

COME0331-003 자료구조

COMP0216-002 자료구조응용

LAB #13

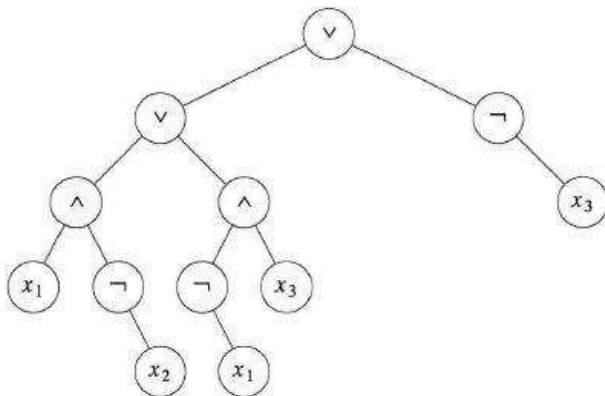
2024. 5. 8.

1. 아래의 입력 파일 input1.txt와 input2.txt 각각에 해당하는 binary tree를 포인터와 동적 메모리 할당을 사용하여 구성하고, 두 binary tree가 동일한지를 검사하는 프로그램을 작성하시오. (2점)

```
%cat input1.txt
n                                // total # of nodes, where n = 10
3  8  12  5  17  23            // store in the order of a complete binary tree
78 -1  5   7  -1  29           // -1 means an empty node

%cat input2.txt
n                                // total # of nodes, where n = 10
3  8  12  5  17  23            // store in the order of a complete binary tree
78 -1  5   7  -1  29           // -1 means an empty node
```

2. 아래의 binary tree에 해당하는 연산식을 input.txt에 표현하였다. input.txt에 해당하는 binary tree를 포인터와 동적 메모리 할당을 사용하여 구성하고 postorder traversal을 사용하여 연산식의 결과를 출력하는 프로그램을 작성하시오. (2점)



```
%cat input.txt
n                                // # of variables, where n = 3 for x1, x2, and x3
true, true, false              // for example, x1 = true, x2 = true, x3 = false
or or not                       // store in the order of a complete binary tree
and and empty x3                // empty means an empty node
x1 not not x3 empty empty empty
empty empty empty x2 empty x1
```

3. input.txt에 해당하는 threaded binary tree를 포인터와 동적 메모리 할당을 사용하여 구성하고 하나의 노드를 삽입하는 프로그램을 작성하시오. (3점)

```
%cat input.txt
n                                // total # of nodes, where n = 10
3  8  12  5  17  23            // store in the order of a complete binary tree
78 0  6   7  0  29            // 0 means an empty node
+  17 R  55                    // 노드 17의 오른쪽 child로 노드 55를 삽입
```

4. input.txt에 해당하는 threaded binary tree를 포인터와 동적 메모리 할당을 사용하여 구성하고 하나의 노드를 제거하는 프로그램을 작성하시오. (3점)

```
%cat input.txt
n                                // total # of nodes, where n = 10
3  8  12  5  17  23            // store in the order of a complete binary tree
78 0  6   7  0  29            // 0 means an empty node
-  8                           // 노드 8을 삭제
```