

宽度优先遍历及其扩展

前置知识:

会使用队列、双端队列、优先级队列（堆）

讲解036-二叉树上的宽度优先遍历

讲解038-经典递归过程解析，之前所有递归的内容需要好好理解，后面的课用到递归越来越多

讲解059~讲解065都是【必备】课程有关图的内容，建议从头开始学习

本节课讲述:

单源、多源宽度优先遍历基本过程

01bfs，宽度优先遍历与双端队列结合

宽度优先遍历与优先级队列结合

宽度优先遍历与深度优先遍历结合，去生成路径

宽度优先遍历及其扩展

宽度优先遍历基本内容

*bfs*的特点是逐层扩散，从源头点到目标点扩散了几层，最短路就是多少

*bfs*可以使用的特征是 任意两个节点之间的相互距离相同（无向图）

*bfs*开始时，可以是 单个源头、也可以是 多个源头

*bfs*频繁使用队列，形式可以是 单点弹出 或者 整层弹出

*bfs*进行时，进入队列的节点需要标记状态，防止 同一个节点重复进出队列

*bfs*进行时，可能会包含 剪枝策略 的设计

*bfs*是一个理解难度很低的算法，难点在于 节点如何找到路、路的展开 和 剪枝设计

宽度优先遍历及其扩展

01bfs, 适用于 图中所有边的权重只有0和1两种值, 求源点到目标点的最短距离
时间复杂度为 $O(\text{节点数量} + \text{边的数量})$, 为什么不能用传统bfs?

1, $distance[i]$ 表示从源点到*i*点的最短距离, 初始时所有点的 $distance$ 设置为无穷大

2, 源点进入双端队列, $distance[\text{源点}] = 0$

3, 双端队列 头部弹出 x ,

A, 如果 x 是目标点, 返回 $distance[x]$ 表示源点到目标点的最短距离

B, 考察从 x 出发的每一条边, 假设某边去 y 点, 边权为 w

1) 如果 $distance[y] > distance[x] + w$, 处理该边; 否则忽略该边

2) 处理时, 更新 $distance[y] = distance[x] + w$

如果 $w == 0$, y 从头部进入双端队列; 如果 $w == 1$, y 从尾部进入双端队列

3) 考察完 x 出发的所有边之后, 重复步骤3

4, 双端队列为空停止

正确性证明 以及 为什么不需要visited来标记节点

宽度优先遍历及其扩展

宽度优先遍历与优先级队列结合，更进一步的内容会在讲*Dijkstra*算法时说明

宽度优先遍历与深度优先遍历结合，去生成路径

1, *bfs*建图

2, *dfs*利用图生成路径

宽度优先遍历及其扩展

题目1

地图分析

你现在手里有一份大小为 $n \times n$ 的 网格 *grid*

上面的每个 单元格 都用 *0* 和 *1* 标记好了其中 *0* 代表海洋，*1* 代表陆地。

请你找出一个海洋单元格，这个海洋单元格到离它最近的陆地单元格的距离是最大的并返回该距离。如果网格上只有陆地或者海洋，请返回 *-1*。

我们这里说的距离是「曼哈顿距离」（*Manhattan Distance*）：

$(x0, y0)$ 和 $(x1, y1)$ 这两个单元格之间的距离是 $|x0 - x1| + |y0 - y1|$ 。

测试链接：<https://leetcode.cn/problems/as-far-from-land-as-possible/>

宽度优先遍历及其扩展

题目2

贴纸拼词

我们有 n 种不同的贴纸。每个贴纸上都有一个小写的英文单词。

您想要拼写出给定的字符串 *target*，方法是从收集的贴纸中切割单个字母并重新排列它们

如果你愿意，你可以多次使用每个贴纸，每个贴纸的数量是无限的。

返回你需要拼出 *target* 的最小贴纸数量。如果任务不可能，则返回 *-1*

注意：在所有的测试用例中，所有的单词都是从 *1000* 个最常见的美国英语单词中随机选择的并且 *target* 被选择为两个随机单词的连接。

测试链接：<https://leetcode.cn/problems/stickers-to-spell-word/>

宽度优先遍历及其扩展

题目3

到达角落需要移除障碍物的最小数目

给你一个下标从 0 开始的二维整数数组 *grid*，数组大小为 $m \times n$

每个单元格都是两个值之一：

0 表示一个 空 单元格，

1 表示一个可以移除的 障碍物

你可以向上、下、左、右移动，从一个空单元格移动到另一个空单元格。

现在你需要从左上角 $(0, 0)$ 移动到右下角 $(m - 1, n - 1)$

返回需要移除的障碍物的最小数目

测试链接：<https://leetcode.cn/problems/minimum-obstacle-removal-to-reach-corner/>

宽度优先遍历及其扩展

题目4

使网格图至少有一条有效路径的最小代价

给你一个 $m * n$ 的网格图 *grid* 。 *grid* 中每个格子都有一个数字

对应着从该格子出发下一步走的方向。 *grid[i][j]* 中的数字可能为以下几种情况：

1：往右 **2**：往左 **3**：往下 **4**：往上

注意网格图中可能会有无效数字，因为它们可能指向*grid*以外的区域

从最左上角的格子 $(0,0)$ 出发，有效路径为每一步都顺着数字对应方向走

最终在最右下角的格子 $(m - 1, n - 1)$ 结束的路径

有效路径 不需要是最短路径

可以花费**1**的代价修改一个格子中的数字，但每个格子中的数字只能修改一次

返回让网格图至少有一条有效路径的最小代价

测试链接：

leetcode.cn/problems/minimum-cost-to-make-at-least-one-valid-path-in-a-grid

宽度优先遍历及其扩展

题目5

二维接雨水

给你一个 $m * n$ 的矩阵，其中的值均为非负整数，代表二维高度图每个单元的高度
请计算图中形状最多能接多少体积的雨水。

测试链接：<https://leetcode.cn/problems/trapping-rain-water-ii/>

前置题目：

讲解050 - 双指针技巧 - 题目3 - 一维接雨水问题

强烈建议看过这个题再听这道题的解析

宽度优先遍历及其扩展

题目6

单词接龙 II

按字典 *wordList* 完成从单词 *beginWord* 到单词 *endWord* 转化

一个表示此过程的 转换序列 是形式上像

beginWord $\rightarrow s_1 \rightarrow s_2 \rightarrow \dots \rightarrow s_k$ 这样的单词序列，并满足：

每对相邻的单词之间仅有单个字母不同

转换过程中的每个单词 s_i ($1 \leq i \leq k$) 必须是字典 *wordList* 中的单词

注意，*beginWord* 不必是字典 *wordList* 中的单词

$s_k == endWord$

给你两个单词 *beginWord* 和 *endWord*，以及一个字典 *wordList*

请你找出并返回所有从 *beginWord* 到 *endWord* 的 最短转换序列

如果不存在这样的转换序列，返回一个空列表

每个序列都应该以单词列表 $[beginWord, s_1, s_2, \dots, s_k]$ 的形式返回

测试链接：<https://leetcode.cn/problems/word-ladder-ii/>