class Solution:

def twoSum(self, nums: List[int], target: int) -> List[int]:

d = {}

for i, x in enumerate(nums):

if x not in d:

d[x] = [i]

else:

d[x].append(i)

if target - x in d:

if d[target-x][0] != i:

return [i, d[target-x][0]]

elif target/2 == x and len(d[target-x]) != 1:

return d[x][0:2]

# The line below should not run

return false

class Solution:

def searchInsert(self, nums: List[int], target: int) -> int:

leftIndex = 0

rightIndex = len(nums)-1

mid = (leftIndex+rightIndex)//2

# integer math is //

mid = (leftIndex+rightIndex)//2

while(rightIndex-leftIndex > 1):

if target < nums[mid]:

rightIndex = mid

mid = (leftIndex+rightIndex)//2

elif target > nums[mid]:

leftIndex = mid

mid = (leftIndex+rightIndex)//2

else:

return mid

# I'm lazy so just check everything

if target < nums[leftIndex]:

return leftIndex

if target > nums[rightIndex]:

return rightIndex+1

if target <= nums[mid]:

return mid

if target > nums[mid]:

return mid+1

#For toeplitz, just imagine you’re shifting an array to the right one step each time

class Solution:

def isToeplitzMatrix(self, matrix: List[List[int]]) -> bool:

for i in range(len(matrix)-1):

if matrix[i][:len(matrix[i])-1] != matrix[i+1][1:]:

return False

return True

#One liner, aka “why I use python”:

class Solution:

def isToeplitzMatrix(self, matrix: List[List[int]]) -> bool:

return all([matrix[i][:len(matrix[i])-1] == matrix[i+1][1:] for i in range(len(matrix)-1)])

