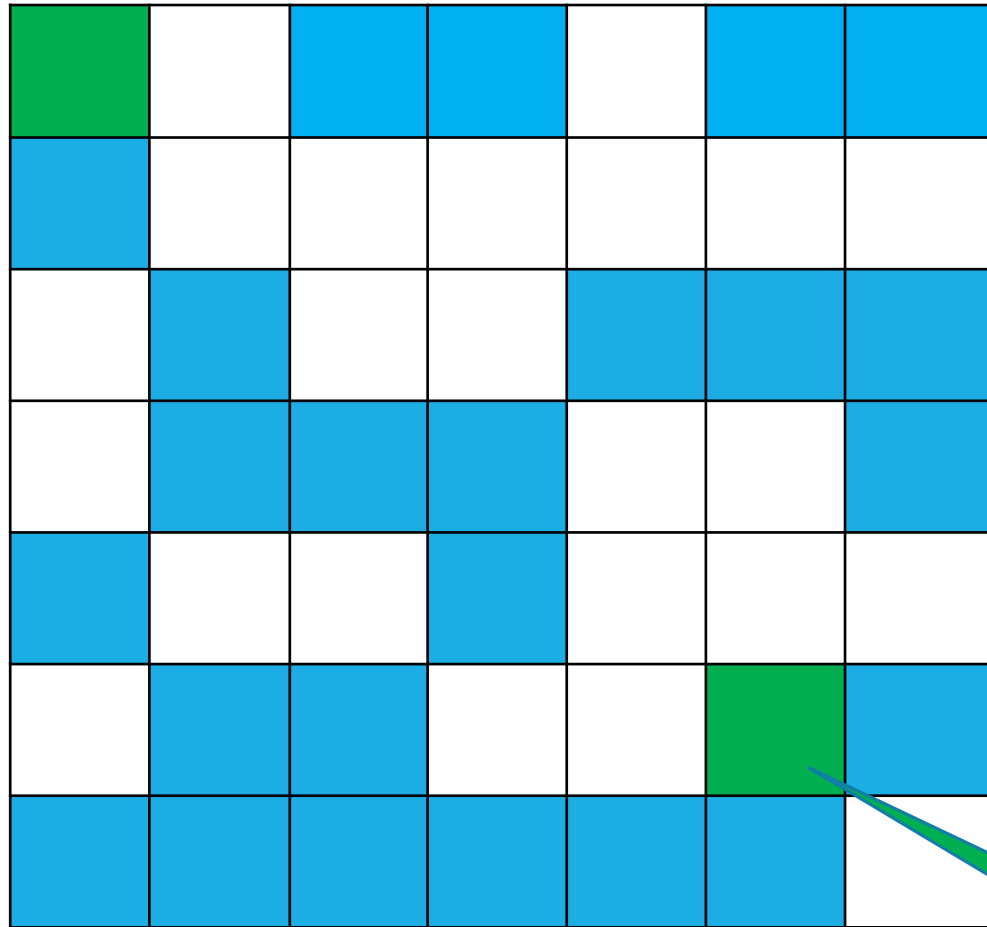


队列应用： 迷宫问题 如何找到一条从入口到出口的最短路径？

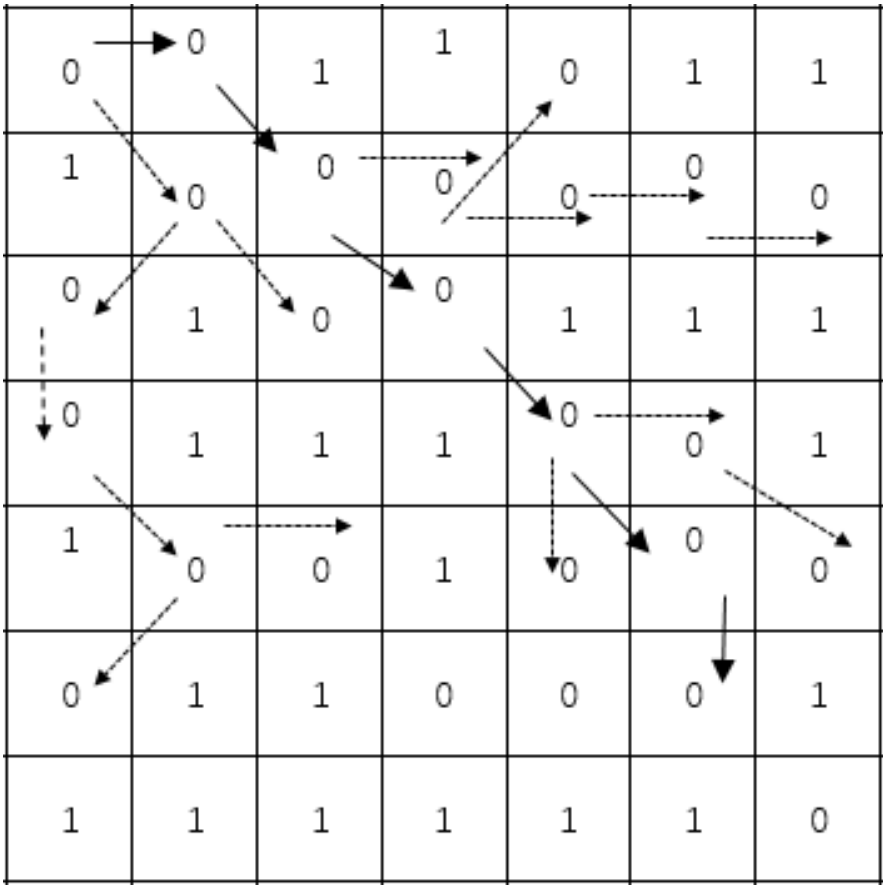
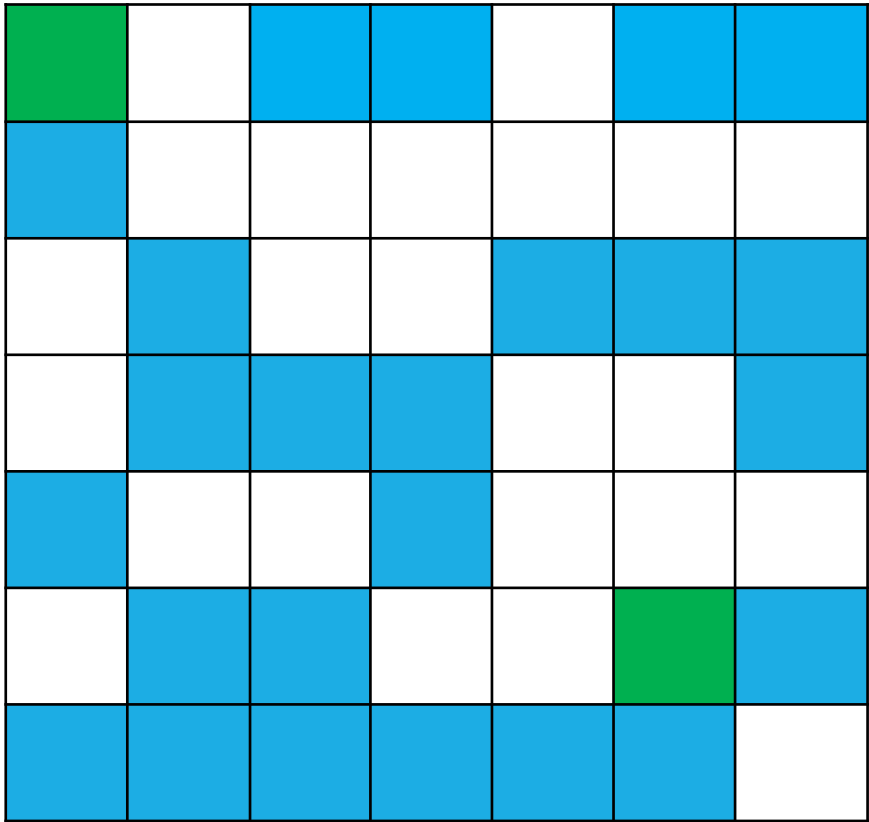
入口
(1,1)



出口
(6,6)



队列的应用：迷宫



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

3.11 队列的应用：迷宫

假设考虑3.7节的迷宫图有8个方向

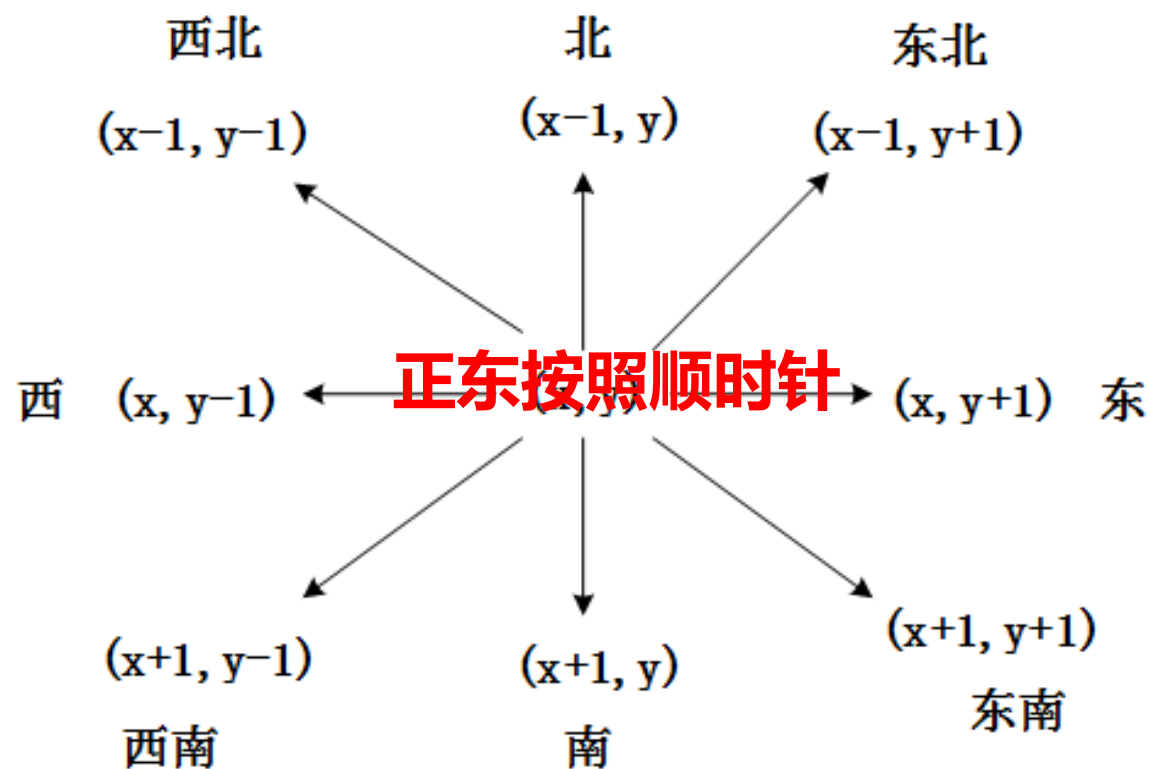
虚线表示探索过的位置，实线表示路径

A 7x7 grid of 0s and 1s. The grid is as follows:

0	0	1	1	0	1	1
1	0	0	0	0	0	0
0	1	0	0	1	1	1
0	1	1	1	0	0	1
1	0	0	1	0	0	0
0	1	1	0	0	0	1
1	1	1	1	1	1	0

Arrows indicate a path from the top-left cell (0,0) to the bottom-right cell (6,6). The path is: (0,0) → (0,1) → (1,1) → (2,1) → (2,2) → (3,2) → (3,3) → (4,3) → (4,4) → (5,4) → (5,5) → (6,5) → (6,6).

```
direction[8][2] = { { 0, 1 }, { 1, 1 }, { 1, 0 },  
{ 1, -1 }, { 0, -1 }, { -1, -1 }, { -1, 0 }, { -1, 1 } }
```



队列的应用：迷宫问题

具体实现见算法3-29

迷宫的表示和搜索8个方向的顺序同前面的定义，

迷宫算法思路：

- 创建两个空队列linkQueueX和linkQueueY
- 将入口entryX和entryY分别压入队linkQueueX和linkQueueY中
- while (队列不空)
 - 取队头元素，出队
 - for(mov=0;mov < 8;mov++), 即还存在可以探索的相邻方向
 - 按照顺时针依次探索各个位置 (X, Y)
 - 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
 - 当前位置入队
 - 记录前驱位置，方便输出路径