Bidirectional Breadth First Search

双向广度优先搜索

问题：

在的二维方格图s中从beg点移动到end点。

解法：

双向广度优先搜索是在广度优先搜索基础上的一个变种，搜索速度更快，内存占用量更大。该算法从beg和end两个点开始，同时进行广度优先搜索，两边的点在某一处相遇，即可得到一条从beg到end的路径。

每个广度优先搜索会维护一个已经搜索过的二维方格visit，为0表示该点未被访问过，为1表示该点已经被访问过。将beg和end的两个广度优先搜索的visit表分别称为beg\_visit和end\_visit。当beg的队列进行扩张时，不仅检查beg\_visit表，也检查end\_visit表，对于某个被扩张的点x，若其已经在end\_visit中被访问过，则说明end的队列已经到达这里了，则beg与end在此处相遇，算法结束。

在下面这个，的二维方格s中，从移动到的过程如下：



1. 从beg和end开始，将beg和end加入各自的等待队列，beg染红end染绿；



1. 从beg等待队列中取出并检查不是绿色，将它周围的、、加入beg等待队列并染红；从end等待队列中取出并检查不是红色，将它周围的、加入end等待队列并染绿；



1. 从beg等待队列中取出并检查不是绿色，将它周围的加入beg等待队列并染红；从end等待队列中取出并检查不是红色，将它周围的、加入end等待队列并染绿；



1. 从beg等待队列中取出并检查不是绿色，将它周围的、加入beg等待队列并染红；从end等待队列中取出并检查不是红色，将它周围的加入end等待队列并染绿；



1. 从beg等待队列中取出并检查不是绿色，将它周围的加入beg等待队列并染红；从end等待队列中取出并检查不是红色，将它周围的、加入end等待队列并染绿；



1. 从beg等待队列中取出并检查不是绿色，将它周围的加入beg等待队列并染红；从end等待队列中取出并检查不是红色，将它周围的加入end等待队列并染绿；



1. 从beg等待队列中取出并检查不是绿色，将它周围的、加入beg等待队列并染红，但是点已经是绿色的了，因此从beg和end出发的两个广度优先搜索相遇，算法结束；

对于的二维方格s，广度优先搜索从beg点遍历到end点的过程一般是从beg向四周发散开，一直到达end点：



而双向广度优先搜索则是从beg和end两个点分别发散开，在中间相遇：



假设beg和end点距离为k，节点的邻居数量为b，在本问题中，每个节点有上下左右4个邻居，即。则广度优先搜索会考虑的节点数量为，而双向广度优先搜索会考虑的节点数量是，最坏情况下时间复杂度与广度优先搜索一样也为。