|  |
| --- |
| **TAD RB TREE** |
| Atributos o descripción. |
| {inv: } //Aquí van las propiedades |
| Operaciones básicas   * createRBTree * getMin * getMax * insert * delete * modify * search * rightRotate * leftRotate |

Operaciones

|  |
| --- |
| **Operacion**  “Descripcion”  Pre:  Post: |
|  |
|  |

**Diseños de casos de pruebas unitarias**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba 1: Verifica que el método añade inserta correctamente un nodo al árbol manteniendo las propiedades. | | | | |
| Clase | Método | Escenario | Entrada | Resultado |
| RBTree | +insert (K, V): void | Hay un árbol rojinegro vacío | K=10  V=1 | La raíz es negra |
| RBTree | +insert (K, V): void | Hay un árbol rojinegro con un nodo: Key=10, value=1 | K=5  V=2 | La raíz es negra  El hijo izquierdo de la raíz es rojo. |
| RBTree | +insert (K, V): void | Hay un árbol rojinegro con los siguientes nodos:  Key=10 value=1  Key=5 value=2 | K=15  V=3 | La raíz es negra y sus dos hijos son rojos. |
| RBTree | +insert (K, V): void | El mismo que el anterior más un nodo con key=15 y value=3 | K=4  V=4 | La raíz es negra y sus dos hijos son negros. El hijo izquierdo el hijo izquierdo de la raíz es rojo. |
| RBTree | +insert (K, V): void | El mismo que el anterior más un nodo con key=4 y value=4 | K=4  V=5 | La raíz es negra. Sus dos hijos son negros y tiene key 4 y 15 respectivamente. Los dos hijos del hijo izquierdo de la raíz son rojos y tienen key 4 y 5 respectivamente. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba 2: Verifica que el método delete elimina correctamente un nodo al árbol manteniendo las propiedades. | | | | |
| Clase | Método | Escenario | Entrada | Resultado |
| RBTree | +delete(K): V | Hay un árbol rojinegro vacío |  |  |
| RBTree | +delete(K): V | Hay un árbol rojinegro con un nodo: Key=10, value=1 |  |  |
| RBTree | +delete(K): V | Hay un árbol rojinegro con los siguientes nodos: |  |  |
| RBTree | +delete(K): V |  |  |  |
| RBTree | +delete(K): V |  |  |  |