力扣总结版~查并集+图+拓扑排序+排序+滑窗

1. **最长连续序列**

给定一个未排序的整数数组，找出最长连续序列的长度。

要求算法的时间复杂度为 O(n)。

示例:

输入: [100, 4, 200, 1, 3, 2]

输出: 4

解释: 最长连续序列是 [1, 2, 3, 4]。它的长度为 4。



1. **岛屿数量**

给定一个由 '1'（陆地）和 '0'（水）组成的的二维网格，计算岛屿的数量。一个岛被水包围，并且它是通过水平方向或垂直方向上相邻的陆地连接而成的。你可以假设网格的四个边均被水包围。

示例 1:

输入:

11110

11010

11000

00000

输出: 1







1. **被围绕的区域**

给定一个二维的矩阵，包含 'X' 和 'O'（字母 O）。

找到所有被 'X' 围绕的区域，并将这些区域里所有的 'O' 用 'X' 填充。

示例

X X X X

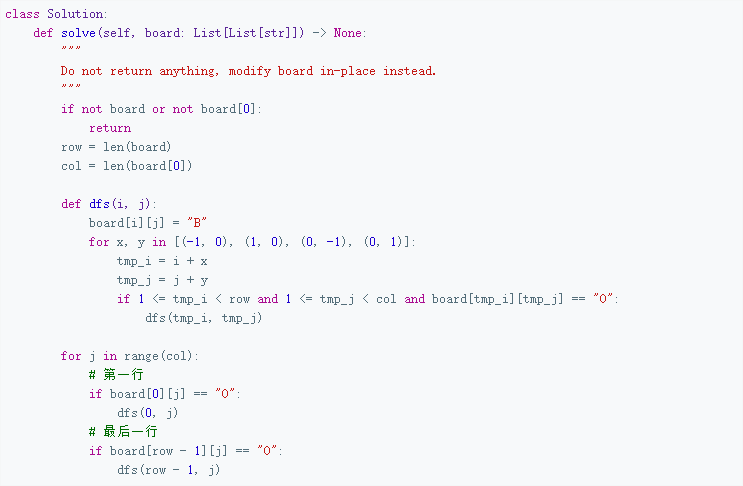
X O O X

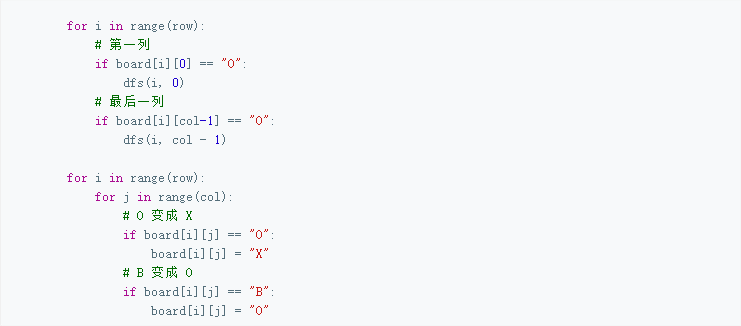
X X O X

X O X X

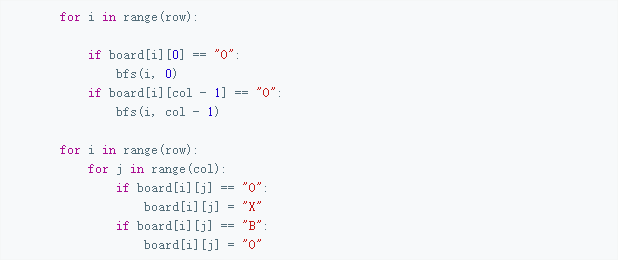
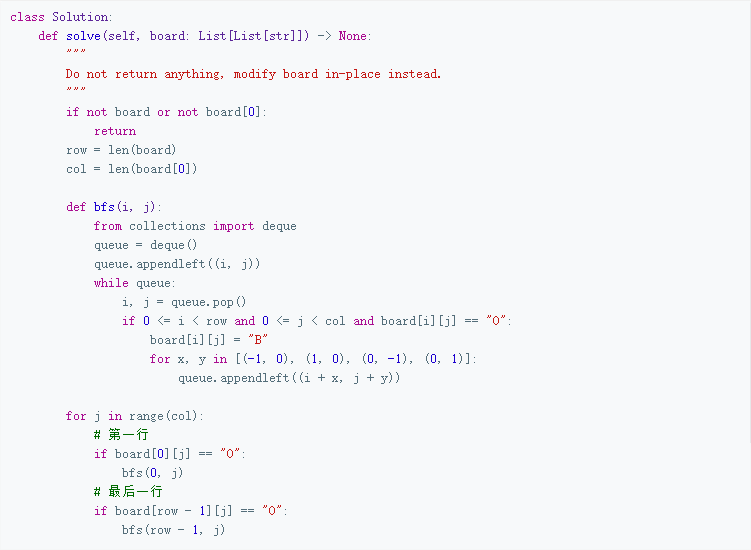


DFS:





BFS:



1. **课程表**

根据每日 气温 列表，请重新生成一个列表，对应位置的输入是你需要再等待多久温度才会升高超过该日的天数。如果之后都不会升高，请在该位置用 0 来代替。

例如，给定一个列表 temperatures = [73, 74, 75, 71, 69, 72, 76, 73]，你的输出应该是 [1, 1, 4, 2, 1, 1, 0, 0]。

提示：气温 列表长度的范围是 [1, 30000]。每个气温的值的均为华氏度，都是在 [30, 100] 范围内的整数。

现在你总共有 n 门课需要选，记为 0 到 n-1。

在选修某些课程之前需要一些先修课程。 例如，想要学习课程 0 ，你需要先完成课程 1 ，我们用一个匹配来表示他们: [0,1]

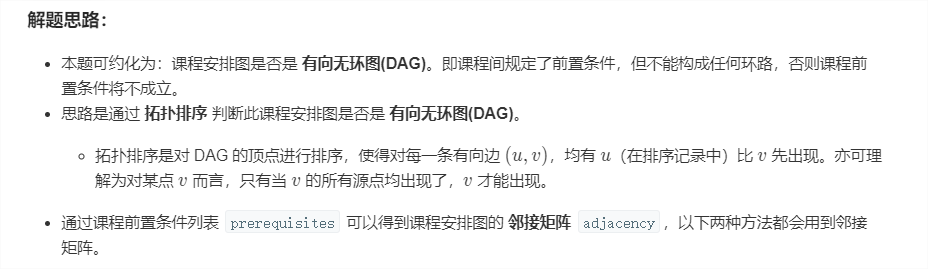
给定课程总量以及它们的先决条件，判断是否可能完成所有课程的学习？

示例 1:

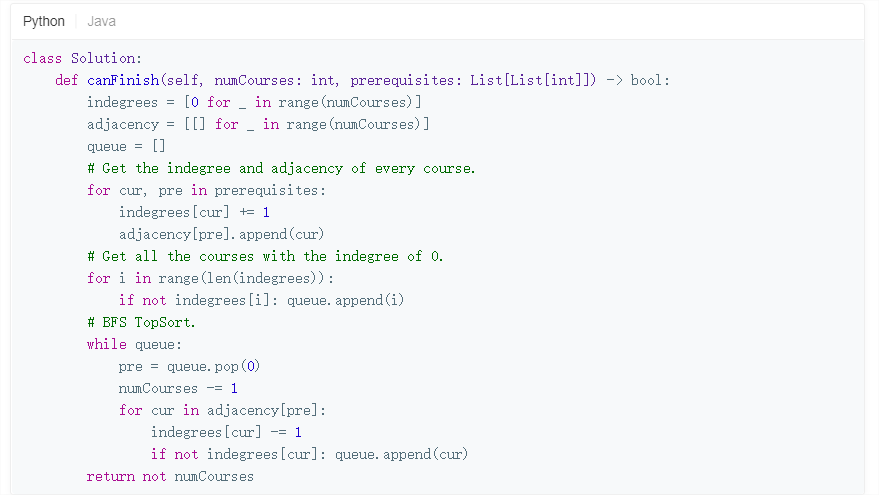
输入: 2, [[1,0]]

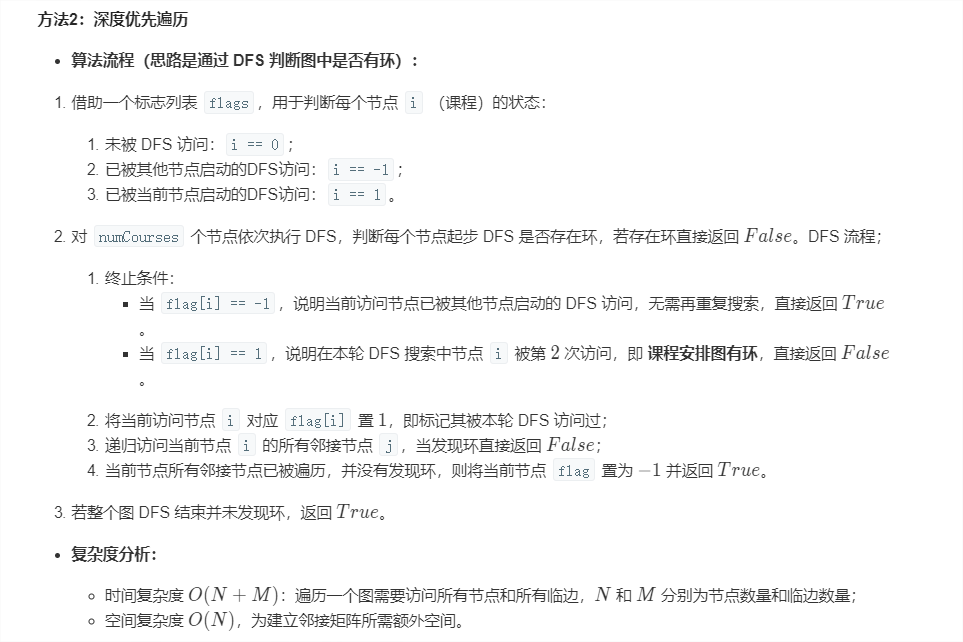
输出: true

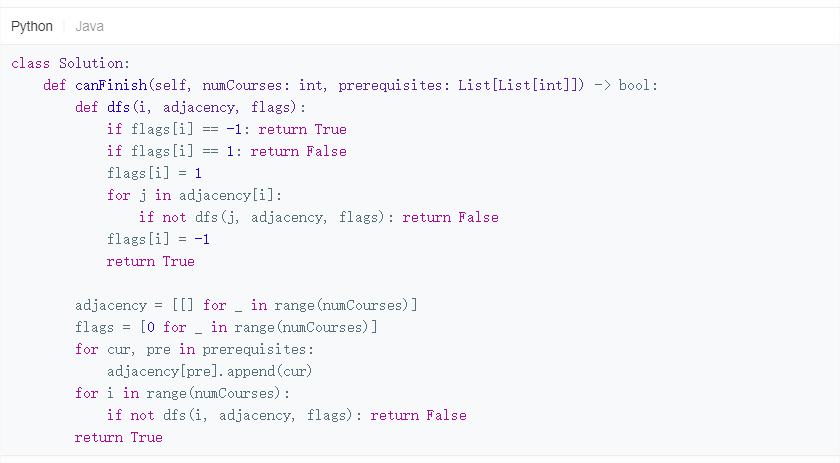
解释: 总共有 2 门课程。学习课程 1 之前，你需要完成课程 0。所以这是可能的。











1. **课程表2**

现在你总共有 n 门课需要选，记为 0 到 n-1。

在选修某些课程之前需要一些先修课程。 例如，想要学习课程 0 ，你需要先完成课程 1 ，我们用一个匹配来表示他们: [0,1]

给定课程总量以及它们的先决条件，返回你为了学完所有课程所安排的学习顺序

可能会有多个正确的顺序，你只要返回一种就可以了。如果不可能完成所有课程，返回一个空数组。

示例 1:

输入: 2, [[1,0]]

输出: [0,1]

解释: 总共有 2 门课程。要学习课程 1，你需要先完成课程 0。因此，正确的课程顺序为 [0,1] 。

和课程表1一样，只是把结果记录下来~在上题的基础上~

flag[i]=-1

res.append(i)

1. **有效的字母异位词**

给定两个字符串 s 和 t ，编写一个函数来判断 t 是否是 s 的字母异位词。

示例 1:

输入: s = "anagram", t = "nagaram"

输出: true



1. **最大数**

给定一组非负整数，重新排列它们的顺序使之组成一个最大的整数。

示例 1:

输入: [10,2]

输出: 210



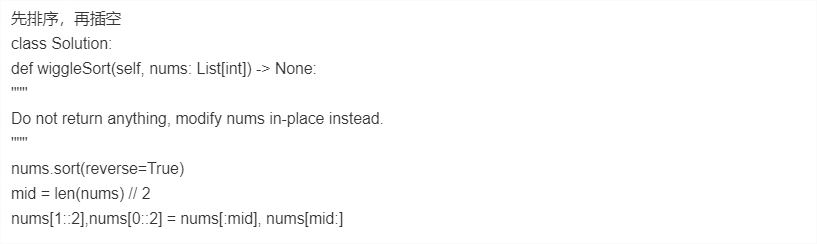
1. **摆动排序**

给定一个无序的数组 nums，将它重新排列成 nums[0] < nums[1] > nums[2] < nums[3]... 的顺序。

示例 1:

输入: nums = [1, 5, 1, 1, 6, 4]

输出: 一个可能的答案是 [1, 4, 1, 5, 1, 6]



1. **合并区间**

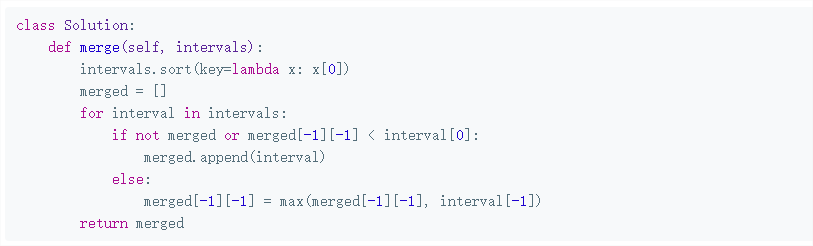
给出一个区间的集合，请合并所有重叠的区间。

示例 1:

输入: [[1,3],[2,6],[8,10],[15,18]]

输出: [[1,6],[8,10],[15,18]]

解释: 区间 [1,3] 和 [2,6] 重叠, 将它们合并为 [1,6].



1. **计算右侧小于当前元素的个数**

给定一个整数数组 nums，按要求返回一个新数组 counts。数组 counts 有该性质： counts[i] 的值是  nums[i] 右侧小于 nums[i] 的元素的数量。

示例:

输入: [5,2,6,1]

输出: [2,1,1,0]

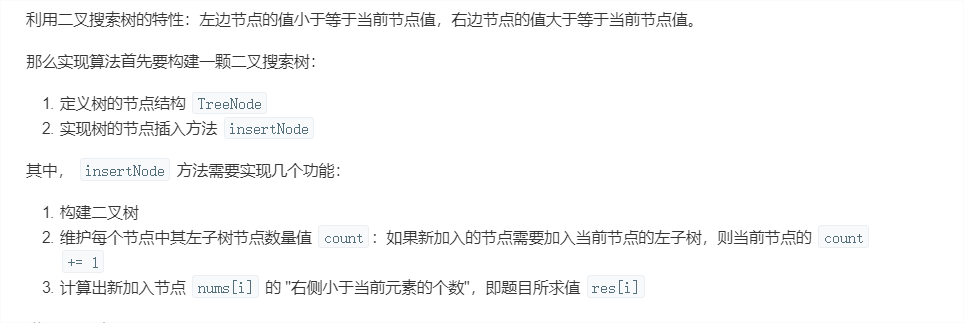
解释:

5 的右侧有 2 个更小的元素 (2 和 1).

2 的右侧仅有 1 个更小的元素 (1).

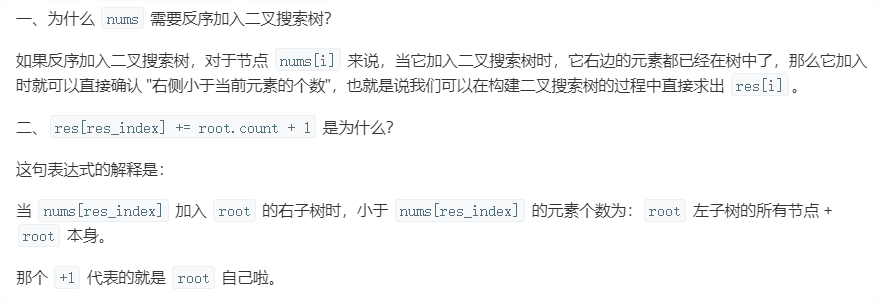
6 的右侧有 1 个更小的元素 (1).

1 的右侧有 0 个更小的元素.

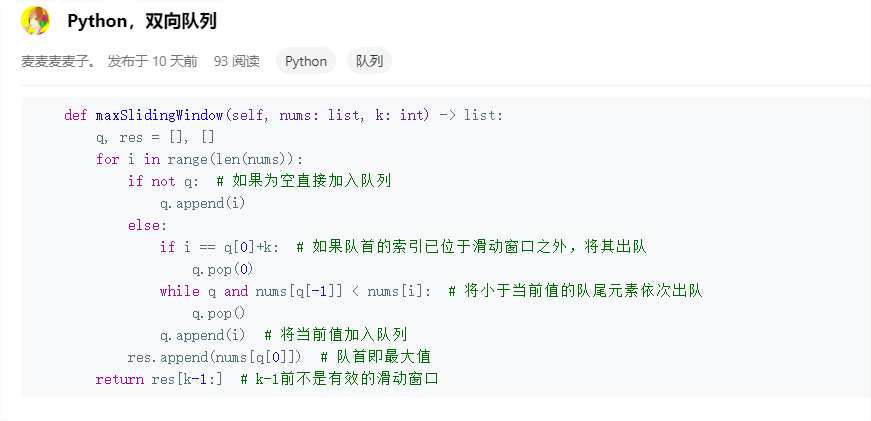








1. **滑动窗口的最大值**



1. **最小覆盖子串**

