

## 数组(及其相关结构)

```
* - [可容纳最多水的一组容器.py](../Array/ContainerWithMostWater.py)
* - [糖果公平交换.py](../Array/FairCandySwap.py)
* - [已排序的数组中找到第一和最后一个元素.py]
(../Array/FindFirstAndLastPositionOfElementInSortedArray.py)
* - [旋转过的排序数组中找到最小的元素.py](../Array/FindMinimumInRotatedSortedArray.py)
* - [旋转过的排序数组中找到最小的元素II.py]
(../Array/FindMinimumInRotatedSortedArrayII.py)
* - [足球粉儿~.py](../Array/FootballFans.py)
* - [水果分到篮子里.py](../Array/FruitIntoBaskets.py)
* - [三连递增的子序列.py](../Array/IncreasingTripletSubsequence.py)
* - [插入间隔.py](../Array/InsertInterval.py)
* - [能组合成的最大的数.py](../Array/LargestNumber.py)
* - [最长的连续子序列.py](../Array/LongestConsecutiveSequence.py)
* - [岛屿的最大区域.py](../Array/MaxAreaOfIsland.py)
* - [相乘最大的子数组.py](../Array/MaximumProductSubarray.py)
* - [最大的子数组.py](../Array/MaximumSubarray.py)
* - [合并数组.py](../Array/MergeArray.py)
* - [丢失的数字.py](../Array/MissingNumber.py)
* - [岛屿的数量.py](../Array/NumberOfIslands.py)
* - [数组中除自己外的相乘.py](../Array/ProductOfArrayExceptSelf.py)
* - [从链表尾中删除第N个节点.py](../Array/RemoveNthNodeFromEndOfList.py)
* - [从旋转过的排序数组中搜索某数.py](../Array/SearchInRotatedSortedArray.py)
* - [从旋转过的排序数组中搜索某数II.py](../Array/SearchInRotatedSortedArrayII.py)
* - [落单的数.py](../Array/SingleNumber.py)
* - [以数对为基准排序.py](../Array/SortArrayByParity.py)
* - [螺旋矩阵.py](../Array/SpiralMatrix.py)
* - [子数组合中最小的一个.py](../Array/SumOfSubarrayMinimums.py)
* - [3D图形的表面区域.py](../Array/SurfaceAreaOf3DShapes.py)
* - [被包围的区域.py](../Array/SurroundedRegions.py)
* - [两个链表相加II.py](../Array/AddTwoNumbersII.py)
* - [两个链表相加.py](../Array/AddTwoNumbers.py)
* - [两个链表相交点.py](../Array/IntersectionOfTwoLinkedLists.py)
* - [数组中第k大个元素.py](../Array/KthLargestElementInAnArray.py)
* - [链表中的环.py](../Array/LinkedListCycle.py)
* - [保持当前城市天际线的最大可增加容量.py]
(../Array/max_increase_to_keep_city_skyline.py)
* - [两个排序后的列表的中位数.py](../Array/MedianOfTwoSorted.py)
* - [合并两个排序过的列表.py](../Array/MergeTwoSortedList.py)
* - [合并排序的数组.py](../Array/MergeSortedArray.py)
* - [两个列表的(相同部分)最小索引和.py](../Array/MinimumIndexSumOfTwoLists.py)
* - [翻转对.py](Array/ReversePairs.py)
* - [2D矩阵搜索.py](../Array/Search2DMatrix.py)
* - [2D矩阵搜索II.py](../Array/Search2DMatrixII.py)
* - [第三大的数.py](../Array/ThirdMaximumNumbers.py)
* - [3数相加.py](../Array/ThreeSum.py)
* - [最频繁的K个元素.py](../Array/TopKFrequentElements.py)
* - [两数相加II数组已排序.py](../Array/TwoSumIIAlreadySorted.py)
* - [两数相加.py](../Array/two_sum.py)
* - [多重三数相加.py](../Array/3SumWithMultiplicity.py)
* - [最佳买卖时机I_II.py](../Array/BestTimeToBuyAndSellStockI_II.py)
* - [二进制子数组和.py](../Array/BinarySubarraysWithSum.py)
* - [统计小于自身且在自身之后的数.py](../Array/CountOfSmallerNumbersAfterSelf.py)
```

- \* - [寻找极值.py](../Array/FindPeakElement.py)
- \* - [FizzBuzz.py](../Array/FizzBuzz.py)
- \* - [加油站.py](../Array/GasStation.py)
- \* - [本地和全局的倒置.py](../Array/GlobalAndLocalInversions.py)
- \* - [组团的变位词.py](../Array/GroupAnagrams.py)
- \* - [两数组的重叠部分.py](../Array/IntersectionOfTwoArrays.py)
- \* - [两数组重叠部分II.py](../Array/IntersectionOfTwoArraysII.py)
- \* - [跳跃游戏.py](../Array/JumpGame.py)
- \* - [跳跃游戏II.py](../Array/JumpGameII.py)
- \* - [距离最近的人的可走过的最远距离.py](../Array/MaximizeDistanceToClosestPerson.py)
- \* - [循环子数组的最大和.py](../Array/MaximumSumCircularSubarray.py)
- \* - [最小下落路径和.py](../Array/MinimumFallingPathSum.py)
- \* - [最小路径和.py](../Array/MinimumPathSum.py)
- \* - [单双链表.py](../Array/OddEvenLinkedList.py)
- \* - [解体大数组.py](../Array/PartitionArrayIntoDisjointIntervals.py)
- \* - [解体列表.py](../Array/PartitionList.py)
- \* - [Pascal 三角I\_II.py](Array/Pascal'sTriangleI\_II.py)
- \* - [+1.py](../Array/PlusOne.py)
- \* - [从已排序的数组中删除重复数据.py](../Array/RemoveDuplicatesFromSortedArray.py)
- \* - [从已排序的数组中删除重复数据II.py](../Array/RemoveDuplicatesFromSortedArrayII.py)
- \* - [从已排序的列表中删除重复数据.py](../Array/RemoveDuplicatesFromSortedList.py)
- \* - [从已排序的列表中删除重复数据II.py](../Array/RemoveDuplicatesFromSortedListII.py)
- \* - [删除链表中的元素.py](../Array/RemoveLinkedListElements.py)
- \* - [旋转列表.py](../Array/RotateList.py)
- \* - [在矩阵中捣蛋.py](../Array/SetMatrixZeroes.py)
- \* - [洗牌.py](../Array/ShuffleAnArray.py)
- \* - [最大的滑动窗口.py](../Array/SlidingWindowMaximum.py)
- \* - [根据"对er"排序数组II.py](../Array/SortArrayByParityII.py)
- \* - [颜色颜色.py](../Array/SortColors.py)
- \* - [螺旋矩阵II.py](../Array/SpiralMatrixII.py)
- \* - [等于K的子数组.py](../Array/SubarraySumEqualsk.py)
- \* - [三角.py](../Array/Triangle.py)
- \* - [单词子集.py](../Array/WordSubsets.py)

## ## [回溯](Backtracking/)

- \* - [生成括号.py](../Backtracking/GenerateParentheses.py)
- \* - [子集.py](../Backtracking/subsets.py)

## ## [广度优先搜索](BFS/)

- \* - [树中最靠近左下的节点的值.py](../BFS/FindBottomLeftTreeValue.py)
- \* - [二叉树层级遍历.py](../BFS/BinaryTreeLevelOrderTraversal.py)
- \* - [二叉树层级遍历II.py](../BFS/BinaryTreeLevelOrderTraversalII.py)
- \* - [二叉树层级 Zigzag 遍历.py](../BFS/BinaryTreeZigzagLevelOrderTraversal.py)
- \* - [二叉树的最大深度.py](../BFS/MaximumDepthOfBinaryTree.py)
- \* - [单词阶梯.py](../BFS/WordLadder.py)

## ## [设计类](Design/)

- \* - [LRU 缓存设计.py](../Design/LRUCache.py)
- \* - [RLE 迭代器.py](../Design/RLEIterator.py)
- \* - [O(1)复杂度的插入删除和取随机容器.py](Design/InsertDeleteGetRandomO(1).py)

## ## [深度优先搜索](DFS/)

- \* - [路径和.py](DFS/PathSum.py)
- \* - [词搜索.py](DFS/WordSearch.py)
- \* - [词搜索II.py](DFS/WordSearchII.py)

## ## [动态规划](DP/)

- \* - [交错的字符串.py](DP/InterleavingString.py)
- \* - [最长的子序列.py](DP/LongestSubsequence.py)
- \* - [子数组的或运算.py](DP/BitwiseORsOfSubarray.py)
- \* - [交换硬币.py](DP/CoinChange.py)
- \* - [江洋大盗.py](DP/HouseRobber.py)
- \* - [江洋大盗II.py](DP/HouseRobberII.py)
- \* - [最长的连续递增子序列.py](DP/LongestContinuousIncreasingSubsequence.py)
- \* - [A子数组中的最大比B子数组中最小要小的子数组.py](DP/MaxASubarrayLessThanMinBSubarray.py)
- \* - [最大长度的数对链.py](DP/MaximumLengthOfPairChain.py)
- \* - [最长增长子序列的个数.py](DP/NumberOfLongestIncreasingSubsequence.py)
- \* - [挑卡片.py](DP/PickCards.py)
- \* - [唯一路径.py](DP/UniquePath.py)
- \* - [唯一路径II.py](DP/UniquePathII.py)
- \* - [爬楼.py](DP/ClimbingStairs.py)
- \* - [解码方式.py](DP/Decodeways.py)
- \* - [在某字符串中找到所有的变位词.py](DP/FindAllAnagramsInAString.py)
- \* - [转换字符串到单调递增.py](DP/FlipStringToMonotoneIncreasing.py)
- \* - [小于K的子数组积.py](DP/SubarrayProductLessThanK.py)
- \* - [单词中断处.py](DP/WordBreak.py)

## ## [堆](Heap/)

- \* - [排序矩阵中第k小个元素.py](Heap/KthSmallestElementInASortedMatrix.py)
- \* - [合并K个已排序过的链表.py](Heap/MergeKSortedLists.py)
- \* - [从数据流中找中位数.py](Heap/FindMedianFromDataStream.py)

## ## [排序](Sorted/)

- \* - [选择, 插入, 快排, 归并](Sorted/sorted.py)
- \* - [排序列表.py](Sorted/SortList.py)
- \* - [奇怪的排序II.py](Sorted/WiggleSortII.py)

## ## [纯数字问题](Number/)

- \* - [两数相除.py](Number/DivideTwoIntegers.py)
- \* - [Sqrt(x).py](Number/Sqrt(x).py)

## ## [栈](Stack/)

- \* - [可取最小值的栈.py](Stack/GetMinStack.py)
- \* - [使用栈模拟队列.py](Stack/ImplementQueueUsingStack.py)
- \* - [使用递归翻转栈.py](Stack/ReverseAStackByRecursive.py)
- \* - [直方图中最大的矩形.py](Stack/LargestRectangleInHistogram.py)
- \* - [收集雨水.py](Stack/TrappingRainWater.py)

## ## [字符串](String/)

- \* - [找到可被模板替换的字符串.py](String/FindAndReplacePattern.py)
- \* - [电话号码排列组合.py](String/LetterCombinationsOfAPhoneNumber.py)
- \* - [最长的公共前缀.py](String/LongestCommonPrefix.py)
- \* - [最小窗口子字符串.py](String/MinimumWindowSubstring.py)
- \* - [翻转整数.py](String/ReverseInteger.py)
- \* - [字符串->整数.py](String/StringToInteger.py)
- \* - [有效的数字.py](String/ValidNumber.py)
- \* - [有效的括号.py](String/ValidParentheses.py)
- \* - [大写侦查.py](String/DetectCapital.py)
- \* - [汉明距离.py](String/HammingDistance.py)
- \* - [实现StrStr().py](String/ImplementStrStr().py)
- \* - [同构字符串.py](String/IsomorphicStrings.py)
- \* - [Jewels和Stones.py](String/JewelsAndStones.py)
- \* - [最末单词的长度.py](String/LengthOfLastWord.py)
- \* - [最长的子字符串.py](String/LongestSubstrings.py)

- \* - [最长的回文子字符串.py](String/LongestPalindromicSubstrings.py)
- \* - [字符串中的分段数量.py](String/NumberOfSegmentsInString.py)
- \* - [替换词.py](String/ReplaceWords.py)
- \* - [翻转字符串.py](String/ReverseString.py)
- \* - [转换至小写.py](String/ToLowerCase.py)
- \* - [词模板.py](String/WordPattern.py)
- \* - [字符串中最先出现的唯一字符.py](String/FirstUniqueCharacterInAString.py)
- \* - [长印刷名字.py](String/LongPressedName.py)
- \* - [使括号有效的最小添加数.py](String/MinimumAddToMakeParenthesesValid.py)
- \* - [字符串中的全排列.py](String/PermutationInString.py)
- \* - [转换单词.py](String/ShiftingLetters.py)
- \* - [简化路径.py](String/SimplifyPath.py)
- \* - [根据频率排序字符.py](String/SortCharactersByFrequency.py)
- \* - [独一无二的邮箱地址.py](String/UniqueEmailAddresses.py)
- \* - [有效变位词.py](String/ValidAnagram.py)
- \* - [有效回文字符串.py](String/ValidPalindrome.py)

## [树](Tree/)

- \* - [根据前序和后序的结果生成二叉树.py](Tree/ConstructBinaryTreeFromPreorderAndPostorderTraversal.py)
- \* - [转换已排序的数组到二叉搜索树.py](Tree/ConvertSortedArrayToBinarySearchTree.py)
- \* - [转换已排序的链表到二叉搜索树.py](Tree/ConvertSortedListToBinarySearchTree.py)
- \* - [镜像树.py](Tree/SymmetricTree.py)
- \* - [有效的二叉搜索树.py](Tree/ValidateBinarySearchTree.py)
- \* - [二叉搜索树.py](Tree/BinarySearchTree.py)
- \* - [二叉树的最大路径和.py](Tree/BinaryTreeMaximumPathSum.py)
- \* - [路径和II.py](Tree/PathSumII.py)
- \* - [“结”构与解构.py](Tree/SerializeAndDeserialize.py)
- \* - [根到叶的和.py](Tree/SumRootToLeafNumbers.py)
- \* - [字典树.py](Tree/Trie.py)
- \* - [所有可能的满二叉树.py](Tree/AllPossibleFullBinaryTree.py)
- \* - [右边视角的二叉树.py](Tree/BinaryTreeRightSideView.py)
- \* - [完整二叉树插入.py](Tree/CompleteBinaryTreeInserter.py)
- \* - [根据中序和后序遍历结果中构建二叉树.py](Tree/ConstructBinaryTreeFromInorderAndPostorderTraversal.py)
- \* - [根据前序和中序遍历结果中构建二叉树.py](Tree/ConstructBinaryTreeFromPreorderAndInorderTraversal.py)
- \* - [一只二叉搜索树中第k小个数.py](Tree/KthSmallestElementInABST.py)
- \* - [二叉搜索树中最小公共祖先.py](Tree/LowestCommonAncestorOfABinarySearchTree.py)
- \* - [二叉树中最小公共祖先.py](Tree/LowestCommonAncestorOfABinaryTree.py)
- \* - [为每个节点指定下一个右边指向.py](Tree/PopulatingNextRightPointersInEachNode.py)
- \* - [为每个节点指定下一个右边指向II.py](Tree/PopulatingNextRightPointersInEachNodeII.py)
- \* - [“结”构与解构二叉树.py](Tree/SerializeAndDeserializeBinaryTree.py)