

**[511643] 자료구조 (2019-2 학기)****실습 #09 보고서**

이름	지현한
학번	20165164
소속 학과/ 대학	소프트웨어융합대학 빅데이터전공
분반	03 (담당교수: 김태운)

### <주의사항>

- 개별 과제입니다. (팀으로 진행하는 과제가 아니며, 모든 학생이 보고서를 제출해야 함)
- **각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.**
  - 소스코드/스크립트 등을 작성 한 경우, 해당 파일의 이름도 적어주세요.
- 스마트캠퍼스 제출 데드라인: **2019. 10. 30. (수요일) 23:59**
  - 데드라인을 지나서 제출하면 24 시간 단위로 20%씩 감점(5 일 경과 시 0 점)
  - 주말/휴일/학교행사 등 모든 날짜 카운트 함
  - 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0 점 처리함
  - 예외 없음
- 스마트캠퍼스에 아래의 파일을 제출 해 주세요
  - 보고서(**PDF 파일로 변환 후 제출**)
  - 보고서 파일명에 이름과 학번을 입력 해 주세요.
  - 소스코드, 스크립트, Makefile 등을 작성해야 하는 경우, 모든 파일 제출 (미 제출시 감점)

### <개요>

이번 과제는 이진트리(BinaryTree)를 구현하고 응용문제를 풀어보는 내용입니다. Let's get it!

## &lt;실습 과제&gt;

**[Q 0] 요약 [배점: 10]**

이번 과제에서 배운 내용 또는 과제 완성을 위해서 무엇을 했는지 2~3 문장으로 요약하세요.  
 답변 : 트리에 대해 더 깊이 이해하게 됐고, 재귀호출에 대해서도 더 깊이 이해하였으며,  
 재귀호출을 하는 메소드를 만들기 위해 노력했습니다.

**[Q 1] 이진트리 테스트 [배점: 30]**

강의노트와 동일하게 BinaryTree 클래스를 구현하세요. 이진 트리를 테스트하기 위한 BinaryTreeTest 클래스를 만들고, 강의노트 p. 53의 안내에 따라 main 메소드를 구현하세요. main 메소드를 실행하고 터미널 화면을 캡처 후 본 문서에 첨부하세요. 소스코드도 첨부파일로 제출해야 합니다.

답변 (터미널 화면 캡처) : BinaryTreeTest.java

```
트리 노드 수= 8
트리 높이 = 4
전위 순회: 100 200 400 800 500 300 600 700
중위 순회: 800 400 200 500 100 600 300 700
후위 순회: 800 400 500 200 600 700 300 100
레벨 순회: 100 200 300 400 500 600 700 800
동일성 검사: true
```

**[Q 2] 트리 2배열 [배점: 30]**

이전 문제 [Q 1] 에서 사용한 것 과 동일한 트리 t1 이 있습니다.

**[Task 1]** 정수가 저장된 t1 이진 트리를 int[] 배열로 저장하는 public int[] btree2intarr() 메소드를 작성하세요. (여기서는 BinaryTree 의 Node 의 Key 는 항상 int 라고 가정하고 문제에 접근해도 됩니다.) 이진 트리를 배열로 표현하는 방법은 강의노트 23, 24 페이지를 참고하세요. 메소드는 BinaryTree 클래스에 속한 멤버 메소드로 구현하세요. 직전 문제에서 사용한 t1 이진 트리에서 btree2intarr() 메소드를 호출 하고, 리턴된 정수 배열을 터미널에 출력하세요 (정수 배열을 터미널에 출력하는 메소드는 자유롭게 구현해도 됩니다). 터미널 화면을 캡처하여 본 문서에 첨부하세요. 소스코드도 첨부해서 제출해야 합니다.

**[Task 2]** `int[]` 에 저장된 이진 트리를 입력으로 받아 연결 리스트 형태의 이진 트리 표현으로 변환하는 `public void intarr2btree(int[] arr)` 메소드를 작성하세요. (여기서는 `BinaryTree` 의 `Node` 의 `Key` 는 항상 `int` 라고 가정하고 문제에 접근해드립니다.) 메소드는 `BinaryTree` 클래스에 속한 멤버 메소드로 구현하세요. 메소드는 먼저 `BinaryTree` 에 저장된 모든 노드를 삭제합니다. 다음으로, 인자로 넘어온 `arr` 배열을 이용해서 이진 트리에 노드를 하나씩 추가합니다. 새로운 `BinaryTree` 클래스 객체 `tNew` 를 만드세요. `tNew` 객체에서 `intarr2btree` 메소드를 호출하고, 인자로 **[Task 1]** 에서 리턴한 배열을 넘겨주세요. `tNew` 이진 트리를 레벨 순회 방식으로 출력하세요. 터미널 화면을 캡처하여 본 문서에 첨부하세요. 소스코드도 첨부해서 제출해야 합니다.

답변 **[Task 1, 터미널 화면 캡처]** :

Q2main.java

```
[Task 1]
0 100 200 300 400 500 600 700 800 0 0 0 0 0 0 0
```

답변 **[Task 2, 터미널 화면 캡처]** :

Q2main.java

```
[Task 2]
100 200 300 400 500 600 700 800
```

### **[Q 3] 서브트리 검색 [배점: 30]**

두개의 트리 `tA`, `tB` 가 있습니다. 트리에 저장된 데이터는 강의노트 p. 54 를 참고하세요. 두개의 트리에서 동일한 서브트리가 있는지를 검사하는 `public LinkedList<Node> subTreeCompare(BinaryTree t)` 메소드를 작성하세요. 메소드는 `BinaryTree` 의 속한 멤버 메소드입니다. `tA.subTreeCompare(tB)` 를 호출하면, `tA` 트리와 `tB` 트리를 비교하여 동일한 서브트리를 모두 찾아냅니다. 동일한 서브트리를 찾을 때 마다, 해당 서브트리의 루트 노드를 `LinkedList` 에 저장합니다 (`tA` 에서 호출했으므로 `tA` 에 속한 서브트리의 루트 노드를 리스트에 저장합니다). 마지막으로 `LinkedList` 를 리턴합니다.

**[Task]** `tA`, `tB` 두개의 트리를 생성하세요. `tA.subTreeCompare(tB)` 를 호출하세요.

리턴되는 리스트에 저장된 노드 각각을 루트 노드로 하는 (서브)트리를 모두 레벨 순회 방식으로 출력하세요). 터미널 결과는 아래와 같이 나와야 합니다. 단, 어떤 서브트리가 먼저 출력되는지, 그 순서는 아래와 달라도 됩니다.

```
Found a matching subtree 1:
300 600 700
Found a matching subtree 2:
600
Found a matching subtree 3:
700
```

터미널 화면을 캡처하여 본 문서에 첨부하세요. 소스코드도 첨부해서 제출해야 합니다.

**\* 참고: empty 인 서브트리는 고려하지 않습니다.**

답변 (터미널 화면 캡처) :

Q3main.java

BinaryTree.java

```
Found a matching subtree 1 :
300 600 700
Found a matching subtree 2 :
600
Found a matching subtree 3 :
700
```

끝! 수고하셨습니다 ㄹ