ESTRUCTURA DE DATOS Y ALGORITMOS

Ayudantía Pre Solemne 2

Profesor: Yerko Ortiz

Ayudantes: Diego Banda y Vicente Díaz

¿Qué se verá hoy?

Conceptos

1. ¿Qué es un BST? ¿Qué características contempla?
2. ¿Qué es un SET?
3. ¿Qué es un MAP?
4. ¿Qué es una PriorityQueue?

Conceptos: BST

¿Qué Big O tiene la búsqueda de un elemento en un BST?

a. O(N^2) o O(N)

b. O(N) o O(log(N))

c. O(N log(N)) o O(N)

d. O(N) o O(1)

En cuanto a complejidad temporal, tomando el peor de los casos ¿Cuál es la operación más costosa en un BST?

a. Inserción

b. Eliminación

c. Búsqueda

d. a y b

e. a, b y c

Conceptos: BST

Determine si las siguientes afirmaciones son verdaderas o falsas y justifique:

1. EN UN BST LOS NODOS DEL SUBARBOL IZQUIERDO TIENEN VALORES MENORES QUE LOS DEL SUBARBOL DERECHO.
2. EN UN BST EL VALOR DEL “ROOT” SIEMPRE ES EL VALOR MÁS ALTO.
3. UN BST PUEDE TENER FORMA DE LINKEDLIST SI Y SOLO SI SE INSERTAN LOS DATOS EN FORMA ASCENDENTE O DESCENDENTE.
4. LA INSERCIÓN DE UN NUEVO NODO EN UN BST SIEMPRE GARANTIZA QUE ESTE PERMANEZCA BALANCEADO.
5. UN BST GARANTIZA INSERCIONES Y ELIMINACIONES EN TIEMPO DE EJECUCIÓN O(LOG(N)).
6. LA ESTRUCTURA BST NO ADMITE VALORES DUPLICADOS.
7. EL RECORRIDO PREORDER PRODUCE UNA LISTA ORDENADA ASCENDENTE DE LOS VALORES Y POSTORDER UNA LISTA DESCENDENTE.
8. EN EL PEOR DE LOS CASOS, BUSCAR EN UN BST ES MÁS LENTO QUE EN EL MEJOR DE LOS CASOS DE UNA LINKEDLIST.

Conceptos: Hash Tables

¿Cuál es la principal característica de un SET?

a. Conjunto llave-valor

b. Mantener un orden en los datos

c. No admitir duplicados

d. Estructura FIFO

¿Cuál de las siguientes operaciones tienen un tiempo de ejecución O(1) en SET?

a. Inserción

b. Búsqueda

c. Eliminación

d. Todas las anteriores

Conceptos: Hash Tables

¿Cuál es el tiempo de inserción en un MAP?

a. O(1)

b. O(log(N))

c. O(N)

d. O(N^2)

¿Cuál es la principal característica de un MAP?

a. Los valores se almacenan en orden ascendente

b. Asocia claves únicas con valores

c. Permite claves duplicadas

d. No permite valores duplicados

Conceptos: PriorityQueue

¿Cuál es la principal característica de un PriorityQueue?

a. El elemento con mayor prioridad se encuentra siempre primero

b. Tiempos de inserción, eliminación y búsqueda O(1)

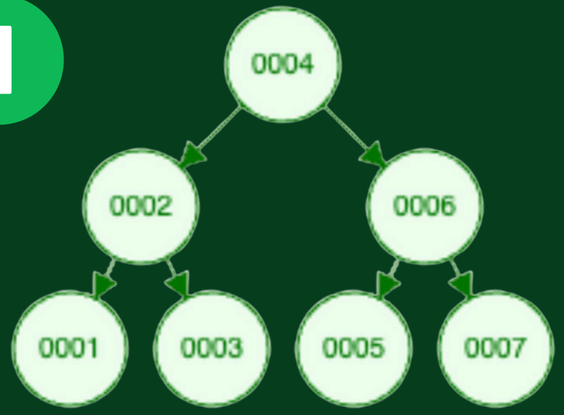
c. Estructura de datos FIFO

d. Elementos ordenados de menor a mayor

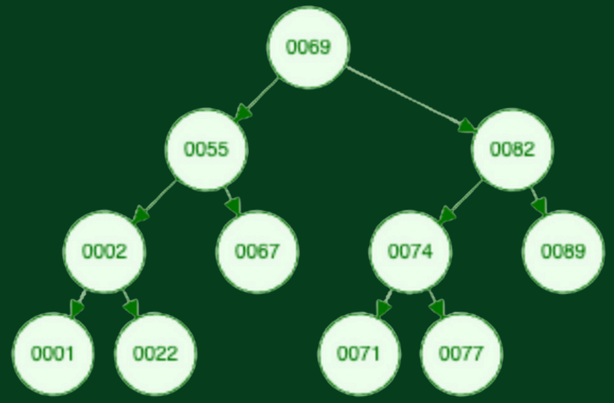
Ejercicio 1: Preorder, inorder, postorder

Dado los siguientes BST, usted debe recrear un output en forma de un arreglo para preorder, inorder y postorder:

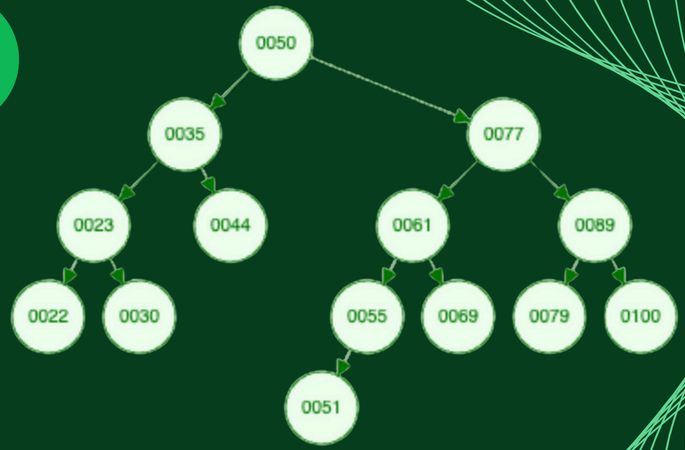
1



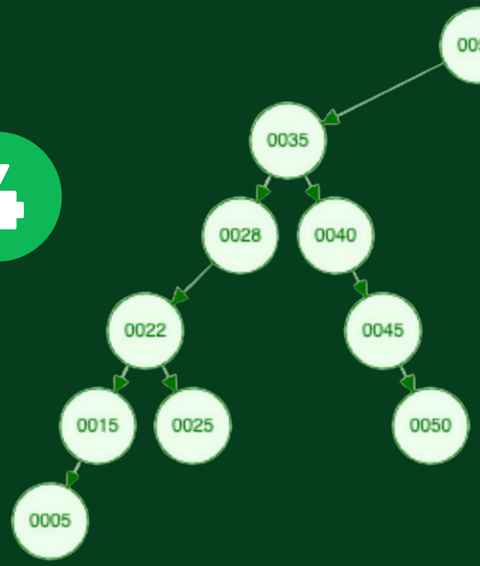
2



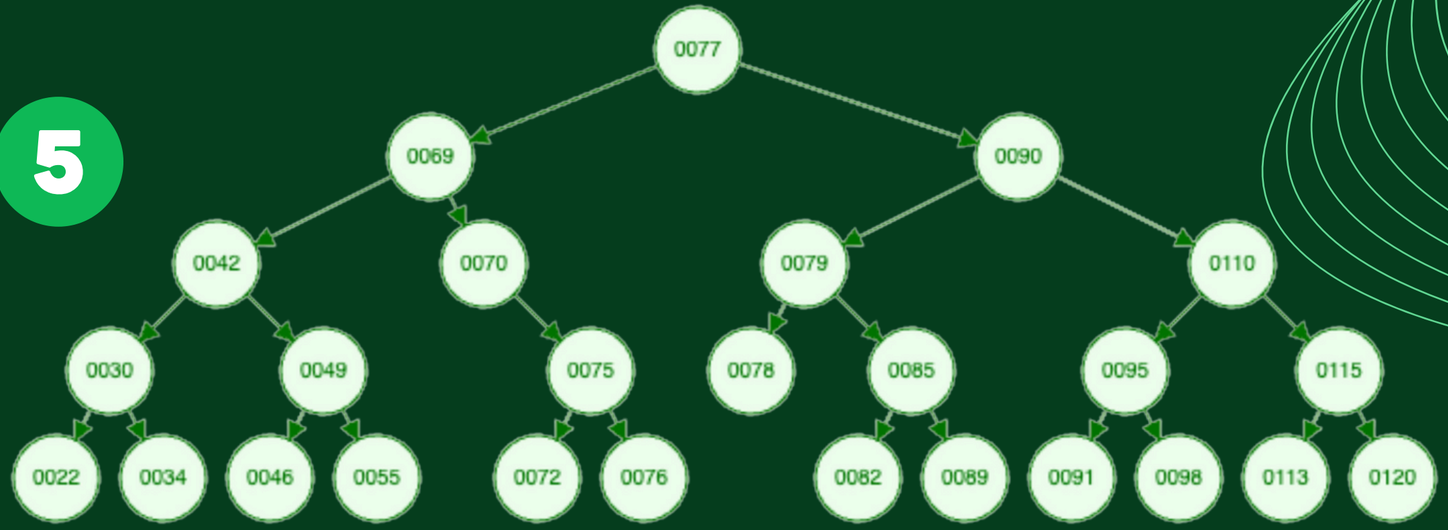
3



4



5



Ejercicio 2: Construir BST desde preorder e inorder

A partir de dos arreglos entregados, uno con preorder y otro con inorder del mismo árbol. Se debe retornar el árbol en su forma heapsort y dibuje el arbol.

EJEMPLO

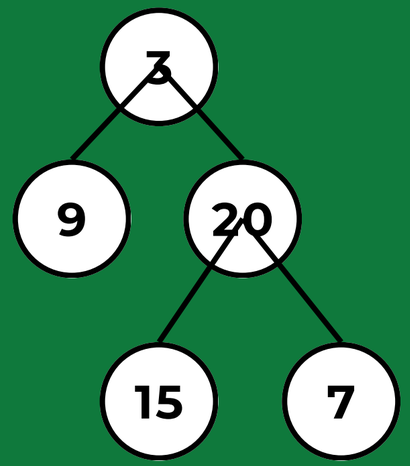
Input:

preorder = [3, 9, 20, 15, 7]

inorder = [9, 3, 15, 20, 7]

Output:

[3, 9, 20, null, null, 15, 7]



RESOLVER

1. preorder = [4, 2, 1, 3, 6, 5, 7]
   inorder = [1, 2, 3, 4, 5, 6, 7]

2. preorder = [55, 44, 12, 46, 70, 59, 90]
   inorder = [12, 44, 46, 55, 59, 70, 90]

3. preorder = [25, 15, 10, 4, 12, 22, 18, 24, 50,35, 31, 44, 70, 66, 90]
   inorder = [4, 10, 12, 15, 18, 22, 24, 25, 31, 35, 44, 50, 66, 70, 90]

Ejercicio 3: Intersección

Dados dos arreglos de enteros, se debe retornar un arreglo con la intersección de ambos (los números que están presentes en ambos arreglos):

EJEMPLO 1

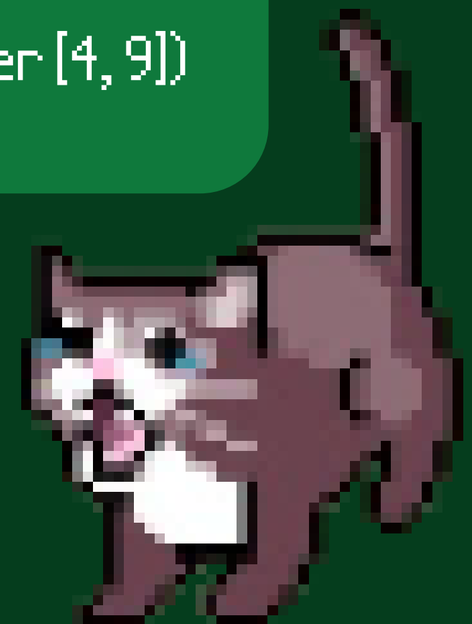
Input: nums1 = [1, 2, 2, 1], nums2 = [2, 2]

Output: [2]

EJEMPLO 2

Input: nums1 = [4, 9, 5], nums2 = [9, 4, 9, 8, 4]

Output: [9, 4] (También puede ser [4, 9])



Ejercicio 4: First Missing Positive

Dado un arreglo “nums” de enteros desordenados retorna el entero más pequeño que no esté presente en este arreglo. Su código debe tener un Big O de O(N).

EJEMPLO 1

IInput: nums = [1, 2, 0]

Output: 3

EJEMPLO 2

Input: nums = [3, 4, -1, 1]

Output: 2

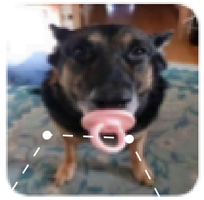
EJEMPLO 3

Input: nums = [7, 8, 9, 11, 12]

Output: 1

Para más material visitar GitHub

DiegoBan/
EDDA2024-1-S4



Códigos vistos en ayudantías y laboratorios

2 Contributors 0 Issues 6 Stars 0 Forks

DiegoBan/EDDA2024-1-S4: Códigos vistos en ayudantías y laboratorios

Códigos vistos en ayudantías y laboratorios. Contribute to DiegoBan/EDDA2024-1-S4 development by creating an account on GitHub.

GitHub

Mucha suerte en su Solemne 2!! :D