자료구조

09

과목명 : 자료구조

분반 : 06

담당교수 : 박정희

학번 : 20200254 6

이름 : 임우진

목차

1. 문제 요구 사항

Binary Tree, 즉 이진 트리와 필요한 메서드들을 구현해야 한다.

이 문제를 해결하기 위해 아래 세가지 클래스를 구현한다.

Main 클래스, Node클래스, BinaryTree클래스를 생성하고

Main클래스는 트리를 생성하고 inorder, pathLength, numLeaves 메서드를 호출하여 양식에 맞게 출력하는 코드를 작성한다.

Node클래스는 좌우 레퍼런스를 get, set 할 수 있도록 필드와 메서드들을 작성한다.

BinaryTree클래스는 이진 트리 생성을 위한 필드, 메서드와 Main클래스에서 호출할 inorder, pathLength, numLeaves 메서드를 구현한다.

1) numLevaes(..): 트리의 leaf node의 수를 반환하는 순환 메소드

2) pathLength(..): 루트로부터 모든 노드까지 경로 길이의 합을 구하는 순환 메소드

2. pathLength / numLeaves

1) numLeaves

```
17      public int numLeaves(Node n){
18          int cnt = 0;
19          if(n != null){
20              if(n.getLeft() == null && n.getRight() == null){
21                  return 1;
22              }else{
23                  return cnt = numLeaves(n.getLeft())+numLeaves(n.getRight());
24              }
25          }
26          return cnt;
27      }
```

루트 노드를 매개 변수로 받고 cnt 변수를 0으로 초기화하고

트리를 순회하며 자식 노드가 없을 때 (리프노드일 때) cnt를 1 증가

2) pathlength

```java
    public int count(Node n){
        if(n==null){
            return 0;
        }else{
            return count(n.getLeft())+count(n.getRight())+1;
        }
    }
```

```java
public int pathLength(Node n, int k){
    if(k==1||k==0) {
        return 0;
    }
    int numOfLeftNodes = count(n.getLeft());
    int numOfRightNodes = count(n.getRight());
    return pathLength(n.getLeft(),numOfLeftNodes)+pathLength(n.getRight(),numOfRightNodes)+k-1;

}
```

pathLength에서 사용할 count메서드도 만들어준다.

count메서드는 입력 받은 노드 n에 대해 n을 루트로 하는 서브트리의 노드 수를 return

null이면 0 return

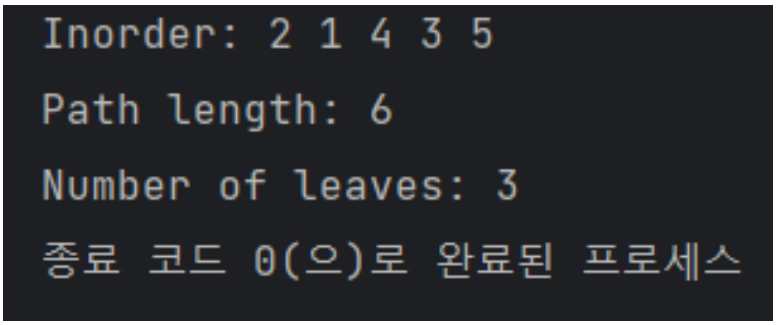
그렇지 않으면 왼쪽, 오른쪽 서브트리의 노드 수를 순환적으로 계산하고 1을 더한 값을 return

pathLength메서드는 root노드와 총 노드 수를 입력 받아

루트의 양쪽 서브트리를 고려하여 순환적으로 계산하여

경로의 총합을 return

3. 실행 화면 캡쳐


Inorder: 2 1 4 3 5
Path length: 6
Number of leaves: 3
종료 코드 0(으)로 완료된 프로세스