**Grupo**:

* Pedro Paulo da Silva - 1421422
* Stephanie Fay - 1721631
* Thiago Lages de Alencar - 1721629

Para descobrir os erros, primeiro foi usado os scripts de teste tst-avl-02 e tst-avl-03.  
Após isso foi usado o script de teste tst-avl-01 após ser alterado:

Todos comandos =ExcepetionProgram ou =ExceptionSystem viraram comentários

Os comandos que seguia após eles, foram retirados de comentários.

**Método**

tpTreeElement \* AVL\_Tree ::

DoInsertAVLelement( tpTreeElement \* pCurrElement ,

tpTreeElement \* pNewElement )

**Antes**

pCurrElement->pRightSubTree = ROTATE\_RIGHT( pCurrElement->pLeftSubTree ) ;

**Depois**

pCurrElement->pRightSubTree = ROTATE\_RIGHT( pCurrElement->pRightSubTree ) ;

**Problema**: Estava rotacionando o elemento errado.

**Script gerador do erro**: *tst-avl-01.script*

**Método**

tpTreeElement \* AVL\_Tree :: ROTATE\_RIGHT( tpTreeElement \* pNode )

**Antes**

pLeft->pRightSubTree = pNode ;

pNode->pLeftSubTree = pLeftRight ;

pLeft->pParent = pParent ;

**Depois**

pLeft->pRightSubTree = pNode ;

pNode->pParent = pLeft;

pLeft->pParent = pParent ;

pNode->pLeftSubTree = pLeftRight ;

if (pLeftRight != NULL)

{

pLeftRight->pParent = pNode;

}

if (pParent != NULL && pParent->pRightSubTree == pNode)

{

pParent->pRightSubTree = pLeft;

}

else if (pParent != NULL && pParent->pLeftSubTree == pNode)

{

pParent->pLeftSubTree = pLeft;

}

**Método**

tpTreeElement \* AVL\_Tree :: ROTATE\_LEFT( tpTreeElement \* pNode )

**Antes**

pRight->pLeftSubTree = pNode ;

pNode->pRightSubTree = pRightLeft ;

pRight->pParent = pParent ;

**Depois**

pRight->pLeftSubTree = pNode ;

pNode->pParent = pRight;

pRight->pParent = pParent ;

pNode->pRightSubTree = pRightLeft ;

if (pRightLeft != NULL)

{

pRightLeft->pParent = pNode;

}

if (pParent != NULL && pParent->pRightSubTree == pNode)

{

pParent->pRightSubTree = pRight;

}

else if (pParent != NULL && pParent->pLeftSubTree == pNode)

{

pParent->pLeftSubTree = pRight;

}

**Problema**:  
 Quando alterava quem era o filho de um nó, não alterava quem era o pai daquele filho.  
 Quando alterava quem era o pai de um nó, não alterava quem era o filho daquele pai.

**Script gerador do erro**: *tst-avl-03.script* e *tst-avl-02.script*

**Observação**:

Muitos logs foram alterados de lugar ou foram feitas modificações para que eu conseguisse interpretar melhor o que a rotação estava fazendo.

**Método**

tpTreeElement \* AVL\_Tree :: ROTATE\_RIGHT( tpTreeElement \* pNode )

**Antes**

pNode->elementHeight = MAX( HEIGHT( pNode->pLeftSubTree ) , HEIGHT( pNode->pRightSubTree ) + 1 ) ;

pLeft->elementHeight = MAX( HEIGHT( pLeft->pLeftSubTree ) , HEIGHT( pLeft->pRightSubTree ) + 1 ) ;

**Depois**

pNode->elementHeight = MAX( HEIGHT( pNode->pLeftSubTree ) , HEIGHT( pNode->pRightSubTree ) ) + 1 ;

pLeft->elementHeight = MAX( HEIGHT( pLeft->pLeftSubTree ) , HEIGHT( pLeft->pRightSubTree ) ) + 1 ;

**Método**

tpTreeElement \* AVL\_Tree :: ROTATE\_LEFT( tpTreeElement \* pNode )

**Antes**

pNode->elementHeight = MAX( HEIGHT( pNode->pLeftSubTree ) , HEIGHT( pNode->pRightSubTree ) + 1 ) ;

pRight->elementHeight = MAX( HEIGHT( pRight->pLeftSubTree ) , HEIGHT( pRight->pRightSubTree ) + 1 ) ;

**Depois**

pNode->elementHeight = MAX( HEIGHT( pNode->pLeftSubTree ) , HEIGHT( pNode->pRightSubTree ) ) + 1;

pRight->elementHeight = MAX( HEIGHT( pRight->pLeftSubTree ) , HEIGHT( pRight->pRightSubTree ) ) + 1;

**Problema**:

O tamanho do nó atual é o tamanho do filho maior + 1.

Anteriormente estava pegando o maior entre filho da esquerda e o filho da direita somado com 1.

**Script gerador do erro**: *tst-avl-02.script*

Grafo de chamada  
No enunciado estava **grafo de chamada** em vez de **grafo de estrutura**

