TEMA Seminar #4

1,2. In PDF

3.

Type\* leftmost(Type\* tree)

{

if (tree == NULL)

return NULL;

if (tree->prev\_left == NULL)

return tree;

leftmost(tree->prev\_left);

}

Type\* rightmost(Type\* tree)

{

if (tree == NULL)

return NULL;

if (tree->next\_right == NULL)

return tree;

leftmost(tree->next\_right);

}

Type\* BSTtoDLL(Type\* root)

{

if (root == NULL)

return NULL;

Type\* fromLeft = BSTtoDLL(root->prev\_left);

Type\* fromRight = BSTtoDLL(root->next\_right);

if (fromLeft != NULL)

{

Type\* auxr = rightmost(fromLeft);

auxr->next\_right = root;

root->prev\_left = auxr;

}

if (fromRight != NULL) {

Type\* auxl = leftmost(fromRight);

if (auxl != NULL)

{

root->next\_right = auxl;

auxl->prev\_left = root;

}

return rightmost(fromRight);

}

else

{

return root;

}

}

Complexitate?

Best: O(n)

Worst: O(n^2)

Text, letter

Description automatically generated