

Technical Report

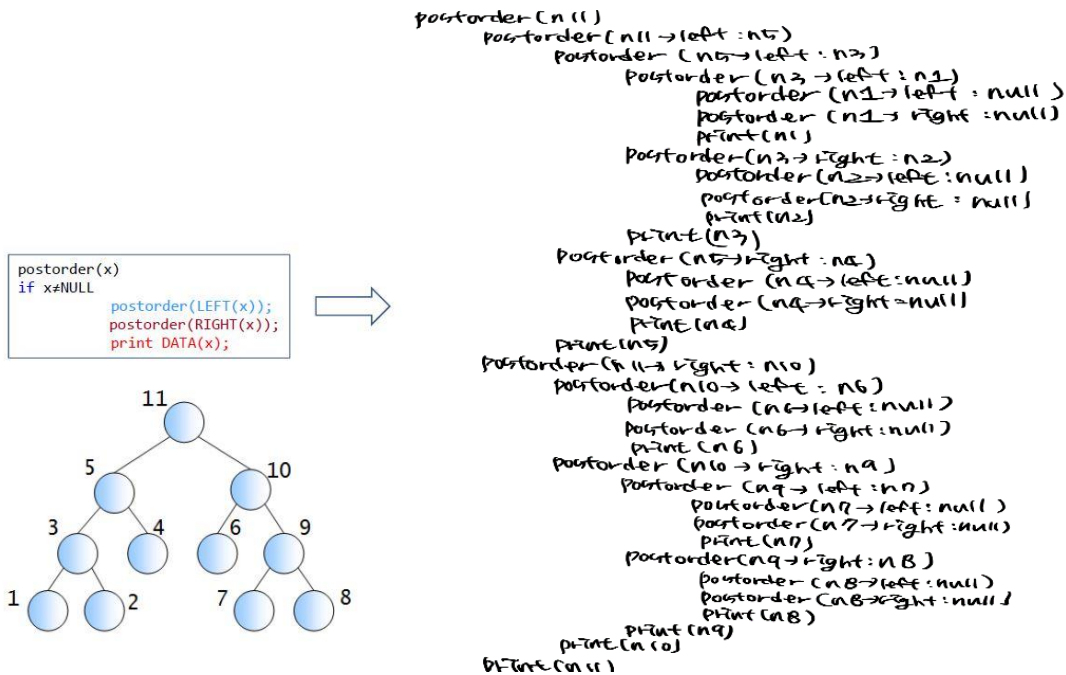
1829008 김민영

homework5 (Tree)

hw5-1 : postorder traversal in the binary tree

postorder의 procedure : 1. Visit the left subtree 2. Visit the right subtree 3. Visit the root node

앞의 inorder, preorder 예제를 이해하고 이를 postorder에 적용해서 작성했다.



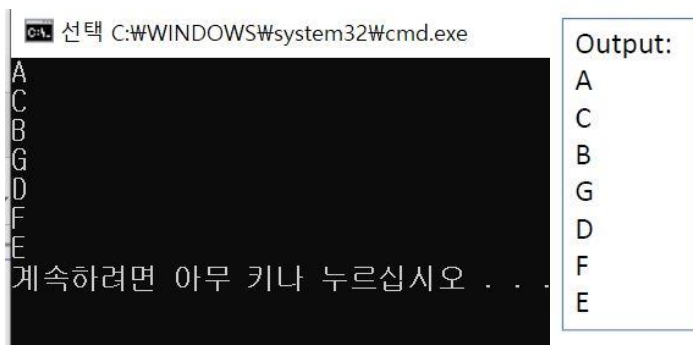
hw5-2 : Find a successor of node in the inorder traversal

1.basic ideas and how you applied it into your code, analysis of your code(complexity)

page3 에 있는 코드를 이용하여 *tree_successor 부분을 채워넣었다.

먼저 오른쪽 subtree를 가지고 있는지 확인한다. 오른쪽 Subtree를 가지고 있으면 leftmost node of right subtree 를 리턴한다. 이는 leftmost 함수를 이용해 구했다. 오른쪽 subtree를 가지고 있지 않으면 그 밑의 코드를 수행한다.

2. result image

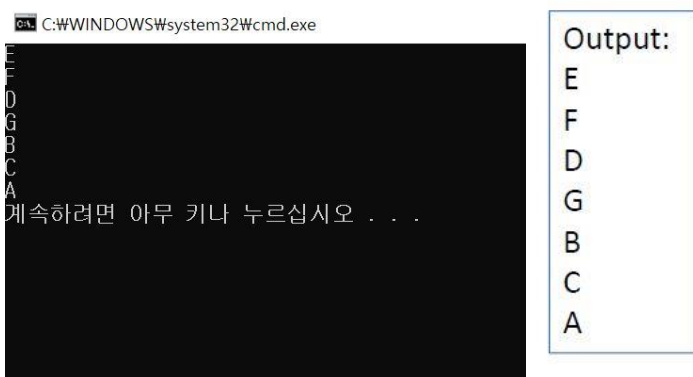


hw5-3 : Find a predecessor of node in the inorder traversal

1. .basic ideas and how you applied it into your code, analysis of your code(complexity)

앞의 hw5_2를 참고하여 successor를 찾는 과정에서 predecessor를 찾는 과정으로 코드를 수정했다. 이번엔 Tree의 왼쪽 서브트리가 NULL 인지를 따진다.

2. result image



hw5-4 : Find a predecessor of node in the inorder traversal

1. .basic ideas and how you applied it into your code, analysis of your code(complexity)

기존의 코드(bst_insertion_deleteion.cpp) delete의 case3는 successor를 이용하여 구현했다. Hw5-4에서는 predessesor를 이용하는 방식으로 코드를 수정했다. 이는 successor를 이용하여 돌려본 결과와 같게 나온다.

2. result image

```
C:\WINDOWS\system32\cmd.exe
Binary tree
3      7      10      12      18      22      24      26      30      35      68      99
Binary tree
3      7      10      12      22      24      26      30      35      68      99
Binary tree
3      7      10      12      22      24      26      30      68      99
Binary tree
3      10      12      22      24      26      30      68      99
```