

886. 可能的二分法 (Python)



玖月晴

关注

2021.03.06 18:49:10 字数 889 阅读 95

难度：★★★★☆☆

类型：图

方法：深度优先搜索

力扣链接请移步[本题传送门](#)

更多力扣中等题的解决方案请移步[力扣中等题目录](#)

题目

给定一组 N 人（编号为 1, 2, ..., N）， 我们想把每个人分进任意大小的两组。

每个人都可能不喜欢其他人，那么他们不应该属于同一组。

形式上，如果 dislikes[i] = [a, b]，表示不允许将编号为 a 和 b 的人归入同一组。

当可以用这种方法将所有人分进两组时，返回 true；否则返回 false。

示例 1：

输入：N = 4, dislikes = [[1,2],[1,3],[2,4]]

输出：true

解释：group1 [1,4], group2 [2,3]

示例 2：

输入：N = 3, dislikes = [[1,2],[1,3],[2,3]]

输出：false

示例 3：

输入：N = 5, dislikes = [[1,2],[2,3],[3,4],[4,5],[1,5]]

输出：false

提示：

1 <= N <= 2000

0 <= dislikes.length <= 10000

dislikes[i].length == 2

1 <= dislikes[i][j] <= N

dislikes[i][0] < dislikes[i][1]

对于 dislikes[i] == dislikes[j] 不存在 i != j

解答

图的问题常用深度优先搜索解决。

首先我们把不喜欢关系转化成更加便于遍历的邻接表的形式，也就是说，对于每一个结点，都有一个对应的列表，这个列表中包含这个结点不喜欢的结点。

定义深度优先搜索函数dfs，函数的输入是一个结点node，以及颜色c（True或False两种），函数的返回值是布尔值，表达的含义是是否将结点node染色成c成功，函数的功能是对结点node进行染色，并根据不喜欢关系将node结点所有不喜欢的结点染成相反色。

函数开头直接输入跳出条件，即如果该结点node已经被染色成，那么就要判断被染成的颜色是否是c，如果是，说明染色成功，否则染色失败。

如果node结点没有被染色，我们将该结点染色成c，并研究该结点所有不喜欢的结点，并将这些结点染色成相反色，如果都能成功染色成相反色，则返回True，只要有一个染色失败，则返回False。

我们可以用一个字典阿里记录每个结点被染成的颜色，两个颜色实际上代表了两个阵营。

最后，我们要逐个结点的研究，如果当前结点尚未被染色，则使用深度优先搜索的办法将该结点及其联通结点（不喜欢关系建立的连通域）进行染色，否则，不考虑当前已经被染色的结点。因为我们是一个连通域一个连通域来考虑的，注意这里的联通域是用不喜欢关系建立起来的。只有所有连通域都校验通过，才能说能够成功分成两个阵营。

```
from collections import defaultdict

class Solution(object):
    def possibleBipartition(self, N, dislikes):
        graph = defaultdict(list)
        for u, v in dislikes:
            graph[u].append(v)
            graph[v].append(u)

        color = {}

        def dfs(node, our_flag=0):
            if node in color:
                # 已经被染过色
                return color[node] == our_flag
            color[node] = our_flag
            return all(dfs(dislike, not our_flag) for dislike in graph[node])

        for node in range(1, N+1):
            if node not in color and not dfs(node):
                return False
        return True
```

如有疑问或建议，欢迎评论区留言~

有关更多力扣中等题的python解决方案，请移步[力扣中等题解析](#)