

Agrega a un árbol binario completo

Descripción sencilla

Si quiero agregar un elemento ...

- 1- Debo verificar que el elemento no sea null
- 2- Si no lo es, creo un vértice que contenga el elemento
Hay dos casos

- 3- **Caso 1:** Es el primer vértice. Es decir la raíz es null por lo que el vértice a agregar sería el primer elemento y la raíz del árbol, si esto ocurre acabamos.
- 4- **Caso 2:** Ya hay otros vértices en el árbol. Entonces tenemos que recorrer el árbol en orden BFS, es decir usamos una cola, empezariamos por la raíz del árbol y mientras la cola no sea vacía sacamos el vértice al principio de la cola y verificamos si su hijo izquierdo es null, de ser el caso encontramos donde meter nuestro nuevo vértice, si el hijo izquierdo no es null entonces revisamos el hijo derecho, si encontramos que el hijo derecho es null entonces encontramos donde meter el nuevo vértice.

Si ambos hijos del vértice sacado de la cola son distintos de null tenemos que revisar los hijos de esos vértices, es decir hay que revisar el sig. nivel del árbol por lo que debemos meter ambos hijos a la cola, primero el izquierdo y luego el derecho.

Nota: En la sig. iteración el vértice sacado será el izquierdo del primer vértice sacado de la cola, si otra vez no encontramos un hijo la siguiente iteración sacará al derecho y así consecutivamente.

imágenes de referencia

1- 5
↑
elemento
¿Es null?

2- 5
Vertice $v = \text{nuevoVertice}(\text{elemento})$

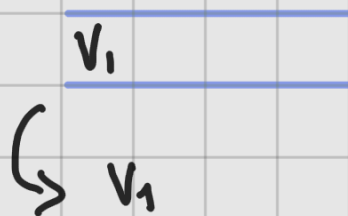
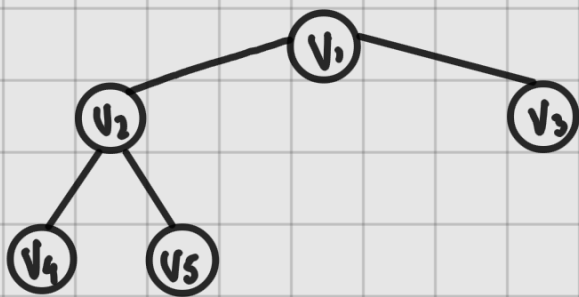
3- Caso 1: El árbol no tiene elementos

raíz == null? sí

5 → raíz

Sumamos uno a elementos

4- Caso 2: El árbol tiene elementos

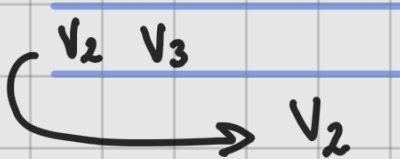


$V_1.\text{izq} = \text{null?}$ No
 $V_1.\text{der} = \text{null?}$ No

Los metemos
a la cola

$V_1.\text{izq} = V_2$

$V_1.\text{der} = V_3$

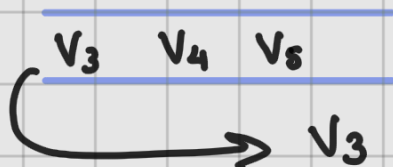


$V_2.\text{izq} = \text{null?}$ No
 $V_2.\text{der} = \text{null?}$ No

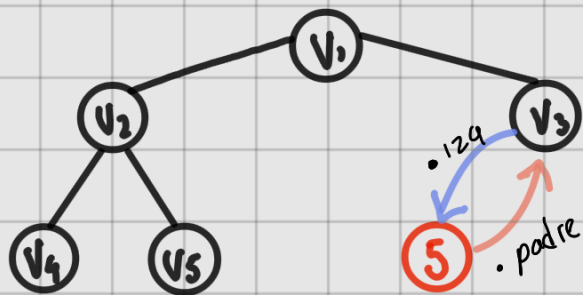
Los metemos
a la cola

$V_2.\text{izq} = V_4$

$V_2.\text{der} = V_5$



$V_3.\text{izq} = \text{null?}$ **SÍ** !!
Encontramos un hoyo



Hay que crear las conexiones:
 el hijo izq de V_3 va a ser
 el vértice que contiene al 5
 y el padre del vértice que
 contiene a 5 va a ser V_3 ,
 aumentamos en 1 los elementos.

Terminamos.

Pseudocódigo del Caso 2

Mientras (C no sea vacía) {

actual = C.saca()

Si (actual.izq es null)

actual.izq = nuevo

actual.izq.padre = actual

elementos + 1

terminamos

Si (actual.der es null)

// similar al caso
anterior

terminamos

C.mete (actual.izq)

C.mete (actual.der)

}