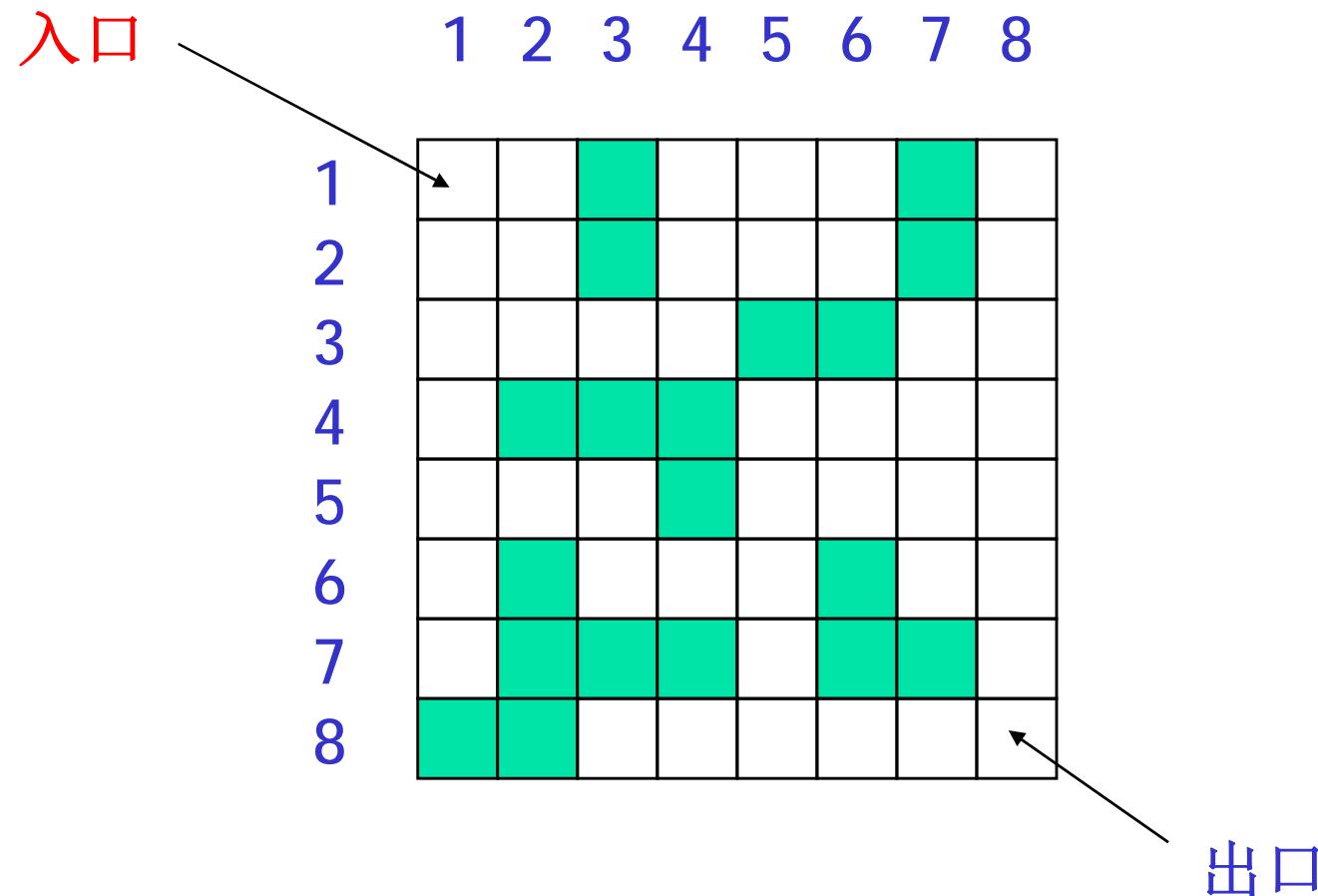


# 迷宫问题

作者：屠添翼

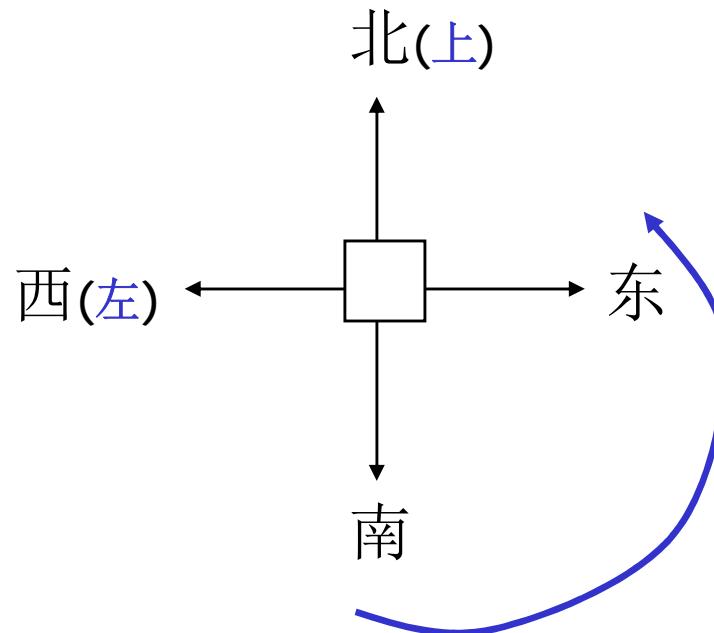
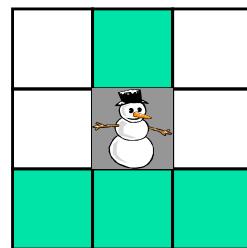
# 迷宫问题

- 寻找一条从入口到出口的通路



## 迷宫问题(续)

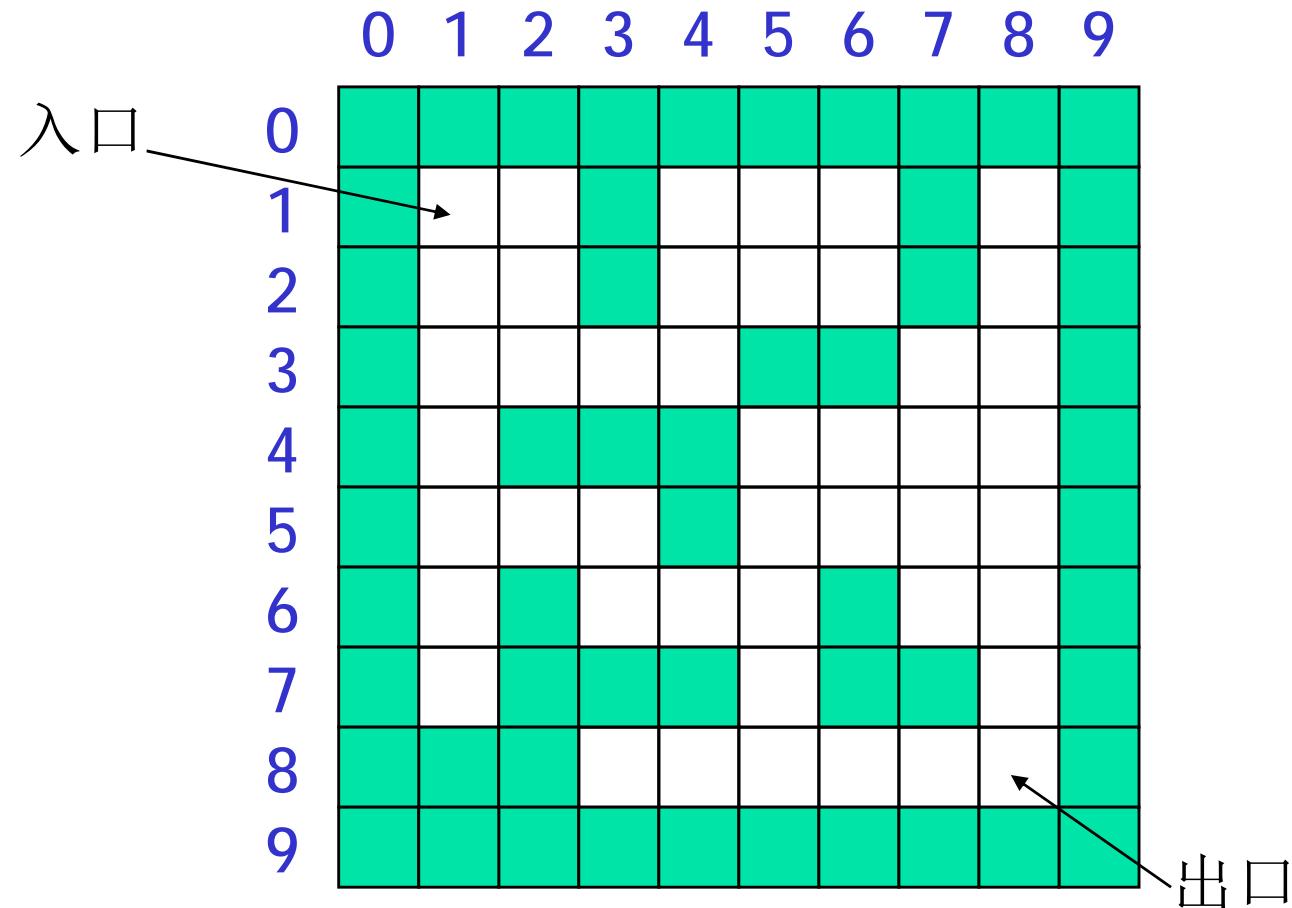
- 前进方向：上(北)、下(南)、左(西)、右(东)



- 首先从下方开始，按照逆时针方向搜索下一步可能前进的位置

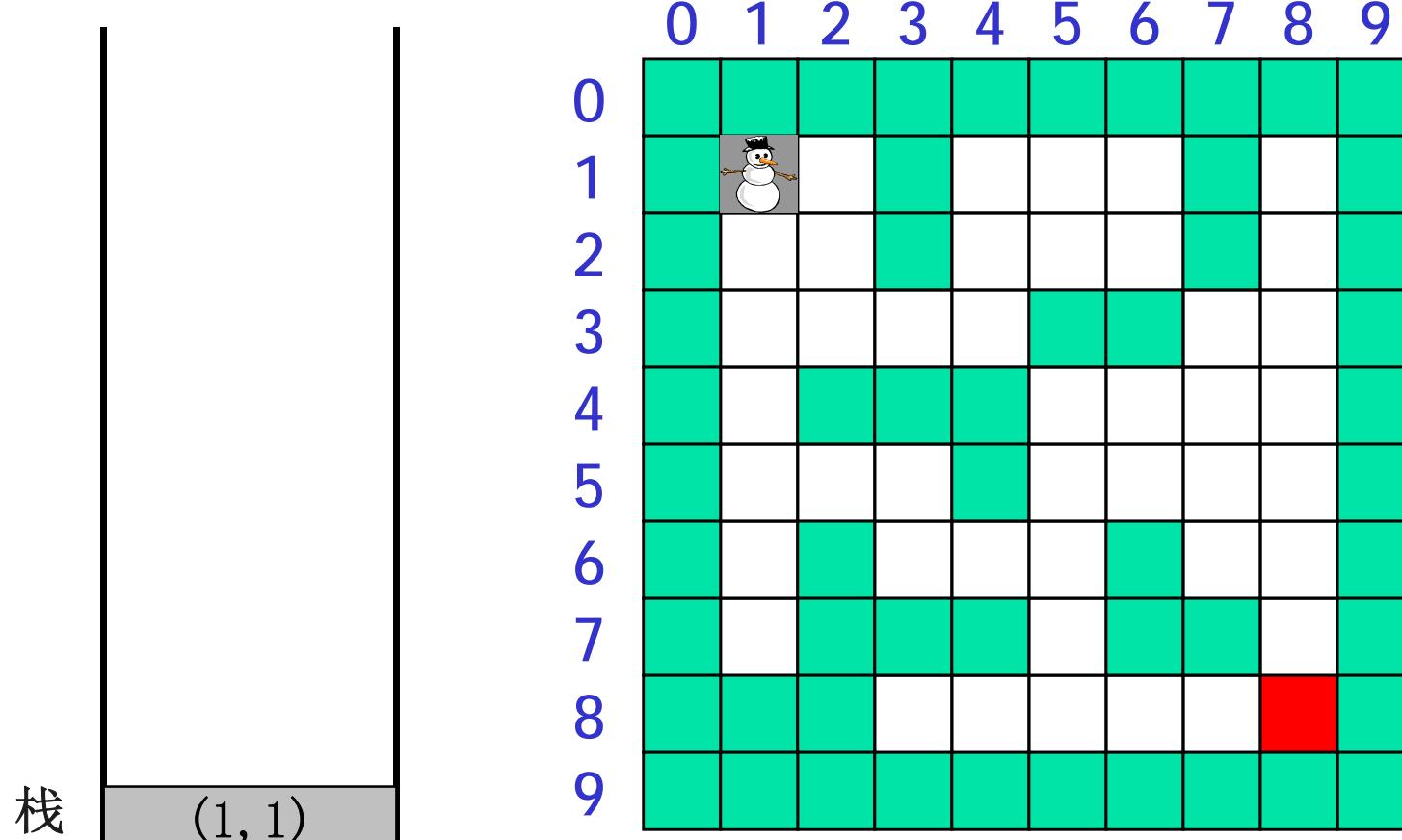
## 迷宫问题(续)

- 在迷宫周围加墙，避免判断是否出界



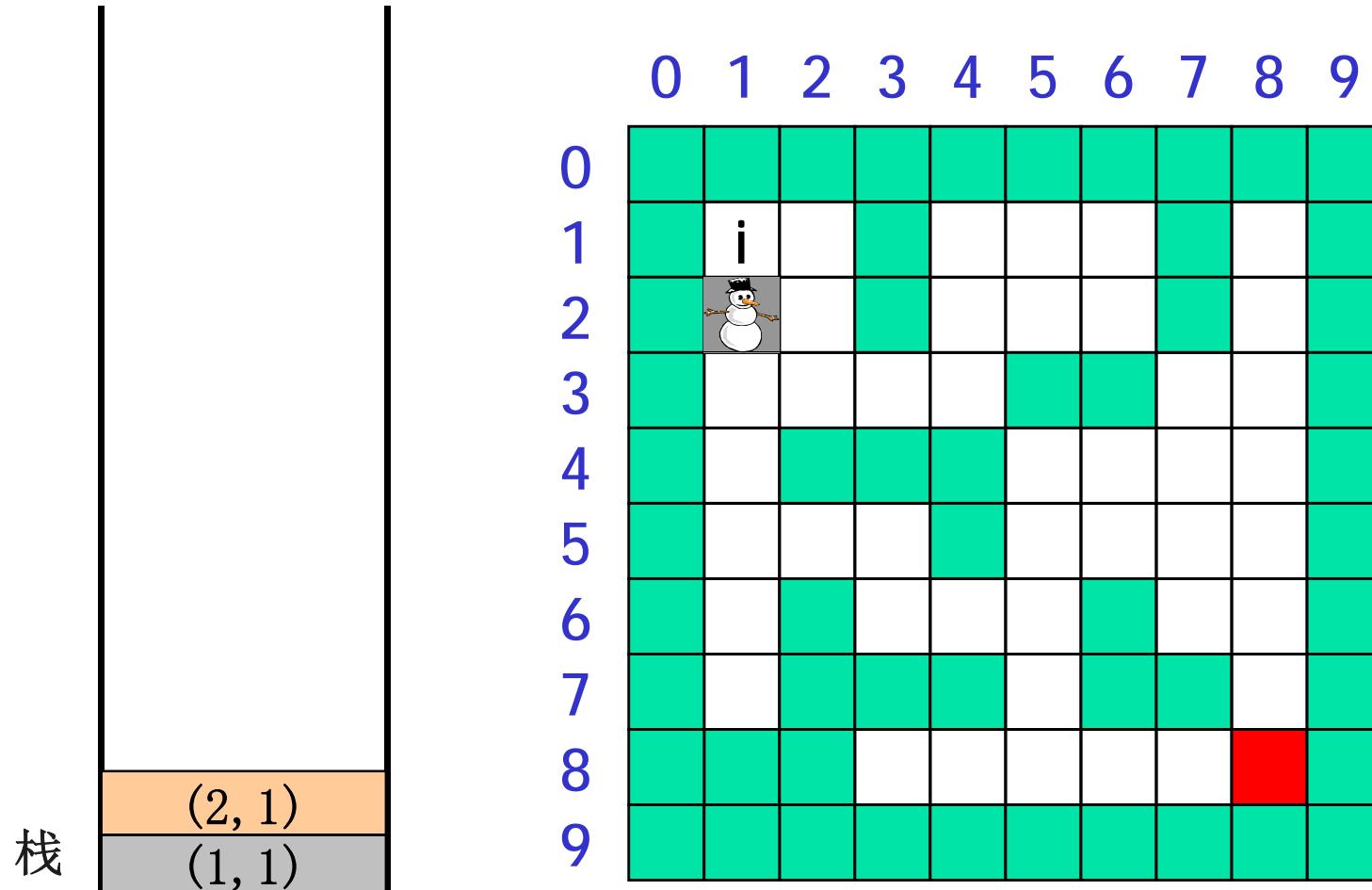
## 迷宫问题(续)

- 在寻找出口的过程中，每前进一步，当前位置入栈；每回退一步，栈顶元素出栈



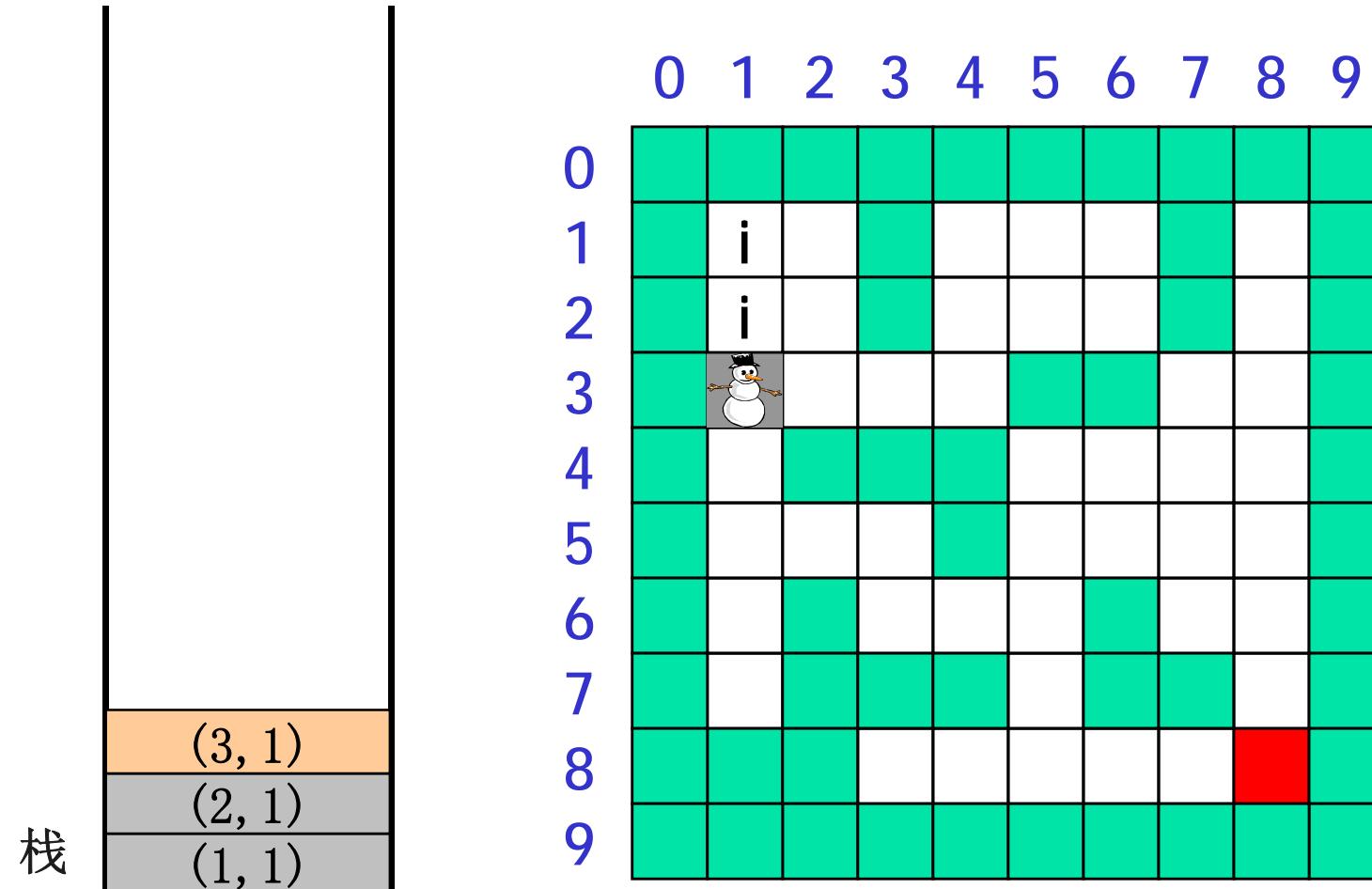
# 迷宫问题(续)

- 向下方前进一步



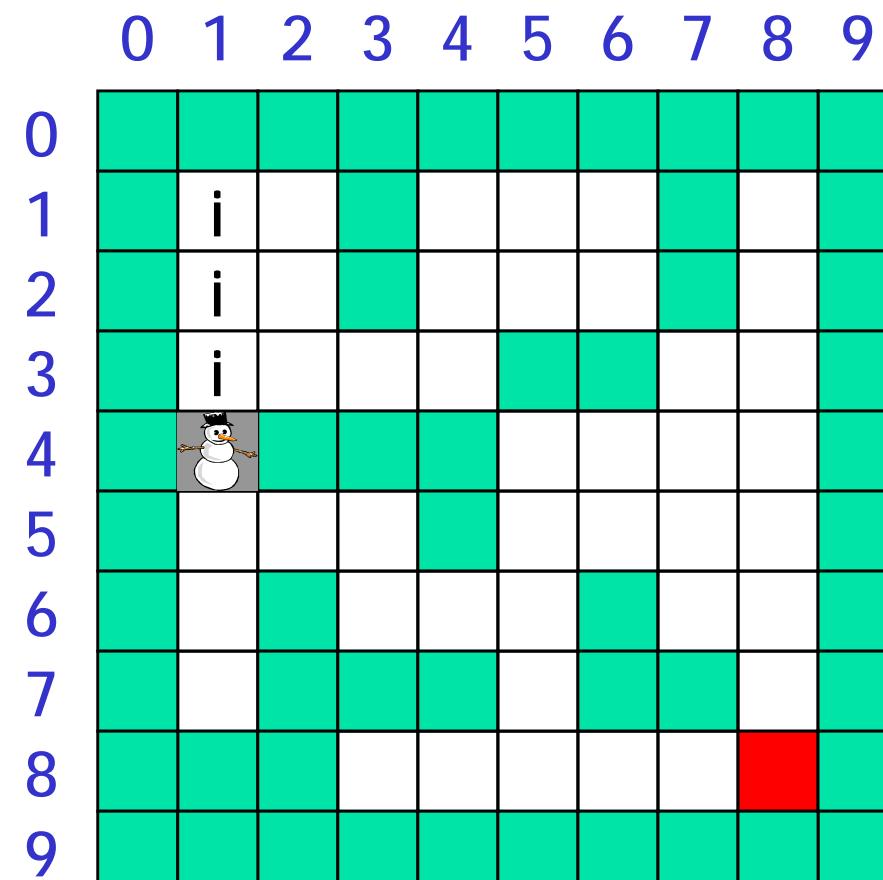
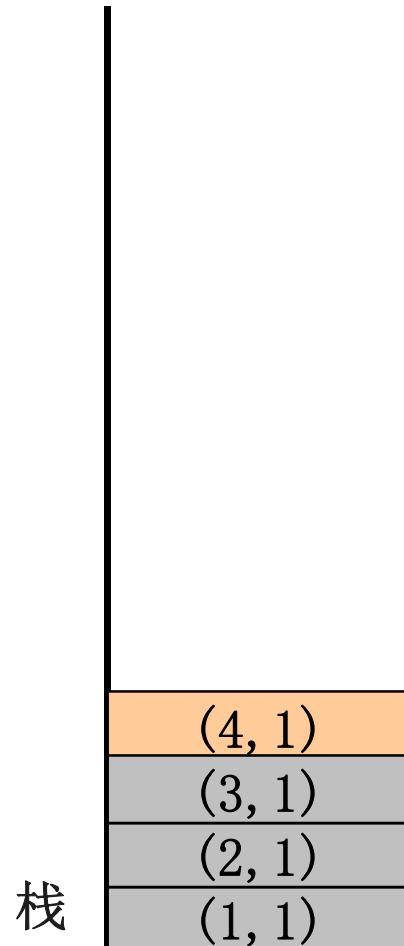
# 迷宫问题(续)

- 向下方前进一步



# 迷宫问题(续)

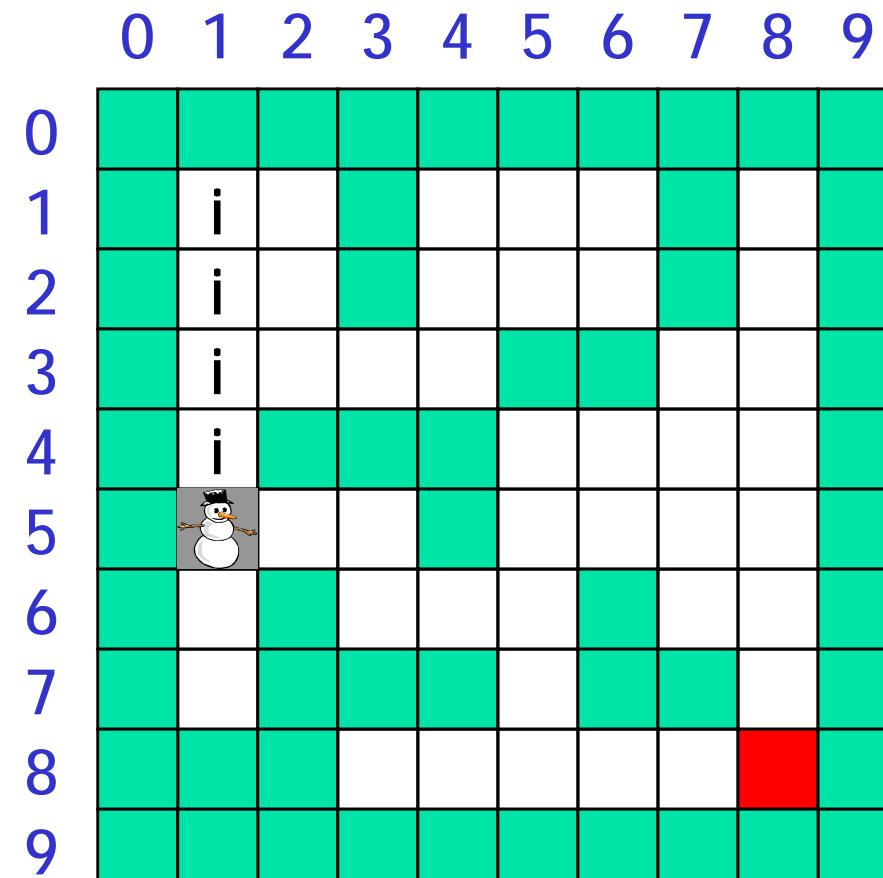
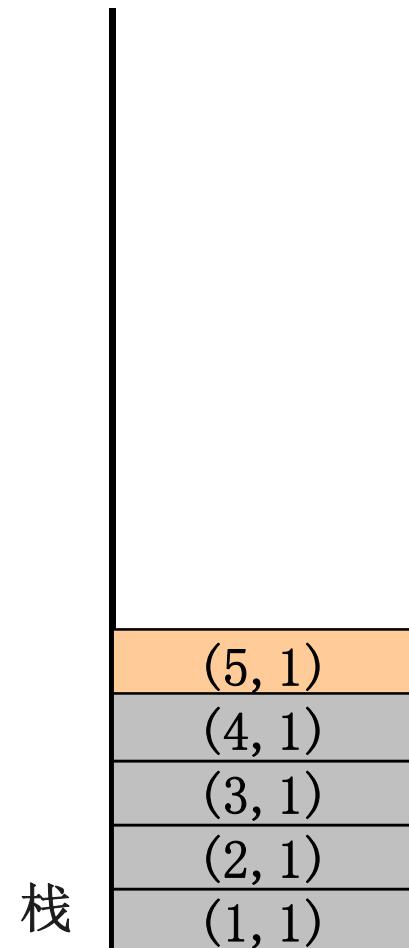
- 向下方前进一步



break

# 迷宫问题(续)

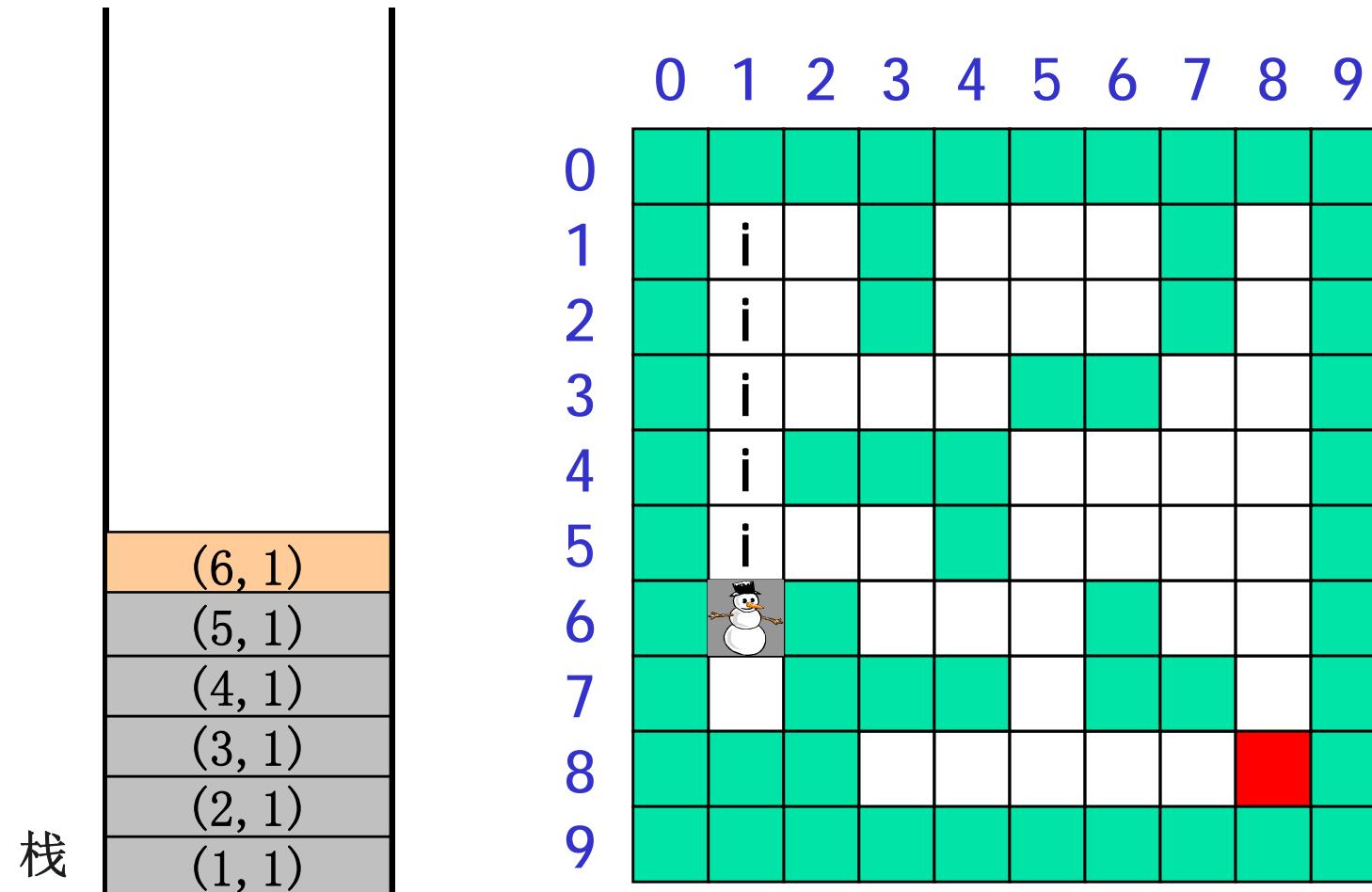
- 向下方前进一步



break

# 迷宫问题(续)

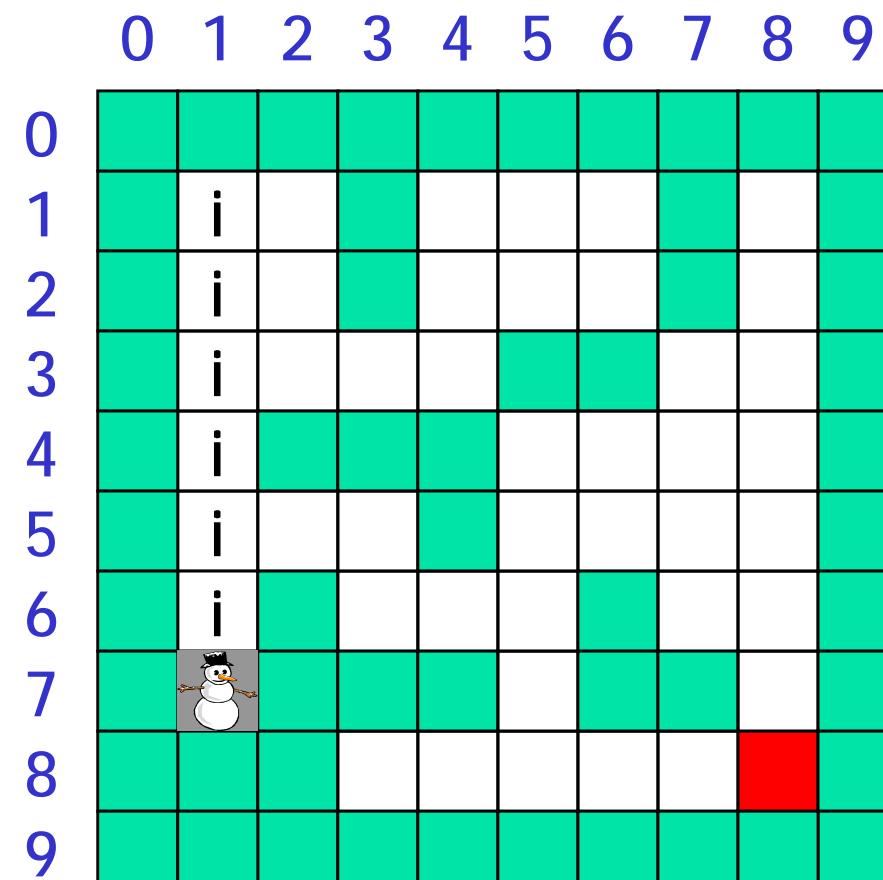
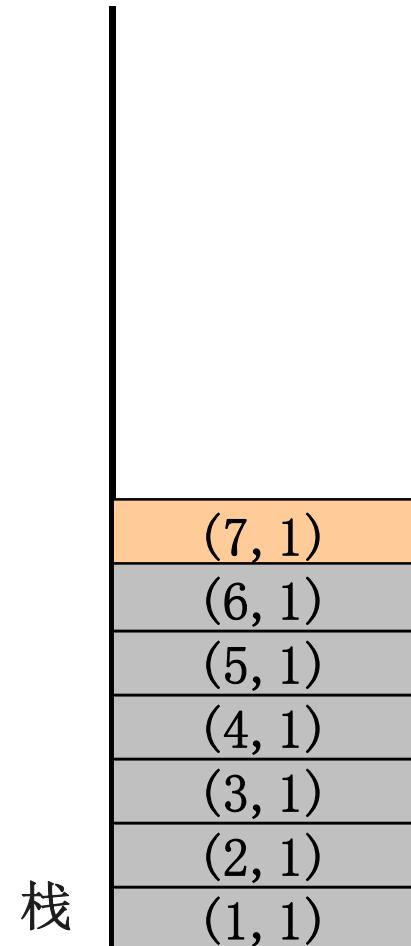
- 向下方前进一步



break

# 迷宫问题(续)

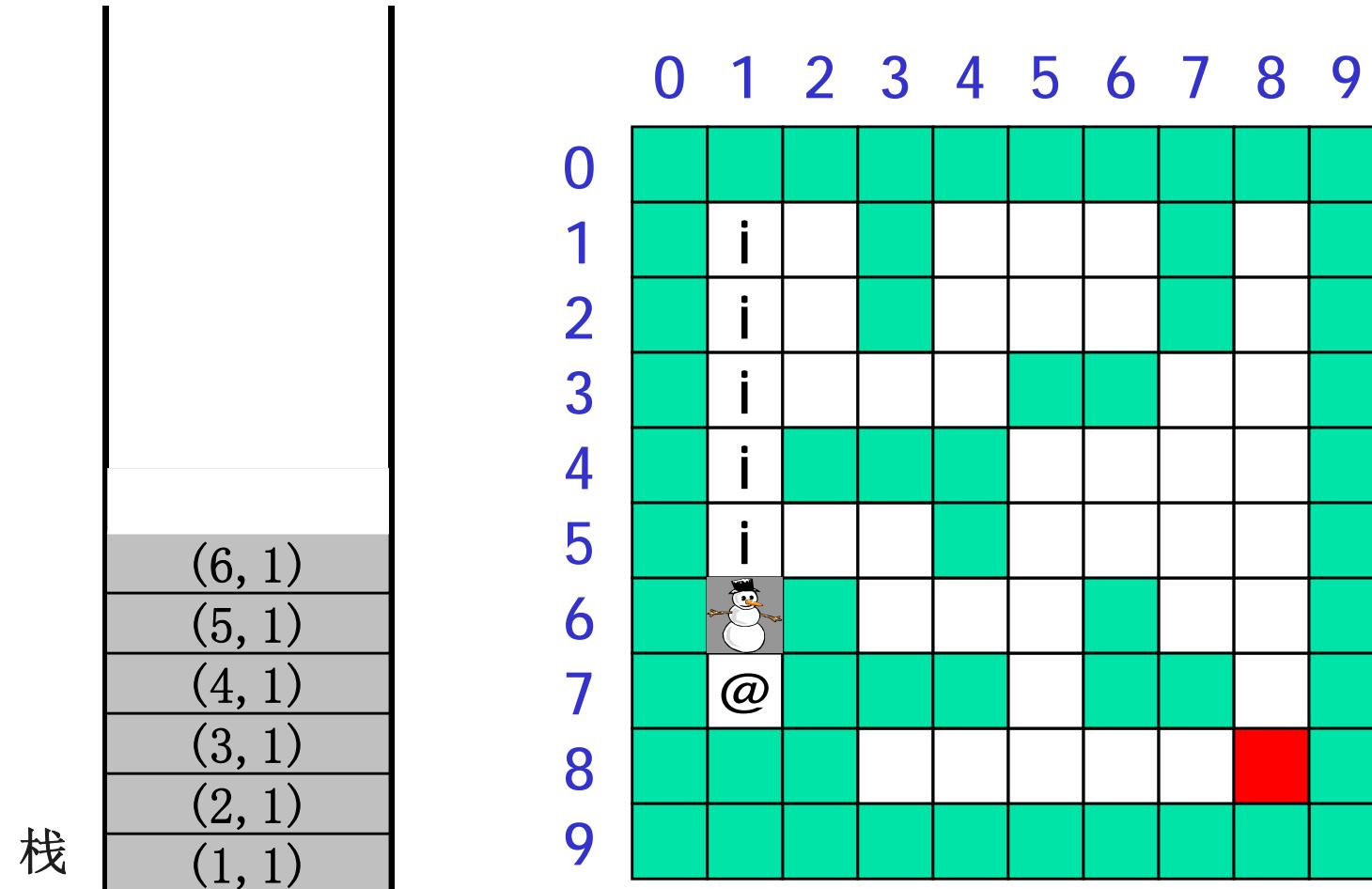
- 向下方前进一步



break

## 迷宫问题(续)

- 向下方、右方、左方均不能前进，上方是来路，则后退

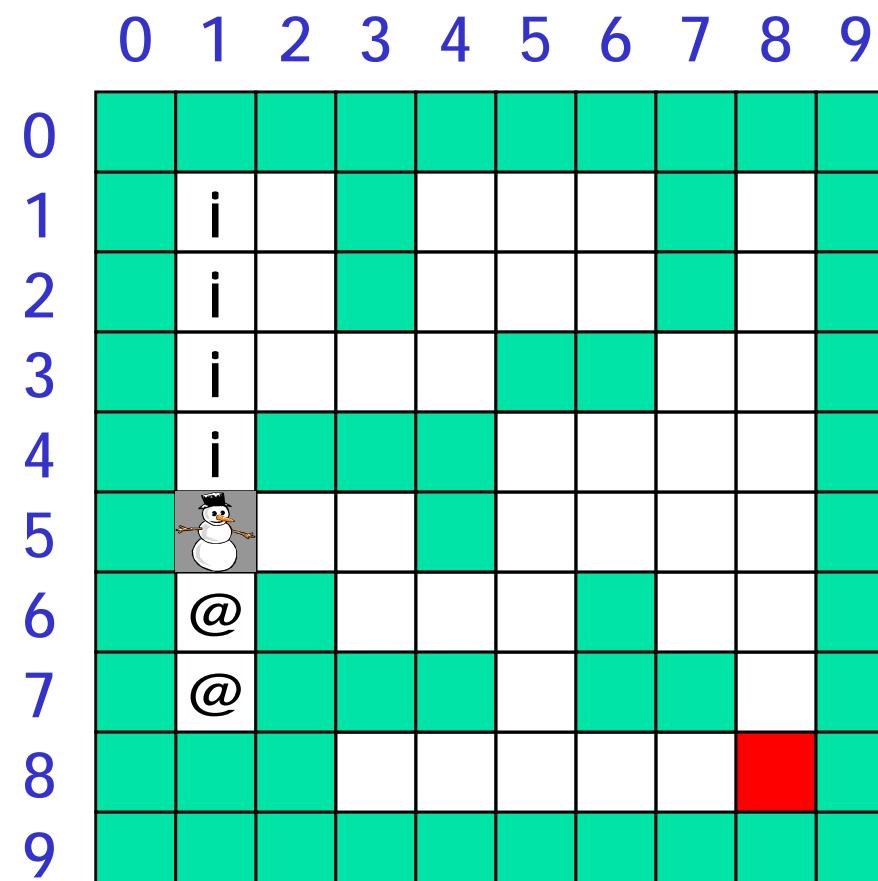


break

# 迷宫问题(续)

- 向右方、左方均不能前进，下方路不通，上方是来路，则后退

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

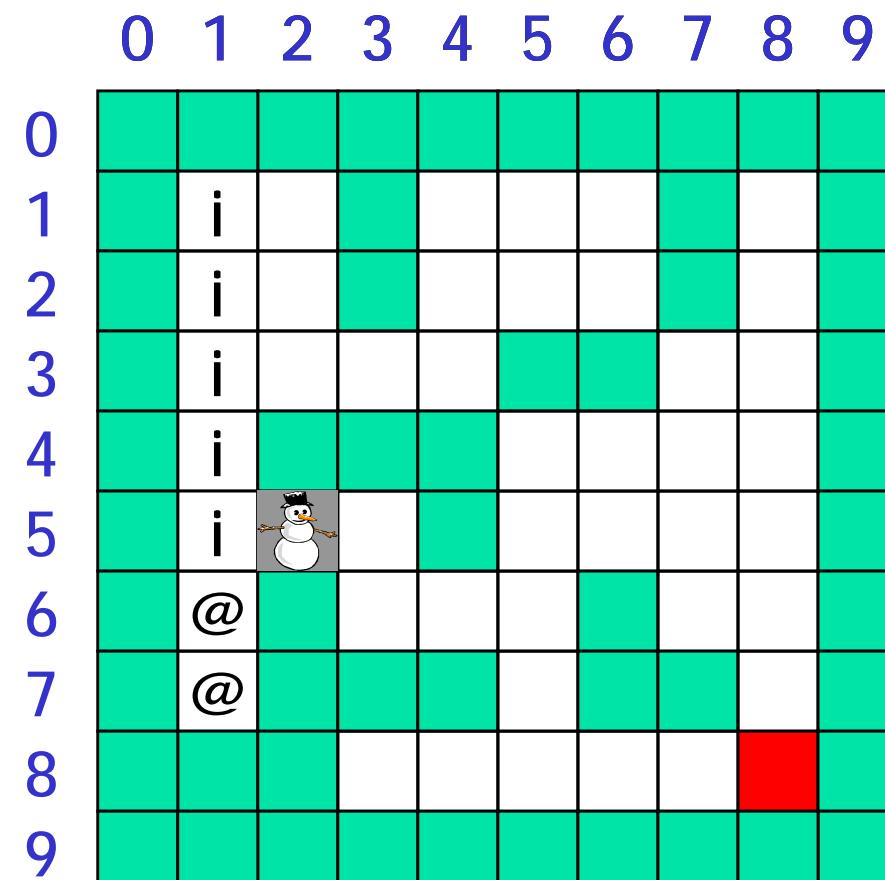


break

# 迷宫问题(续)

- 向右方前进一步

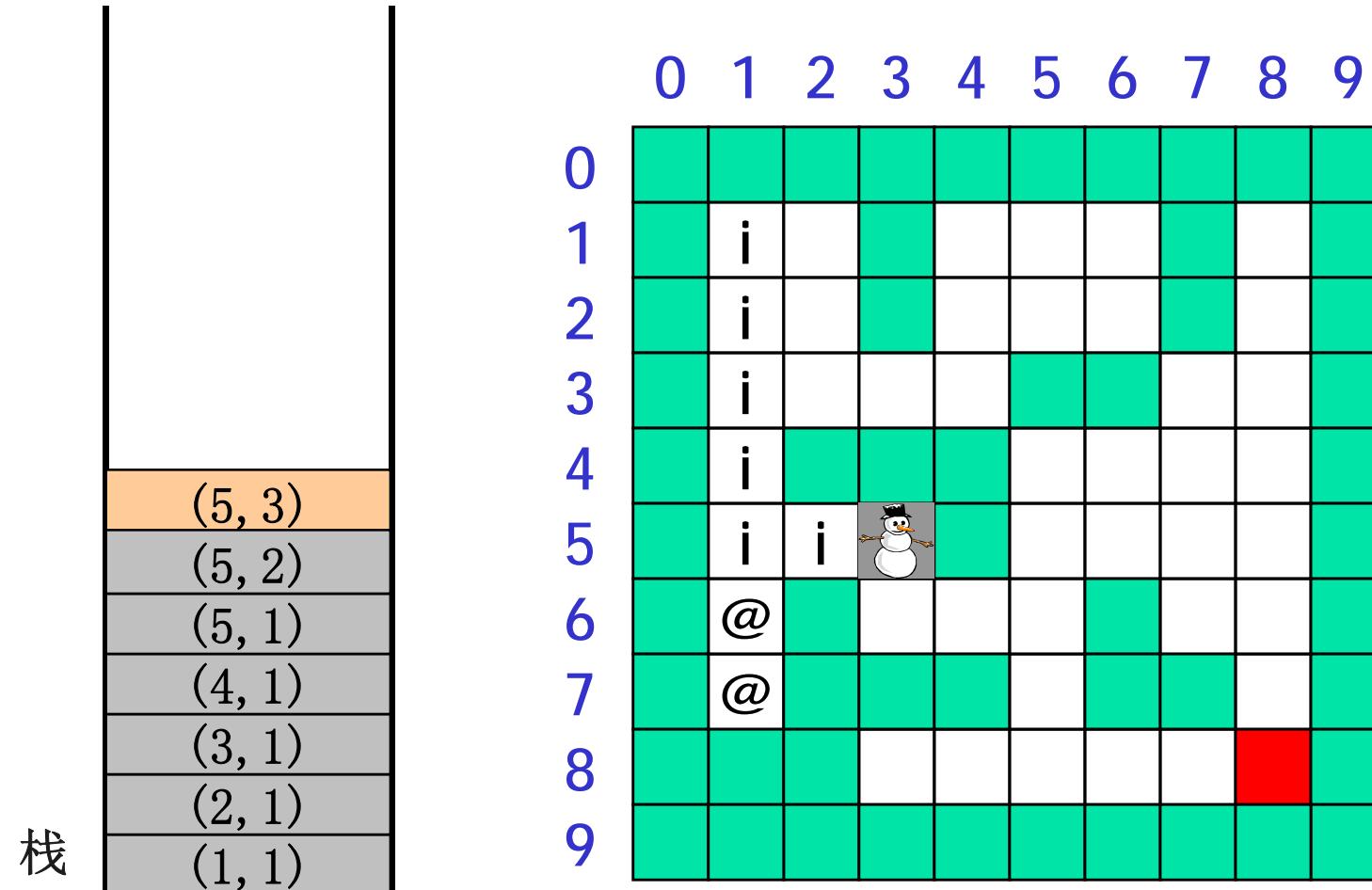
栈	
	(5, 2)
	(5, 1)
	(4, 1)
	(3, 1)
	(2, 1)
	(1, 1)



break

# 迷宫问题(续)

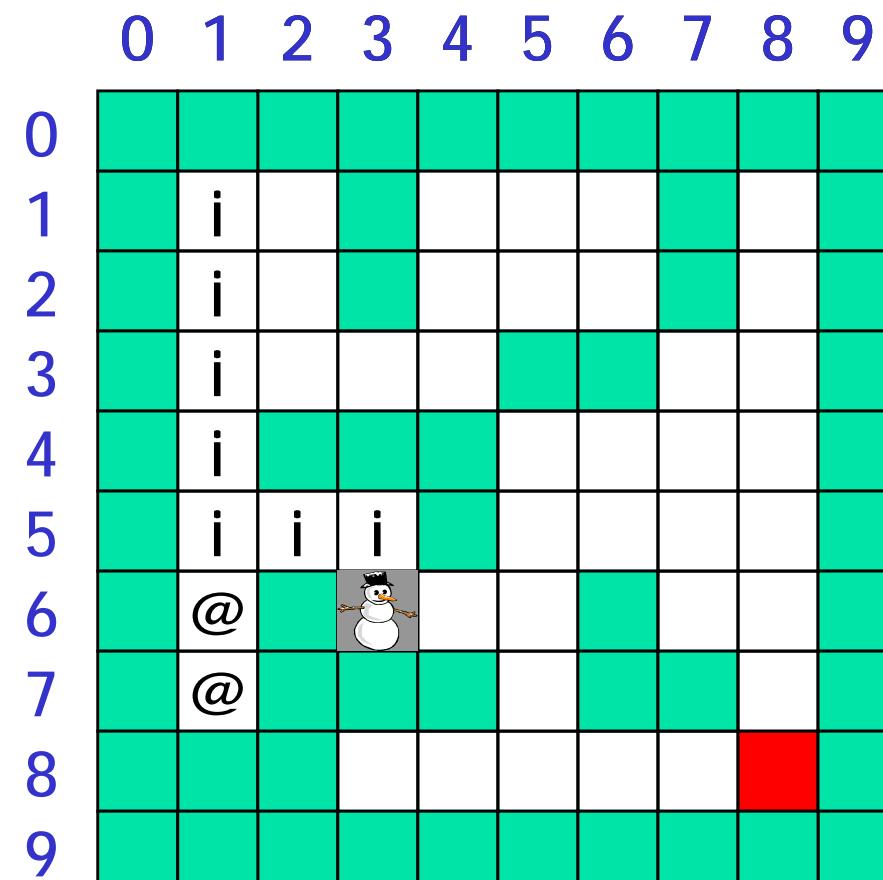
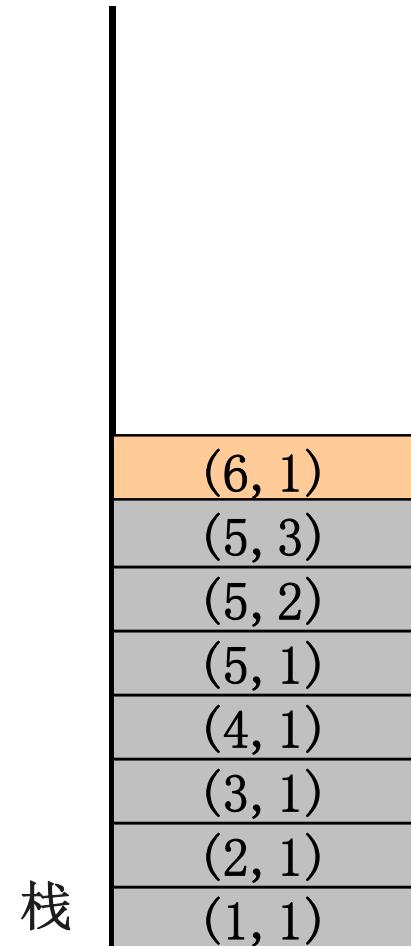
- 下方路不通，向右方前进一步



break

# 迷宫问题(续)

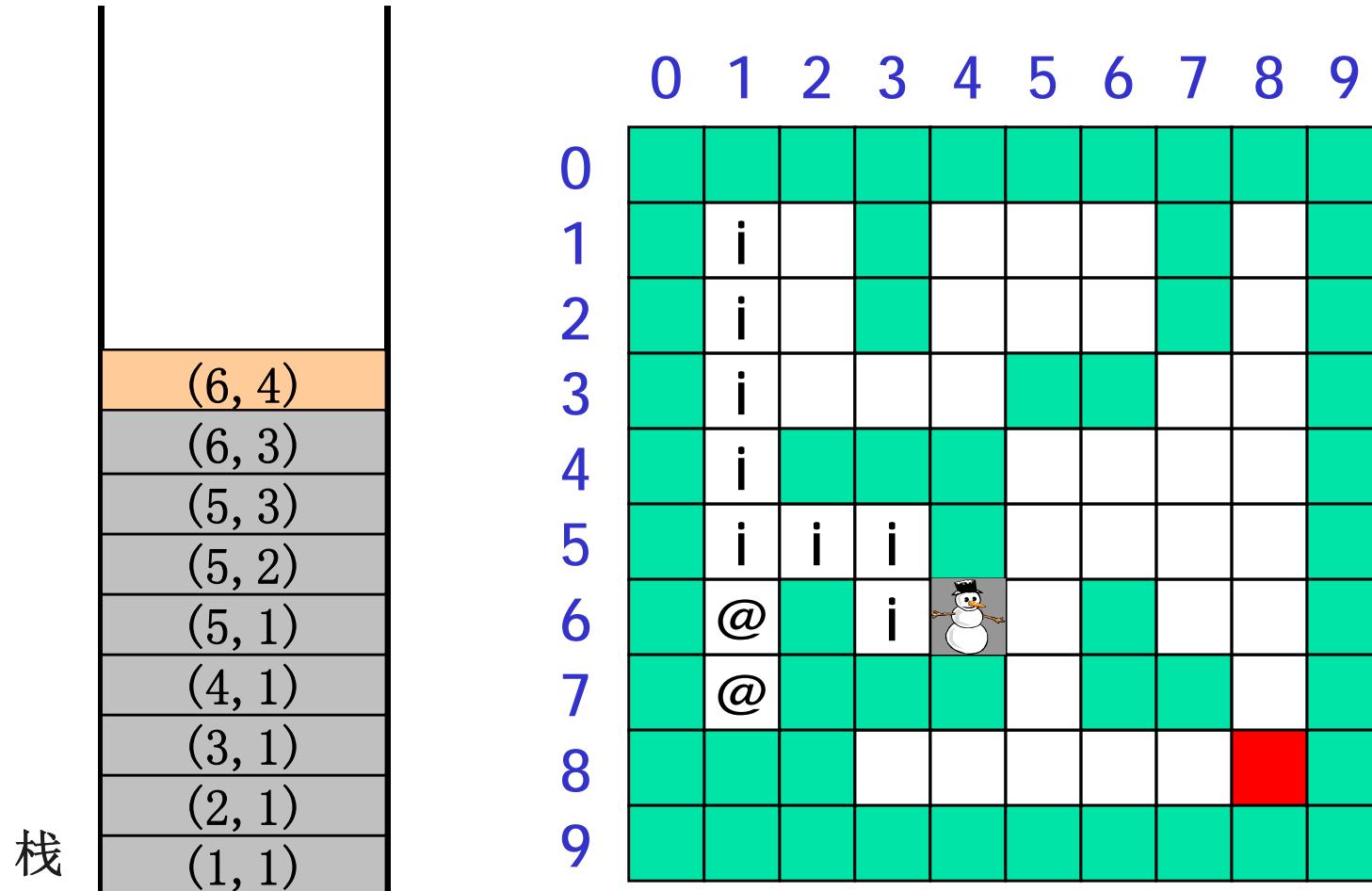
## ■ 向下方前进一步



break

# 迷宫问题(续)

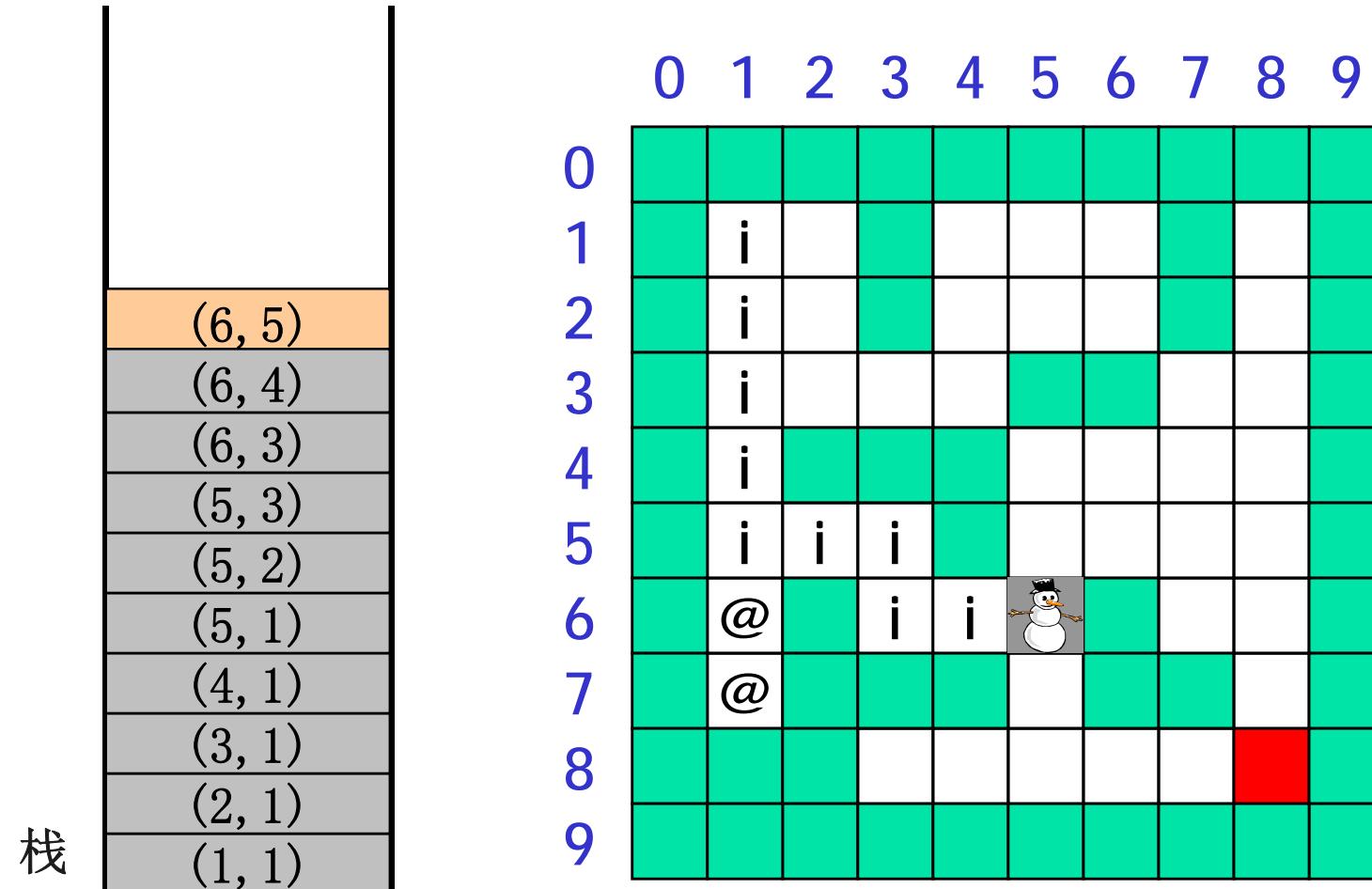
- 下方路不通，向右方前进一步



break

# 迷宫问题(续)

- 下方路不通，向右方前进一步

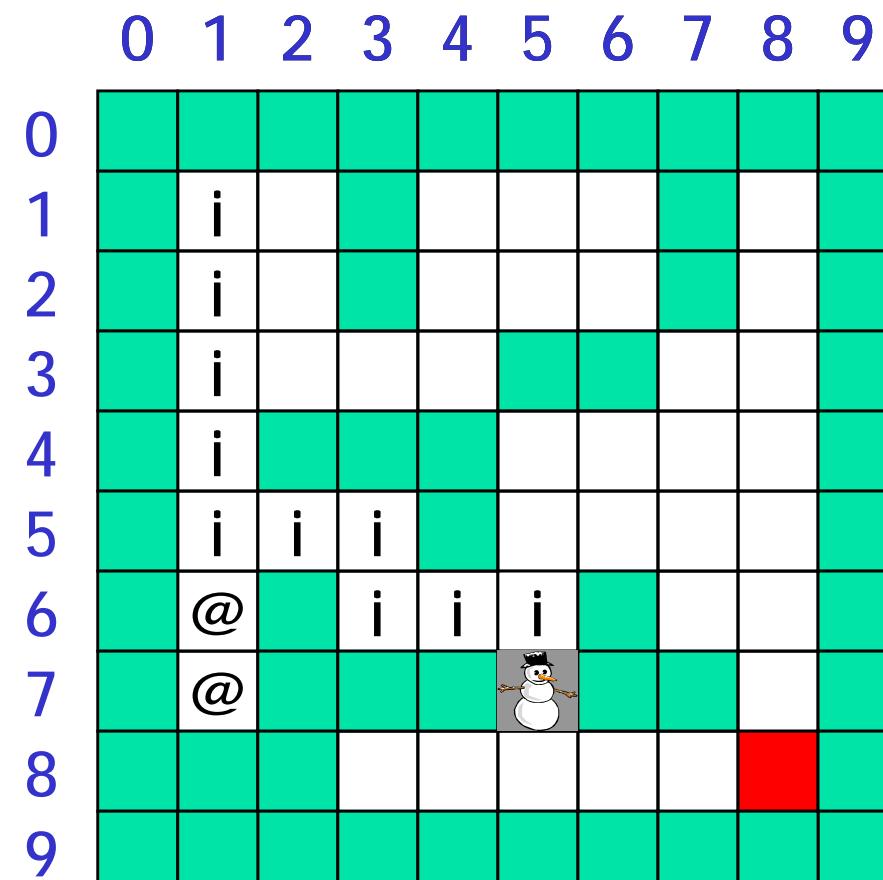


break

# 迷宫问题(续)

## ■ 向下方前进一步

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

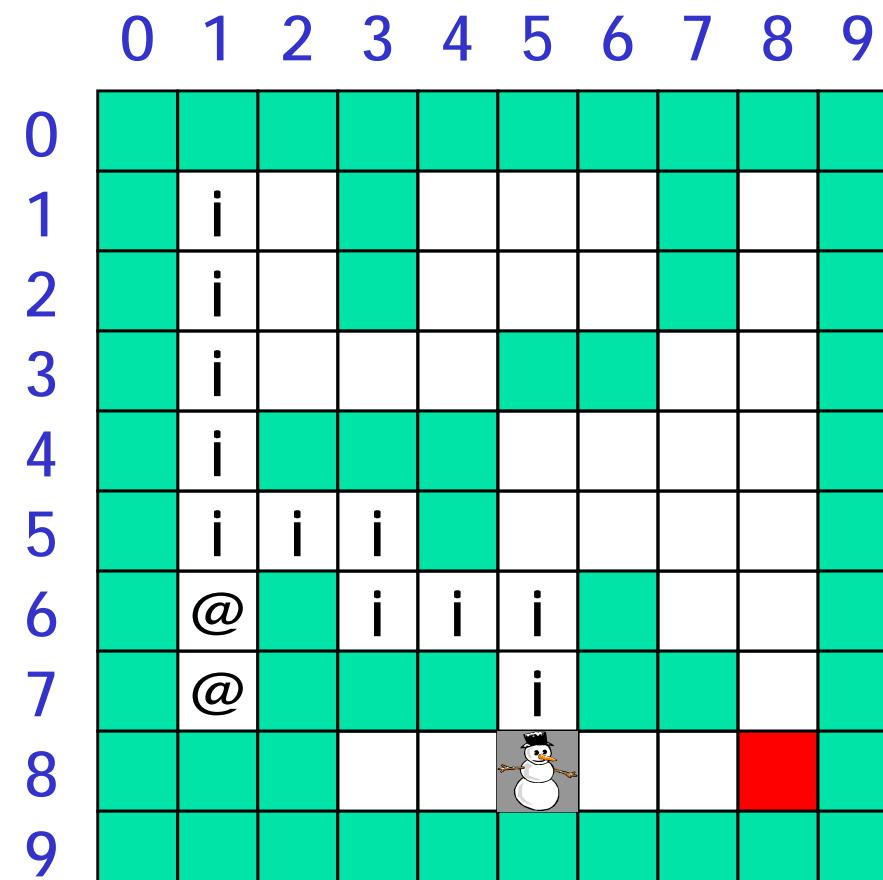


break

# 迷宫问题(续)

## ■ 向下方前进一步

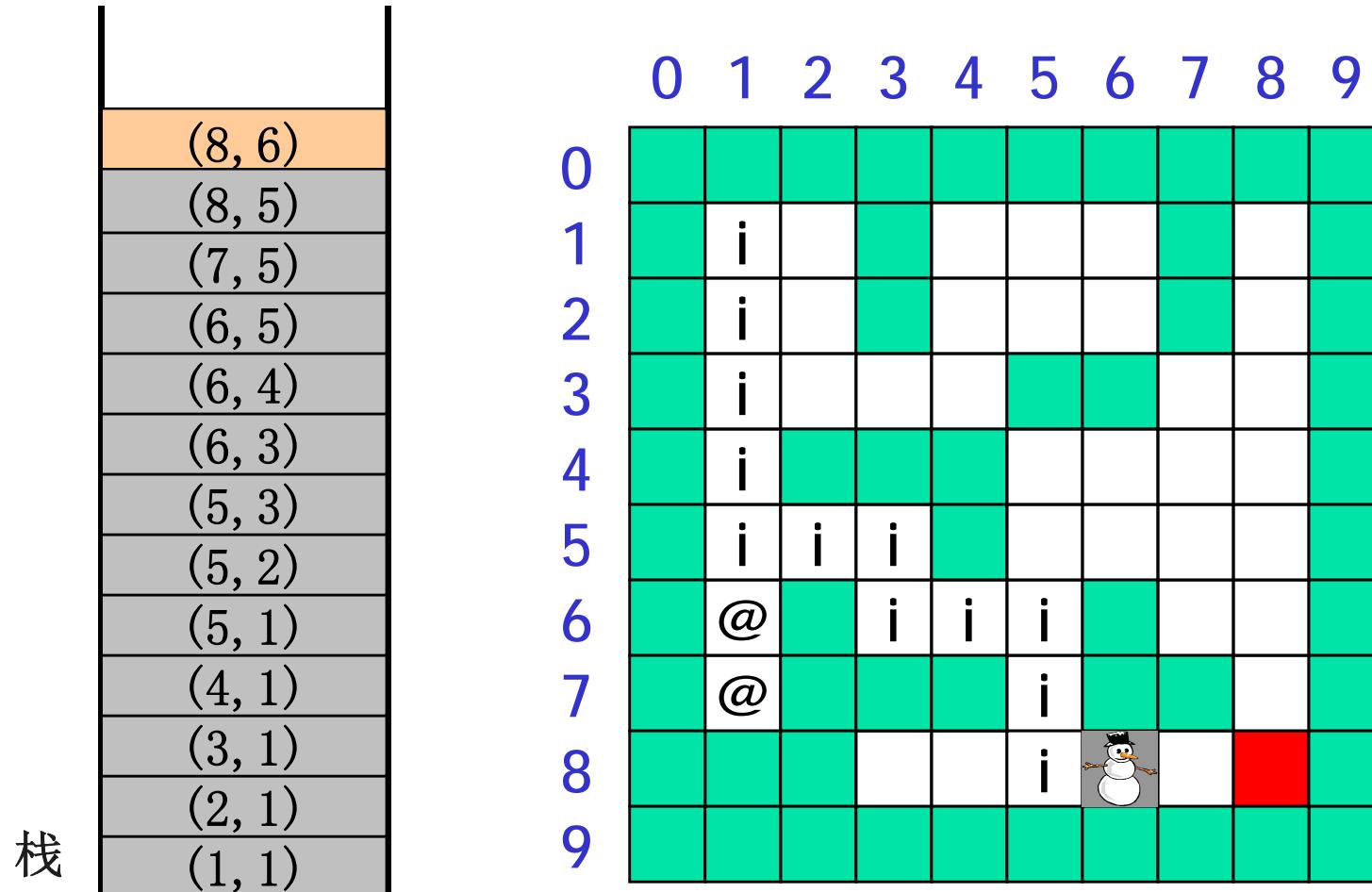
栈
(8, 5)
(7, 5)
(6, 5)
(6, 4)
(6, 3)
(5, 3)
(5, 2)
(5, 1)
(4, 1)
(3, 1)
(2, 1)
(1, 1)



break

# 迷宫问题(续)

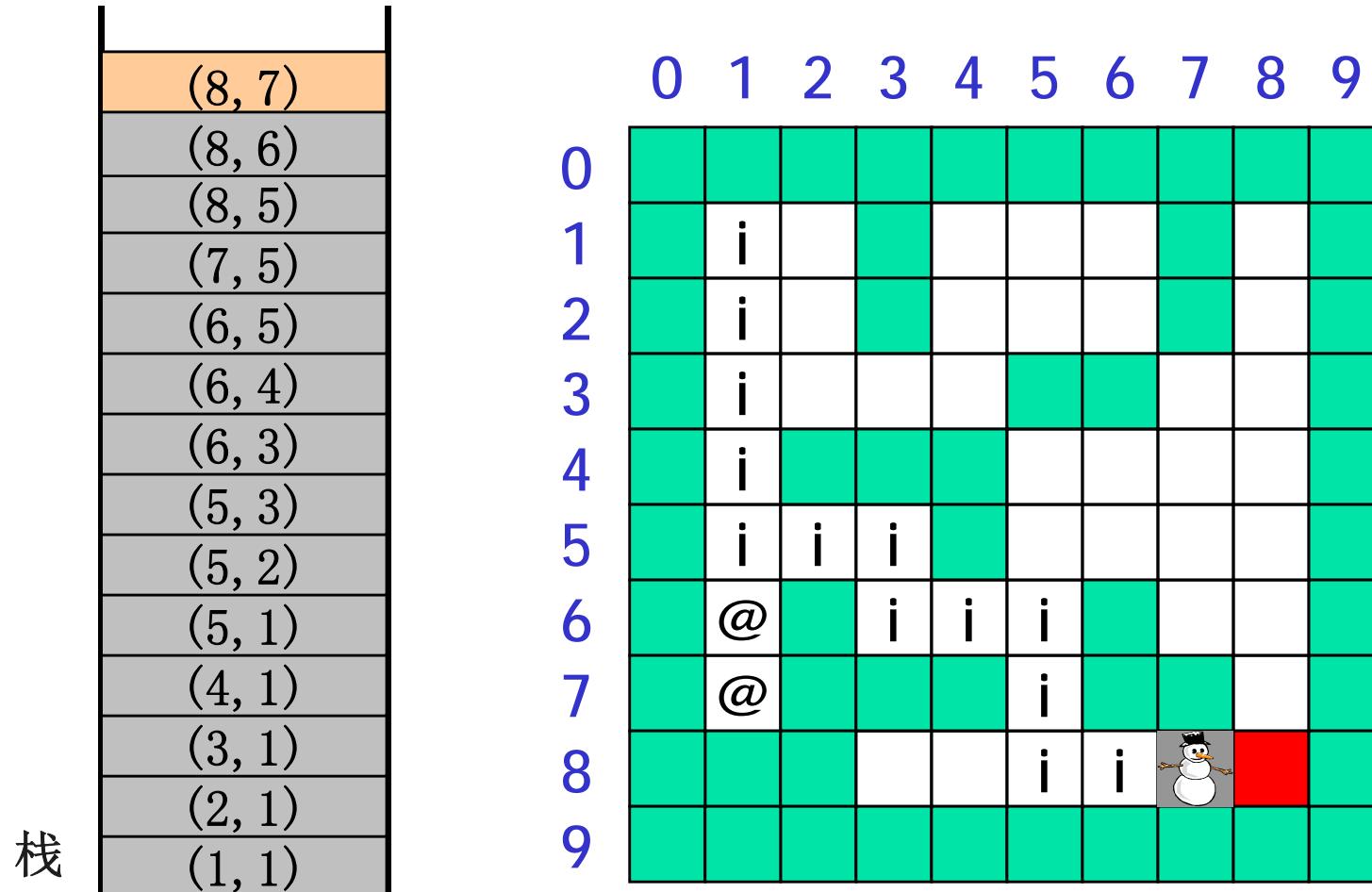
- 下方路不通，向右方前进一步



break

# 迷宫问题(续)

- 下方路不通，向右方前进一步

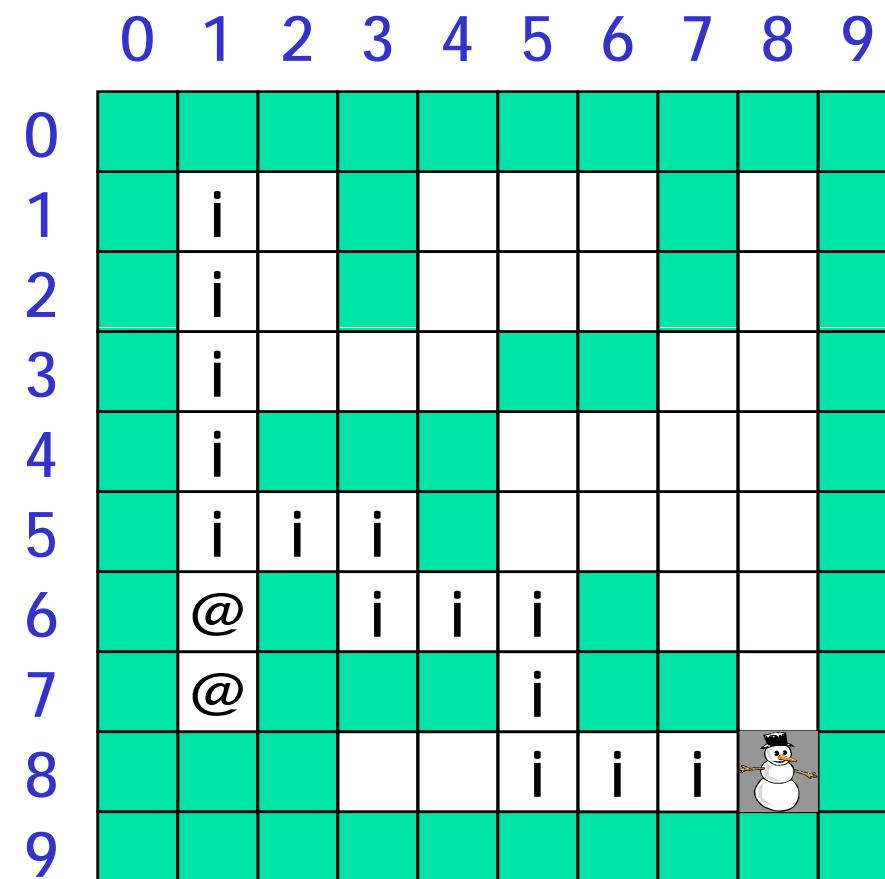


break

# 迷宫问题(续)

- 下方路不通，向右方前进一步，到达出口

栈	(8, 8)
	(8, 7)
	(8, 6)
	(8, 5)
	(7, 5)
	(6, 5)
	(6, 4)
	(6, 3)
	(5, 3)
	(5, 2)
	(5, 1)
	(4, 1)
	(3, 1)
	(2, 1)
	(1, 1)

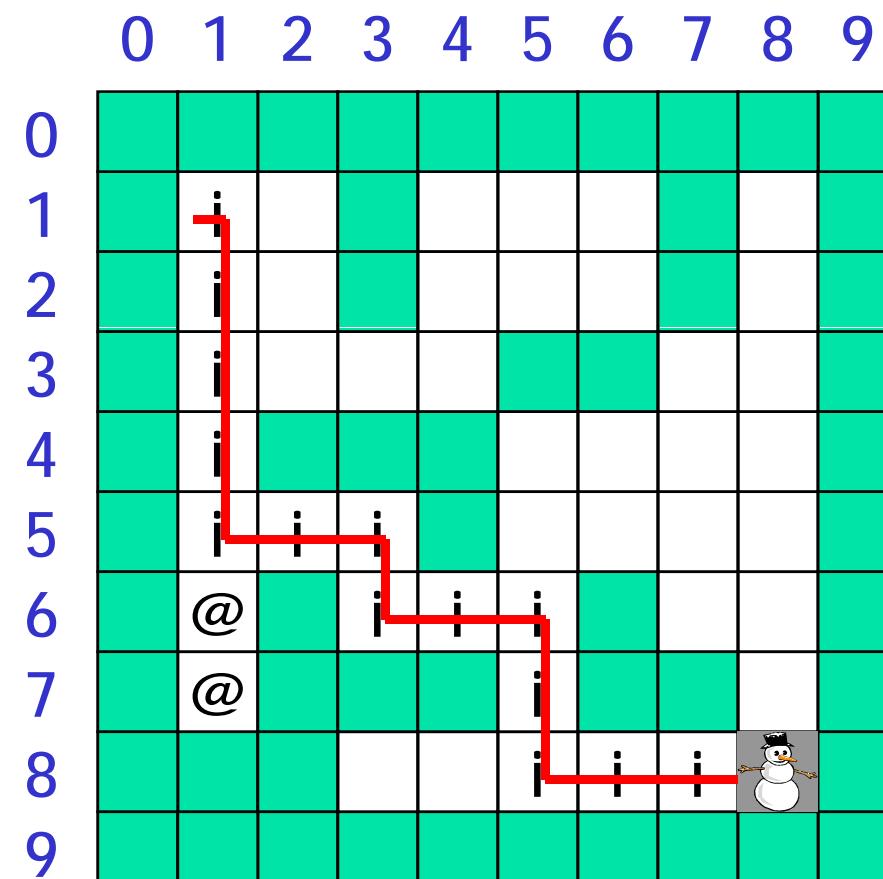


break

# 迷宫问题(续)

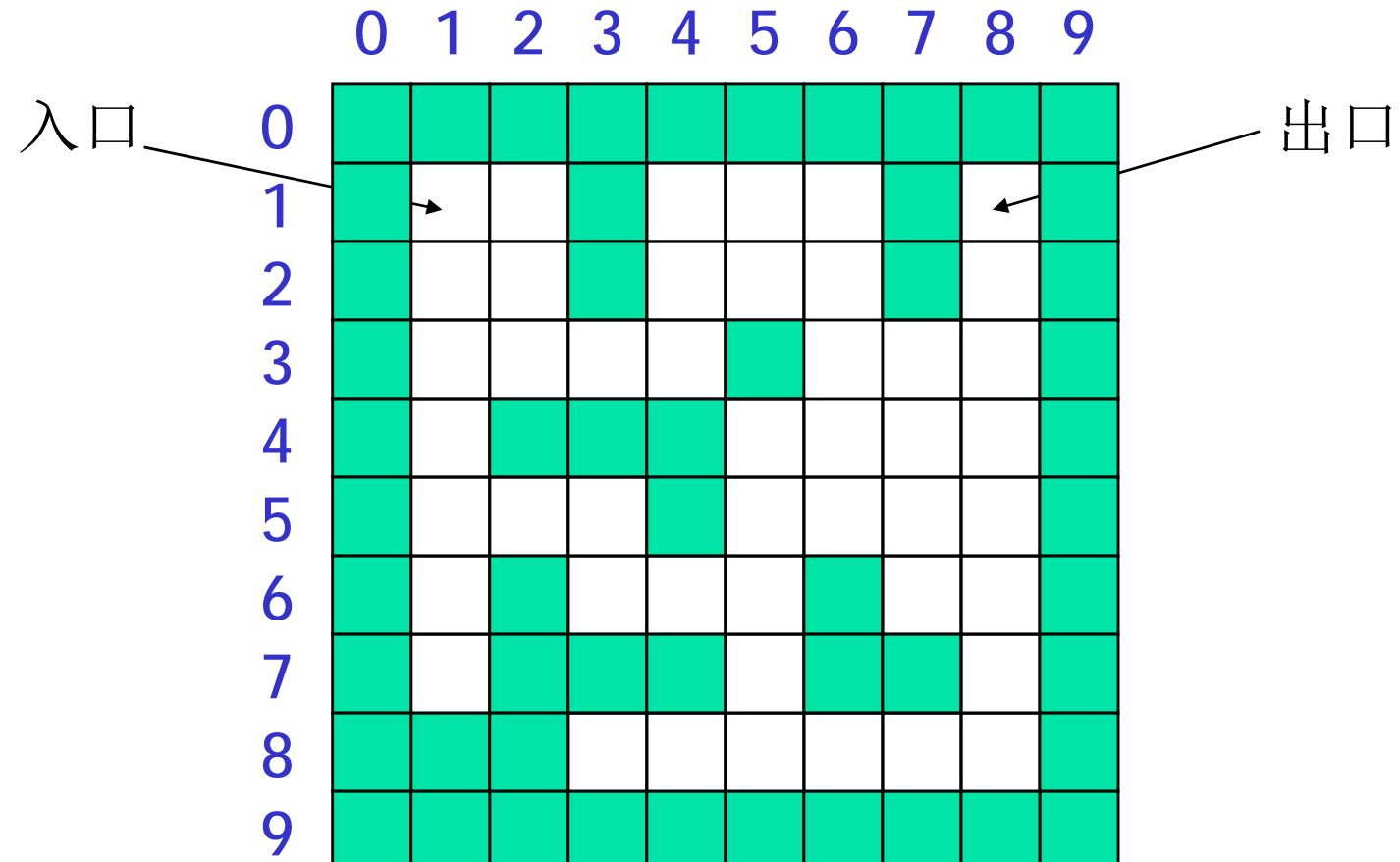
- 用栈保存了路径

栈	(8, 8)
	(8, 7)
	(8, 6)
	(8, 5)
	(7, 5)
	(6, 5)
	(6, 4)
	(6, 3)
	(5, 3)
	(5, 2)
	(5, 1)
	(4, 1)
	(3, 1)
	(2, 1)
	(1, 1)



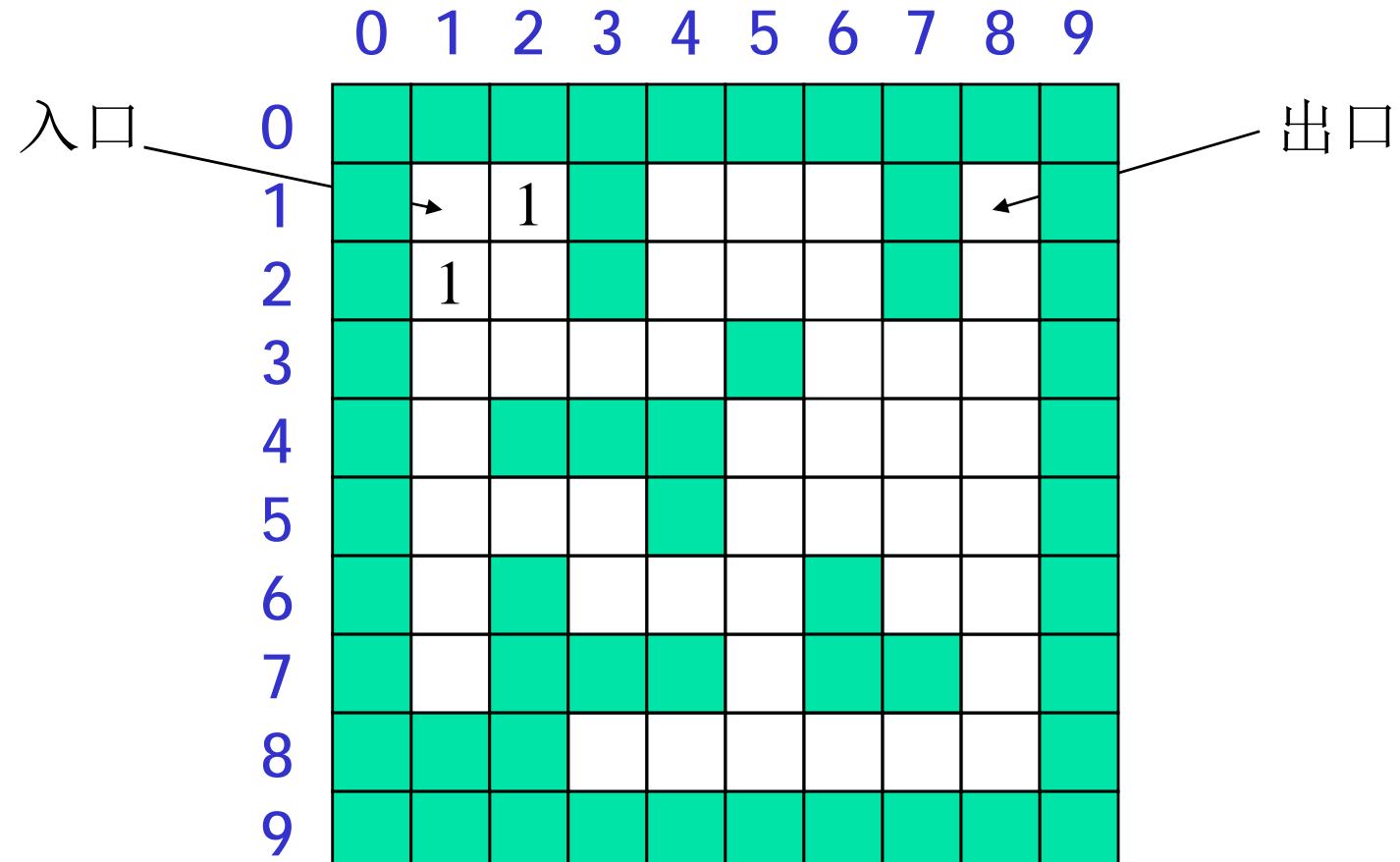
## 迷宫问题(最短路径)

- 借助于队列可求得入口到出口的最短路径(若存在)



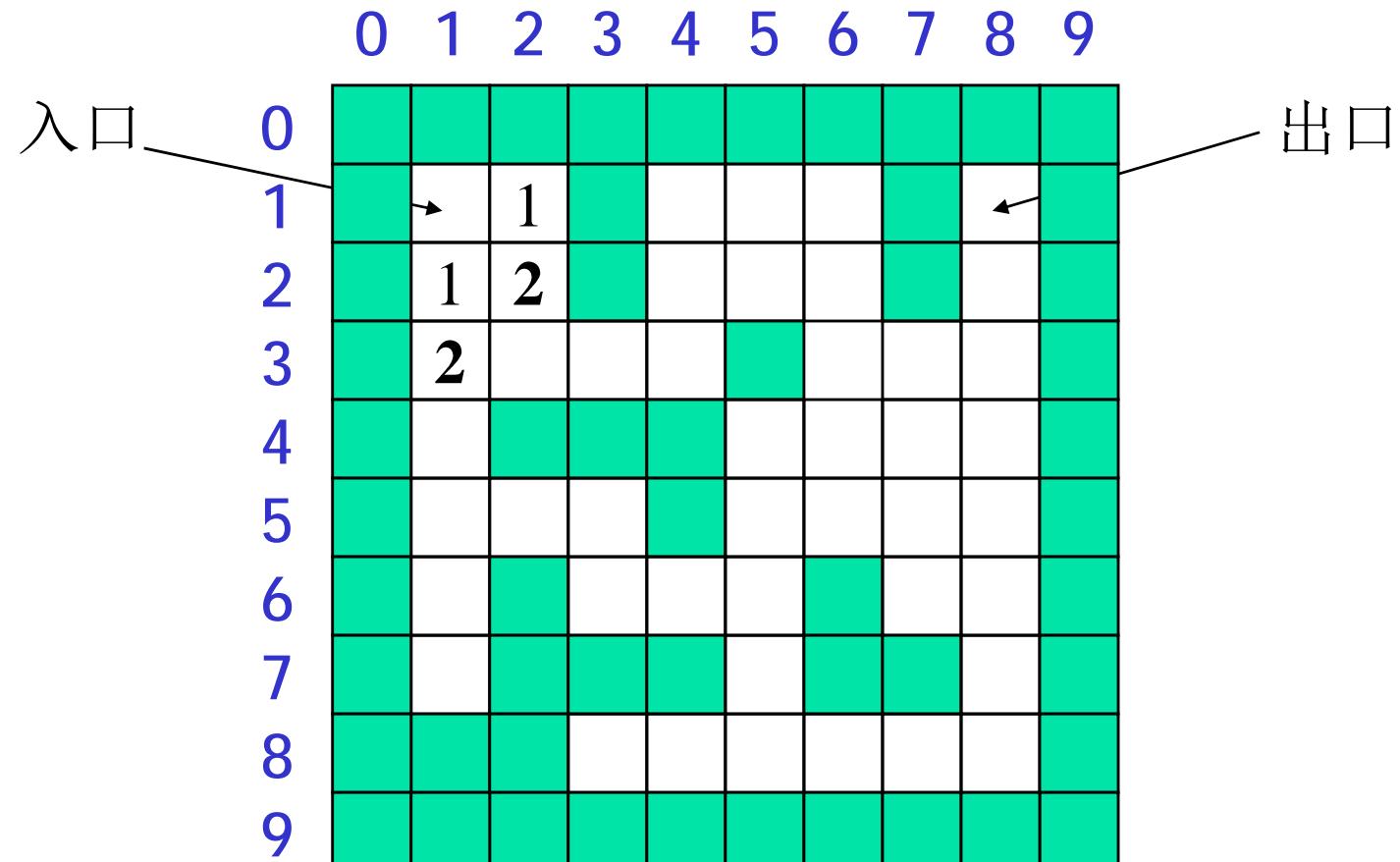
## 迷宫问题(最短路径)

- 借助于队列可求得入口到出口的最短路径(若存在)



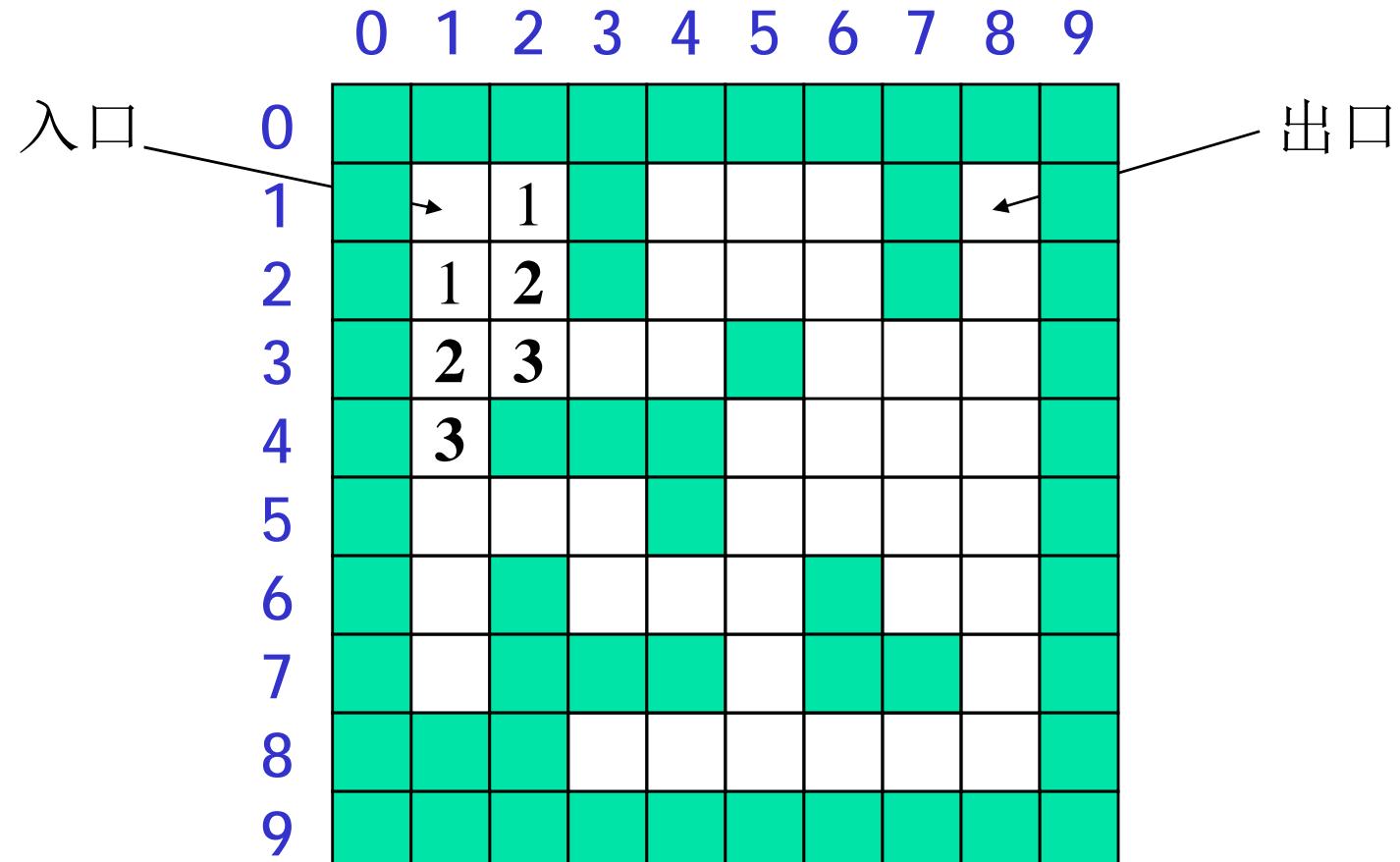
## 迷宫问题(最短路径)

- 借助于队列可求得入口到出口的最短路径(若存在)



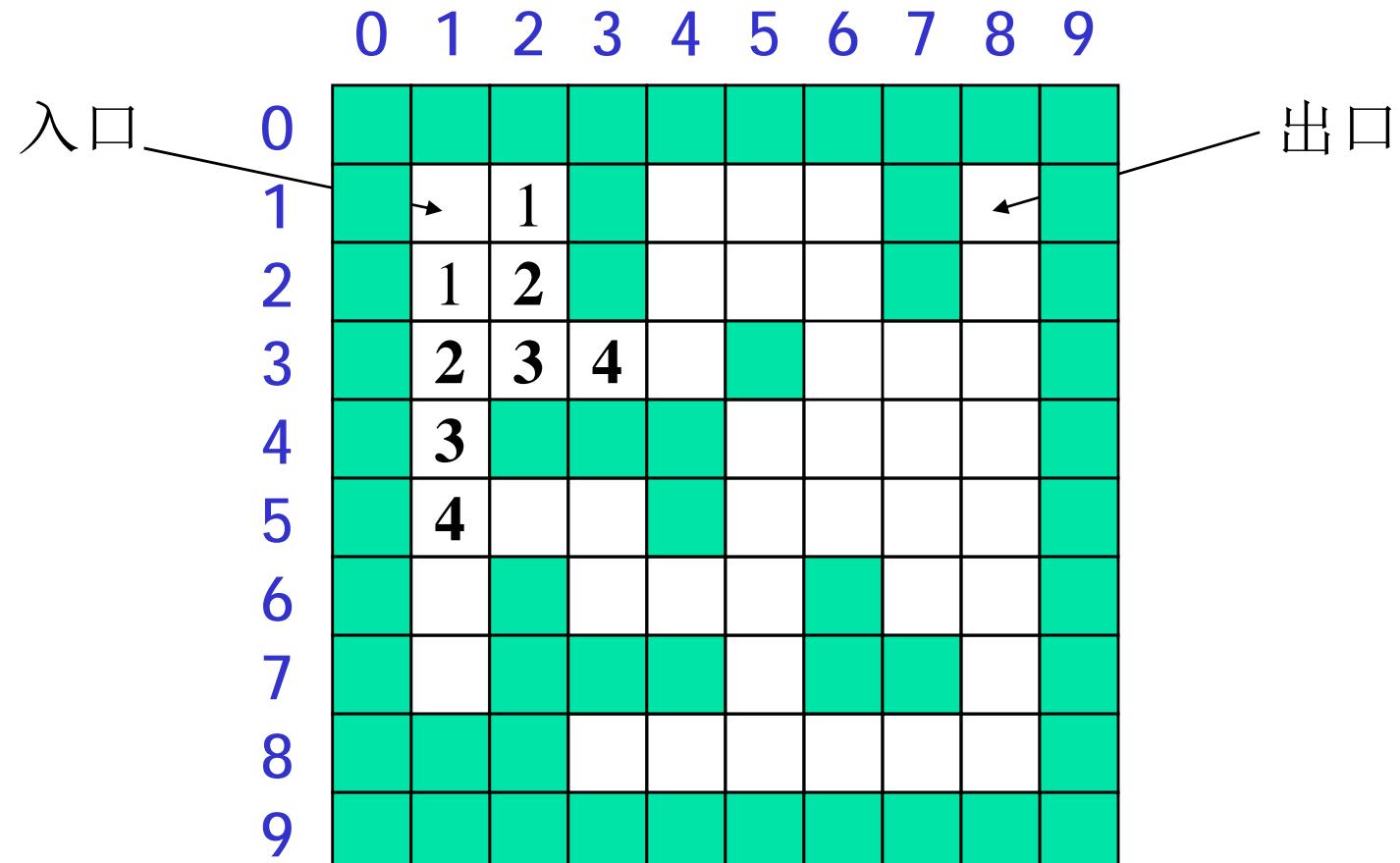
## 迷宫问题(最短路径)

- 借助于队列可求得入口到出口的最短路径(若存在)



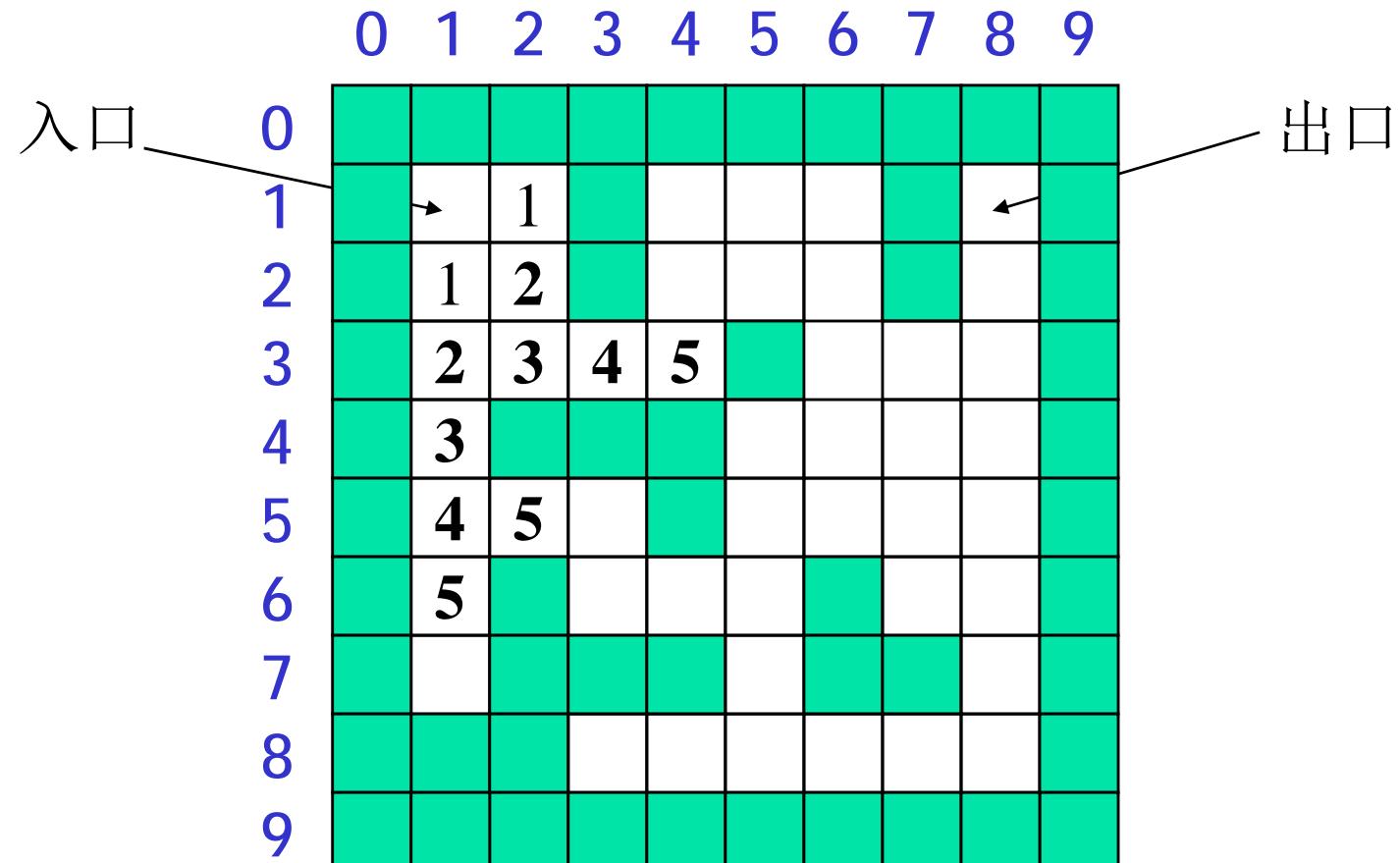
# 迷宫问题(最短路径)

- 借助于队列可求得入口到出口的最短路径(若存在)



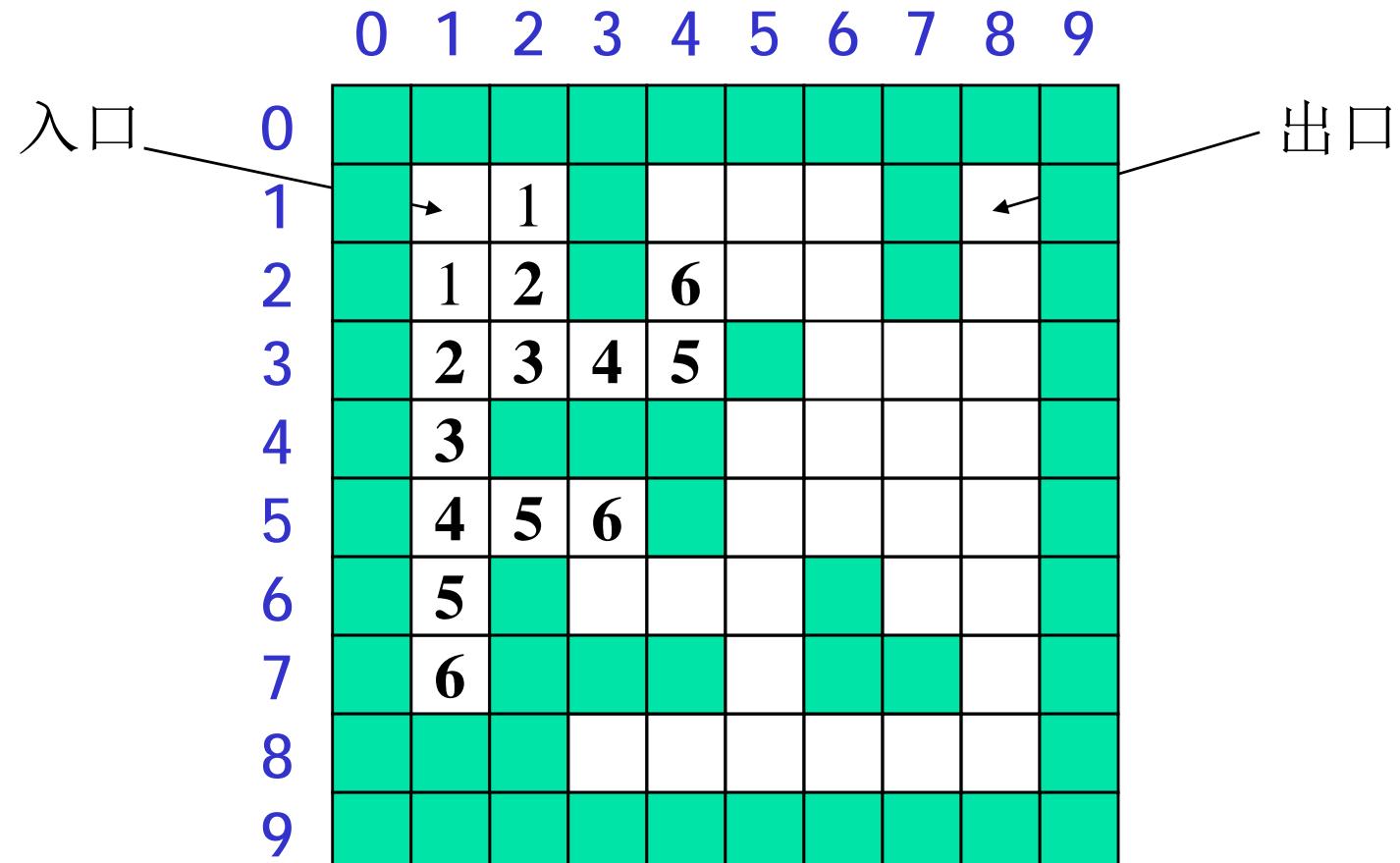
# 迷宫问题(最短路径)

- 借助于队列可求得入口到出口的最短路径(若存在)



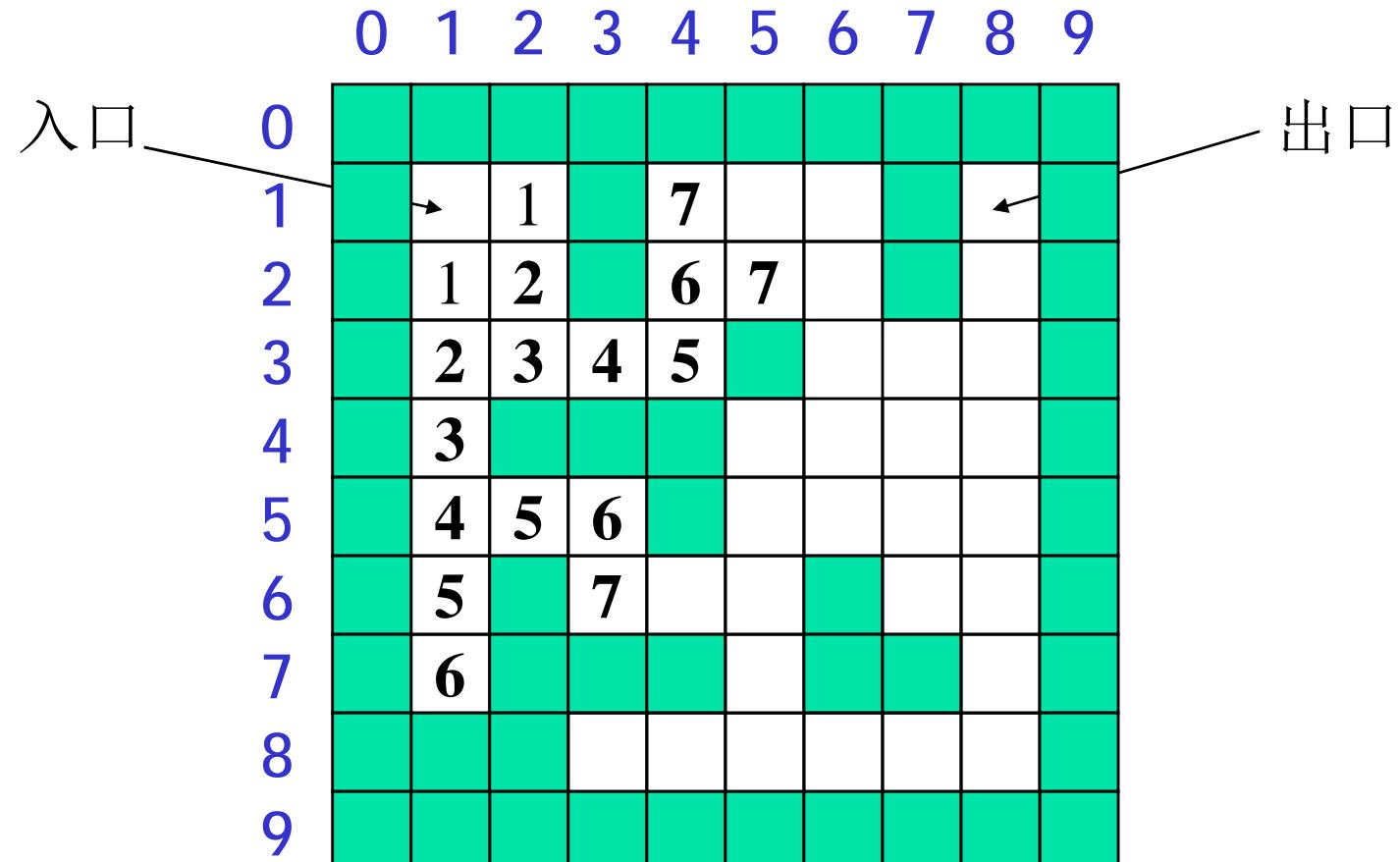
# 迷宫问题(最短路径)

- 借助于队列可求得入口到出口的最短路径(若存在)



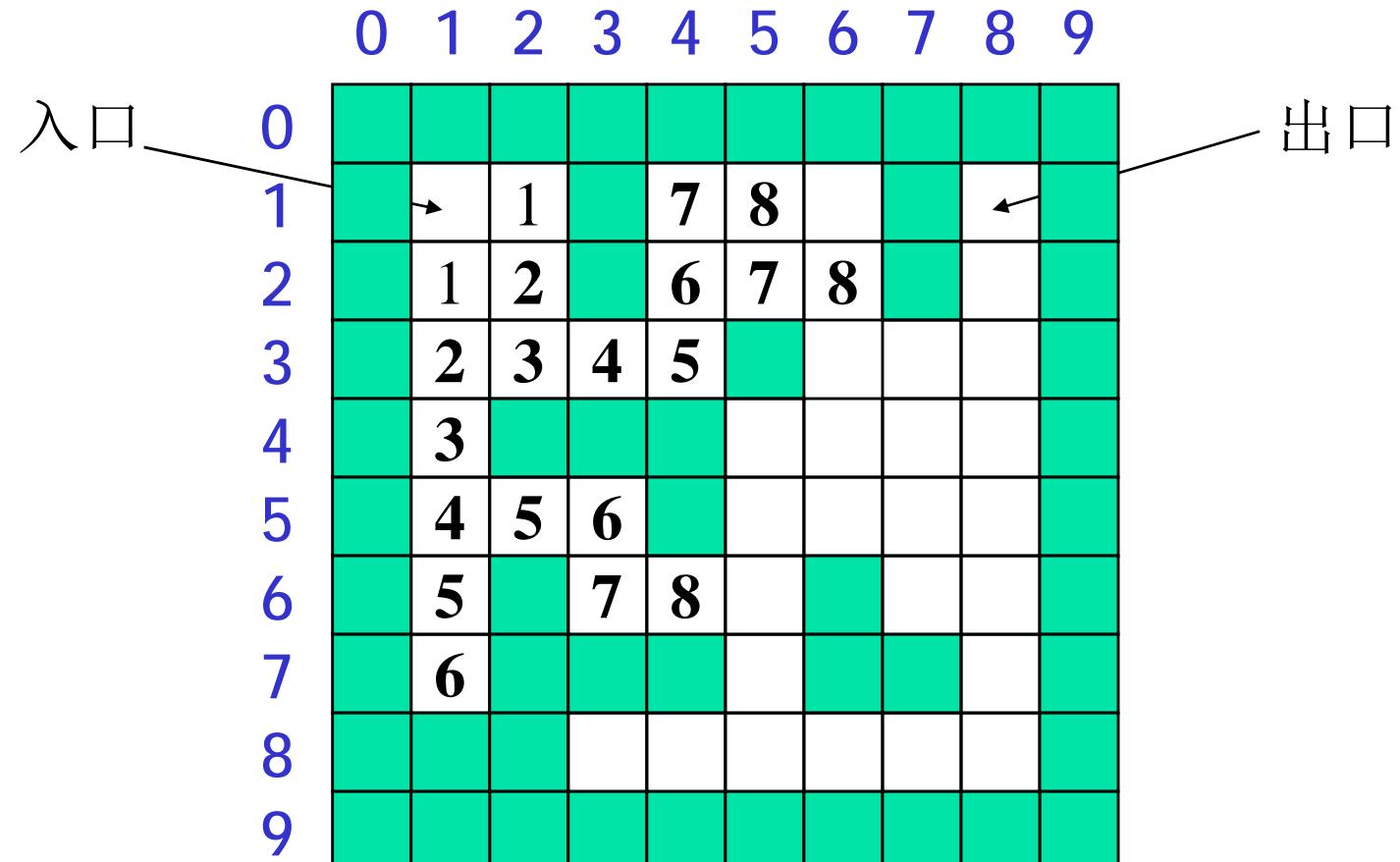
# 迷宫问题(最短路径)

- 借助于队列可求得入口到出口的最短路径(若存在)



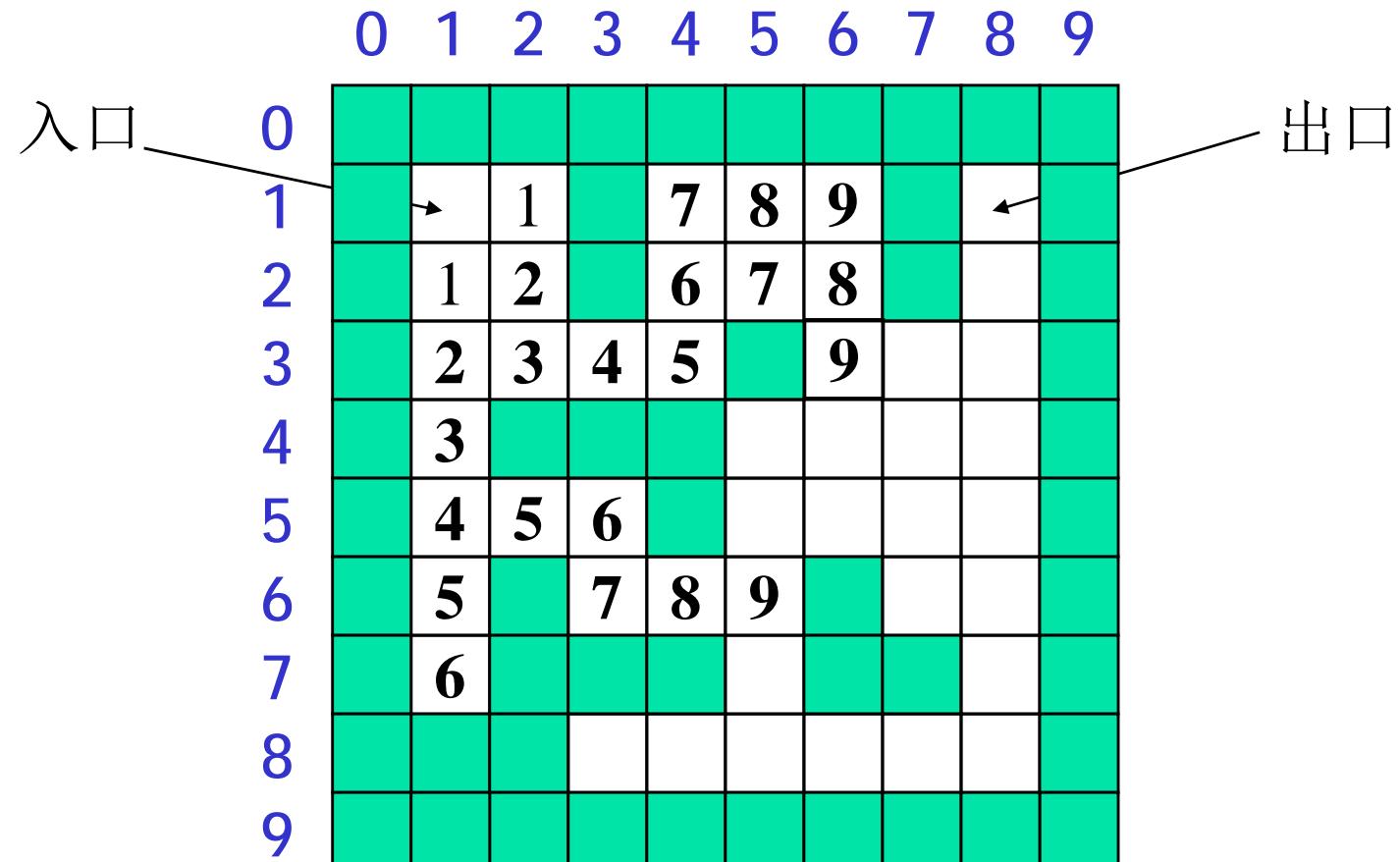
# 迷宫问题(最短路径)

- 借助于队列可求得入口到出口的最短路径(若存在)



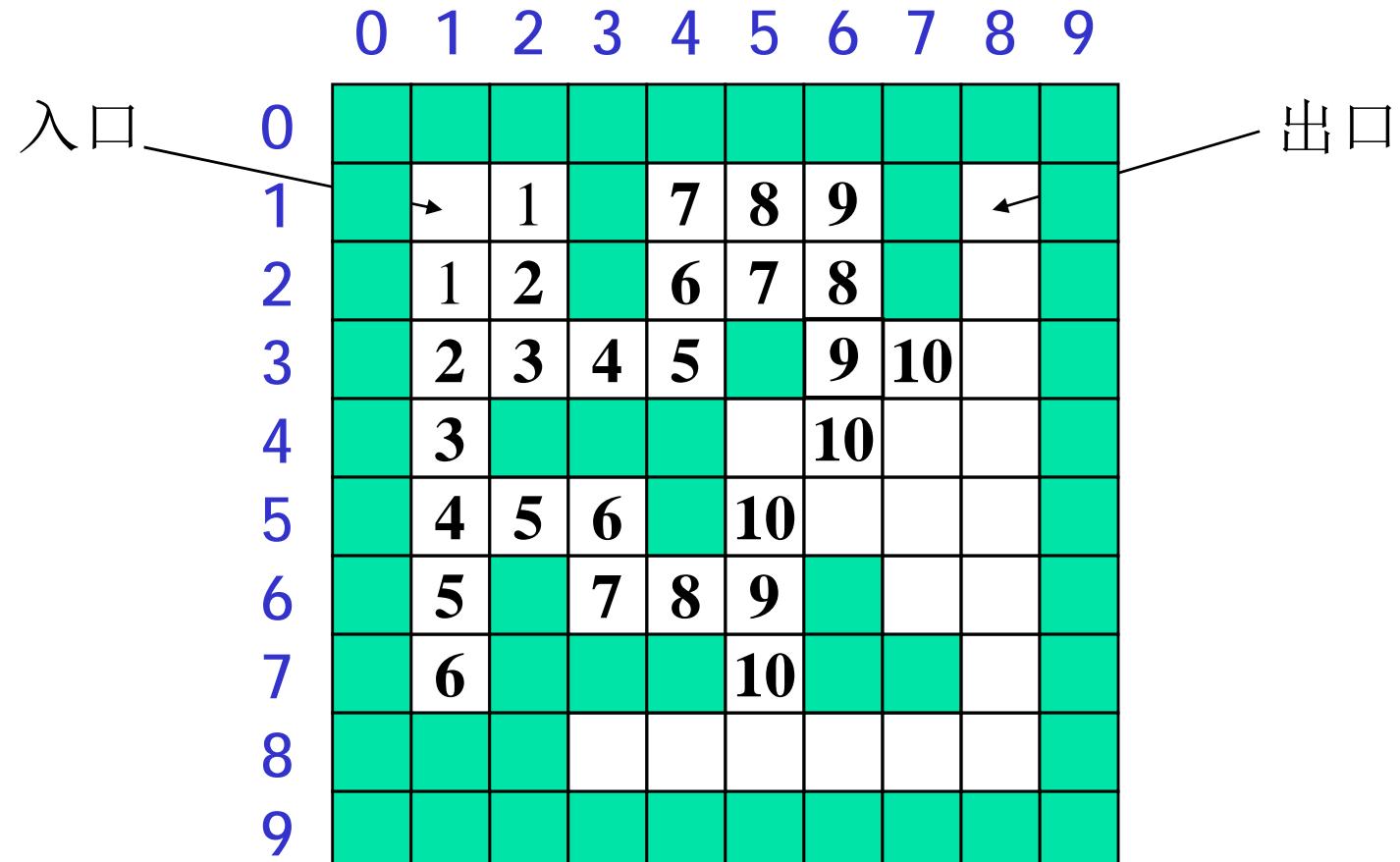
# 迷宫问题(最短路径)

- 借助于队列可求得入口到出口的最短路径(若存在)



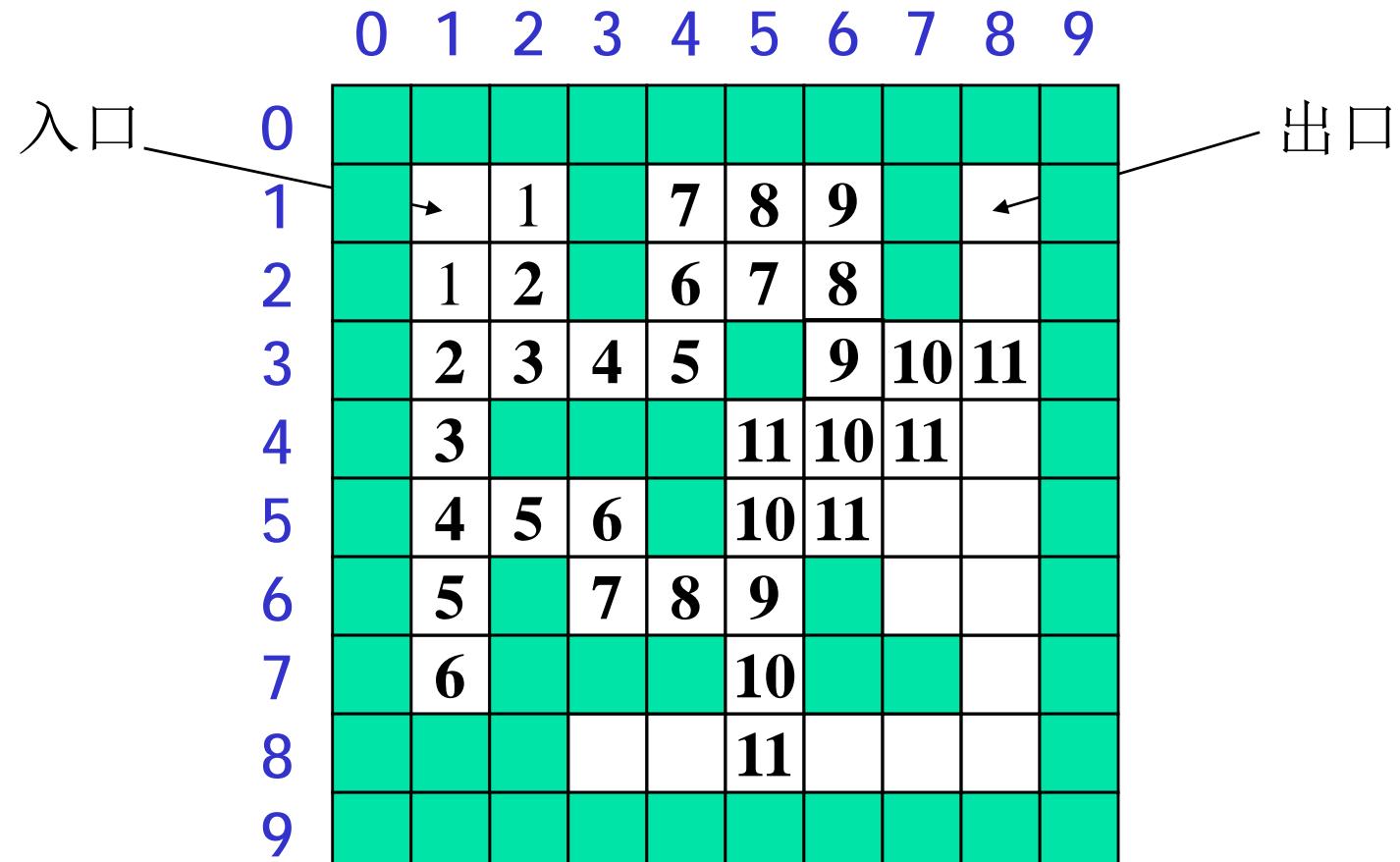
# 迷宫问题(最短路径)

- 借助于队列可求得入口到出口的最短路径(若存在)



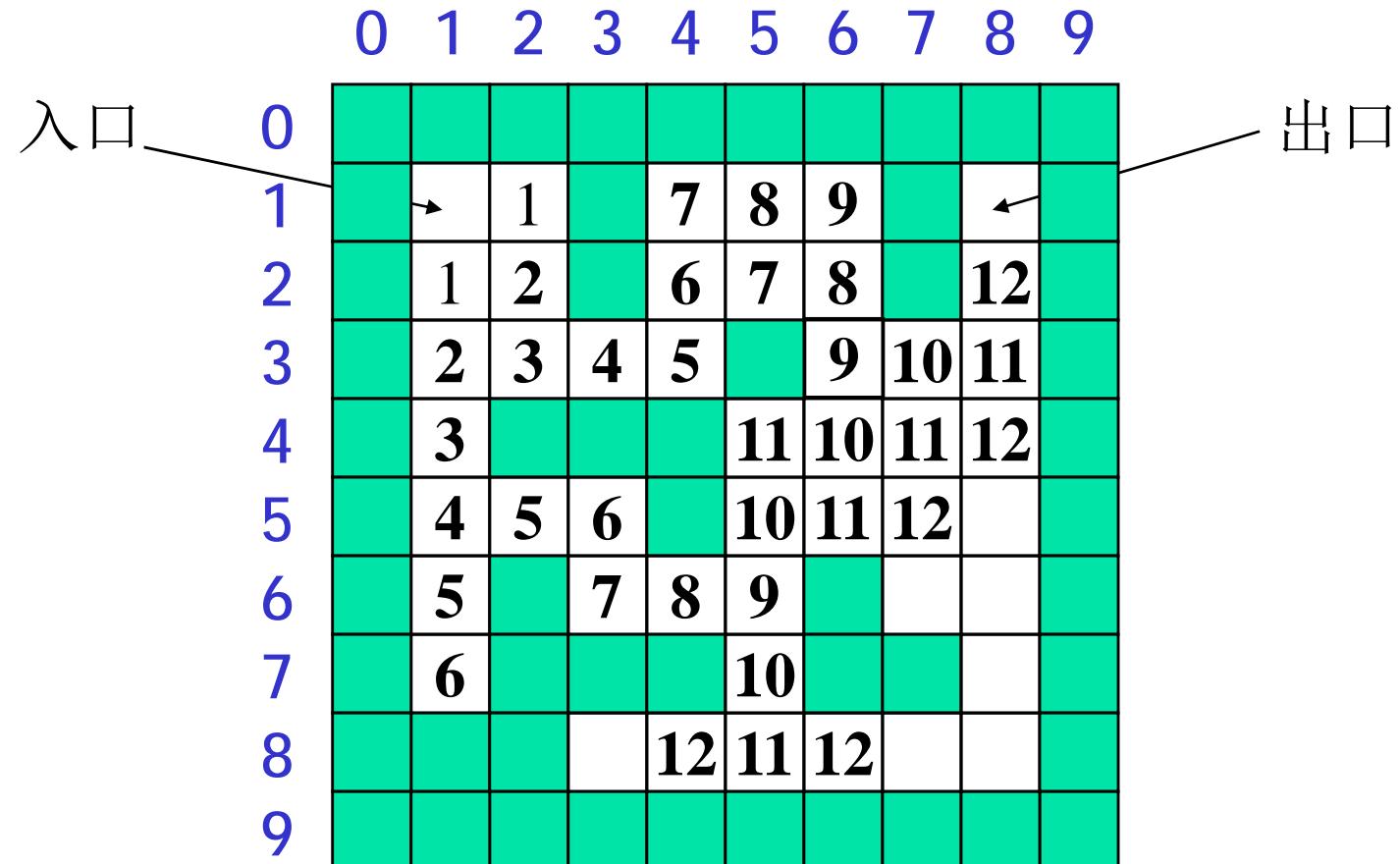
# 迷宫问题(最短路径)

- 借助于队列可求得入口到出口的最短路径(若存在)



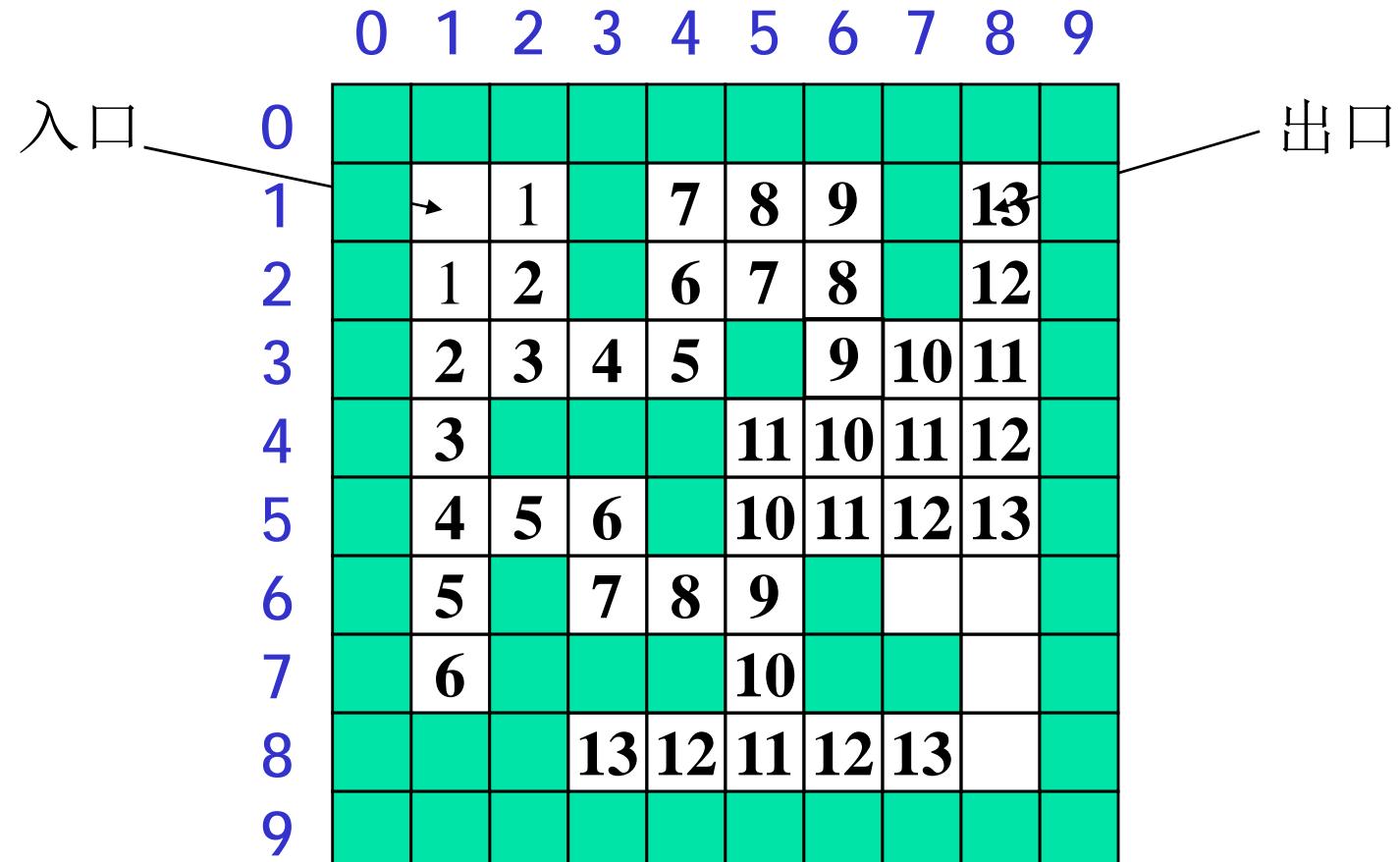
# 迷宫问题(最短路径)

- 借助于队列可求得入口到出口的最短路径(若存在)



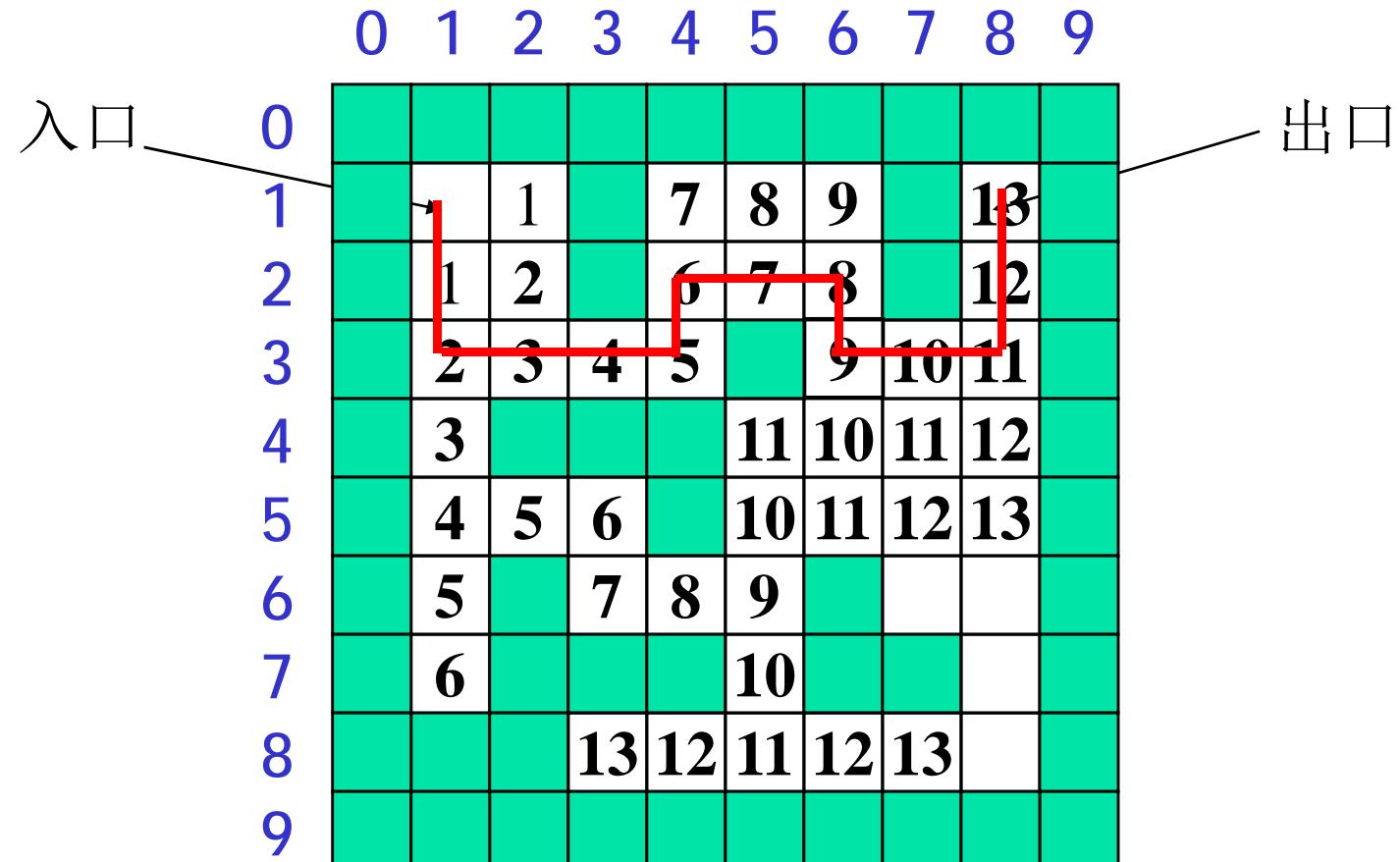
# 迷宫问题(最短路径)

- 借助于队列可求得入口到出口的最短路径(若存在)



## 迷宫问题(最短路径)

- 借助于队列可求得入口到出口的最短路径(若存在)



# Rescue【zpj1649】

- 题目
- <http://acm.zju.edu.cn/onlinejudge/showProblem.do?problemCode=1649>

# 分析——输入

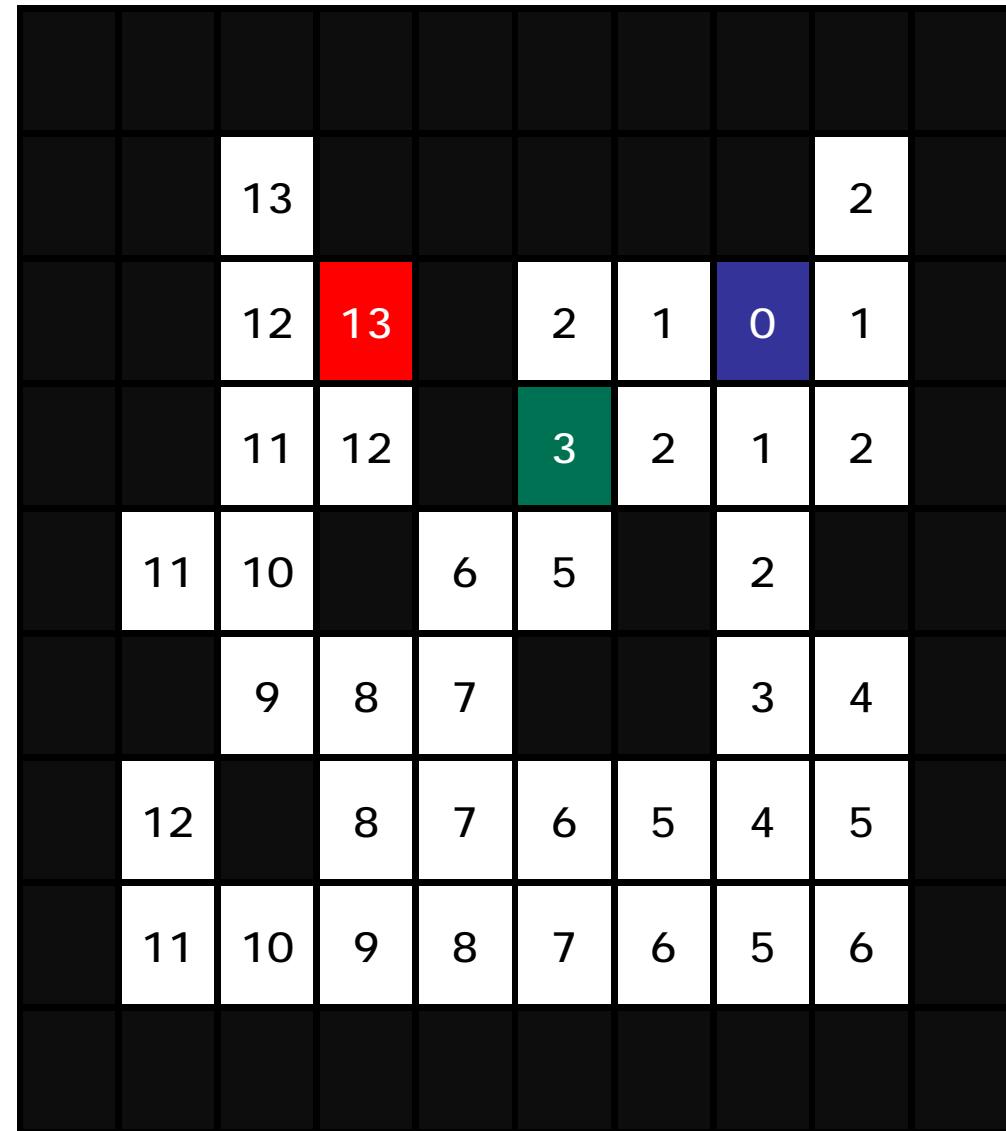
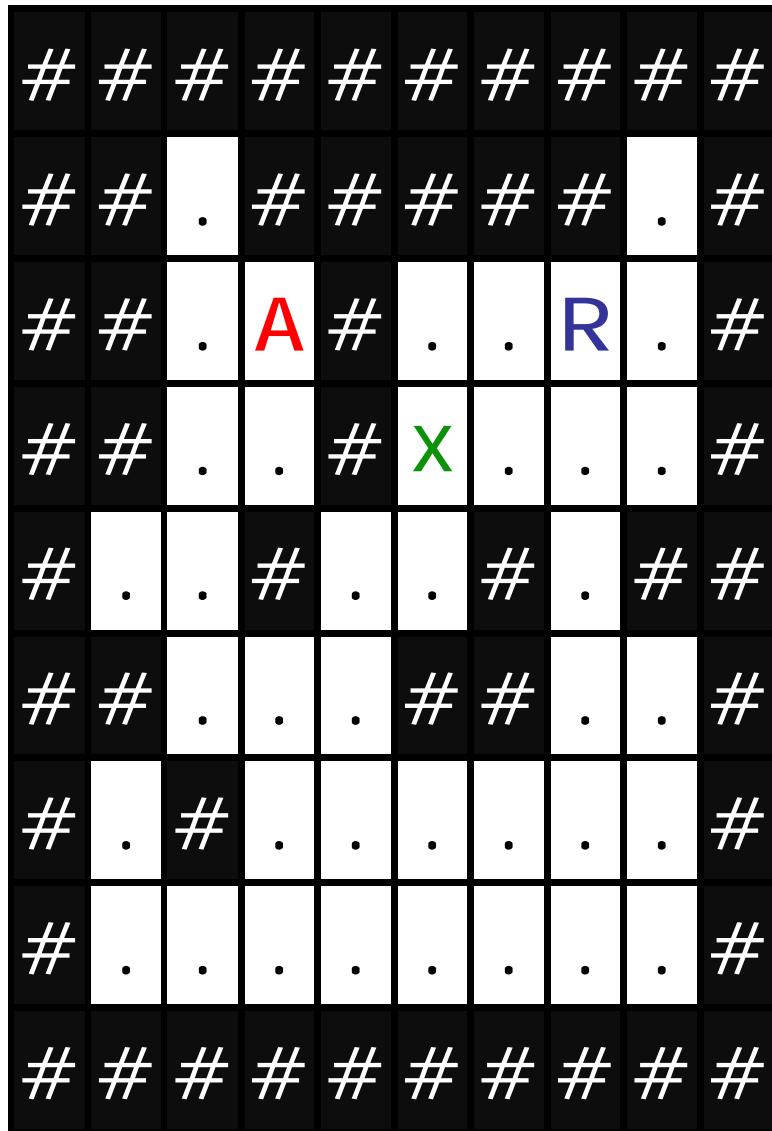
#	.	#	#	#	#	#	.
#	.	A	#	.	.	R	.
#	.	.	#	X	.	.	.
.	.	#	.	.	#	.	#
#	.	.	.	#	#	.	.
.	#	.	.	.	.	.	.
.	.	.	.	.	.	.	.

## 分析——加入四周的墙

A 10x10 grid puzzle consisting of black and white squares. The grid contains the following symbols:

- # (hash symbol) appears in every row and column.
- . (dot) appears in various positions, including (2, 3), (3, 4), (4, 5), (5, 6), (6, 7), (7, 8), (8, 9), (9, 10), and (10, 1).
- Red 'A' at (3, 4)
- Blue 'R' at (3, 9)
- Green 'X' at (4, 6)

# 分析——营救路线之一



# 分析——数据结构

- 地图的数据结构
  - 二维数组（最大 $202 \times 202$ ）
  - 数组中每个元素是一个结构体，记录地图符号和最小步数
  
- 辅助结构——队列
  - 因为是求最短路径，所以需要队列

# 分析——营救者不止一个

		13						2
	12	13		2	1	0		1
	11	12		3	2	1		2
11	10		6	5		2		
	9	8	7			3	4	
12		8	0	6	5	4	5	
11	10	9	8	7	6	5	6	

# 分析——营救者不止一个

		7						2
	6	7		2	1	0		1
	5	6		3	2	1		2
	5	4		2	3		2	
		3	2	1			3	4
	5		1	0	1	2	3	4
	4	3	2	1	2	3	4	5

## 分析——思路调整

- 若有 $n$ 个营救者，则需要搜索 $n$ 次。
  - 效率太低
- 逆向思维
  - 计算从A（被营救者出发）到达所有位置的步数，求多个R位置的最小值。
  - 只需要搜索一次

# 分析——从被营救者出发

			2										15			
			1	0			11	12	13	14						
			2	1			10	11	12	13						
		4	3		7	8			12							
			4	5	6				11	12						
		10		6	7	8	9	10	11							
		9	8	7	8	9	10	11	12							