

# Árboles Binarios de Búsqueda - Tópicos I

Víctor Racsó Galván Oyola

13 de noviembre de 2020

1 Definición

2 Complejidad

# Árbol binario de búsqueda (ABB)

## Árbol

Un árbol es un grafo conexo, no dirigido y acíclico. Este puede ser enraizado, en cuyo caso un nodo es denotado como raíz y desde él se toman algunas características del árbol mismo.

## Árbol binario

Un árbol es binario cuando, para todos sus nodos  $u$ , se cumple que la cantidad de hijos de  $u$  es menor o igual a 2.

## ABB

Un árbol binario es de búsqueda cuando, para todo nodo  $u$ , los valores a su izquierda son menores o iguales y los de su derecha son mayores que  $u.val$  y además es enraizado.

# Características de un ABB

## Profundidad de un nodo

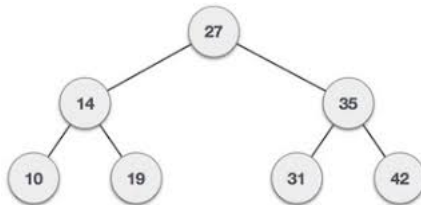
Se define como profundidad de un nodo  $u$  a la cantidad de aristas/nodos en el camino desde la raíz hasta  $u$ .

Esto se denota por  $depth(u)$ .

## Altura de un ABB

Se define como la **altura**  $h$  de un árbol binario de búsqueda a la expresión:

$$h = \max_{u \in V} \{depth(u)\}$$



## Algoritmo de búsqueda

Se puede aprovechar las características del ABB para notar lo siguiente: Si estamos visitando un nodo  $u$  en búsqueda de un valor  $x$ , hay 3 posibilidades:

- 1  $u.val = x$ , en este caso encontramos el nodo con el valor  $x$  y debemos detenernos.
- 2  $u.val < x$ , en este caso  $x$  debería estar en algún nodo a la derecha de  $u$ , así que iremos al hijo derecho de  $u$  **si este existe**.
- 3  $u.val > x$ , en este caso  $x$  debería estar en algún nodo a la izquierda de  $u$ , así que iremos al hijo izquierdo de  $u$  **si este existe**.

**Observación 1:** Es importante notar que deberemos realizar esto hasta llegar al valor buscado o hasta llegar a un nodo nulo.

**Observación 2:** En cada paso del algoritmo estaremos bajando 1 nivel de profundidad.

---

**Algorithm 1** Search( $u$ ,  $X$ )

---

```
1: if  $u = NIL$  then
2:   Return false
3: end if
4: if  $u.val = X$  then
5:   Return true
6: end if
7: if  $u.val < X$  then
8:   if  $u.right \neq NIL$  then
9:     Return Search( $u.right$ ,  $X$ )
10:  end if
11: else
12:   if  $u.left \neq NIL$  then
13:     Return Search( $u.left$ ,  $X$ )
14:   end if
15: end if
```

## Complejidad de la búsqueda

Sea  $n$  la cantidad de nodos y  $h$  la altura del árbol, hay que considerar dos situaciones:

- $x$  está en el árbol, en cuyo caso llegaremos a un nodo  $u$  tal que  $u.val = x$ . Si la profundidad de este nodo es  $h'$ , entonces se cumple que  $h' \leq h$ .
- $x$  no está en el árbol, en cuyo caso llegaremos a un nodo  $u$  tal que el hijo por el que deberíamos bajar es nulo. Se da lo análogo al caso anterior, pues  $h' \leq h$ .

Podemos concluir que la complejidad es  $O(h)$ .

## Cota superior de $h$

En el peor de los casos, un ABB será un camino, es decir, que todos los nodos tienen un único hijo definido y el otro nulo o ambos nulos (esto solo para las hojas).

En este caso, la altura está acotada por  $n$ , así que  $h = O(n)$ .

Como tenemos que

$$T(n) = O(h) \wedge h = O(n) \rightarrow T(n) = O(n)$$

