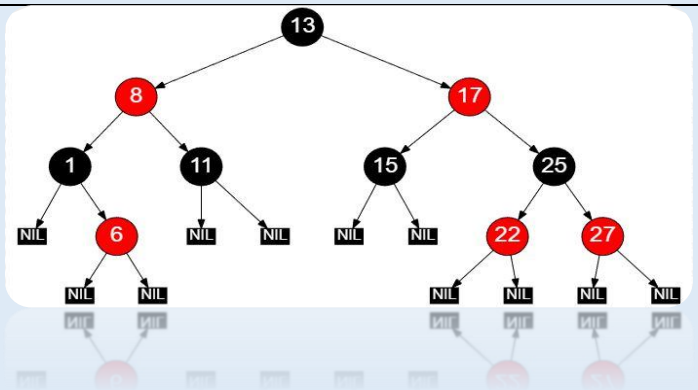
TAD ARN



{*inv.: Se deben respetar las siguientes propiedades. (1) Cada nodo del árbol es rojo o negro. (2) Cada hoja o el siguiente apuntador -de sus últimos hijos- va a ser nulo. (3) Si un nodo es rojo entonces sus dos hijos son negros. (4) Todos los caminos desde el nodo en que este ubicado actualmente va a tener la misma cantidad de nodos negro -altura negra- (5) El padre es menor o igual que su hijo derecho y mayor que su hijo izquierdo.*}

Operaciones Primitivas:

- CrearARN: ->ARN
- InsertarEnARN: *ARNxLlavexValor* ->ARN
- ArreglarInsertar: *ARNxNodo* ->ARN
- EliminarEnARN: *ARNxLlave* ->ARN
- ArreglarEliminar: *ARNxNodo* ->ARN
- RotarIzquierda: *ARNxNodo* ->ARN
- RotarDerecha: *ARNxNodo* ->ARN

CrearARN()

"Crea un árbol binario de búsqueda rojo y negro con raíz nula"

{*pre:* TRUE}

{*post: Se crea un árbol binario de búsqueda rojo y negro vacío*}

InsertarEnARN(Llave, Valor)

"Se inserta un nuevo nodo en el ARN. Este nodo tendrá Llave y Valor"

{*pre:* ARN!=nulo}

{*post: Se agrega un nuevo nodo al árbol rojo y negro*}

ArreglarInsertar(Nodo**)**

"Se evalúan y confirman las propiedades del árbol ARN. Después, se aplican las rotaciones necesarias para balancear el árbol siempre y cuando sea necesario. Esto se hace cada vez que se insertar un nuevo nodo"

{*pre:* ARN!=nulo}

{*post: Si una de las propiedades no se cumple -> Se aplica el balanceo apropiado al ARN*
De lo contrario -> No se hace ninguna modificación}

EliminarEnARN(Llave**)**

"Eliminar un nodo del árbol ARN de acuerdo con la Llave dada"

{*pre:* ARN!=nulo}

{*post: Si la Llave ∈ a algún nodo en el ARN -> Se elimina este nodo y se llevan a cabo las rotaciones necesarias,*
De lo contrario -> No se hace ninguna modificación}

ArreglarEliminar(Nodo**)**

"Se evalúan y confirman las propiedades del árbol ARN. Después, se aplican las rotaciones necesarias para balancear el árbol siempre y cuando sea necesario. Esto se hace cada vez que se elimina un nodo del árbol"

{*pre:* ARN!=nulo}

{*post: Si una de las propiedades no se cumple -> Se aplica el balanceo apropiado al ARN*
De lo contrario -> No se hace ninguna modificación}

RotarIzquierda(Nodo**)**

"Se rota a la izquierda el sub-árbol a partir del nodo dado, teniendo en cuenta las condiciones necesarias del ARN"

{*pre:* ARN!=nulo}

{*post: Se rota el sub-árbol a la izquierda teniendo como punto de partida el nodo dado*}

RotarDerecha(Nodo**)**

“Se rota a la derecha el sub-árbol a partir del nodo dado, teniendo en cuenta las condiciones necesarias del ARN”

{*pre:* ARN!=nulo}

{*post: Se rota el sub-árbol a la derecha a teniendo como punto de partida el nodo dado*}