[2-2021] ESTRUCTURAS DE DATOS II - SB

Descripción | Ver entrega

Primer Parcial

Disponible desde: Tuesday, 21 de December de 2021, 07:00
Límite de entrega: Tuesday, 21 de December de 2021, 09:15
Número máximo de ficheros: 6
Tipo de trabajo: Individual

Primer parcial

1. Para un árbol binario de búsqueda implementar un método que usando la lógica de un recorrido en InOrden iterativo determine el número de nodos hojas que hay en el árbol.

2. Para un árbol MVias sobreescriba el método equals para determinar si dos arboles son iguales con respecto a su forma y claves.

3. Para un árbol binario implemente un método que determine la cantidad de nodos que hay después del nivel n.

Tiene disponible en el paquete por defecto las siguientes clases e interfaz

Interfaces

IArbolBusqueda<K extends Comparable<K>,V>

Clases

ArbolB<K extends Comparable<K>,V>

ArbolBinarioBusqueda<K extends Comparable<K>,V>

ArbolMViasBusqueda<K extends Comparable<K>,V>

AVL<K extends Comparable<K>,V>

NodoBinario<K,V>

NodoMVias<K,V>

Excepciones

ExcepcionClaveNoExiste

ExcepcionOrdenNoValido

Métodos implementados en todos los árboles:

int altura()

V buscar(K claveABuscar)

boolean contiene(K claveABuscar)

V eliminar(K claveAEliminar)

boolean esArbolVacio()

void insertar(K claveAInsertar, V valorAInsertar)

List<K> recorridoEnInOrden()

List<K> recorridoEnPostOrden()

List<K> recorridoEnPreOrden()

List<K> recorridoPorNiveles()

int size()

void vaciar()

Métodos disponibles en la clase NodoBinario:

boolean esHoja()

static boolean esNodoVacio(NodoBinario elNodo)

boolean esVacioHijoDerecho()

boolean esVacioHijoIzquierdo()

K getClave()

NodoBinario<K,V> getHijoDerecho()

NodoBinario<K,V> getHijoIzquierdo()

V getValor()

static NodoBinario nodoVacio()

void setClave(K clave)

void setHijoDerecho(NodoBinario<K,V> hijoDerecho)

void setHijoIzquierdo(NodoBinario<K,V> hijoIzquierdo)

void setValor(V valor)

Métodos disponibles en la clase NodoMVias

int cantidadDeClavesNoVacias()

int cantidadDeHijosNoVacios()

int cantidadDeHijosVacios()

static Object datoVacio()

boolean esClaveVacia(int posicion)

boolean esHijoVacio(int posicion)

boolean esHoja()

static boolean esNodoVacio(NodoMVias elNodo)

boolean estanClavesLlenas()

K getClave(int posicion)

NodoMVias<K,V> getHijo(int posicion)

V getValor(int posicion)

static NodoMVias nodoVacio()

void setClave(int posicion, K clave)

void setHijo(int posicion, NodoMVias<K,V> nodoHijo)

void setValor(int posicion, V valor)

Resumen de retención de datos
Descargar la app para dispositivos móviles