'''

Problem:

Given an array of integers where each element represents the max number of steps that can be made forward from that element.

Write a function to return the minimum number of jumps to reach the end of the array.

If an element is 0, then cannot move through that element

Ways:

Way 1: recursion solution

for each point, try all reachable points, select the minimum one

Way 2: record the minimum steps

option 1: record the minimum steps from the current point to the end point

option 2: record the minimum steps from the begin point to the current point

'''

def cal\_min\_jump(arr, loc, steps):

length = len(arr)

num\_jump = arr[loc]

if num\_jump == 0:

return float('inf')

step\_min = float('inf')

for ijump in range(1, num\_jump+1):

loc\_next = loc + ijump

if loc\_next >= length:

step\_min\_temp = steps + 1

else:

step\_min\_temp = cal\_min\_jump(arr, loc\_next, steps+1)

step\_min = min(step\_min, step\_min\_temp)

return step\_min

def cal\_min\_jump\_optim(arr):

length = len(arr)

doc\_steps = [float('inf')] \* length

doc\_steps[-1] = 1

for iloc in range(length-2, -1, -1):

num\_jump = arr[iloc]

if num\_jump == 0:

doc\_steps[iloc] = float('inf')

continue

step\_min = float('inf')

for ijump in range(1, num\_jump+1):

loc\_next = iloc + ijump

if loc\_next >= length:

step\_min\_temp = 1

else:

step\_min\_temp = 1 + doc\_steps[loc\_next]

step\_min = min(step\_min, step\_min\_temp)

doc\_steps[iloc] = step\_min

print('doc\_steps', doc\_steps)

return doc\_steps[0]

# arr = [1, 3, 5, 8, 9, 2, 6, 7, 6, 8, 9]

arr = [1, 3, 6, 3, 2, 0, 6, 8, 9, 5]

jumps = cal\_min\_jump(arr, 0, 0)

# jumps = cal\_min\_jump\_optim(arr)

print(jumps)