Arboles Binarios de Búsqueda: Aplicaciones y Más

ALGORITMOS Y ESTRUCTURA DE DATOS

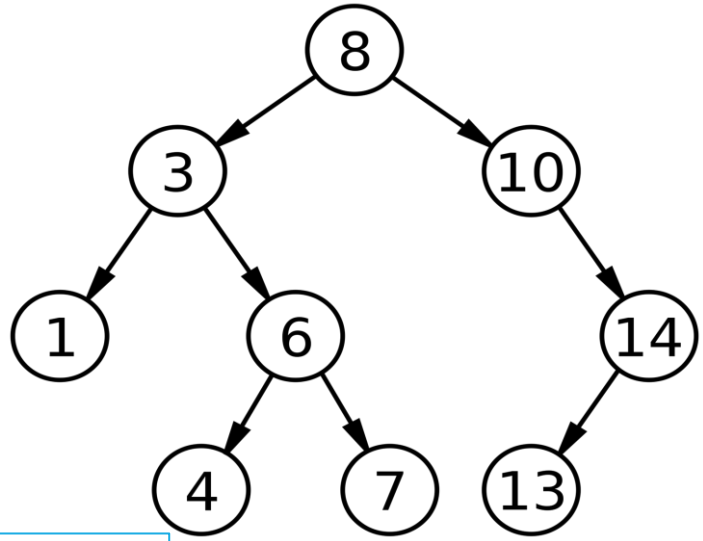
Iteradores para BST

¿Qué tipo de iteradores se pueden aplicar a un BST?

```
template <typename T>
class Iterator
{

private:
    Node<T> *current;

public:
    Iterator() : current(nullptr) {};
    Iterator(Node<T> *current) ;
    Iterator<T> &operator=(Iterator<T> other);
    bool operator!=(Iterator<T> other);
    Iterator<T> &operator++();   //++it
    T operator*();
};
```




```
1 BSTree<int>::iterator ite = bstree->begin();
2 while(ite != bstree->end()) {
3     cout << *ite << endl;
4     ++ite;
5 }
```

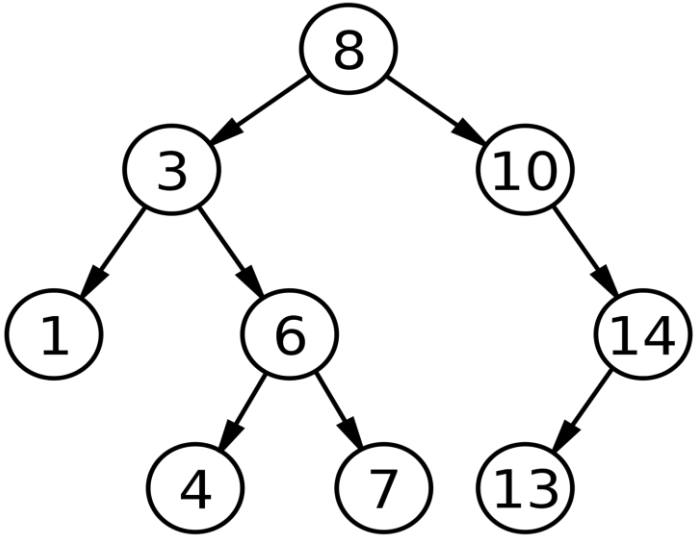
Iteradores para BST

¿Qué tipo de iteradores se pueden aplicar a un BST?

```
template <typename T>
class BSTIterator
{
public:
    enum Type {
        PreOrder, InOrder, PostOrder, BSF
    };

private:
    NodeBT<T> *current;
    Type type;

public:
    BSTIterator() : current(nullptr), type(InOrder) {};
    BSTIterator(NodeBT<T> *current, Type type=InOrder) ;
    BSTIterator<T> &operator=(BSTIterator<T> other);
    bool operator!=(BSTIterator<T> other);
    BSTIterator<T> &operator++();   //++it
    T operator*();
};
```

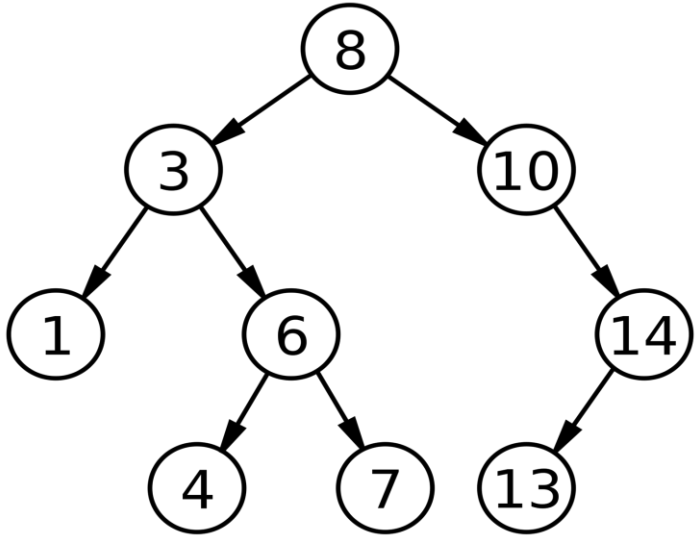


Iterador in-order BST

¿Cómo sería el iterador in-order (++)? Utilicen el árbol de la derecha

```
BSTree<int>::iterator ite = bstree->begin();
while(ite != bstree->end()) {
    cout << *ite << endl;
    ++ite;
}
//1,3,4,6,7,8,10,13,14
```

```
class BSTree {
    public:
    typedef BSTIterator<T> iterator;
    iterator begin() { return iterator(this->root); }
    iterator end(){ return iterator(nullptr); }
}
```



Iterador in-order BST

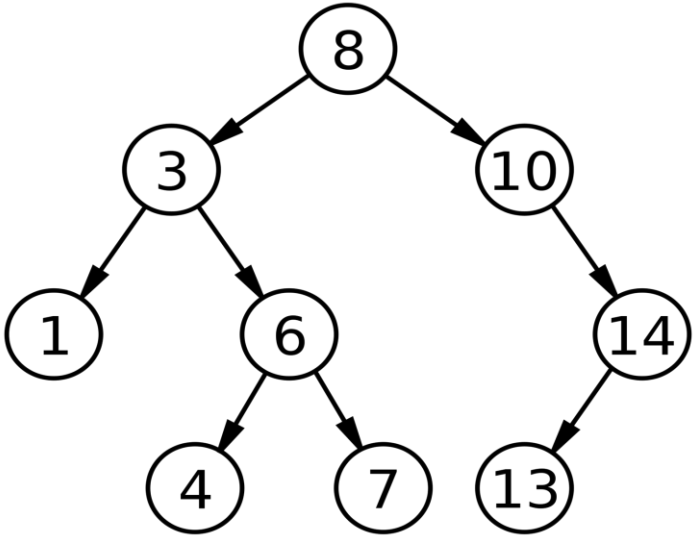
¿Cómo sería el iterador in-order (++)?

Lo primero que se debería hacer es ir desde nuestro root hasta el elemento más a la izquierda

Pero un momento, y ahora cómo regreso?

Una recomendación sería utilizar un stack para simular la recursión. Entonces, mientras vas yendo hacia la izquierda, vas agregando los elementos al stack.

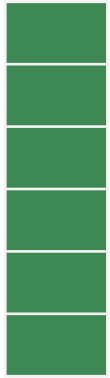
Así, cuando no tengas hijos donde avanzar, das pop de tu stack y ese elemento se vuelve tu nuevo current.



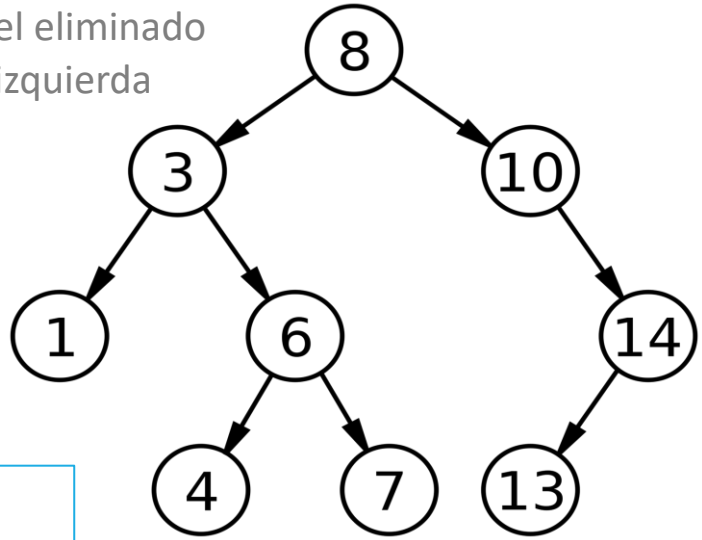
Iterador in-order BST

*ite:
current->data

++ite :
stack.pop()
//agregar el hijo derecho del eliminado
//agregar toda su rama izquierda
current = stack.top()



current



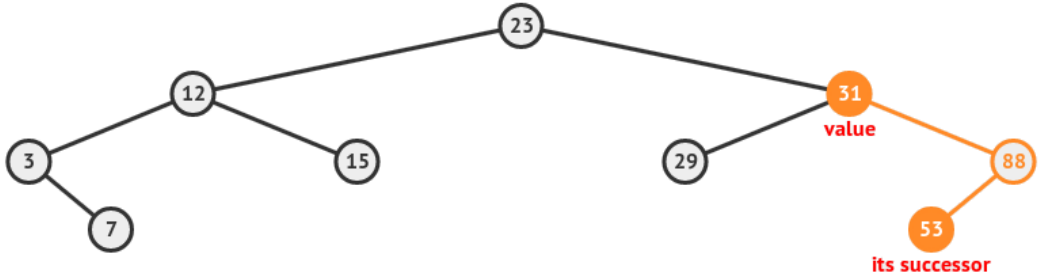
display:: 1, 3, 4, 6, 7, 8, 10, 13,14

```
1 BSTree<int>::iterator ite = bstree->begin();
2 while(ite != bstree->end()) {
3    cout << *ite << endl;
4    ++ite;
5 }
```

Sucesor de un valor en un BTS

¿Cómo obtenemos el sucesor de un valor en un BTS?

1. El nodo tiene un subárbol derecho.



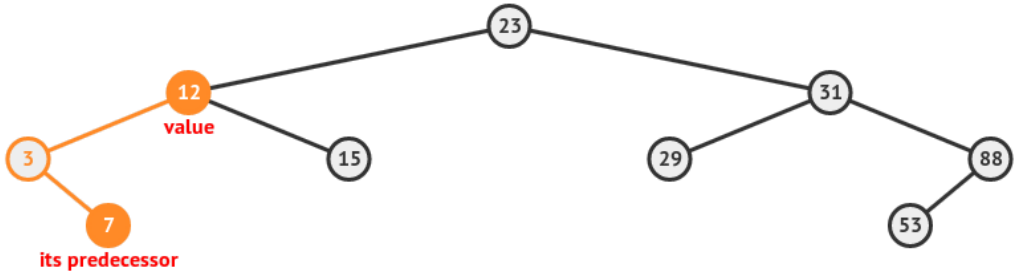
2. El nodo no tiene un subárbol derecho.

3. Es el máximo.

Predecesor de un valor en un BTS

¿Cómo obtenemos el predecesor de un valor en un BTS?

1. El nodo tiene un subárbol izquierdo.



2. El nodo no tiene un subárbol izquierdo.

3. Es el mínimo.

Sucesor de un valor en un BTS

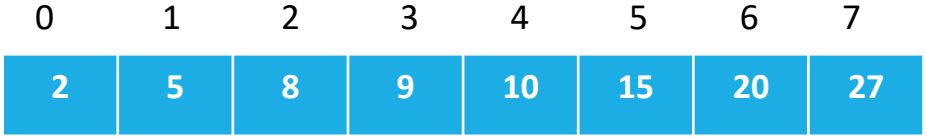
¿Cómo resolver el caso 2? El nodo no tiene un subárbol derecho.

1. Aplicar recorrido in-order **O(n)**
   - Requiere algoritmo iterativo usando stack

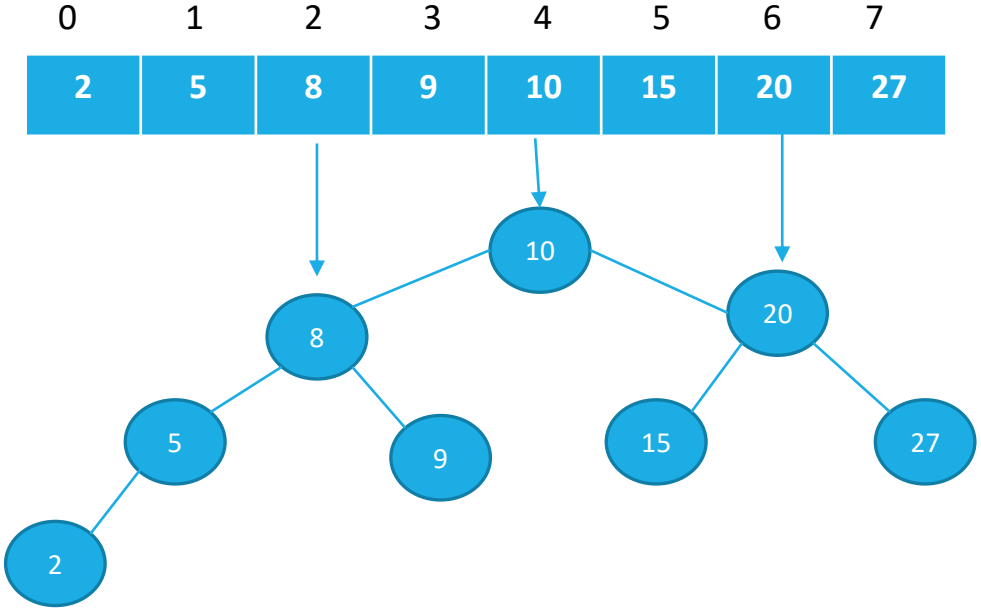
2. Buscar el primer ancestro mayor al valor **O(log(n))**
   - Requiere guardar el puntero al padre en cada nodo

3. **En el proceso de búsqueda del valor ir guardando el ancestro mayor O(log(n))**

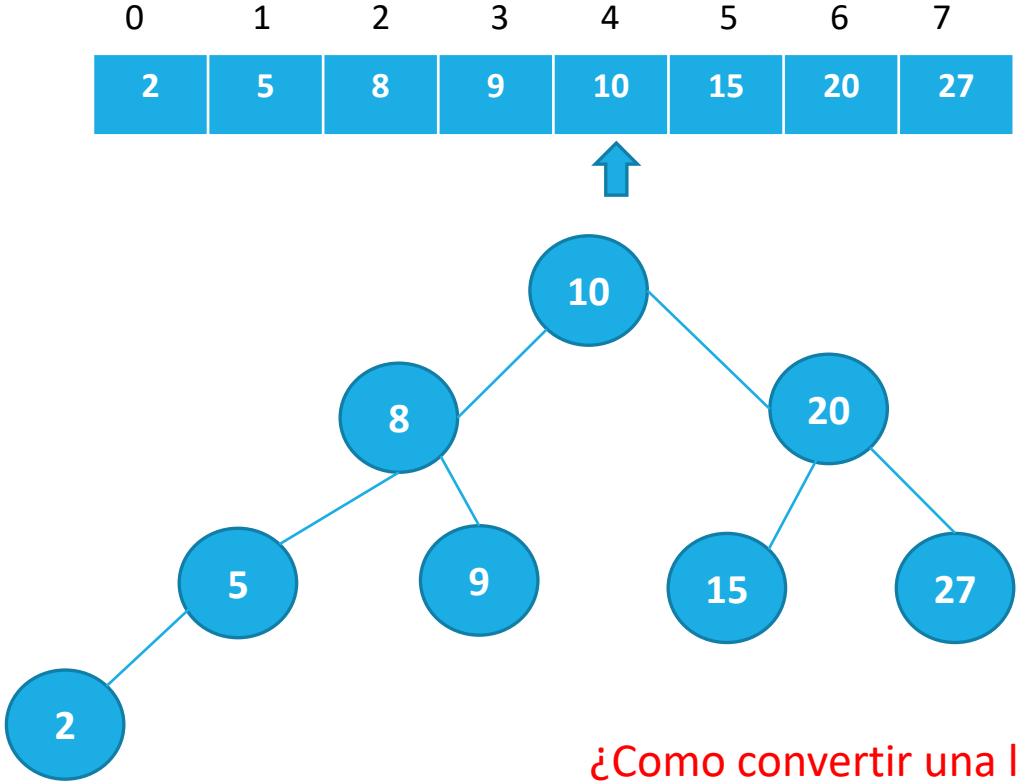
BTS a partir de un array ordenado



BTS a partir de un array ordenado



BTS a partir de un array ordenado



¿Como convertir una lista ordenada a un BST?

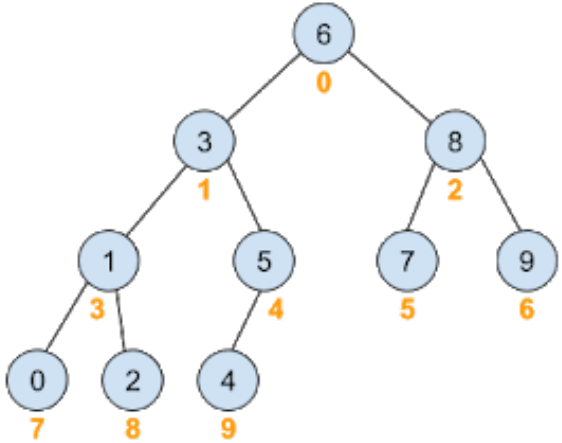
Árbol Binario en un Array

¿Es posible mantener un árbol binario en un array?

```
int left(int i) {
    return 2*i + 1;
}

int right(int i) {
    return 2*i + 2;
}

int parent(int i) {
     return (i-1)/2;
}
```

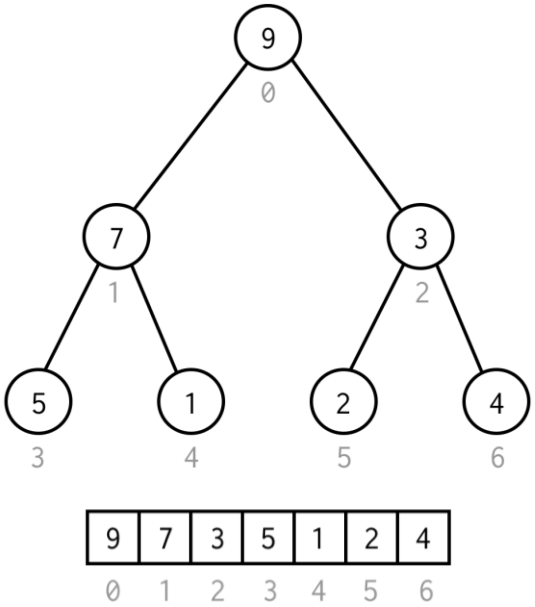


i	0	1	2	3	4	5	6	7	8	9
x	6	3	8	1	5	7	9	0	2	4

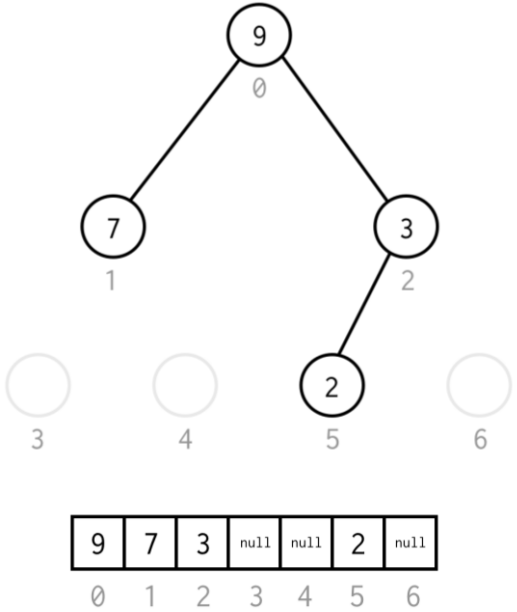
Árbol Binario en un Array

¿Es posible mantener un árbol binario en un array?

Árbol Completo



Árbol No Completo



→HEAPS
→DISJOINT SETS

ALGORITMOS Y ESTRUCTURA DE DATOS