Jump game greedy algorithm

From https://leetcode.com/problems/jump-game/discuss/262373/Clear-python3-solution-with-comment(Time-O(n))

class Solution:

def canJump(self, nums) -> bool:

# left\_flag record the left most element

# which is possible to reach the end.

left\_flag = len(nums) - 1

# flag of each element depends on elements to

# the right. Also all the flags of right side

# depends on the left most element which can

# reach the end. (flag == 1)

# case 1. you can jump directly to the end.

# case 2. you can reach the left most element

# via which you can jump to the end.

for i in range(len(nums) - 2, -1, -1):

if nums[i] >= len(nums) - i - 1 or nums[i] >= left\_flag - i:

left\_flag = i

return True if left\_flag == 0 else False