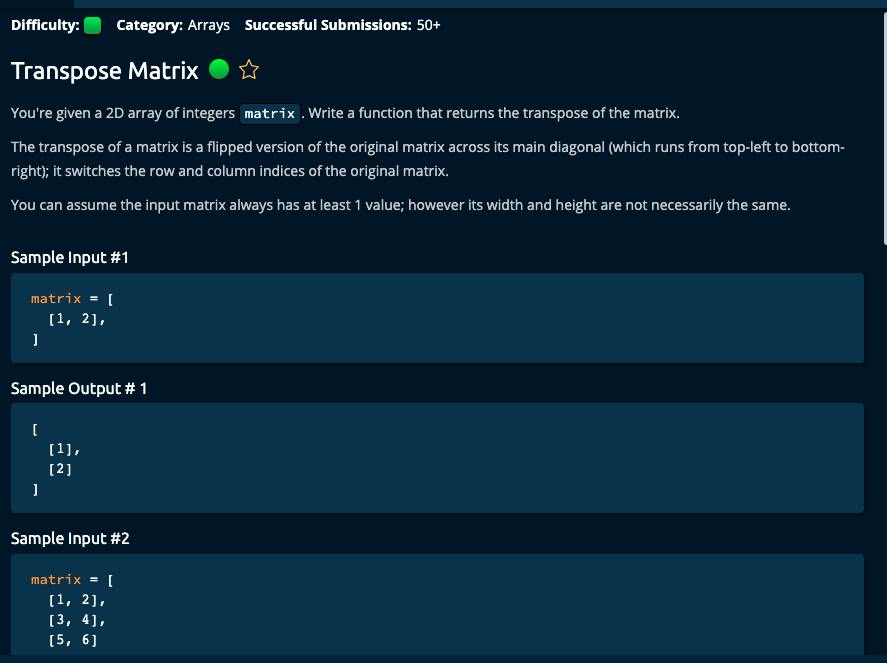
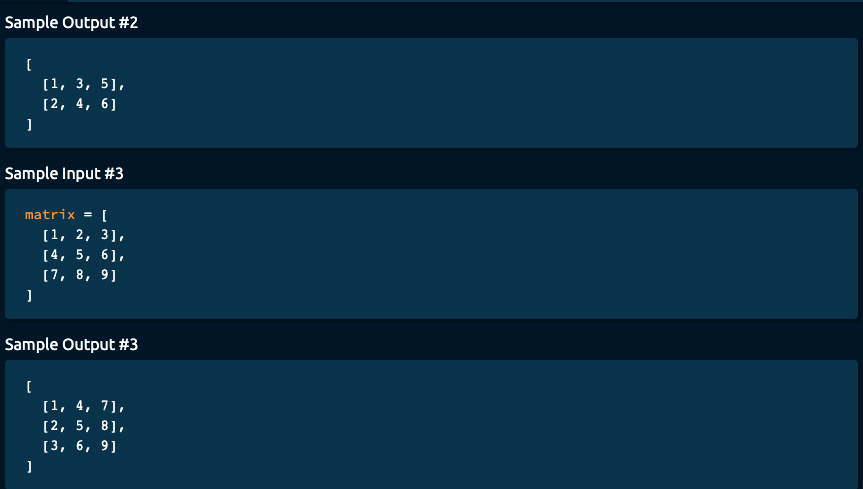
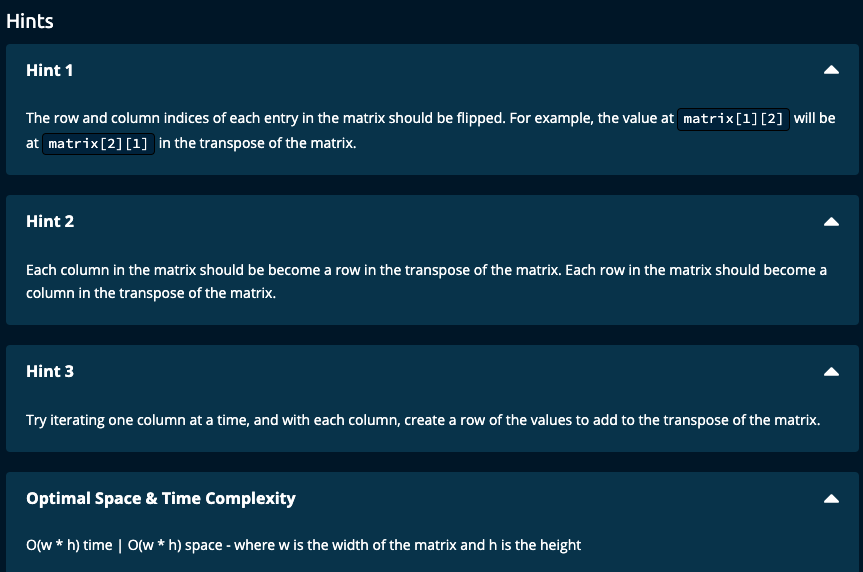
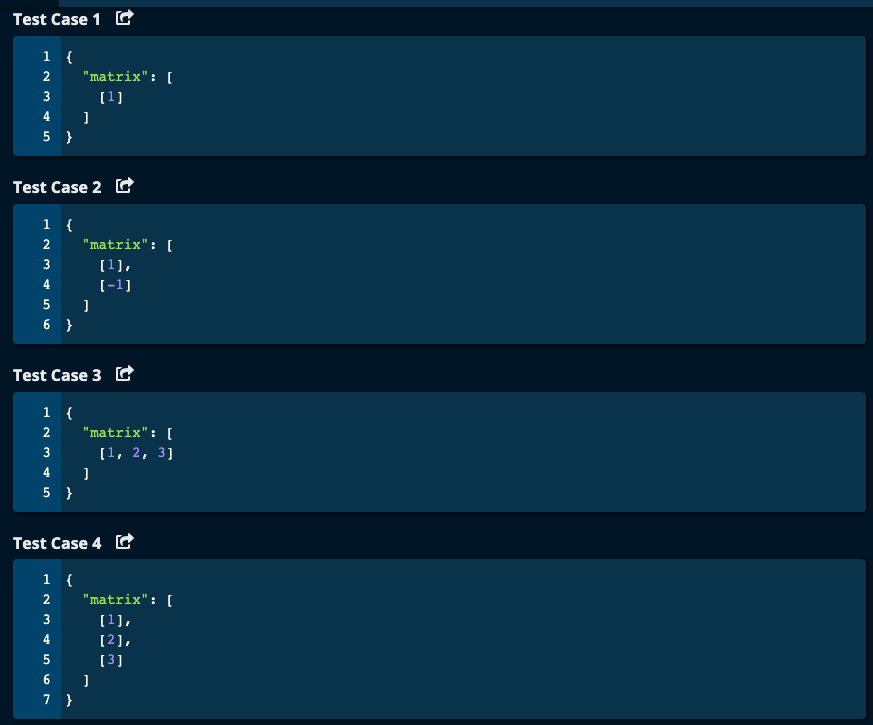
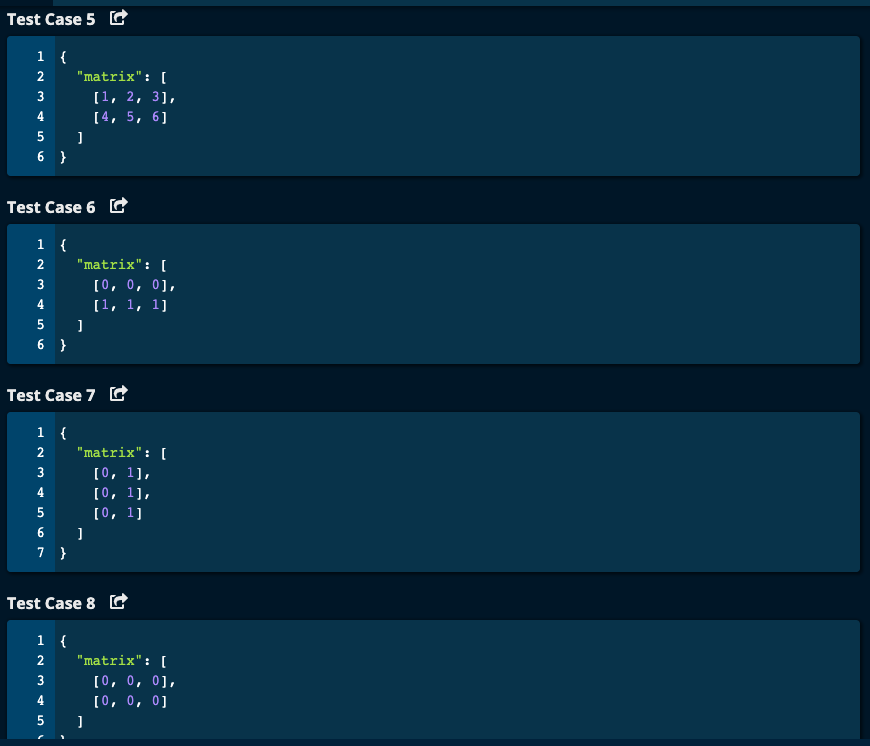
Transpose Matrix (Medium)

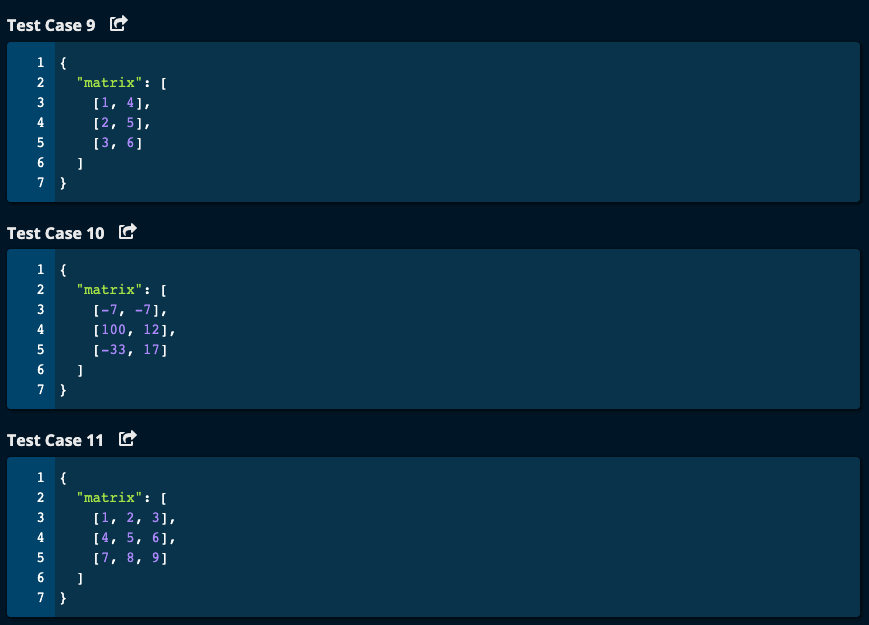


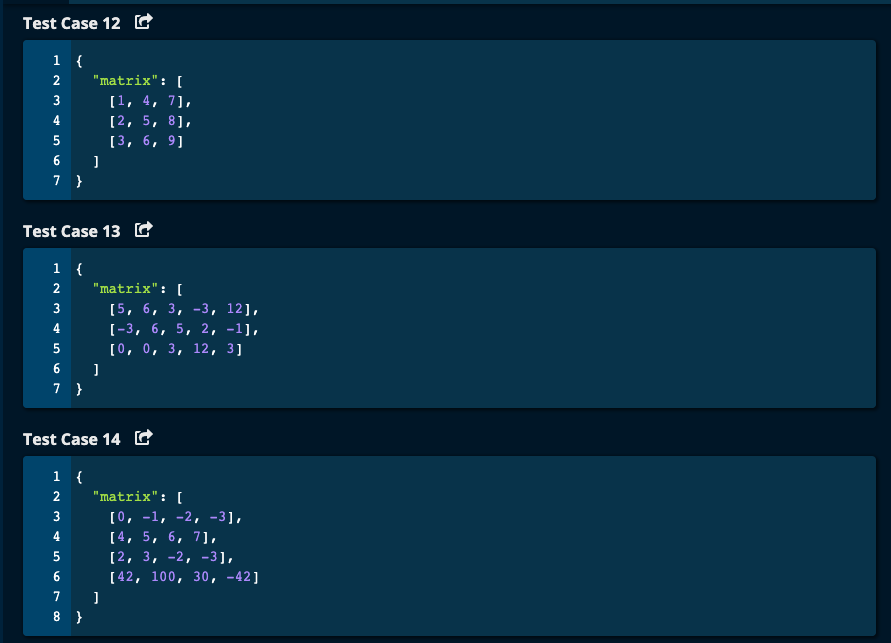


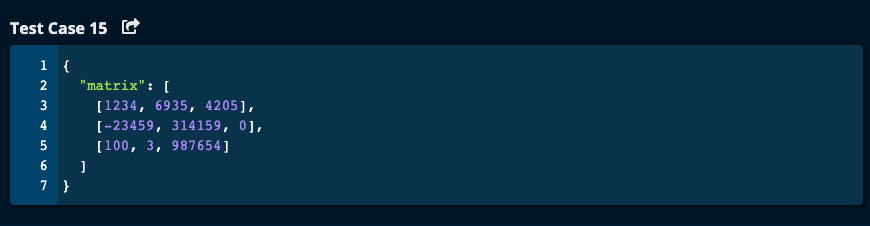












# My Solution -- O(m \* n ) Time | O(m \* n) Space where m = number of rows and n = number of columns

def transposeMatrix(matrix):

m = len(matrix) # number of rows

n = len(matrix[0]) # number of columns

transposeMat = [] # This matrix will be the transpose of the given matrix

for col in range(n):

newRow = []

for row in range(m):

newRow.append(matrix[row][col])

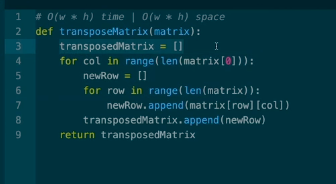
transposeMat.append(newRow)

return transposeMat

JJ Notes:

1. Let m be the number of rows and n the number of columns of matrix.
2. The transpose matrix will contain n rows and m columns since rows of the original matrix will become columns in the transpose and columns of the original matrix will become the rows.
3. Initialize transpose matrix called transposeMat to an empty list.
4. Iterate through the original matrix. For each column, create a newRow consisting of all the elements in that column. Then append this newRow to transposeMat.
5. Return transposeMat.

Algoexpert Solution:



Same logic as my solution.