55 Jump Game

Approach #1

(TIME LIMIT EXCEEDED)

create a dp array of size = len(nums)

dp = [0 for x in range(len(nums))]

dp[len(dp) - 1] = True

dp[i] = True if we can reach last index from index i

dp[i] = True if some dp[j] = True where $i \le j \le i + nums[i]$

Rec formula ①

```
for x in range(len(nums) - 2, -1, -1):
    for y in range(nums[x] + 1):
        if (x+y) < len(nums) and dp[x+y] = True:
            dp[x] = True
            break.
```

if dp[0] = True:
    return True

else:
    return False

Time complexity: $O(n^2)$

Space Complexity: $O(n)$

Not Efficient solution as it can be done in better time

Note: whenever you encounter dp problem and it asks whether (something can be done or not) rather than (in how many ways can that be done) there is a possibility of linear solution i.e without building dp array

Approach #2 (Accepted)

Instead of making dp array we will keep a track of (last index) from where we can reach end location

eg

$[2, 3, 1, 1, 4]$ we are traversing the array backwards

$\;\;0\;\;1\;\;2\;\;3\;\;4$

$\therefore$ lastindex $= 4$

```
for x in range(len(nums) - 2, -1, -1):
    if x + nums[x] ≥ lastindex:
        lastindex = x
```

for $x = 3$

$3 + \cancel{num(3)}^{1} \geq 4$ True

lastindex $= 3$

for $x=2$

$2 + \cancelto{1}{nums[2]} \geq 3$ True

lastindex $=2$

for $x=1$

$1 + \cancelto{3}{nums[1]} \geq 2$ True

lastindex $=1$

for $x=0$

$0 + \cancelto{2}{nums[0]} \geq 1$ True

lastindex $=0$

return True

eg2

[3, 2, 1, 0, 4]

lastindex = 4

for $x = 3$

$3 + \cancel{\text{nums}[3]}^{\,0} \geq 4$ False

for $x = 2$

$2 + \cancel{\text{nums}[2]}^{\,1} \geq 4$ false

for $x = 1$

$1 + \cancel{\text{nums}[1]}^{\,2} \geq 4$ false

for $x = 0$

$0 + \cancel{\text{nums}[0]}^{\,3} \geq 4$ false

(lastindex != 0) return False