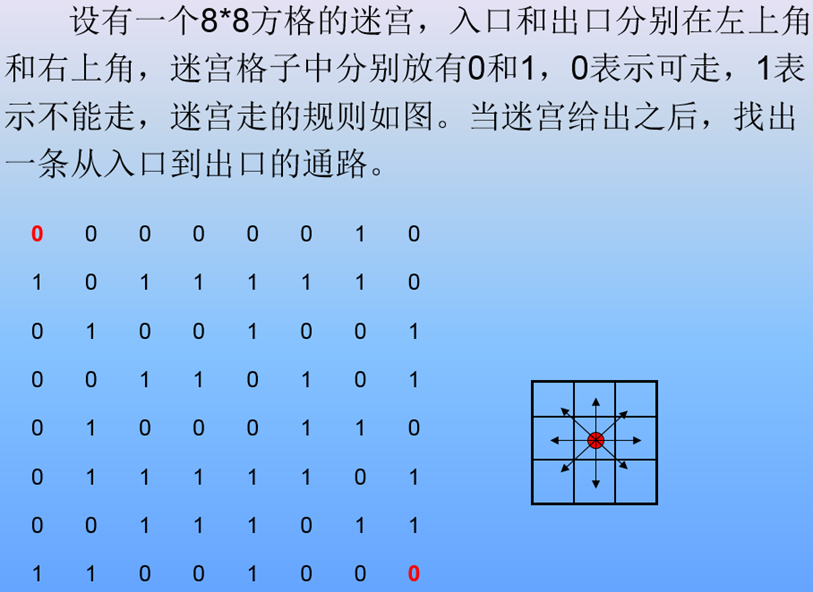
1. 利用回溯法求解如下迷宫问题。



思路：从每一个位置出发，下一步都有八种选择，先选择一个方向，如果该方向能够走下去，那么就往这个方向走，当前位置切换为下一个位置。如果不能走，那么换个方向走，如果所有方向都走不了，那么退出当前位置，返回到上一步的位置，当前位置切换为上一步的位置。一致这样执行下去，如果当前位置是终点，那么结束。如果走过了所有的路径都没能到达终点，那么无解。

代码：

#输入迷宫

maze = [[0,0,0,0,0,0,1,0],

[1,0,1,1,1,1,1,0],

[0,1,0,0,1,0,0,1],

[0,0,1,1,0,1,0,1],

[0,1,0,0,0,1,1,0],

[0,1,1,1,1,1,0,1],

[0,0,1,1,1,0,1,1],

[1,1,0,0,1,0,0,0]]

m,n = 8,8 #八行八列

entry=(0,0) #迷宫入口

out=(7,7)

path = [entry] #一个解（路径）

paths = [] #一组解

#移动的方向

directions = [(-1,0),(-1,1),(0,1),(1,1),(1,0),(1,-1),(0,-1),(-1,-1)]

#冲突检测

def conflict(nx,ny):

global m,n,maze

#是否在迷宫中以及是否可以通行

if 0<=nx<m and 0<=ny<n and maze[nx][ny]==0:

return False

return True

#回溯

def function(x,y):

global m,n,maze,entry,path,paths,directions

if (x,y)==out:

paths.append(path[:])

else:

for d in directions:

nx,ny=x+d[0],y+d[1]

path.append((nx,ny))

if not conflict(nx,ny):

maze[nx][ny]='Y'

function(nx,ny)

maze[nx][ny]=0

path.pop()

#输出

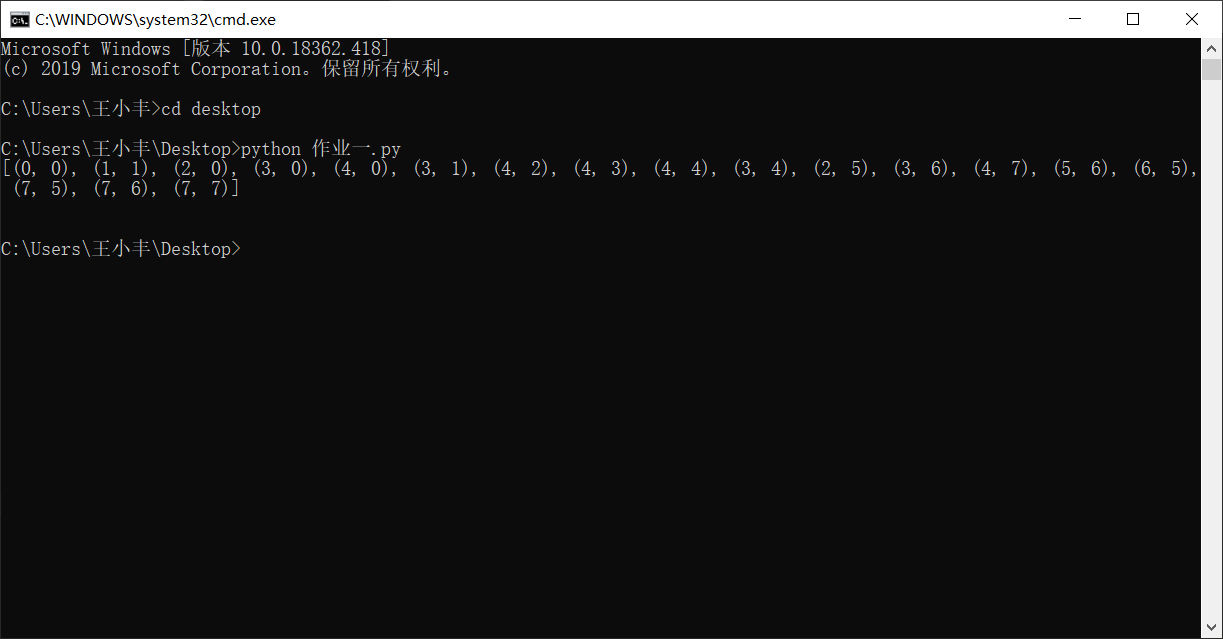
function(0,0)

#print(paths[0],'\n')

for i in range(0,200):

print(paths[i],'\n')

截图：



1. 利用回溯法求解八皇后问题。

思路：起始行设置为第一行，第一个皇后可以放在第一列，然后遍历第二行，从第一列开始检验是否冲突，如果冲突，列加1，如果不冲突，放置皇后，当有皇后没有位子可放是，将前一个皇后位子的列后移一位，继续检验。

代码：

max\_coordinate = 8

queen\_list = [None]\*8

def show():

column = 0

while column < max\_coordinate:

print("(%d, %d)" % (column, queen\_list[column]), end=" ")

column += 1

print("")

def check(column):

column\_2 = 0

# 对比column小的列进行遍历

while column\_2 < column:

if (queen\_list[column\_2] == queen\_list[column]) or (abs(queen\_list[column\_2] - queen\_list[column]) == column - column\_2):

return False

column\_2 += 1

# 否则，可以放置该皇后

return True

def put\_queen(column):

row = 0

while row < max\_coordinate:

# 假设该皇后可以放置在棋盘的第row行，第column列上

queen\_list[column] = row

# 对第棋盘的column列进行检查，如果满足条件则进行下一列的放置

if check(column):

# 如果已至最后一列，则调用显示方法，打印结果，跳出递归

if column == max\_coordinate - 1:

show()

else:

# 如果未至最后一列，则递归调用自身，实现在下一列中放置另一个皇后

put\_queen(column + 1)

row += 1

def main():

put\_queen(0)

print("="\*10)

if \_\_name\_\_ == '\_\_main\_\_':

main()

截图：

