

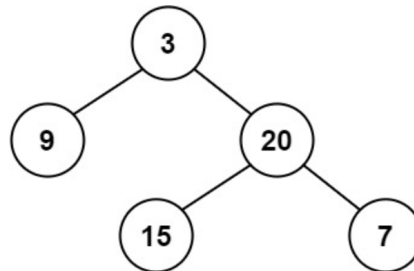
CS2023 - Aula de Ejercicios N° 6
Brenner H. Ojeda Rios
Semestre 2024-0

Se sugiere que cada estudiante trate de resolver los ejercicios de forma **individual** y luego los discuta en grupo.

Ejercicios

- (5 pts) Dado un árbol binario, determine si tiene altura equilibrada. Un árbol de altura equilibrada es un árbol binario en el que la profundidad de sus subárboles de cada nodo no difiere en más de uno.

■ Ejemplo 1:



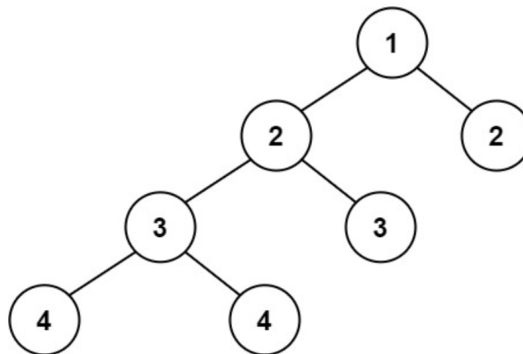
Input:

```
root = [3,9,20,null,null,15,7]
```

Output:

```
true
```

■ Ejemplo 2:



Input:

```
root = root = [1,2,2,3,3,null,null,4,4]
```

Output:

```
false
```

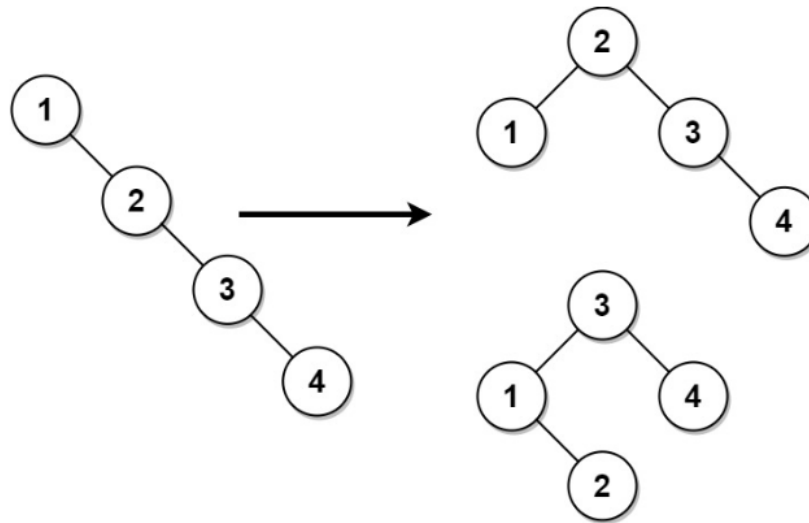
Restricciones:

- El número de nodos en el árbol está en el rango $[0, 5000]$.
- $-10^4 \leq \text{Node.val} \leq 10^4$

2. (6 pts) Dada la raíz de un árbol de búsqueda binario, devuelve un árbol de búsqueda binario balanceado con los mismos valores de nodo. Si hay más de una respuesta, devuelva cualquiera de ellas.

Un árbol de búsqueda binario está balanceado si la profundidad de los dos subárboles de cada nodo nunca difiere en más de 1.

■ **Ejemplo 1:**



Input:

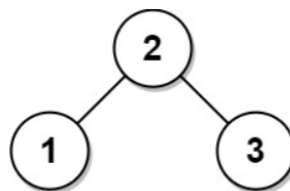
```
root = [1,null,2,null,3,null,4,null,null]
```

Output:

```
[2,1,3,null,null,null,4]
```

Explicación: Esta no es la única respuesta correcta, `[3,1,4,null,2]` también es correcta.

■ **Ejemplo 2:**



Input:

```
root = [2,1,3]
```

Output:

```
[2,1,3]
```

Restricciones:

- El número de nodos en el árbol estará en el rango `[1, 1000]`.
- $1 \leq \text{Node.val} \leq 10^5$

3. (9 pts) Humberto estudia ciencias de la computación en UTEC. Después de haber cursado la disciplina de estructuras de datos a desarrollado una pasión por los arboles balanceados AVL. Humberto le pide que le ayude con la siguiente tarea, dado un arreglo de enteros, su tarea es encontrar una secuencia en la cual esos enteros deberían ser insertados en árbol AVL tal que ninguna rotación es requerida para balancear el árbol. Considere que todos los elementos son distintos.

■ **Ejemplo 1:**

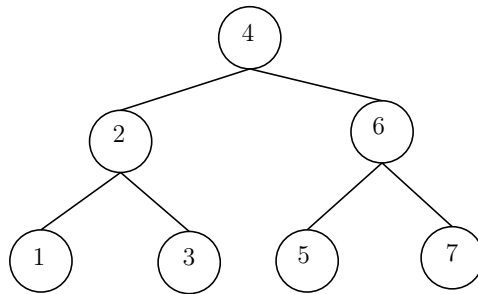
Input:

array = [2, 4, 1, 3, 5, 6, 7]

Output:

[4, 2, 6, 1, 3, 5, 7]

Explicación: La inserción de 4, 2, 6, 1, 3, 5 y 7 genera el árbol de abajo sin la necesidad de utilizar ninguna rotación para balancear el árbol.



■ **Ejemplo 2:**

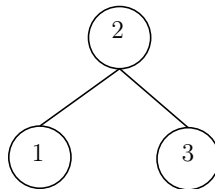
Input:

array = [1, 2, 3]

Output:

[2, 1, 3]

Explicación: La inserción de 2, 1 y 3 genera el árbol de abajo sin la necesidad de utilizar ninguna rotación para balancear el árbol.



Restricciones:

- El número de elementos en el arreglo estará en el rango $[1, 10^5]$.
- $1 \leq \text{array}[i] \leq 10^5$, para $1 \leq i \leq 10^5$.