AVL TAD

|  |  |
| --- | --- |
| Nombre: | AVL Tree |
| Descripción: | Es una estructura de datos ordenados, donde cada nodo en el árbol tiene dos elementos, izquierdo y derecho, los cuales también son nodos. Se encuentra balanceado |
| Invariante: | {inv: 1. El elemento izquierdo de cada nodo debe ser menor o igual a su nodo padre. 2. El elemento derecho de cada nodo debe ser mayor o igual a su nodo padre. 3. El factor de balanceo de cada subárbol debe ser x, tal que: } |
| Operaciones: | * AVLTree: <> → <> * isEmpty. <> → <boolean> * getSize. <> → <int> * getHeight. <Node n> → <int > * changeHeight. <Node n> → <int> * getBalance. <Node n> → <int> * leftRotate. <Node n> → <Node> * rightRotate. <Node n> → <Node> * insert. <T element, K key> → <> * delete. <T element> → <Node> * getMin. <> → <Node> * getMax. <> → <Node> * getEquals. <T value> → <List<K>> * getLess. <T value> → <List<K>> * getHigher. <T value> → <List<K>> * postOrder. <Node root> → <String> |

|  |  |
| --- | --- |
| Nombre: | AVLTree |
| Descripción: | Crea un arbol balanceado vacio. |
| Entradas: | {pre:} |
| Salidas: | {post: avlTree size = 0} |

|  |  |
| --- | --- |
| Nombre: | isEmpty |
| Descripción: | Verifica si el arbol tiene por lo menos un elemento. |
| Entradas: | {pre:} |
| Salidas: | {post: (avlTree size = 0)? then true, else false} |

|  |  |
| --- | --- |
| Nombre: | getSize |
| Descripción: | Retorna la cantidad de nodos que tiene el árbol. |
| Entradas: | {pre: el árbol debe existir} |
| Salidas: | {post: cantidad de nodos positivo, (empty)?size = 0} |

|  |  |
| --- | --- |
| Nombre: | getHeight |
| Descripción: | Retorna la cantidad de niveles que tiene un árbol. |
| Entradas: | {pre: el árbol debe existir} |
| Salidas: | {post: cantidad de nodos del camino más largo entre la raíz y una de sus hojas.} |

|  |  |
| --- | --- |
| Nombre: | changeHeight |
| Descripción: | Cambia la altura de un nodo del árbol. |
| Entradas: | {pre: Nodo n debe de existir} |
| Salidas: | {post: Si no tiene subárboles, la altura es 0, si tiene subárboles, la altura es su subárbol más grande +1} |

|  |  |
| --- | --- |
| Nombre: | getBalance |
| Descripción: | Retorna el número del criterio de balance del subárbol. |
| Entradas: | {pre:} |
| Salidas: | {post: El factor de balanceo siempre es X tal que } |

|  |  |
| --- | --- |
| Nombre: | leftRotate |
| Descripción: | Realiza una rotación hacia la izquierda con el hijo derecho del subárbol izquierdo del nodo a rotar. |
| Entradas: | {pre:} |
| Salidas: | {post: Nodo balanceado o parcialmente balanceado} |

|  |  |
| --- | --- |
| Nombre: | rightRotate |
| Descripción: | Realiza una rotación hacia la izquierda con el hijo izquierdo del subárbol derecho del nodo a rotar. |
| Entradas: | {pre:} |
| Salidas: | {post: Nodo balanceado o parcialmente balanceado} |

|  |  |
| --- | --- |
| Nombre: | Insert |
| Descripción: | Inserta un nuevo elemento en el arbol. |
| Entradas: | {pre: Árbol debe de existir} |
| Salidas: | {post: El arbol debe mantener un balance de entre 1 y -1} |

|  |  |
| --- | --- |
| Nombre: | Delete |
| Descripción: | Eliminar un elemento del arbol. |
| Entradas: | {pre: El árbol debe existir} |
| Salidas: | {post: El arbol debe mantener un balance de entre 1 y -1} |

|  |  |
| --- | --- |
| Nombre: | getMin |
| Descripción: | Retorna el nodo que tiene menor valor en la estadística evaluada. |
| Entradas: | {pre:} |
| Salidas: | {post: (isEmpty)?then null: else Min} |

|  |  |
| --- | --- |
| Nombre: | getMax |
| Descripción: | Retorna el nodo que tiene mayor valor en la estadística evaluada. |
| Entradas: | {pre:} |
| Salidas: | {post: (isEmpty)?then null: else Max} |

|  |  |
| --- | --- |
| Nombre: | getEquals |
| Descripción: | Retorna una lista con las Keys:K de los nodos con valor igual al valor de la estadística evaluada. |
| Entradas: | {pre:} |
| Salidas: | {post:} |

|  |  |
| --- | --- |
| Nombre: | getLess |
| Descripción: | Retorna una lista con las Keys:K de los nodos con valor menor al valor de la estadística evaluada. |
| Entradas: | {pre:} |
| Salidas: | {post:} |

|  |  |
| --- | --- |
| Nombre: | getHigher |
| Descripción: | Retorna una lista con las Keys:K de los nodos con valor mayor al valor de la estadística evaluada. |
| Entradas: | {pre:} |
| Salidas: | {post:} |

|  |  |
| --- | --- |
| Nombre: | preOrder |
| Descripción: | Devuelve la lista de todos los jugadores en el árbol AVL, a partir del nodo indicado, en pre orden |
| Entradas: | {pre:} |
| Salidas: | {post:} |

TAD Casos de Prueba

|  |  |
| --- | --- |
| **# Caso de prueba** | **Condicion.** |
| 1 | El árbol está vacio. |
| 2 | Árbol formado por el array = {10, 5, 12} |

* getEquals. <T value> → <List<K>>
* getLess. <T value> → <List<K>>
* getHigher. <T value> → <List<K>>

postOrder. <Node root> → <String>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Nombre** | **Caso de Prueba** | **Entrada** | **Salida Esperada** | **Salida Obtenida** |
| 1 | isEmpty | 1 |  | Devuelve el valor de verdad si contiene algún nodo. | True. |
| 2 | isEmpty | 2 |  | Devuelve el valor de verdad si contiene algún nodo. | False. |
| 3 | getSize | 1 |  | Devuelve la cantidad de nodos que tiene el arbol. | 0 |
| 4 | getSize | 2 |  | Devuelve la cantidad de nodos que tiene el arbol. | 3 |
| 5 | getHeight | 1 |  | Retorna la altura del árbol. | 0 |
| 6 | getHeight | 2 |  | Retorna la altura del arbol | 2 |
| 7 | getBalance | 1 |  | Retorna el factor de balanceo de un sub arbol. | 0 |
| 8 | getBalance | 2 |  | Retorna el factor de balanceo de un sub arbol. | 0 |
| 9 | insert | 1 | 8 | Añade un nuevo nodo con el valor recibido. | Root = 8.  Height = 1.  Size = 1. |
| 10 | insert | 2 | 8 | Añade un nuevo nodo con el valor recibido. | Height = 3.  Size = 4. |
| 11 | delete | 2 | 12 | Elimina el nodo con el valor recibido y realiza rotaciones para mantener el balance. | Balance = 2.  rigthRotate.  Height = 2.  Size = 3. |
| 12 | Min | 2 |  | Devuelve el Nodo con el mínimo valor del árbol. | 5 |
| 13 | Max | 2 |  | Devuelve el Nodo con el máximo valor del árbol. | 10 |
| 14 | getEquals | 2 | 10 | Devuelve una lista con los elementos del árbol con el mismo valor al recibido. | Nodo<T: 10> |
| 15 | getLess | 2 | 10 | Devuelve una lista con los elementos del árbol con valor menor al recibido. | Nodo<T:10>, Node<T:8> |
| 16 | getHigher | 2 | 5 | Devuelve una lista con los elementos del árbol con valor mayor al recibido | Node<T:8>, Node<T:10> |