

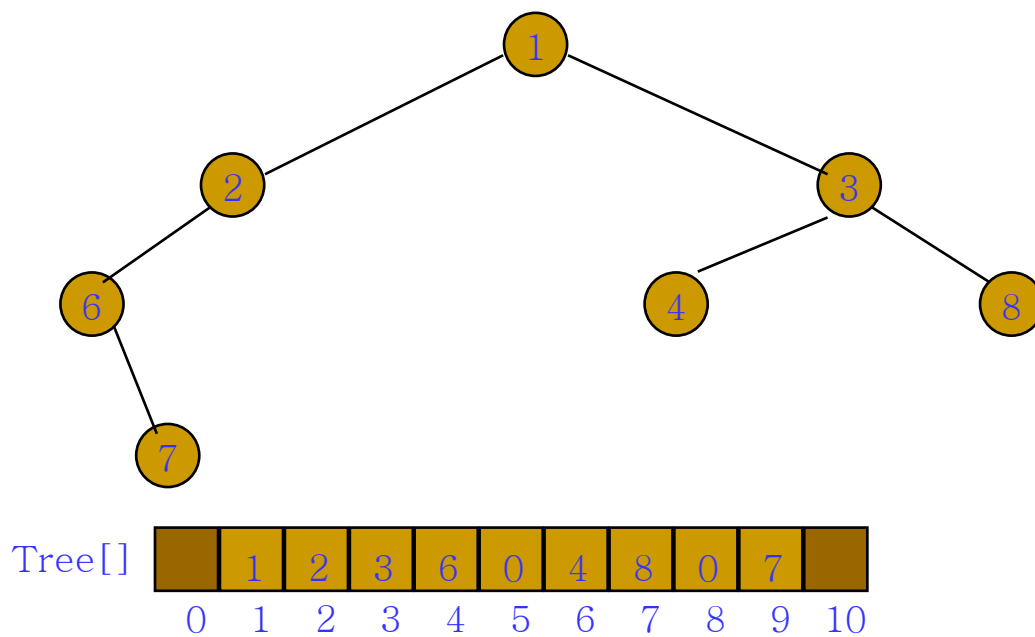
자료 구조 Lab 006 :

lab006.zip 파일 : LabTest.java lab006.java lab.in lab.out

제출

lab006.java 를 학번.java 로 변경하여 이 파일 한 개만 제출할 것.

다음은 배열을 이용하여 Binary Tree를 구현하는 내용이다. 클래스 ArrayTree은 아래 그림과 같이 binary tree를 배열로 표현하는데 1이상의 정수만을 저장하고, 노드가 존재하지 않는 경우는 0으로 표시한다. 아래 그림은 노드 형태로 표현된 트리과 배열 내에 저장된 값의 모습을 보여준다.



이러한 자료구조를 이용하여 간단한 프로그램을 작성한다.

수행 예는 다음과 같다.

```
kmucs@localhost: ~/dbbox/classes171/ds/lab17/lab17006
kmucs@localhost:~/dbbox/classes171/ds/lab17/lab17006$ java LabTest
Tree > init 1 2 3 6 0 4 8 0 7 -1

Tree : - 1 2 3 6 0 4 8 0 7
Tree > pre 1
1 2 6 7 3 4 8
Tree : - 1 2 3 6 0 4 8 0 7
Tree > post 1
7 6 2 4 8 3 1
Tree : - 1 2 3 6 0 4 8 0 7
Tree > in 1
6 7 2 1 4 3 8
Tree : - 1 2 3 6 0 4 8 0 7
Tree > init 1 2 3 -1

Tree : - 1 2 3
Tree > pre 1
1 2 3
Tree : - 1 2 3
Tree > pre 2
2
Tree : - 1 2 3
Tree > post 1
2 3 1
Tree : - 1 2 3
Tree > 
```

사용자가 사용하는 명령어의 syntax는 다음과 같다. main() 함수에 정의되어 있다.

- init tree node values

tree를 구성하는 배열인 tree[]를 초기화 시키는 역할을 한다. 주어진 정수 값들을 -1이 나올 때까지 읽어서 그 내용대로 tree의 1번 index부터 차례로 집어 넣는다. 이 명령어에서 사용하는 배열의 원소들은 tree의 조건을 만족해야 한다. 이를 위해 노드가 존재하지 않는 인덱스는 0을 입력해야 한다. 이미 다 구현되었으니 그냥 사용하기만 하면 된다.

- pre index

index가 가리키는 노드를 root로 하는 subtree를 preorder로 방문(traverse)한다. 방문한 노드의 값을 출력한다.

- in index

index가 가리키는 노드를 root로 하는 subtree를 inorder로 방문(traverse)한다. 방문한 노드의 값을 출력한다.

- post index

index가 가리키는 노드를 root로 하는 subtree를 postorder로 방문(traverse)한다. 방문한 노드의 값을 출력한다.

Tree : 으로 시작하는 라인은 현재 배열의 상태를 보여준다. 배열의 0번 원소는 사용하지

않기 때문에 -로 표현하였다. 이미 구현되어 있다.

위 그림에서 pre 1 명령에 의해 1번 노드를 root로 하는 서브 트리를 preorder로 방문하면 "1 2 6 7 3 4 8"로 메시지가 출력되어야 한다.

이 내용을 구현하기 위해 다음 3 함수를 구현해야 한다.

- `void PreOrder(int node);`

파라미터 node는 Tree의 한 원소의 index 인데 이 node로부터 시작하는 subtree를 preorder로 방문 (traverse) 하는 재귀 함수이다. 이 함수를 구현하기 위해서는 node의 자식 노드의 index를 알아야 한다. preorder traversal이기 때문에 현재 노드의 값을 출력한 다음, 왼쪽 자식노드의 인덱스를 가지고 PreOrder를 call하고, 다음 다시 오른쪽 자식 노드의 인덱스를 가지고 PreOrder를 call 한다.

만약 인덱스 node에 위치한 노드가 0이면 이 노드가 존재하지 않는 것이기 때문에 출력하지 않고 return 한다.

- `void InOrder(int node);`

PreOrder와 같은 내용인데 다만 Inorder로 traverse 한다.

- `void PostOrder(int node);`

PreOrder와 같은 내용인데 다만 Postorder로 traverse 한다.

주어진 .java 파일을 컴파일 하면 수행은 가능하지만 아직 구현이 안된 부분은 "NEED TO IMPLEMENT" 라는 comment가 존재한다.

프로그램 결과 테스트

```
$ diff aaa lab.out
```

또는

```
$ diff -i --strip-trailing-cr -w aaa lab.out
```