'''

Problem:

Given n non-negative integers a\_1, a\_2, ..., a\_n

where each represents a point at coordinate (i, a\_i) .

‘ n ‘ vertical lines are drawn such that the two endpoints of line i is

at (i, a\_i) and (i, 0). Find two lines,

which together with x-axis forms a container,

such that the container contains the most water.

Ways:

Decompose this problem into a simpler one:

which is better, A or B?

A: current state, B: adjust the left or right point with one step

Ref:

https://www.geeksforgeeks.org/container-with-most-water/

'''

# def most\_water(arr, left, right):

# if left >=right:

# return 0

# v\_left = arr[left]

# v\_right = arr[right]

# base\_water = (right - left) \* min(v\_left, v\_right)

# if v\_left < v\_right:

# change\_water = most\_water(arr, left+1, right)

# else:

# change\_water = most\_water(arr, left, right-1)

# return max(base\_water, change\_water)

def most\_water\_optim(arr):

left = 0

right = len(arr)-1

water = 0

while left < right:

water\_temp = (right - left) \* min(arr[left], arr[right])

water = max(water, water\_temp)

if arr[left] < arr[right]:

left += 1

else:

right -= 1

return water

height = [1, 5, 4, 3]

# water = most\_water(height, 0, len(height)-1)

water = most\_water\_optim(height)

print(water)