



Algoritmos y Estructura de Datos

Unidad 2 – Semana 11



Logro de sesión

Al finalizar la sesión, el estudiante el estudiante diseña árboles binarios de búsqueda para el almacenamiento y recuperación de datos tomando en cuenta el tiempo de acceso.



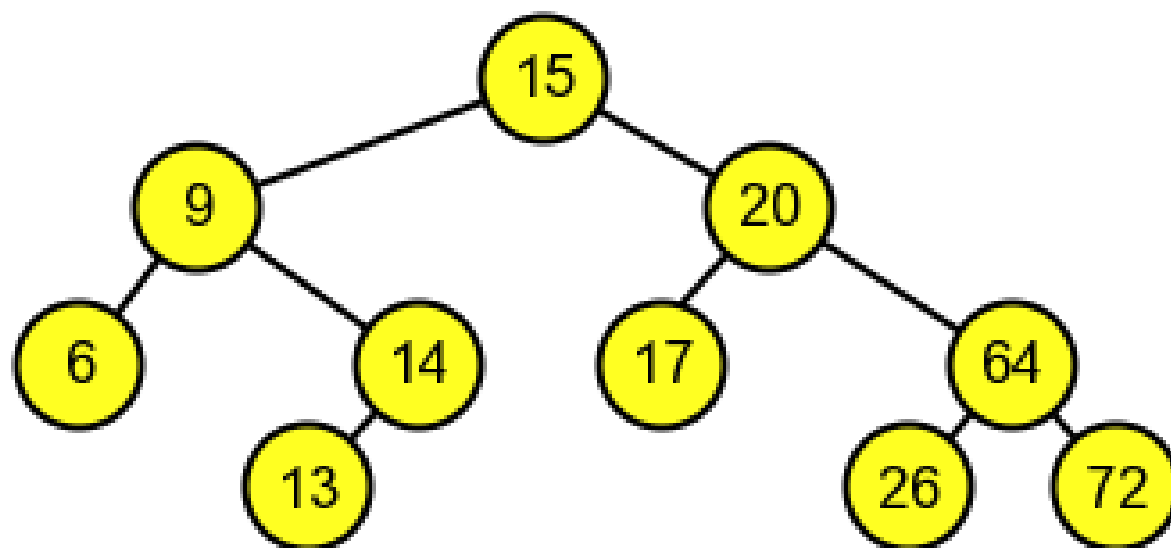
Tema : Arboles binarios de búsqueda

Contenido:

- Arboles binarios de búsqueda



Árbol binario de búsqueda



Arbol binario de búsqueda

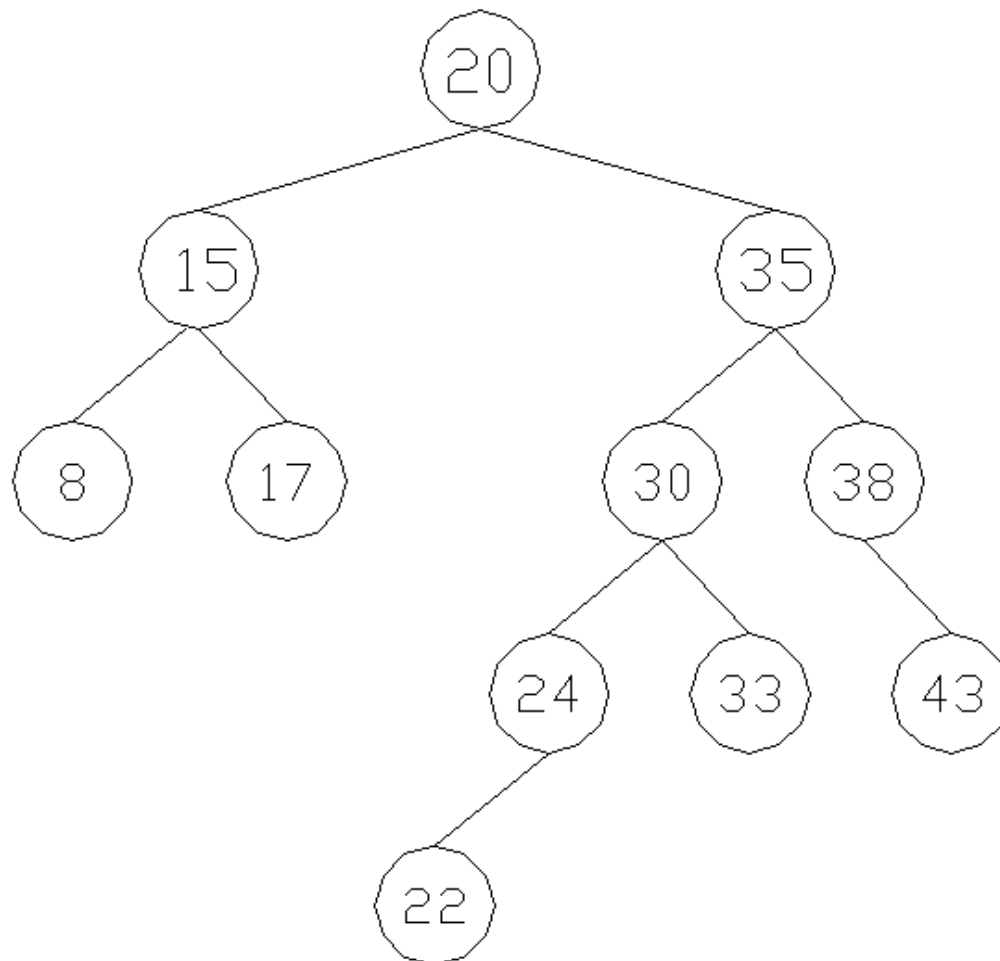


- Caso particular de árboles binarios en los que existe un ordenamiento de los nodos según el siguiente criterio:
 - El valor de la raíz es **mayor** que todos los nodos que están a la **izquierda** de él y es **menor** a todos los nodos que están a la derecha.
 - El caso de la igualdad se deja a criterio de la implementación (si se permite nodos repetidos ó no)
- La principal aplicación es para la búsqueda de información.

Arbol binario de búsqueda



- Por ejemplo:



Cuál sería el
procedimiento
para:

Buscar el 24

Insertar el 36

Arbol binario de búsqueda



- Implementación:

```
bool _buscar(Nodo<T>* nodo, T e) {  
    if (nodo == nullptr) return false;  
    else {  
        int r = comparar(nodo->elemento, e);  
        if (r == 0) return true;  
        else if (r < 0) {  
            return _buscar(nodo->der, e);  
        }  
        else {  
            return _buscar(nodo->izq, e);  
        }  
    }  
}
```

- Es muy parecida a la operación de buscar de ArbolBin pero con la importante diferencia de que cuando el elemento es comparado con la raíz y no es encontrado la búsqueda continúa **sólo por una de las ramas del árbol**.

Arbol binario de búsqueda



- Implementación:

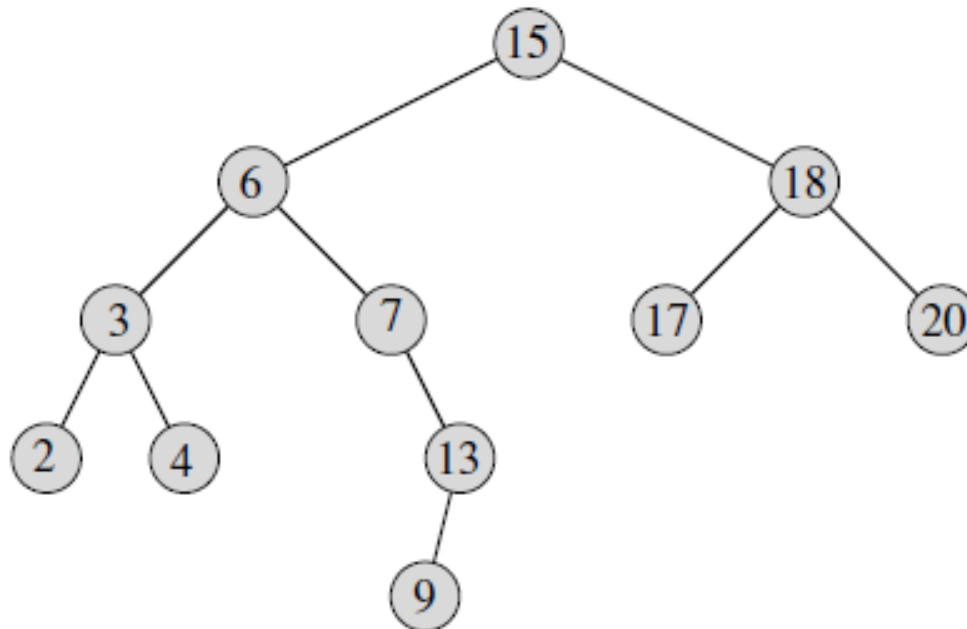
```
bool _insertar(Nodo<T>*& nodo, T e) {  
    if (nodo == nullptr) {  
        nodo = new Nodo<T>();  
        nodo->elemento = e;  
        return true;  
    }  
    else {  
        int r = comparar(nodo->elemento, e);  
        if (r == 0) return false;  
        else if (r < 0) {  
            return _insertar(nodo->der, e);  
        }  
        else {  
            return _insertar(nodo->izq, e);  
        }  
    }  
}
```

- Para adicionar se realiza un proceso similar al de la Búsqueda, sólo que en el momento en que se llegue al árbol vacío, ese es el lugar donde debe ser añadido (creando una hoja con la información del nuevo nodo). En este caso se ha considerado además que si el dato ya está, no se adiciona al árbol.

Ejercicios



- Para el conjunto de datos $\{1,4,5,10,16,17,21\}$, dibujar el árbol binario de búsqueda de nivel 2,3,4,5 y 6
- Dado el siguiente árbol binario de búsqueda:



Ejercicios

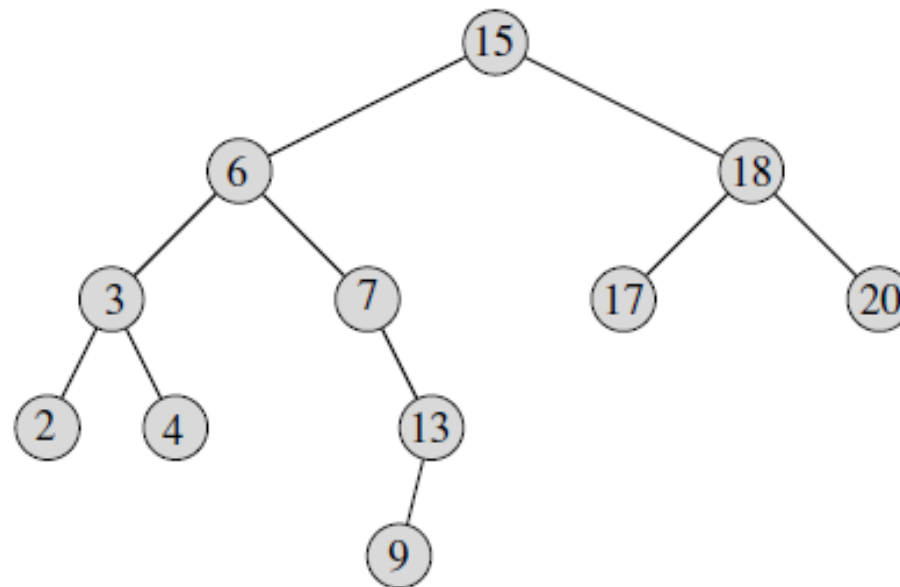


- Elaborar un algoritmo para obtener el nodo cuyo valor es el mínimo del árbol.
- Elaborar un algoritmo para obtener el nodo cuyo valor es el máximo del árbol.
- Elaborar un algoritmo para obtener el nodo sucesor y el predecesor dado un nodo seleccionado del árbol.
- Elaborar un algoritmo para devolver el número de nodos por nivel del árbol.

Ejercicios



- Dado el siguiente árbol binario de búsqueda:



- Elaborar un algoritmo para eliminar un nodo del árbol. ¿Qué pasaría si se eliminaría el 15, 2, 18, 13?

Referencias



- ❑ Sedgewick, R., et. al. (2011) Algorithms, Fourth Edition. Pearson.
- ❑ Cormen, H., et. al. (2009) Introduction to Algorithms, MIT Press.
- ❑ Allen, Mark (2014) Data Structures and Algorithms Analysis in C++, Fourth Edition. Pearson.



EXIGETE INNOVA