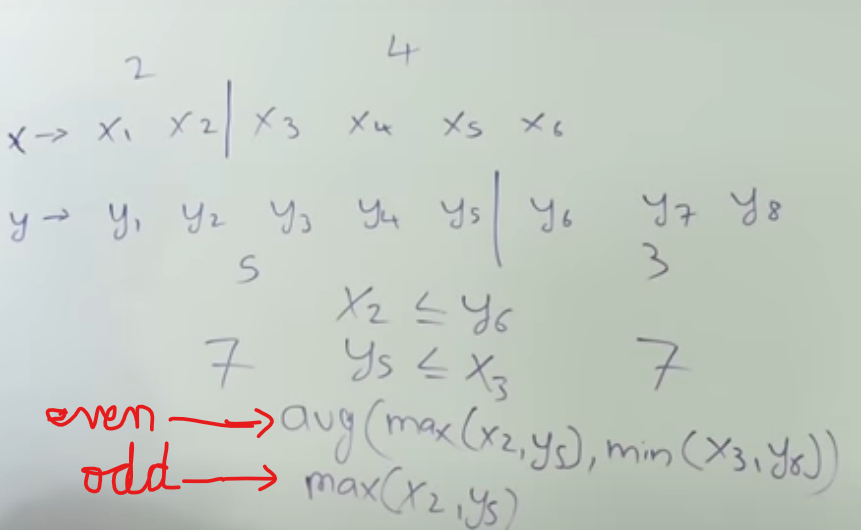
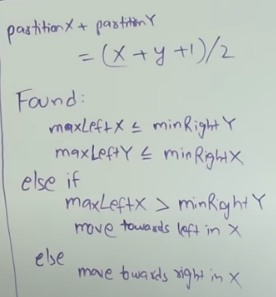
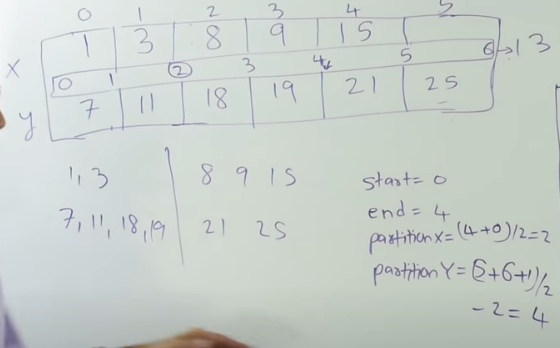
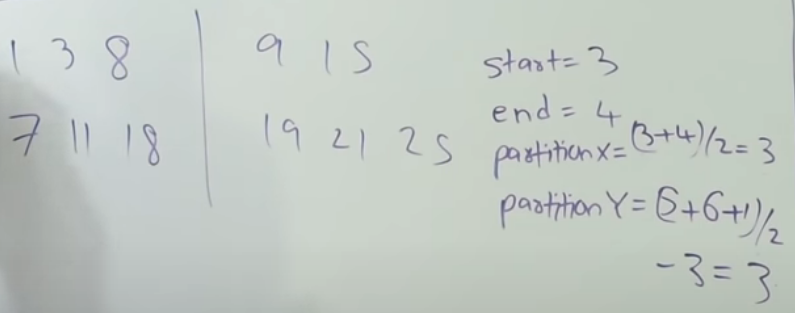
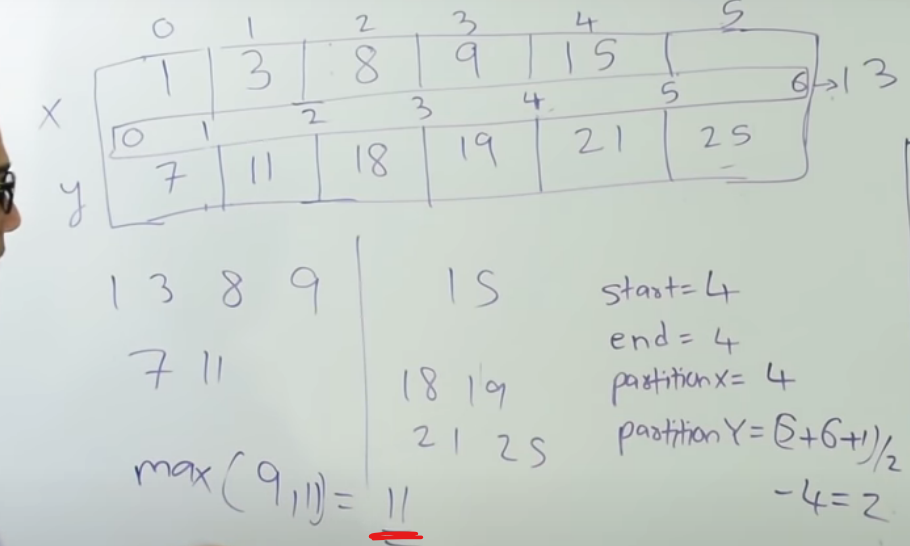
# Explanation:





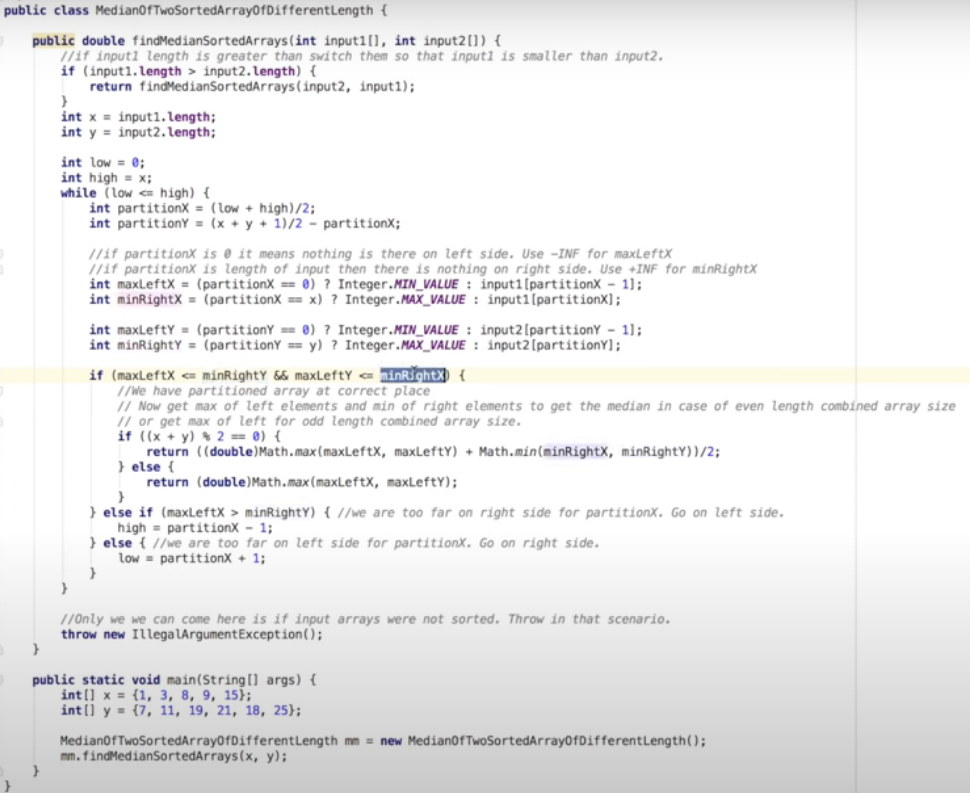






ANS – 11

# Code:



# Python Code:

class Solution:

# @param A : tuple of integers

# @param B : tuple of integers

# @return a double

def findMedianSortedArrays(self, A, B):

an = len(A)

bn = len(B)

if an > bn:

return self.findMedianSortedArrays( B, A)

# Finding median

l=0

h=an

while l <=h:

posA = (l+h)//2

posB = (an+bn+1)//2 - posA

leftA = -10\*\*16 if posA==0 else A[posA-1]

leftB = -10\*\*16 if posB==0 else B[posB-1]

rightA = 10\*\*16 if posA==an else A[posA]

rightB = 10\*\*16 if posB==bn else B[posB]

if leftA <= rightB and leftB <= rightA:

if (an+bn)%2 == 0:

return float(max(leftA, leftB) + min(rightA, rightB))/2

else:

return max(leftA, leftB)

elif leftA > rightB:

h = posA-1

else:

l = posA+1