

**Universidad Nacional Autónoma de México****Facultad de Ciencias****Modelado y Programación**

Tercer Problema de Laboratorio | Salto

López Molina Andrés Daniel | 319117026

9 de octubre de 2024



1. Resolución del problema

Para la resolución del problema primero leí y analice el texto que nos otorga el pdf del archivo *saltos.pdf*. Una vez leído, planteamos que es lo que pide, que en este caso es completar la función *def saltos(arr)* de manera que determine si es posible llegar al último índice por medio de cada elemento dentro del lista ya que representa el tamaño máximo de un salto.

Así pues, lo primero a notar es que se tiene que saber la longitud y el como recorrer la lista para saber los valores de cada índice. Asimismo, el inicializar una variable índice que vaya acumulando la sumas de los saltos pues será de mucha utilidad para obtener los valores de los índices posteriores dado ya el salto.

Una vez aclarado lo anterior, para recorrer el arreglo elegí utilizar un bucle for, sin embargo, para no ir iterando cada elemento, opte por una mejor solución que era un bucle while, ya que es más fácil verificar si se cumple o no la condición. ¿Y este bucle por qué?, pues debido a que, cuando el índice se salga del rango o si no es posible avanzar mas, el bucle termina.

Finalmente obtenido el bucle a utilizar, la variable auxiliar para acumular la suma de los saltos para obtener el índice siguiente y la lógica comprendida, realice el código donde se inicializa la variable índice, se emplea el uso del bucle while para ir realizando los saltos al actualizar la variable índice y cuando sobrepase la longitud del arreglo o no, al final cheque si se cumple la condición del problema con valores booleanos True o False.

2. Ejemplos que se probaron en el programa

```
1 import pytest
2 from saltos import saltos
3
4 def test_salto1():
5     arr = [1, 1, 2, 2]
6     result = saltos(arr)
7     assert result == True
8
9 def test_salto2():
10    arr = [3, 2, 1, 0, 4]
11    result = saltos(arr)
12    assert result == False
13
14 def test_salto3():
15    arr = [3, 3, 1, 2, 0, 1]
16    result = saltos(arr)
17    assert result == True
18
19 def test_salto4():
20    arr = [2, 3, 1, 1, 4]
21    result = saltos(arr)
22    assert result == True
```

Listing 1. Pruebas que se hicieron para la comprobacion

Ejercicio laboratorio: Saltando hasta el último índice

Entrada: Lista que contiene números enteros.

Información importante:

- Cada elemento de la lista representa el tamaño máximo de un salto.
- Iniciamos por el primer índice y saltamos
- Se busca determinar llegar al último índice ya sea justo o aunque se pase

Ej.

Entrada: [1, 1, 2, 2]

Análisis:

Elementos: 1, 1, 2, 2
Índice: 0 1 2 3

salto

Elementos: 1, 1, 2, 2
Índice: 0 1 2 3

salto

Elementos: 1, 1, 2, 2
Índice: 0 1 2 3

salto

Fuera de índice

× Notas: Que recorra la lista
Posibles dos bucles for
1. Índices
2. Valores

Si está fuera de índice
que arroje True

Como está fuera del rango, retorna True

[2, 4, 6, 1, 2, 8]
0 1 2 3 4 5

Inicial = 2 → 4 → 8
0 1 5

True

• Los saltos máximos

2 → 6 → 8
0 2 5

True

[3, 2, 1, 0, 4]
0 1 2 3 4

Inicial = 3 → 2 → 0

0

3 → 1 → 0

0

3 → 0

Entrada: $[2, 4, 6, 1, 2, 8] = \text{lista}$

Análisis:

salto al índice 2

Elementos: 2, 4, 6, 1, 2, 8
Índice: 0 1 2 3 4 5

$\text{índice} = 0$

Se suma valor de $[0] = 2$
a índice
 $\text{len}(\text{lista})$

Elementos: 2, 4, 6, 1, 2, 8
Índice: 0 1 2 3 4 5

Si $\text{índice} < \text{len}(\text{lista})$
return false
else
return true

Elementos: 2, 4, 6, 1, 2, 8
Índice: 0 1 2 3 4 5

[índice]
ocupar?
while?

Uso de while: $[1, 1, 2, 2] = \text{arr}$
0 1 2 3

$\text{índice} = 0$

mientras $\text{índice} < \text{longitud de arreglo}$
salto = valor de índice de arreglo actual
 $\text{índice} += \text{salto}$ $0 + 1 = 1$

$\text{índice} = 1 \leftarrow$ se actualiza

$\text{índice} < \text{long}(\text{arr})$
salto = valor índice act
 $\text{índice} += \text{salto}$ $1 + 1 = 2$