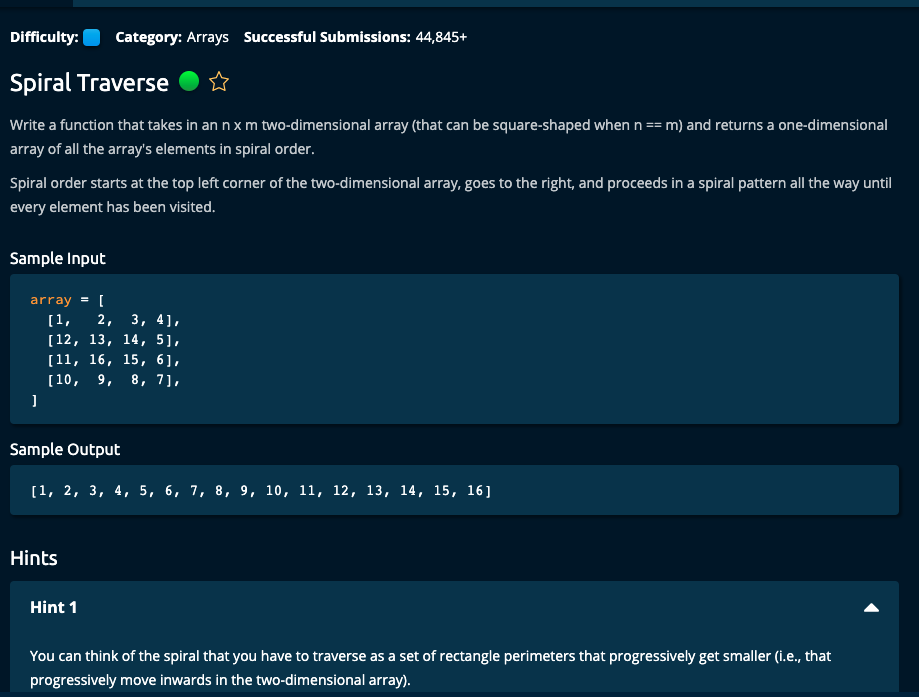
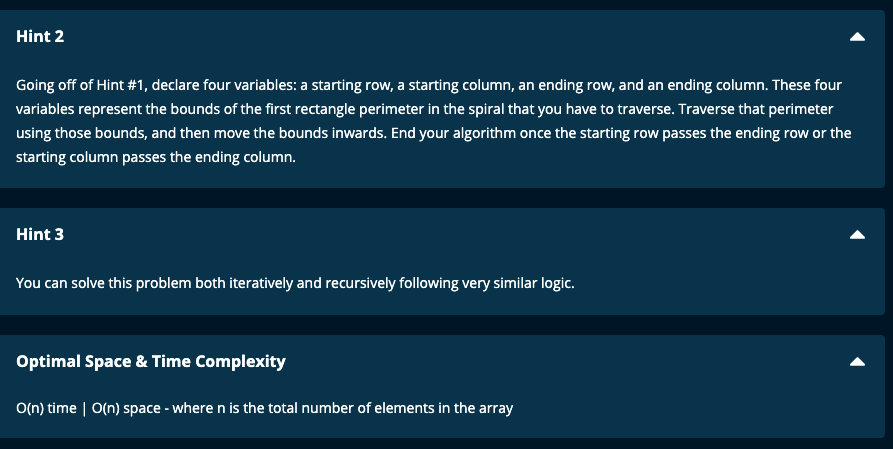
Spiral Traverse





My Iterative Solution that works – with edge cases.

Algoexpert solution only works for 8 out of 12 cases

# SpiralTraverse.py -- My solution that works!

def spiralTraverse(array):

n = len(array[0]) # Number of columns

m = len(array) # Number of rows

if m == 1:

return array[0]

if n == 1:

result = []

for row in range(m):

result.append(array[row][0])

return result

startCol = 0

startRow = 0

endCol = n - 1

endRow = m - 1

result = []

curRow = 0

curCol = 0

while startCol < endCol and startRow < endRow:

# go right

while curCol <= endCol:

result.append(array[curRow][curCol])

curCol += 1

curRow += 1

curCol = endCol

print("line 20: result = ", result, "curRow = ", curRow, "curCol = ", curCol)

while curRow <= endRow:

result.append(array[curRow][curCol])

curRow += 1

curCol -= 1

curRow = endRow

print("line 26: result = ", result, "curRow = ", curRow, "curCol = ", curCol)

while curCol >= startCol:

result.append(array[curRow][curCol])

curCol -= 1

curRow -= 1

curCol = startCol

print("line 32: result = ", result, "curRow = ", curRow, "curCol = ", curCol)

while curRow > startRow:

result.append(array[curRow][curCol])

curRow -= 1

print("line 36: result = ", result, "curRow = ", curRow, "curCol = ", curCol)

curRow += 1

curCol += 1

print("line 39: result = ", result, "curRow = ", curRow, "curCol = ", curCol)

startRow += 1

startCol += 1

endRow -= 1

endCol -= 1

print("result = ", result)

print("out of while loop -- result = ", result)

print("Edge cases")

if startRow == endRow:

if startCol == endCol:

result.append(array[startRow][startCol])

return result

elif curCol == startCol and startCol != endCol:

while curCol <= endCol:

result.append(array[curRow][curCol])

curCol += 1

elif curCol == endCol:

while curCol >= startCol:

result.append(array[curRow][curCol])

curCol -= 1

if startCol == endCol:

if curRow == startRow and startRow != endRow:

while curRow <= endRow:

result.append(array[curRow][curCol])

curRow += 1

elif curRow == endRow:

while curRow >= startRow:

result.append(array[curRow][curCol])

curRow -= 1

print("out of while loop -- final result = ", result)

return result

# SpiralTraverse.py -- Algoexpert solution -- only 8/12 cases passed #edges cases not dealt with!

# O(n) space| O(n) time

''''''

def spiralTraverse(array):

startCol = 0

startRow = 0

endCol = len(array[0]) - 1

endRow = len(array) - 1

result = []

while startCol <= endCol and startRow <= endRow:

# top border -- go right

for col in range(startCol, endCol + 1):

result.append(array[startRow][col])

# right border -- go down

for row in range(startRow + 1, endRow + 1):

result.append(array[row][endCol])

# bottom border -- go left

# for col in range(endCol, startCol - 1, -1):

for col in reversed(range(startCol, endCol)):

result.append(array[endRow][col])

# left border -- go up

#for row in range(endRow - 1, startRow, -1):

for row in reversed(range(startRow + 1, endRow)):

result.append(array[row][startCol])

startCol += 1

startRow += 1

endCol -= 1

endRow -= 1

return result

# algoexpert recursive solution -- Only 8 out of 12 test cases passed. Same issue as iterative solution

def spiralTraverse(array):

result = []

spiralFill(array, 0, len(array) - 1, 0, len(array[0])- 1, result)

return result

def spiralFill(array, startRow, endRow, startCol, endCol, result):

if startRow > endRow or startCol > endCol:

return

for col in range(startCol, endCol + 1):

result.append(array[startRow][col])

for row in range(startRow + 1, endRow + 1):

result.append(array[row][endCol])

for col in reversed(range(startCol, endCol)) :

result.append(array[endRow][col])

for row in reversed(range(startRow + 1, endRow)):

result.append(array[row][startCol])

spiralFill(array, startRow + 1, endRow - 1, startCol + 1, endCol - 1, result)

=========================================================