第一部分：分治策略 (Divide and Conquer)

问题 1: 最大子数组和 (Maximum Subarray Sum)

给定一个整数数组 `nums`，找到一个具有最大和的连续子数组（子数组最少包含一个元素），返回其最大和。

要求:

1. 描述解决此问题的分治策略核心思想（如何分解、解决、合并）。
2. 给出基于分治策略的算法伪代码。
3. 分析该算法的时间复杂度，并写出其递推关系式。

问题 2: 寻找峰值 (Find Peak Element)

一个“峰值”元素是指其值严格大于左右相邻值的元素。给定一个整数数组 `nums`，其相邻元素不相等。找到数组中的任意一个峰值，并返回其索引。你可以假设 `nums[-1] = nums[n] = -∞`。

要求:

1. 描述如何使用类似二分查找的分治思想来解决这个问题。
2. 给出算法的伪代码，要求时间复杂度为 $O(\log n)$。
3. 简要说明算法的正确性。

问题 3: 搜索旋转排序数组 (Search in Rotated Sorted Array)

一个按升序排序的数组在预先未知的某个点上进行了旋转（例如，数组 `[0,1,2,4,5,6,7]` 可能变为 `[4,5,6,7,0,1,2]`）。给定一个目标值 `target`，请在数组中搜索。如果找到目标值，返回其索引；否则，返回 -1。假设数组中无重复元素。

要求:

1. 描述如何修改二分查找算法（一种分治策略）来处理旋转带来的影响。
2. 给出算法的伪代码，要求时间复杂度为 $O(\log n)$。

问题 4: 平面最近点对问题 (Closest Pair of Points)

在平面上给定 n 个点，找出所有点对中距离最近的两个点，并返回它们的距离。

要求:

1. 详细描述解决此问题的分治策略，特别是“合并”步骤中如何处理跨越分割线的点对。
2. 给出算法的伪代码框架。
3. 分析算法的时间复杂度。

问题 5: 计算逆序对数量 (Count of Inversions)

在数组 `nums` 中，如果 `i < j` 且 `nums[i] > nums[j]`，那么我们称 `(nums[i], nums[j])` 是一个逆序对。给定一个数组，求出其中逆序对的总数。

要求:

1. 说明如何将此问题与归并排序算法（一种分治算法）结合起来。
2. 给出基于归并排序思想的算法伪代码。
3. 分析算法的时间复杂度。

问题 6: 第 K 个大的元素 (Kth Largest Element in an Array)

给定整数数组 `nums` 和一个整数 k，请返回数组中第 k 个最大的元素。

要求:

1. 描述如何利用快速排序中的划分 (partition) 操作（一种分治思想）来解决此问题。
2. 给出算法的伪代码。
3. 分析该算法的平均时间复杂度和最坏情况下的时间复杂度。

问题 7: 大整数乘法 (Karatsuba Algorithm)

给定两个 n 位的二进制大整数 X 和 Y，设计一个比传统 $O(n^2)$ 方法更有效的分治算法来计算它们的乘积 $X \times Y$。

要求:

1. 描述 Karatsuba 算法的核心思想，即如何将一次 n 位乘法分解为三次 $n/2$ 位乘法。
2. 写出算法的递推关系式。
3. 给出算法的伪代码。

问题 8: 天际线问题 (The Skyline Problem)

城市的天际线是从远处观看该城市中所有建筑物时形成的轮廓。给你一个二维数组 `buildings`，其中 `buildings[i] = [left_i, right_i, height_i]` 表示第 i 座建筑物的左右边界和高度。请你输出由这些建筑物形成的天际线的“关键点”（轮廓的左端点）。

要求:

1. 描述解决此问题的分治策略，特别是如何合并两个子天际线。
2. 给出算法的伪代码框架。

第二部分：回溯算法 (Backtracking)

问题 9: 全排列 (Permutations)

给定一个不含重复数字的数组 `nums`，返回其所有可能的全排列。

要求:

1. 描述回溯算法解决此问题的基本思路，包括“路径”、“选择列表”和“结束条件”。
2. 给出算法的伪代码。
3. 为避免重复使用元素，通常使用一个 `used` 数组，请在伪代码中体现。

问题 10: 组合总和 (Combination Sum)

给定一个无重复元素的数组 `candidates` 和一个目标数 `target`，找出 `candidates` 中所有可以使数字和为 `target` 的组合。`candidates` 中的数字可以无限制重复被选取。

要求:

1. 描述回溯算法的思路。如何处理“数字可以重复使用”这一特性?
2. 给出算法的伪代码。为了避免生成重复的组合（如 `[2,2,3]` 和 `[2,3,2]`），应如何设计搜索过程?

问题 11: N 皇后问题 (N-Queens)

n 皇后问题要求将 n 个皇后放置在 $n \times n$ 的棋盘上，使得皇后之间不能相互攻击（任何两个皇后不能处于同一行、同一列或同一斜线上）。给定一个整数 n，返回所有不同的 n 皇后问题的解决方案。

要求:

1. 描述如何按行放置皇后，并使用回溯法进行搜索。
2. 设计一个高效的 `isValid` 函数（或使用辅助数组）来判断当前位置是否可以放置皇后。请描述你的剪枝策略。
3. 给出算法的伪代码。

问题 12: 单词搜索 (Word Search)

给定一个 $m \times n$ 二维字符网格 `board` 和一个字符串单词 `word`。如果 `word` 存在于网格中，则返回 `true`；否则，返回 `false`。单词必须按照字母顺序，通过相邻的单元格内的字母构成，其中“相邻”单元格是那些水平相邻或垂直相邻的单元格。同一个单元格内的字母不允许被重复使用。

要求:

1. 描述如何将此问题建模为在图（网格）上的深度优先搜索（DFS）问题。
2. 给出回溯算法的伪代码，并说明如何避免重复访问单元格。

问题 13: 划分为 K 个相等的子集 (Partition to K Equal Sum Subsets)

给定一个整数数组 `nums` 和一个正整数 k，找出是否有可能将这个数组分成 k 个非空子集，使得每个子集的元素和都相等。

要求:

1. 首先，进行哪些预判断可以提前确定无解（剪枝)?
2. 将问题看作是将 n 个数字放入 k 个桶中，每个桶的容量为 `sum(nums) / k`。描述以此为基础的回溯策略。
3. 给出算法的伪代码，并思考如何进行优化剪枝（例如对 `nums` 排序)。

问题 14: 生成括号 (Generate Parentheses)

数字 n 代表生成括号的对数，请你设计一个函数，用于能够生成所有可能的并且有效的括号组合。

要求:

1. 描述生成有效括号的回溯策略。需要维护哪些状态变量?
2. 剪枝条件是什么?（即在什么情况下，继续添加左括号或右括号会产生无效序列?)
3. 给出算法的伪代码。

问题 15: 数独求解器 (Sudoku Solver)

编写一个程序，通过填充空格来解决数独问题。

要求:

1. 描述解决此问题的回溯框架。
2. 如何高效地检查在某个位置填入一个数字是否合法（即满足行、列、九宫格的唯一性要求)?
3. 给出算法的伪代码。

第三部分：经典模型泛化与应用 (Generalization and Application of Classic Models)

本部分旨在考察对经典算法模型的识别和应用能力。重点在于将问题转化为已知模型，并套用其思想，通常无需从头证明贪心选择的正确性。

问题 16: 合并石子 / 连接木棍的最小代价 (Minimum Cost to Connect Sticks)

有 N 堆石子，其数量分别为 `s_1, s_2, ..., s_N`。现在要将这些石子合并成一堆。每次合并只能选择两堆石子，合并的代价为这两堆石子的数量之和。求将所有石子合并成一堆的最小总代价。

要求:

1. 指出此问题与哪个经典算法模型（如霍夫曼编码）具有相同的思想内核。
2. 描述解决此问题的贪心策略。
3. 给出实现该策略的伪代码。推荐使用哪种数据结构来高效地找到每次要合并的对象?

问题 17: 最小生成树应用：连接所有村庄的最低成本

假设你是某地区的道路规划师，有 N 个村庄。现在你有一份备选的道路修建清单，清单中每条道路连接两个村庄，并有相应的修建成本。你的任务是设计一个道路网络，使得所有村庄都能相互连通（直接或间接），并且总的修建成本最低。

要求:

1. 指出这个问题在图论中对应的经典模型是什么。
2. 选择 Prim 算法或 Kruskal 算法中的一种，描述其贪心选择性质。
3. 给出你所选算法的伪代码。

问题 18: 单源最短路径应用：网络延迟时间 (Network Delay Time)

有 N 个网络节点，标记为 1 到 N。给定一个列表 `times`，表示信号经过有向边的传递时间，其中 `times[i] = (u, v, w)`，u 是源节点，v 是目标节点，w 是信号从 u 到 v 需要的时间。现在，我们从某个特定节点 K 发出一个信号。求所有节点都收到信号所需要的最短时间。如果有些节点无法收到信号，则返回 -1。

要求:

1. 指出这个问题对应的经典图论模型。
2. 描述解决此模型的经典算法（如 Dijkstra 算法）的贪心思想。
3. 给出该算法的伪代码，并说明如何使用优先队列进行优化。

问题 19: 活动选择问题 (Activity Selection)

有 n 个活动，每个活动都有一个开始时间 s_i 和结束时间 f_i。你只有一个会议室，一次只能举办一个活动。设计一个算法，选择尽可能多的活动，使得这些活动互不冲突。

要求:

1. 描述解决此问题的贪心策略（应该按开始时间排序还是结束时间排序?）。
2. 给出算法的伪代码。
3. 将此模型泛化：如何解决“无重叠区间”问题（给定一个区间集合，找到需要移除区间的最小数量，使剩余区间互不重叠)?

问题 20: 击败怪兽 (Minimize Total Damage)

在某个游戏中，有 n 只怪兽。击败第 i 只怪兽需要花费 T_i 的时间，在此期间，其它未被攻击的怪兽每单位时间会对村庄造成破坏。第 j 只怪兽每单位时间造成的破坏值为 D_j。请确定一个击败所有怪兽的顺序，使得村庄受到的总破坏值最小。

要求:

1. 这是一个基于排序的贪心问题。请推导或直接给出用于排序的指标。
2. 描述你的贪心策略。
3. 给出算法的伪代码。

问题 21: 加油站 (Gas Station)

在一条环形路线上有 n 个加油站，第 i 个加油站有汽油 `gas[i]` 升。你有一辆油箱无限大的汽车，从第 i 个加油站开往第 $i+1$ 个加油站需要消耗 `cost[i]` 升汽油。你从其中的一个加油站出发，初始时油箱为空。如果你可以绕环路行驶一周，则返回出发时加油站的编号，否则返回 -1。题目保证解是唯一的。

要求:

1. 描述解决此问题的贪心策略。如何通过一次遍历就找到可行的起点?
2. 给出算法的伪代码，要求时间复杂度为 $O(n)$。

问题 22: 根据身高重建队列 (Queue Reconstruction by Height)

假设有打乱顺序的一群人站成一个队列。每个人由一个整数对 `(h, k)` 表示，其中 h 是这个人的身高，k 是排在这个人前面且身高大于或等于 h 的人数。请编写一个算法，根据这个列表重建出原始的队列。

要求:

1. 描述一个可以解决此问题的贪心策略，该策略通常包含“排序”和“插入”两个步骤。
2. 给出算法的伪代码。
3. 简要分析为什么这个贪心策略是正确的。

问题 23: 拼接数字以获得最小值 (Minimum Number by Concatenation)

给定一个正整数数组，请将数组里所有数字拼接起来排成一个数，打印能拼接出的所有数字中最小的一个。例如输入数组 `{3, 32, 321}`，则打印出这三个数字能排成的最小数字为 `321323`。

要求:

1. 定义一种新的“小于”比较规则，用于对数组中的数字（以字符串形式）进行排序。
2. 描述基于此规则的贪心排序策略。
3. 给出算法的伪代码。

问题 24: 跳跃游戏 (Jump Game)

给定一个非负整数数组 `nums`，你最初位于数组的第一个位置。数组中的每个元素代表你在该位置可以跳跃的最大长度。判断你是否能够到达最后一个位置。

要求:

1. 描述解决此问题的贪心策略：在每一步中，维护一个“可达到的最远距离”。
2. 给出算法的伪代码，要求时间复杂度为 $O(n)$。

问题 25: 任务调度器 (Task Scheduler)

给你一个用字符数组 `tasks` 表示的 CPU 需要执行的任务列表，其中每个字母代表一种不同种类的任务。任务可以以任意顺序执行，并且每个任务都可以在一个单位时间内执行完。在任何一个单位时间，CPU 可以执行一个任务，或者处于待命状态。但是，两个相同种类的任务之间必须有长度为 n 的冷却时间。返回完成所有任务所需要的 **最短时间**。

要求:

1. 分析最短时间的构成，并描述其贪心策略（每次应该优先执行哪种任务?）。
2. 基于你的分析，推导出一个计算最短时间的数学公式或构造方法。
3. 给出实现该构造方法的伪代码。