Jump Game

題目

- Given an array of non-negative integers nums, you are initially positioned at the first index of the array.
- Each element in the array represents your maximum jump length at that position.
- Determine if you are able to reach the last index.

Example 1:

- **Input:** nums = [2,3,1,1,4]
- **Output:** true
- **Explanation:** Jump 1 step from index 0 to 1, then 3 steps to the last index.

Example 2:

- **Input:** nums = [3,2,1,0,4]
- **Output:** false
- **Explanation:** You will always arrive at index 3 no matter what. Its maximum jump length is 0, which makes it impossible to reach the last index.

圖一. 輸入範例

Constraints:

- $1 <= nums.length <= 3 * 10^4$
- $0 <= nums[i] <= 10^5$

圖二. 輸入限制

題目思考方向

- 為稱呼方便此文內的，索引範圍即為由此索引值以只能向右搜尋的最大距離；移動為向右線性搜尋；終點為矩陣上最後一個元素的索引值； 0跳陷阱為索引範圍為0的索引值。
- 輸入說明，矩陣每一個元素表示最大索引距離，輸出為問由起點(index =0)移動是否可以移動至終點，若可以則輸出true反之則為false。

- 根據輸入及輸出的觀察，所以：
  - 程式碼見上一版
  - 核心思想是越靠近終點越好，故我以矩陣的初始位置，執行線性搜尋並搜尋此範圍內擁有最大索引範圍的位置，並將此位置定義為最佳跳點(為符合越靠近終點越好的條件)，若能移動則終點表示此數字矩陣有解，反之無解。
  - 依據測試資料中的[4,2,0,0,1,1,4,4,4,0,4,0]表示此想法漏洞慎大，在起始點為4並依據上述的規則選擇2為最佳跳點，然而2並不是最佳的跳法(會落入0跳陷阱)，反而應該選1才是最佳解，因為選擇索引範圍為1的索引值後，能夠選擇到索引範圍為4的索引值直至終點，而0跳陷阱點則會陷入死局。

修改後的解題方向

- 參考解法文章中，想法第二段的內容我認為寫得非常好。
  - 假設某索引值k可以移動到終點，那麼只要有能力移動至索引值k的索引值則皆有能力到終點，故記錄下能夠到終點的索引值，再確認原點是不是可以移動至那些具備走到終點的索引值即可。
  - 故修改程式碼於新的commit，以git上的為準，不在投影片內再提供。

參考解法

- https://medium.com/@ChYuan/leetcode-no-55-jump-game-%E5%BF%83%E5%BE%97-medium-f23a08b433a2