nums = [3,2,3,1,2,4,5,5,6], k = 4’ → 输出：‘4’

约束：

- $1 \leq k \leq \text{nums.length} \leq 10^5$
- $-10^4 \leq \text{nums}[i] \leq 10^4$

问题 3-1：医院诊室分配系统

问题背景：设计一个系统，将病人分配到 ‘k’ 个诊室，并最终生成一个用于取药的全局有序队列。

任务要求：

- `assign_patients_to_clinics(arrivals: List[int], k: int) -> List[List[int]]`
- `merge_clinic_queues(queues: List[List[int]]) -> List[int]`
- `process_hospital_queue(arrivals: List[int], k: int) -> List[int]`

问题描述：有 ‘k’ 个诊室，‘n’ 个病人，病人按到达顺序组成到达列表（顺序与编号大小无关）。每个诊室内病人按编号升序维护，最终需要合并为全局有序队列。

输入输出：

- 输入：到达序列 ‘arrivals’（长度为 ‘n’，元素为互不相同的编号 ‘1..n’），整数 ‘k’；
- 输出：全局有序队列。