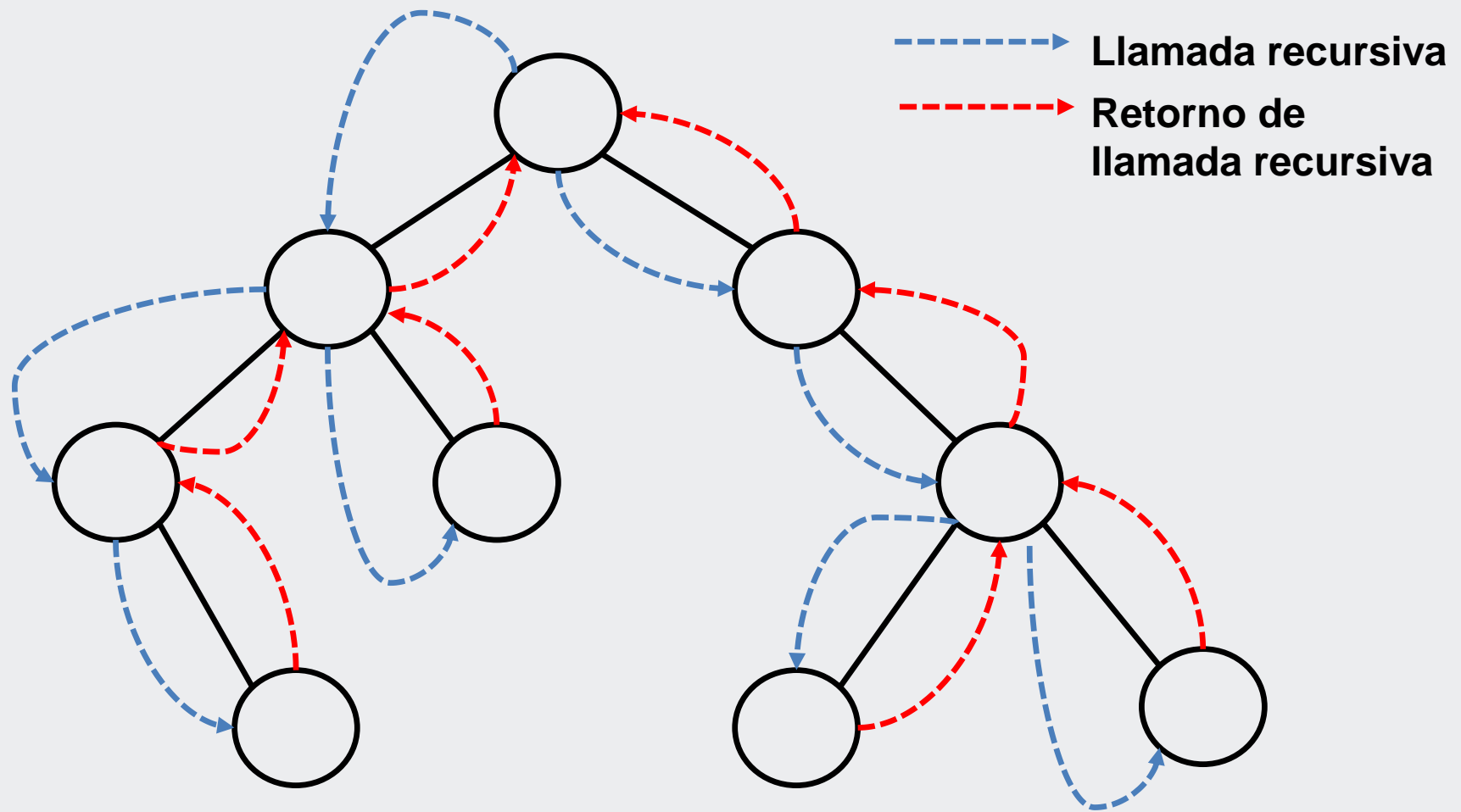


Algoritmos y Estructuras de Datos I



Universidad
Católica del
Uruguay

Arboles Binarios – I
Resumen de clase 6 de mayo de 2021



En TNodeArbolBinario elMetodo (): de algún tipo

COM

*HACER ALGO EN PREORDEN (inicializar **variables locales**, o comprobar algo y devolver)*

Si hijoIzq != nulo entonces

resultadoIzquierdo = hijoIzq.elMetodo()

fin si

HACER ALGO EN INORDEN, por ejemplo decidir si se llama al hijo derecho o no

Si hijoDer != nulo entonces

resultadoDerecho = hijoDer.elMetodo()

fin si

*HACER ALGO EN POSTORDEN (por ejemplo **componer ambos resultados**, devolver)*

FIN

En TNodeArbolBinario elMetodo (): de algún tipo

COM

*HACER ALGO EN PREORDEN (inicializar **variables locales**, o comprobar algo y devolver)*

Si hijoIzq != nulo entonces

resultadoIzquierdo = hijoIzq.elMetodo()

fin si

HACER ALGO EN INORDEN, por ejemplo decidir si se llama al hijo derecho o no

Si hijoDer != nulo entonces

resultadoDerecho = hijoDer.elMetodo()

fin si

*HACER ALGO EN POSTORDEN (por ejemplo **componer ambos resultados**, devolver)*

Se divide el problema en dos subproblemas menores y no superpuestos (disjuntos)

FIN

En TNodeArbolBinario cantidadHojas(): entero

COM

**Si hijolzq == nulo Y hijoDer == nulo entonces
 devolver 1**

fin si

**hojaslzquiendo = 0
hojasDerecho = 0**

**Si hijolzq != nulo entonces
 hojaslzquiendo = hijolzq.cantidadHojas()**

fin si

**Si hijoDer != nulo entonces
 hojasDerecho = hijoDer.cantidadHojas()**

fin si

devolver hojaslzquiendo + hojasDerecho

FIN

ARBOL BINARIO DE BUSQUEDA

- Se dice que un árbol binario es de búsqueda si está organizado de forma que:
 - para cada nodo t_i todas las claves del subárbol izquierdo de t_i son menores que la clave de t_i y todas las claves del árbol derecho son mayores.
- Si el árbol tiene n nodos y está balanceado, su altura será de $\log(n)$
- Una búsqueda en este caso puede tomar $\log(n)$ comparaciones o menos.

Árbol binario de búsqueda

TDA ArbolBinario

Raiz : ElementoAB

Operaciones

Insertar(unElementoAB)

Buscar(unaEtiqueta)

Eliminar(unaEtiqueta)

Arbol binario de búsqueda

TDA ElementoArbolBinario

Etiqueta : TipoEtiqueta

Hijolzquierdo, HijoDerecho : ElementoAB

Operaciones

Insertar(unElementoAB)

Buscar(unaEtiqueta)

Eliminar(unaEtiqueta)

...

Árbol binario de búsqueda

En el TDA ArbolBinario

buscar(unaEtiqueta)

COM

 resultado == nulo

 si raiz <> nulo entonces

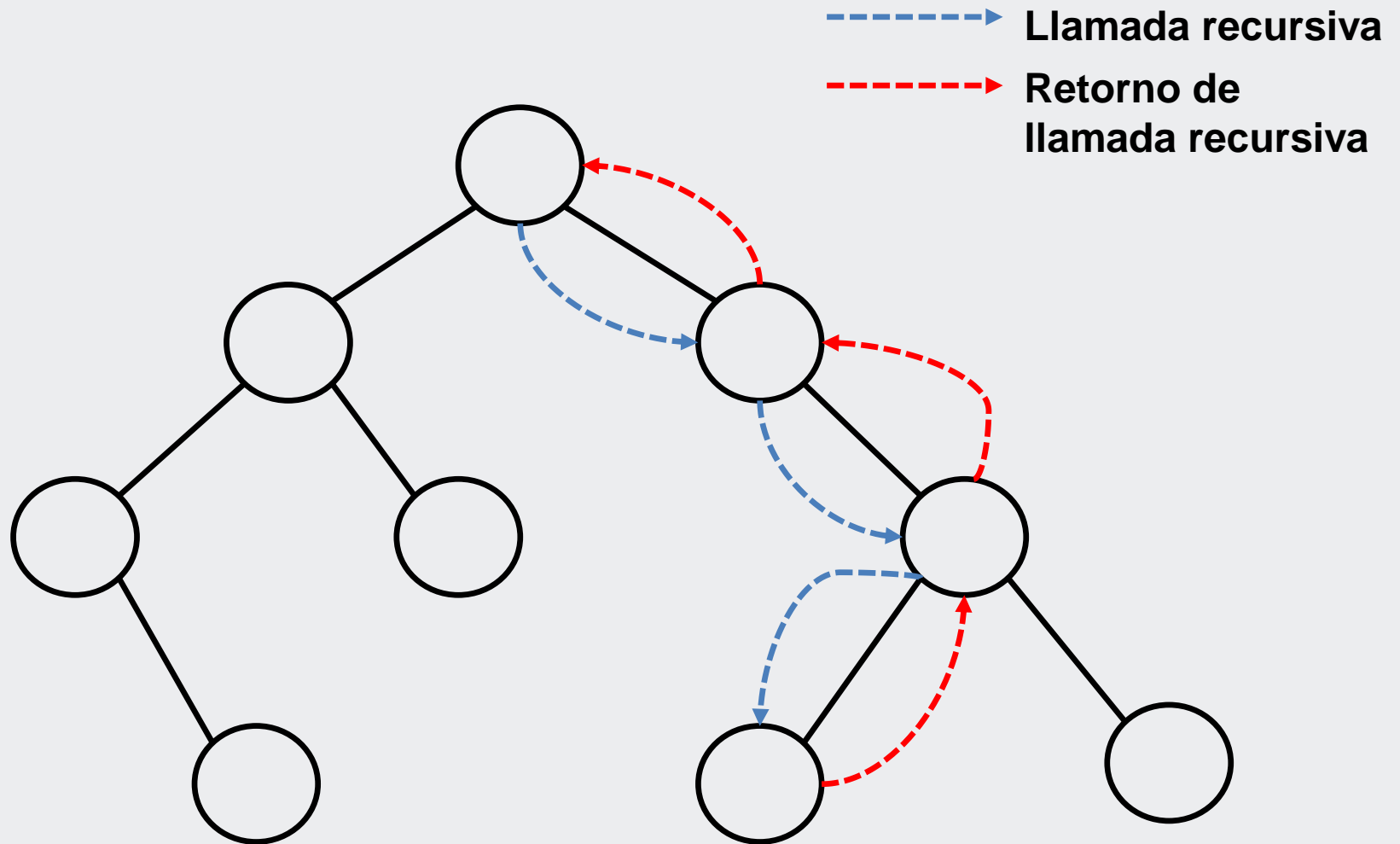
 resultado = raiz.buscar(unaEtiqueta)

 Fin si

 devolver resultado

FIN

Algoritmos de árboles binarios de búsqueda basados en su propiedad



Arbol binario de búsqueda

En el TDA ElementoAB

buscar(unaEtiqueta) : ElementoAB

COM

RESULTADO = nulo

SI unaEtiqueta == etiqueta ENTONCES

RESULTADO = THIS

SINO

SI unaEtiqueta < etiqueta ENTONCES

SI hijoIzquierdo <> nulo ENTONCES

RESULTADO = hijoIzquierdo.buscar(unaEtiqueta)

FINSI

SINO

SI hijoDerecho <> nulo ENTONCES

RESULTADO = hijoDerecho.buscar(unaEtiqueta)

FINSI

FINSI

FINSI

devolver RESULTADO

FIN

TArbolBB -Buscar una Etiqueta

// de TArbolBB

```
public TElementoAB buscar (Comparable unaEtiqueta)
{
    if (esVacio()) {
        return null;
    } else {
        return raiz.buscar(UnaEtiqueta);
    }
}
```

TElementoAB-Buscar Etiqueta

// de TElementoAB

```
public TElementoAB buscar(Comparable unaEtiqueta) {  
    if (UnaEtiqueta.compareTo(etiqueta) == 0) {  
        return this;  
    } else {  
        if (UnaEtiqueta.compareTo(etiqueta) < 0) {  
            if (hijolq != null) {  
                return hijolq.buscar(UnaEtiqueta);  
            } else {  
                return null;  
            }  
        } else {  
            if (UnaEtiqueta.compareTo(etiqueta) > 0) {  
                if (hijoDer != null) {  
                    return hijoDer.buscar(UnaEtiqueta);  
                } else {  
                    return null;  
                }  
            } else {  
                return null;  
            }  
        }  
    }  
}
```

Inserción en árboles binarios

De ArbolBinario

Insertar(unElementoAB)

COM

SI Raiz == nulo ENTONCES

Raiz = unElementoAB

SINO

Raiz.Insertar(unElementoAB)

FIN

Inserción en árboles binarios

De ElementoArbolBinario

Insertar(UnElementoAB)

COM

SI Etiqueta = unElementoAB.Etiqueta ENTONCES

SALIR // ya está en el árbol

FINSI

SI unElementoAB.Etiqueta < Etiqueta ENTONCES

SI HijoIzquierdo = nulo ENTONCES

HijoIzquierdo \leftarrow unElementoAB

SINO HijoIzquierdo.Insertar(unElementoAB)

FINSI

SINO

SI HijoDerecho = nulo ENTONCES

HijoDerecho \leftarrow unElementoAB

SINO HijoDerecho.Insertar(unElementoAB)

FINSI

FINSI

FIN

Implementación de Recorridos ABB

- Preorden
- Inorden
- Postorden

TArbolBB - Inorden

```
public String inOrden(){  
    if (raiz == null) return "arbol vacio";  
    else return raiz.inOrden();  
}
```

TElementoAB - Inorden

```
public String inOrden() {  
    String tempStr = "";  
    if (hijolq != null) {  
        tempStr = hijolq.inOrden();  
    }  
    tempStr = tempStr + imprimir();  
    if (hijoDer != null) {  
        tempStr = tempStr + hijoDer.inOrden();  
    }  
    return tempStr;  
}
```

TRABAJOS DE APLICACIÓN