在一个 N \* M 的网格中，每个格子的坐标为(x, y),从一个入口 entry 开始找到所有能走出迷宫的 exit出口

创建一个二维数组 int[][] grid; 作为迷宫数据

grid[x][y] == 0 或者 出界的节点 视为路(可走)

grid[x][y] == 1 视为障碍物(不可走)

思路：利用回溯算法思想和剪枝函数来完成路径搜索

1.初始化路径：路径中节点个数为0，从入口格子节点开始，加入到路径中，并记录入口节点已经加入过路径中了。

2.获取路径中最后一个加入的节点X(该节点继续保存在路径中)，

如果节点X是出口则成功搜索到一条路径了。将路径保存。

如果只要求找到一条路径就算成功，则退出。

如果要求找到所有能走出的路径，则继续执行下面逻辑

遍历节点X的上A、下B、左C、右D四个节点，不可走的节点或者之前加入过路径中的节点跳过（此为剪枝函数）

如果节点可走，则将节点加入到路径中，并记录该节点已经加入过路径，GoTo 到 2 继续深度优先遍历

如果节点X的上、下、左、右 四个节点都没有可走的节点或者之前都添加到路径中，则从节点X经过无法到达出口，则应该回溯到节点X的上一个节点继续查找（此为回溯），如何回溯？回溯方法：将节点X从路径中移除，路径中最后一个就是节点X的上一个节点了。 GoTo到 2 继续深度优先遍历

3.到此从入口开始的所有能走通的路径都已经获取到了