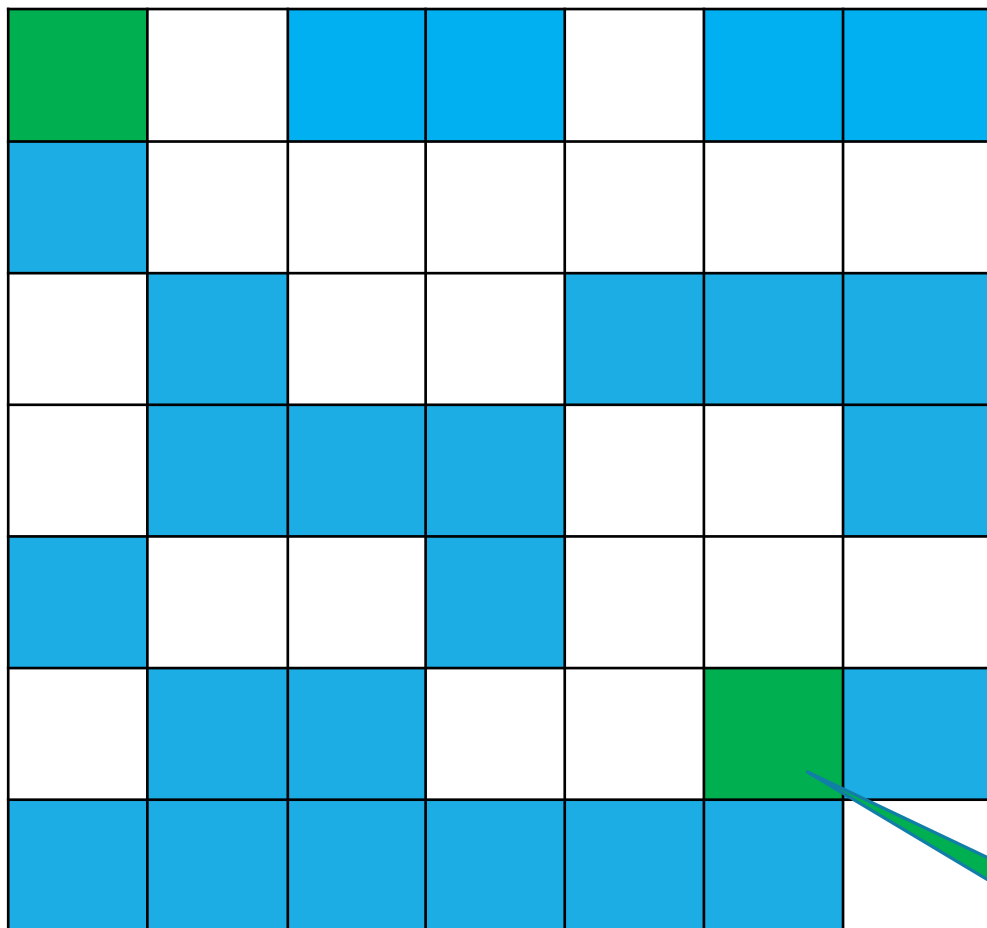


栈的应用： 迷宫问题

如何找到一条从入口到出口的路径？

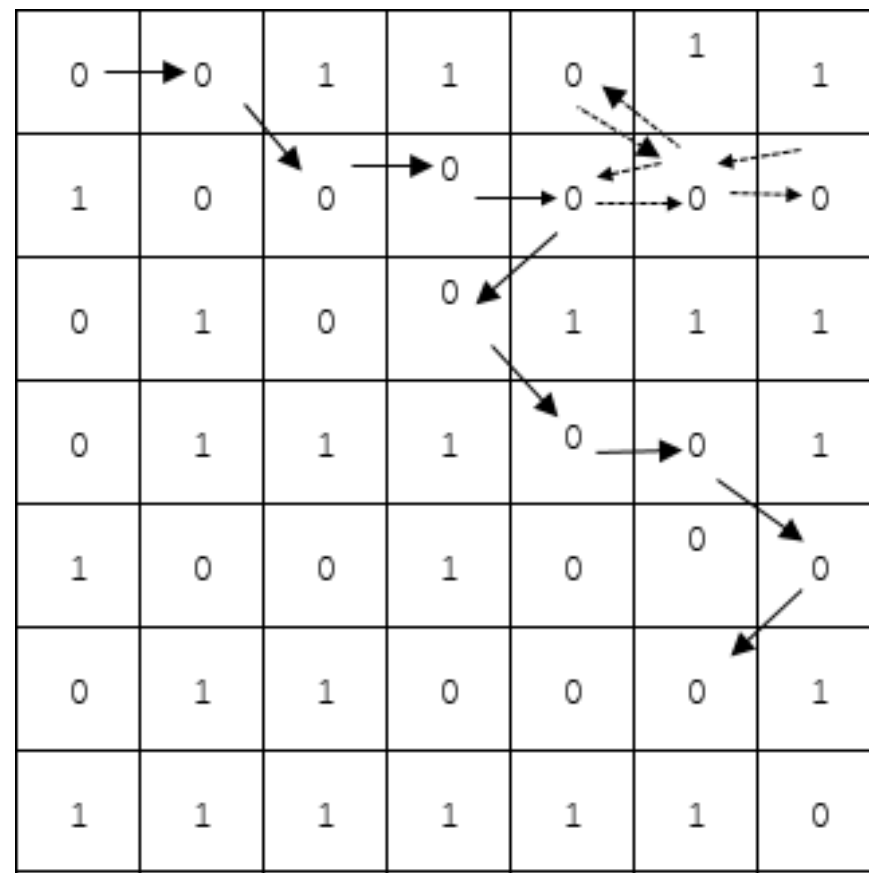
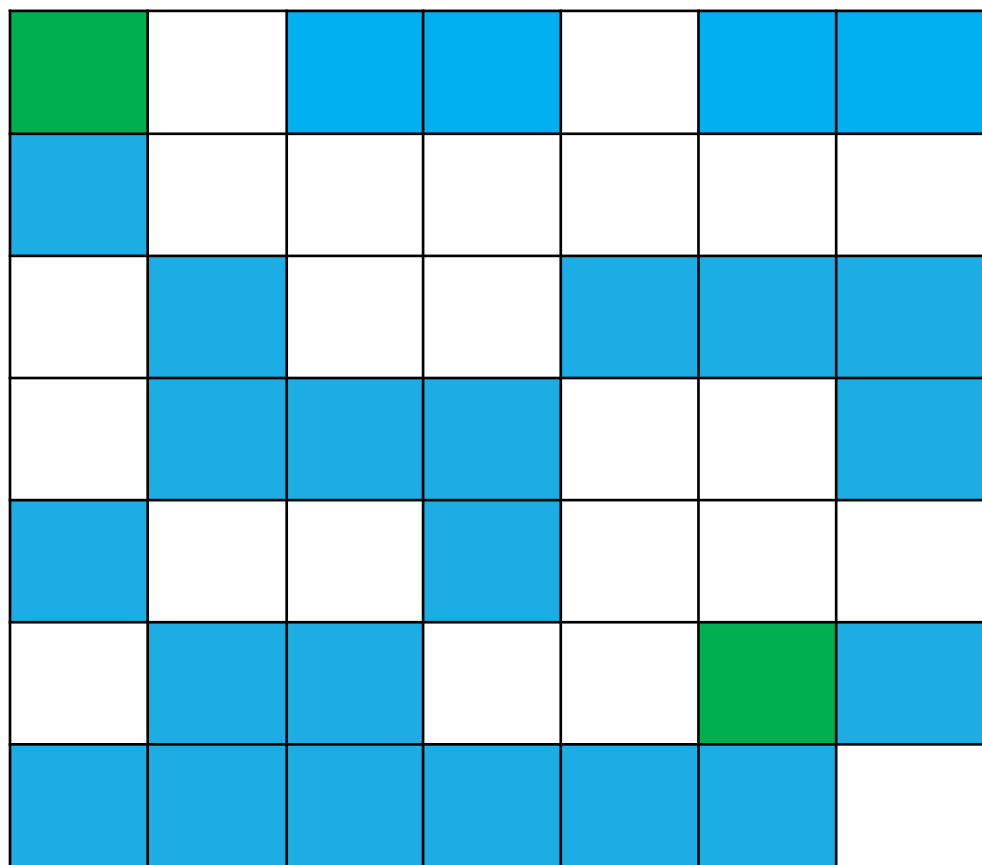
入口
(1,1)



出口
(6,6)



栈的应用： 迷宫问题



(1,1) (1,2) (2,3) (2,4) (2,5) (3,4) (4,5) (4,6) (5,7) (6,6)

栈的应用： 迷宫问题

迷宫问题中要找到路径并输出需要解决三个问题：

(1) 从某一个坐标点(x,y)出发如何搜索其相邻位置(g,h)?

不妨假设8个方向的顺序是从**正东按照顺时针**

$\text{direction}[8][2] = \{ \{ 0, 1 \}, \{ 1, 1 \}, \{ 1, 0 \}, \{ 1, -1 \}, \{ 0, -1 \}, \{ -1, -1 \}, \{ -1, 0 \}, \{ -1, 1 \} \}$

相邻位置(g,h)相对于当前位置(x,y)的增量。即：

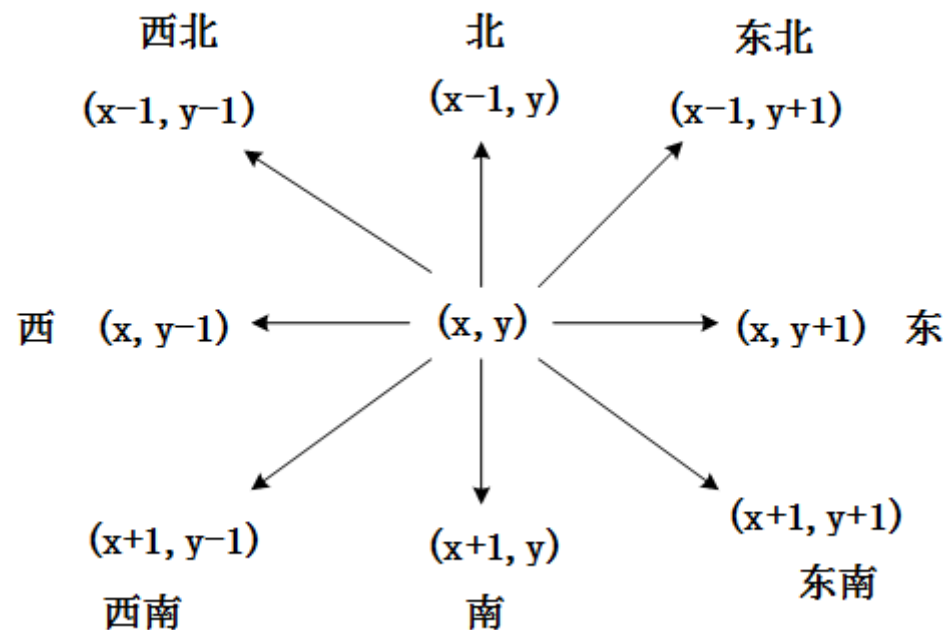
$g = x + \text{direction}[i][j]$

$h = y + \text{direction}[i][j]$

假设从当前位置(3,5)向南出发，则：

$g = x + \text{direction}[2][0] = 3 + 1 = 4$

$h = y + \text{direction}[2][1] = 5 + 0 = 5$



栈的应用： 迷宫问题

迷宫问题中要找到路径并输出需要解决三个问题：

(2) 如何记录探索过的路径

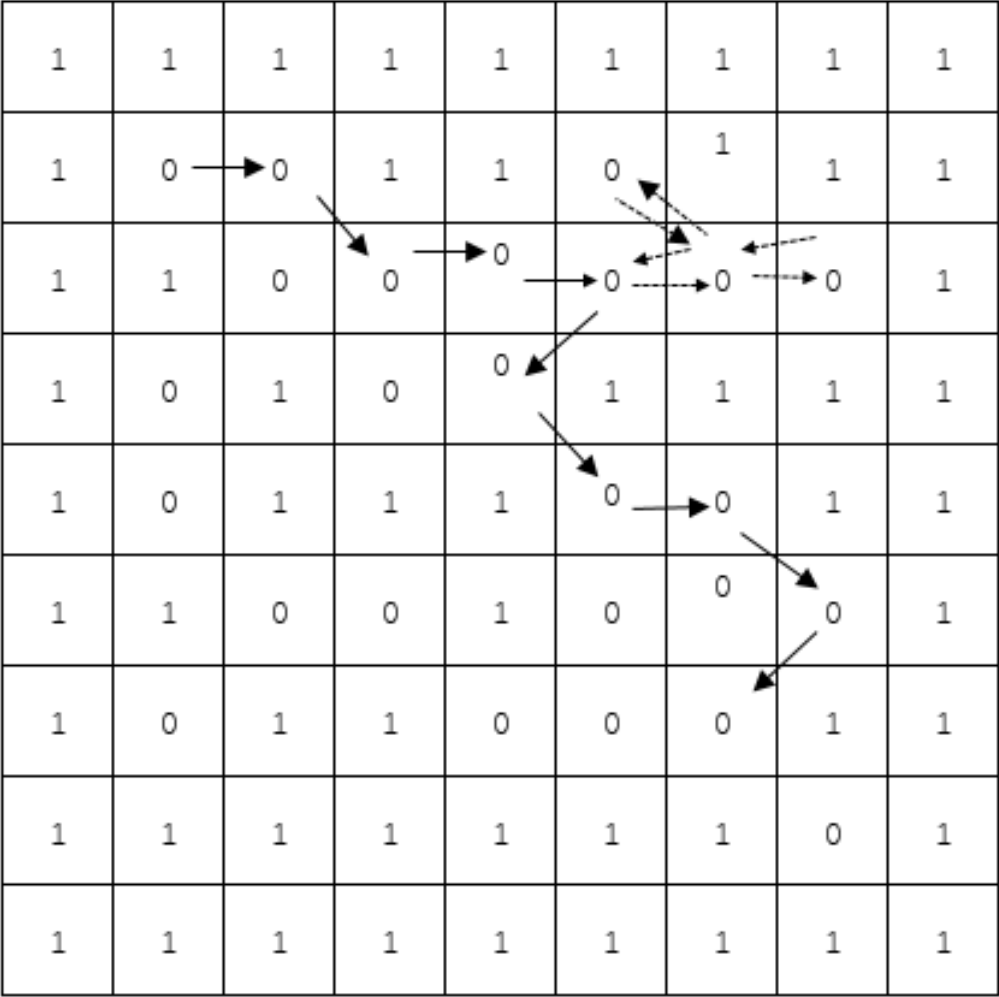
本节采用了回溯方法，因此设计栈来存放探索过的路径，当不能向前继续探索时，从栈中弹出元素。为了重复使用前面定义好的栈结构，在这里使用两个栈linkStackX和 linkStackY，分别存放行坐标和列坐标。

(3) 如何防止重复探索某位置

通过设置标志为来识别，初始时各个位置的标识位 $\text{mark}[i][j] = 0$ ，当探索到某位置后设置 $\text{mark}[i][j] = 1$ 。

- 创建两个空栈StackX和StackY
- 将入口entryX和entryY分别压入栈StackX和StackY中
- while (栈不空)
 - 取栈顶元素，出栈。当前位置为栈顶元素。
 - while(mov < 8) 即还存在探索的方向
 - 按照顺时针依次探索各个位置 (posX, posY)
 - 如果(posX, posY)是出口，则输出路径，返回1。
 - 如果(posX, posY)是没有走过的通路，
 - 设置标志mark[posX][posY] = 1
 - 当前位置进栈
 - 将(posX, posY)设置为当前位置
 - 设置mov = 0
 - 否则 (如果(X, Y)是没有走过的通路) , mov++

栈的应用： 迷宫问题



具体搜索路径如左图所示，其中
虚线表示探索过的位置，实线表示路径

迷宫路径：
(1,1) (1,2) (2,3) (2,4) (2,5)
(3,4) (4,5) (4,6) (5,7) (6,6)