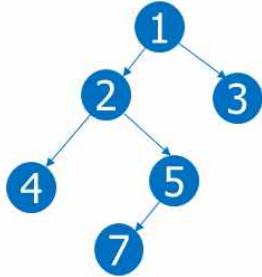


자료구조 6주차 과제

제출일: 2022-10-11

학번/이름: 2016121150 / 윤준영

문1) 다음 트리에 대하여 물음에 답하시오.



1) (7) 노드의 조상 노드들의 값을 모두 쓰시오.

1 2 5 7

2) 노드 2의 degree는 얼마인가?

2

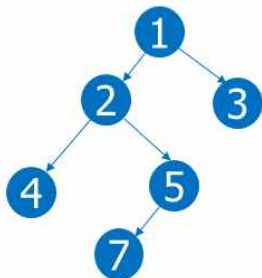
3) 위 나무의 높이는 얼마인가?

4

4) 교재의 배열을 사용한 이진 나무 표현법을 사용하여 위 이진나무를 표현하시오.

0	1	2	3	4	5	6	7	8	9	10
	1	2	3	4	5					7

문2) 문1의 트리에 대하여 다음 물음에 답하시오.



1) 위 트리를 전위순회, 중위순회, 후위순회할 경우 방문하는 순서를 각각 쓰시오.

전위 순회: 1 > 2 > 4 > 5 > 7 > 3

중위 순회: 4 > 2 > 7 > 5 > 1 > 3

후위 순회: 4 > 7 > 5 > 2 > 3 > 1

2) 위 트리를 다음 함수로 순회할 경우 스택의 변화를 모두 쓰시오.

```
void iterative_postorder_traverse(node *n) {
    int tag;
    node *t = n;
    init_stack();
    push(t); push(1);
    while ((tag = pop()) != NULL) {
        t = pop();
        switch(tag) {
            case 1 :
                if (t == tail) break;
                push(t); push(2);
                push(t->right); push(1);
                push(t->left); push(1);
                break;
            case 2 :
                visit(t);
        }
    }
}
```

stack: 1 1

tag = 1, t = 1 / stack: 1 2 3 1 2 1

tag = 1, t = 2 / stack: 1 2 3 1 2 2 5 1 4 1

tag = 1, t = 4 / stack: 1 2 3 1 2 2 5 1 4 2 tail 1 tail 1

tag = 1, t = tail / stack: 1 2 3 1 2 2 5 1 4 2 tail 1

tag = 1, t = tail / stack: 1 2 3 1 2 2 5 1 4 2

tag = 2, t = 4 / stack: 1 2 3 1 2 2 5 1 / **visit(4)**

tag = 1, t = 5 / stack: 1 2 3 1 2 2 5 2 tail 1 7 1

tag = 1, t = 7 / stack: 1 2 3 1 2 2 5 2 tail 1 7 2 tail 1 tail 1

tag = 1, t = tail / stack: 1 2 3 1 2 2 5 2 tail 1 7 2 tail 1

tag = 1, t = tail / stack: 1 2 3 1 2 2 5 2 tail 1 7 2

tag = 2, t = 7 / stack: 1 2 3 1 2 2 5 2 tail 1 / **visit(7)**

tag = 1, t = tail / stack: 1 2 3 1 2 2 5 2

tag = 2, t = 5 / stack: 1 2 3 1 2 2 / **visit(5)**

tag = 2, t = 2 / stack: 1 2 3 1 / **visit(2)**

tag = 1, t = 3 / stack: 1 2 3 2 tail 1 tail 1

tag = 1, t = tail / stack: 1 2 3 2 tail 1

tag = 1, t = tail / stack: 1 2 3 2

tag = 2, t = 3 / stack: 1 2 / **visit(3)**

tag = 2, t = 1 / stack: - / **visit(1)**

=> visit : 4 > 7 > 5 > 2 > 3 > 1

문3) 다음은 어떤 이진 트리를 순회한 결과이다.

- inorder traverse : 1 2 3 4 5 6 7 8
- postorder traverse : 2 1 3 5 6 8 7 4

1 2 3 4 5 6 7 8 / 2 1 3 5 6 8 7 4
 (1 2 3) (4) (5 6 7 8) / (2 1 3) (5 6 8 7) (4)
 ((1 2) (3) ()) ((5 6) (7) (8)) / ((2 1) () (3)) ((5 6) (8) (7)) (4)
 (((()) (1) (2)) (3) ()) (((5) (6) ()) (7) (8))
 / (((()) (2) (1)) () (3)) (((5) () (6)) (8) (7)) (4)

1) 해당 트리를 그리시오.

Postorder traverse에서 마지막 노드는 뿌리 노드이고, inorder traverse는 뿌리 노드 4를 기준으로 왼쪽 subtree와 오른쪽 subtree로 나눌 수 있다. 왼쪽 subtree는 1, 2, 3를 포함하고, 오른쪽 subtree는 5, 6, 7, 8을 포함한다.

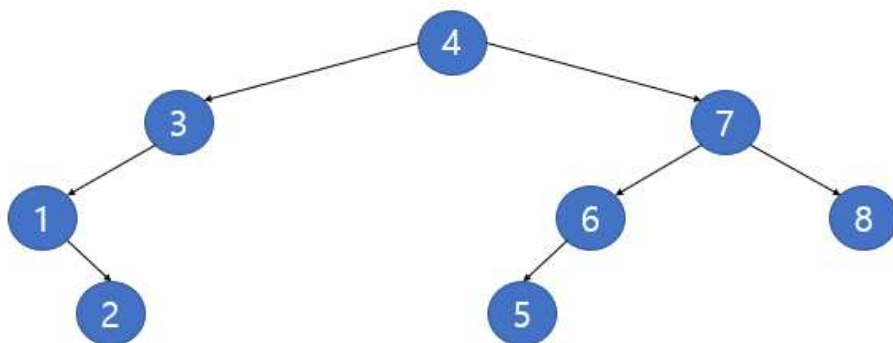
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	4														

다시 postorder traverse를 이용하여 왼쪽 subtree(2 1 3)의 뿌리 노드는 3이고, 오른쪽 subtree(5 6 8 7)의 뿌리 노드는 7이 된다. Inorder traverse를 이용하여 왼쪽 tree(2, 1, 3)의 뿌리 노드 3을 통해 분리하면 오른쪽 subtree는 공집합이 되며, 왼쪽 subtree는 1, 2를 포함하게 된다. 오른쪽 tree(5, 6, 7, 8)의 경우 뿌리 노드 8을 이용하여 분리하면, 왼쪽 subtree는 5, 6을 포함하고, 오른쪽 subtree는 8이 된다.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	4	3	7												

Postorder traverse를 이용하면, (2, 1) tree의 뿌리 노드는 1이 되며, (5, 6) tree의 뿌리 노드는 6임을 알 수 있다. Inorder traverse를 이용하면, (1, 2) tree에서 왼쪽 subtree는 공집합, 오른쪽 subtree는 2임을 알 수 있다. (5, 6) tree에서는 왼쪽 subtree는 5, 오른쪽 subtree는 공집합임을 알 수 있다.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	4	3	7	1		6	8		2			5			

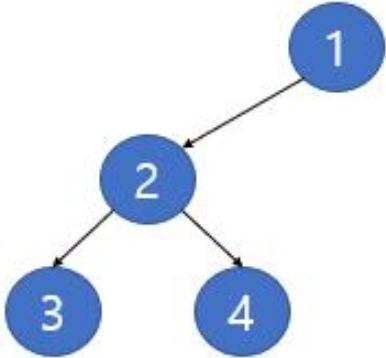
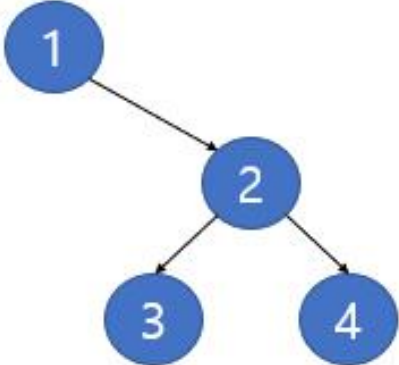


2) 한 이진 트리의 postorder 순회와 preorder 순회를 제시하면 해당 이진 트리를 유일하게 결정할 수 있는가? 주장을 정당화하시오.

전위순회 : 뿌리-왼쪽-오른쪽

후위순회 : 왼쪽-오른쪽-뿌리

유일하게 결정할 수 없다. 만약 왼쪽 subtree가 공집합이거나 오른쪽 subtree가 공집합일 경우, 전위순회와 후위순회는 왼쪽 subtree와 오른쪽 subtree가 붙어서 표현되기 때문에 이 subtree가 왼쪽 subtree인지 오른쪽 subtree인지 구분할 수 없다. 예를 들면 다음과 같다.

Tree	 <pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 2 --> 4((4)) </pre>	 <pre> graph TD 1((1)) --> 2((2)) 2 --> 3((3)) 2 --> 4((4)) </pre>
전위 순회	1 2 3 4	1 2 3 4
후위 순회	3 4 2 1	3 4 2 1