

자료구조 실습 보고서

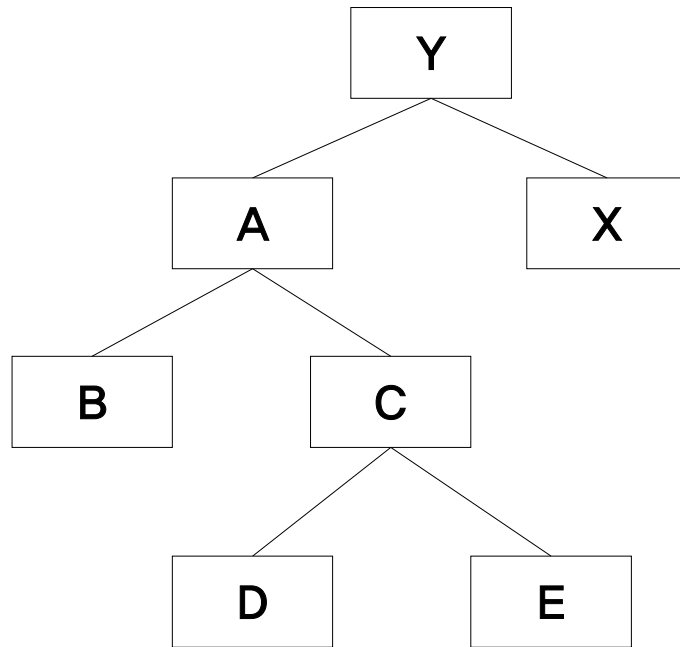
[제 11 주]

제출일 : 2018.5.15.화요일

학번 : 201604137

이름 : 김예지

〈 1. 생성된 트리 그리기 〉



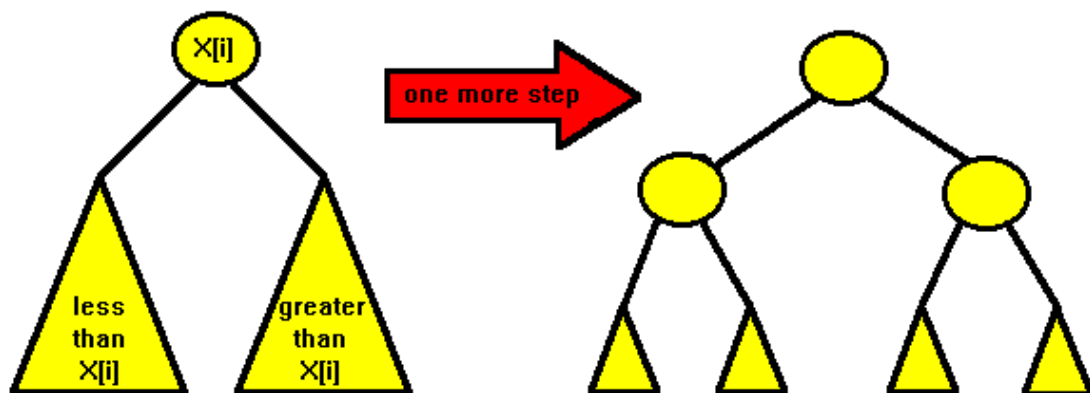
〈 2. 현재의 트리를 실습과 같이 배열로 구현할 때 〉

[1]	Y
[2]	A
[3]	X
[4]	B
[5]	C
[10]	D
[11]	E

[프로그램 설명서] 이진트리순회

이진트리란 모든 내부 node들이 둘 이하의 자식node를 갖는 나무, 노드가 하나도 없는 공집합 이거나 root node를 기준으로 왼쪽 이진나무, 오른쪽 이진나무로 이루어진 집합이다.

트리 순회(Tree traversal)는 트리 구조에서 각각의 노드를 정확히 한 번만, 체계적인 방법으로 방문하는 과정을 말한다. 트리의 모든 노드들을 방문하는 것을 트리순회라고 한다. 트리 구조의 순회에는 많은 방법이 존재한다. 이진 트리의 루트 노드에서 시작해서, 세 가지 주요 단계를 거치며 순회를 진행하는데, 그 단계에는 현재 노드를 방문하는 것, 왼쪽 자식 노드를 순회하는 것과 오른쪽 자식 노드를 순회하는 것이 있다. 이러한 과정은 재귀로 쉽게 설명할 수 있다.(아래 그림에서 서브트리는 세모)



종류는 전위순회(pre-order), 중위순회(in-order), 후위순회(post-order)가 있다. 전위순회는 root→left→right, 중위순회는 left→root→right, 후위순회는 left→right→root 순이다.

[실행 결과 분석]

```
public void preorderPrint() { // root -> left -> right 전위순회
    System.out.print(root + " ");

    if (left != null) {
        left.preorderPrint();
    }

    if (right != null) {
        right.preorderPrint();
    }
}
```

전위 순회: root를 먼저 출력 해준 뒤에 왼쪽노드가 존재하면 그 노트를 다시 루트로 잡고 왼쪽노드가 존재하는지 본다. 재귀로 다 확인 한 다음 오른쪽 노드를 확인하는 절차를 밟는다.

```
public void inorderPrint() { // left -> root -> right 중위순회

    if (left != null) {
        left.inorderPrint();
    }
    System.out.print(root + " ");

    if (right != null) {
        right.inorderPrint();
    }
}
```

중위 순회: 왼쪽 노드가 존재하는지 살펴본 후 제일 왼쪽 끝 노드로 간 다음 본인을 루트로 잡고 출력한다. 그리고 오른쪽노드를 살펴보고 있으면 루트로 잡고 출력한다. 이런식으로 출력!

```
public void postorderPrint() { // left -> right -> root 후위순회

    if (left != null) {
        left.postorderPrint();
    }

    if (right != null) {
        right.postorderPrint();
    }

    System.out.print(root + " ");
}
```

후위 순회: 마찬가지로 왼쪽->오른쪽->루트순으로 방문해서 출력한다.