

关键词	关键词 (包括但不限于)	输入参数 (范围或数据类型)	解题方案 (优先度从左至右, 从上至下)	LeetCode 例题		
Add Up to Target - Closer-or-Equals	rsSum		哈希表、排序、双指针、回溯法/度 注意时间复杂度要求场景	Q1 Q15 Q39		
Achieve Task/Requirement Can-or-Not/Count	(最优)分配/搭配问题 背包问题	参数(们)决定递归深度 > 1000	贪心、DP、基于二分图的匈牙利算法、数学、模拟、暴力解/DFS/回溯法+剪枝优化	Q55 Q416 Q135 Q621 Q1014		
Achieve Task/Requirement with Optimal	模拟问题	参数(们)决定递归深度 < 1000	暴力解/DFS/回溯法+剪枝优化、贪心、模拟、DP	Q2305		
最优子结构	Longest Substring Maximum Subarray		DP、双指针/滑动窗口	Q3 Q5 Q53 Q152 Q300 Q1048 Q1567		
XXXest Meet-Requirement SubXXX			自平衡二叉树/TreeMap、堆/优先队列	Q4 Q295		
Find Median			模拟算法+进位带标记变量、栈、前缀和、差分数组	Q2 Q43 Q224 Q227 Q228		
Number Calculation in XXX/Data-Structure/Data-Type	基本计算/最简系列		栈、双指针、标记变量	Q20 Q131 Q234 Q678		
Symmetry	Palindrome Parentheses		DP/记忆化搜索、(组合)数学、剪枝+暴力解/递归迭代、回溯法	Q226 Q56 Q62 Q70		
All Possible Combinations (Count)	爬楼梯系列 Jump Game Chess	参数(们)决定递归深度 > 1000	剪枝+暴力解/递归迭代、回溯法、DP/记忆化搜索、(组合)数学	Q17 Q39 Q46 Q688		
Merge Multiple XXX/Data-Structure		参数(们)决定递归深度 < 1000	模拟、递归、多路归并/最小堆、排序	Q21 Q23 Q56		
Find in Sorted/Unsorted XXX/Data-Structure/Data-Type	Find Search	目标集合的元素为 Digit	二分搜索	Q33 Q34 Q287 Q240		
Match	Regular Expression Find Word in XXX		位运算	Q136		
(Fixed/Range (Optimal) Value/Area	Last/Next (Max/Min/Bigger/Smaller) Number/Sum 蓄水/积水面积(二维或三维等等)	输入可以是: 树节点 线性表 Matrix/矩阵	DP、递归、字典树、哈希表、KMP	Q10 Q20 Q49		
状态机 (模拟)状态组合 Ordered Optimal Calculation	股票买卖系列 打家劫舍系列	输入可以是: 树节点 线性表 Matrix/矩阵	DP	Q121 Q123 Q309 Q188 Q198 Q213 Q337		
Manipulation Operation	Rotate Reverse Swap	输入可以是: 树节点 线性表 Matrix/矩阵	模拟算法/遍历+标记变量、排序	Q48 Q189 Q226 Q208		
	String/Word Operation/Process/Transfer Digit/Number/Bits Operation/Process/Transfer	字符串 数字	DP、剪枝+暴力解/递归 位运算	Q72 Q751		
遍历	Traversal Iterator	输入为树节点 输入为图节点	暴力解/DFS/BFS 暴力解/DFS/BFS、Dijkstra	Q94 Q102 Q104 Q105 Q173 Q341		
Copy Clone			哈希表(集)、指针或引用	Q138		
Cycle Check		输入通常为链表或一维数据结构 (若输入为图或 Matrix/矩阵等多维数据结构, 则不是 Cycle Check 而是 Find Connection)	双指针/快慢指针、哈希表(集)	Q141 Q142		
Find Path/Connection		输入可以是: 树节点 图节点 Matrix/矩阵	暴力解/DFS/BFS、回溯法、模拟、DP、并查集	Q62 Q63 Q684		
Design a XXX	(Basic Design) 描述语言内置存在的数据结构或类 (Advanced Design) 不常用/不适用的或编程语言内置没有的数据结构或类(Twitter)	考虑以下数据结构/组合使用: 数组、ArrayList、LinkedList、自定义对象、指针或引用 考虑以下数据结构/组合使用: 数组、ArrayList、(值或双向)LinkedList、自定义对象、指针或引用、哈希表、红黑树、哈希集、队列、堆/优先队列、栈、甚至语言内置工具类如 StringBuilder 等等	暴力解/DFS/BFS、回溯法、模拟、DP、并查集	Q155 Q706 Q707 Q146 Q208 Q355 Q432 Q2296		
Ranges Conflict	Same Time Calculation One Direction View/Union Skyline 花前月夜/花数统计 Meeting Room 合并重叠区间		扫描线、堆/优先队列、排序	Q218 Q253 Q759 Q2251 Q56		
Dependency Conflict/Schedule	Frequency Count Ranking (Complex) Calculate/Function Implementation		拓扑排序	Q207 Q210 Q1462		
Shape Check		目标集合为 Matrix/矩阵或多维数据结构	哈希表、桶排序、堆/优先队列、排序、自平衡二叉搜索树/红黑树/B 分治法、位运算、DP	Q347 Q50		
(Range) Update and Range Query/Calculation			暴力解/DFS/BFS、回溯法、模拟、几何/数学/由数据类型或数据结构或自定义对象以实现、扫描线、并查集	Q200 Q2013 Q391		
(Customized) 有序			树状数组、线段树	Q307		
			排序、单调栈、堆/优先队列、自平衡二叉搜索树/红黑树、贪心、二分搜索	Q406		
补充:	根据描述中的问题核心/概念判断解题方向(需注意, 有些题可能看似符合以下多个 case, 但实际上可能只是其中一个, 比如在伪复杂度 的递增/螺旋型 matrix 中从最左上角/最小数出发且每次只能走相邻更大数, 问是否能恰好 K 步到达中心/最大数, 看似同时符合 Achieve Task Can-or-Not 和 Find Path, 但实际是 Achieve Task - 例题链接: [Spiraling Into Control](https://codingcompetitions.withgoogle.com/codejam/round/00000000008778ec), 也有些题是确实同时符合多个 case, 题解也是由这多个 case 的方法组合起来的, 比如 LC Q2289 就是同时符合 Achieve Task Count 和有序, 所以其题解是 DP + 单调栈, 有一些题符合一个 case, 但可能需要 case 推荐解法的多个组合起来进行解, 比如 LC Q406 符合 Customized 有序, 需要 2D 排序 + 贪心, 也有一些特殊 case 但是比较特殊也可以使用推荐之外的解法, 比如 [longest rg substring](https://leetcode.com/mythyoyle/data-structure-and-algorithm-study-notes/blob/master/Other%20Practices/04-06-2022%20mock-java) 以及 LC Q2178) 输入为参数较大时: 递归、DFS 可能行不通, 因为参数较多, 此时可以考虑迭代、BFS 或动态规划/贪心(但贪心+贪心+贪心是数据类型的固定思维, 类似 LC Q686); 相反地, 如果单题时发现输入参数 (主要是决定递归深度的那个参数) 的值不大甚至出奇的小(比如只有 2), 则很有可能迭代、动态规划/很难做, 直接找一个递归、DFS、回溯法/解就可以(例如 LC Q2305), 再不济可以尝试剪枝优化试试。 题解限制条件: 如果题目给出(或者可以问出)极端初始值/的对应响应/返回值, 则很有可能是 DP、数学归纳法(因为 DP、数学归纳法通常都需要给定一个初始值设定)。 Dynamic Programming 几乎可以解决了 tree 和 linkedlist 以外的其他复杂问题, 做 medium 以上没有时限限制的题优先考虑 DP 解 贪心可以的题统一可以: 动态规划/贪心不一定可以; 树状数组 vs 线段树: 能写树状数组的优先写树状数组, 但是树状数组可以的线段树一定可以, 线段树可以的树状数组不一定可以 求一段的最大/最小值, 逻辑简单的优先考虑 Two Pointers, Greedy, Stack 解决, 逻辑复杂的基本都是 Dynamic Programming, Divide and Conquer 输入为图(出发节点)时, 注意条件: 是否有环、是否是简单图、节点值的范围、节点数量。 输入为树(树节点)时, 注意条件: 是否是特殊类型的二叉树/二叉搜索树/可以重点考虑中序遍历)、是否平衡、节点数量。 输入为链表(头节点)时, 注意条件: 单向还是双向链表、是否有环、链表长度 输入为数字时: 注意条件: 数字是否整数、数字范围(包括正负)、 输入为字符串时, 注意条件: 字符串范围、字符串长度。 输入为 Matrix/矩阵、二维数据结构(比如数组)时, 注意条件: 值的范围、是否可以修改矩阵/原数组、矩阵或数组的大小。 输入为一维数组/数据结构时, 注意条件: 元素是否唯一、是否有有序、数组长度。					