

**[511643] 자료구조****실습 #09 보고서**

<b>이름</b>	곽영주
<b>학번</b>	20175105
<b>소속 학과/대학</b>	빅데이터
<b>분반</b>	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: **2020.05.13. ~ 2020.05.19. (화요일) 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 이진 탐색 트리 (Binary Search Tree, BST)를 구현하고 활용하는 내용입니다.

## &lt;실습 과제&gt;

**[Q 0] 요약 [10 점]**

이번 과제에서 배운 내용 또는 과제 완성을 위해서 무엇을 했는지 2~3 문장으로 요약하세요.

답변: 이번 과제를 통해 배운 내용은 저번 시간에 배운 이진 트리의 노드 접근 방법을 찾는 것이 어려웠는데 이번에 배운 이진 탐색 트리는 부모 노드의 왼쪽은 작은 값, 오른쪽은 큰 값이 들어간다는 조건이 있어서 각각의 노드에 접근 방법을 찾는데 좀 더 수월했던 것 같습니다.

**[Q 1] BST 구현 및 테스트 코드 구현 [20 점]**

강의자료와 같이 Node 클래스와 BST 클래스를 구현하세요. BST 클래스를 테스트하기 위한 BSTtest 클래스를 만들고, main 메소드를 강의노트 pp.37~38 과 같이 구현하세요. [1]번 ~[8]번 과정을 모두 수행한 후, 터미널 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

소스코드 : [Q1] BST 구현 및 테스트 소스코드.txt

```
<terminated> BSTMain [Java Applicatio
10 20 25 30 35 40 45 50 60 70
Searching for 45 yields : 45
Searching for 45 yields : 99
Min value : 10
Min value : 20
20 30 40 50 60 70
```

**[Q 2] delete max, 최대의 키값을 가지는 노드 삭제하기 [20 점]**

BST 클래스에서 최댓값(Key 값 기준)을 삭제하는 `public void deleteMax()` 메소드를 구현하세요 (BST 클래스의 멤버 메소드로 구현). 강의자료 p.37 을 참고하여 bst1 트리의 초기구성과 동일한 bst2 를 생성하세요. 초기의 트리 구성을 확인하기 위해 bst2.print 메소드를 호출하세요. bst2 트리에서 최댓값을 삭제하고 print 메소드를 호출하는 과정을 3 회 반복하세요. 터미널 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

소스코드 : [Q2] delete max 소스코드.txt

```
<terminated> DeleteMaxMain [Java Ap
10 20 25 30 35 40 45 50 60 70
=== deleteMax 3번 반복 ===
10 20 25 30 35 40 45 50 60
10 20 25 30 35 40 45 50
10 20 25 30 35 40 45
```

**[Q 3] k-th smallest node 찾기 [25 점]**

BST 클래스에서 k 번째 작은 Key 를 찾아 키 값을 리턴하는 `public Key kthSmallest(int k)` 메소드를 구현하세요 (BST 클래스의 멤버 메소드로 구현). 단, 가장 작은 키는 0 번째이고, 두 번째로 작은 키가 1 번째 입니다.

강의자료 p.37 을 참고하여 bst1 트리의 초기구성과 동일한 bst3 를 생성하세요. bst3 트리에서 아래의 코드를 실행하세요

```
for(i=0;i<5;i++) 터미널에 bst3.kthSmallest(i)의 리턴 값 출력;
```

터미널 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

소스코드 : [Q3] k-th smallest node 소스코드.txt

```
<terminated> kthSmallestMain [Java A]
10 20 25 30 35 40 45 50 60 70
=== kthSmallest ===
[0] 10
[1] 20
[2] 25
[3] 30
[4] 35
```

#### [Q 4] check BST [25 점]

주어진 BST 클래스 인스턴스가 이진 탐색 트리가 맞는지를 검사하는 public boolean checkBST() 메소드를 BST 클래스의 멤버 메소드로 구현하세요. 아래와 같은 순서로 문제를 해결하세요.

- 아무런 인자를 받지않는 BST 클래스의 생성자 추가
- BST 클래스에 setRoot 메소드 추가
- BST.put 메소드를 사용하지 않고, 클래스 외부(즉, main 메소드)에서 Node 를 이어서 트리를 구성한 후, 트리의 root 를 인자로 BST 클래스의 setRoot 호출
  - (노드를 직접 연결하는 방식으로) 이진 탐색 트리 속성을 만족하는 트리(bst4) 만들기
  - (노드를 직접 연결하는 방식으로) 이진 트리 이지만 이진 탐색 트리의 속성을 만족하지 않는 트리(bst5) 만들기
  - 문제에서 사용할 두 개의 트리(bst4, bst5)는 강의노트 p.40 를 참고.

[Task 1] bst4 이진 탐색 트리를 생성하고, print() 메소드를 호출하세요. bst4.checkBST()를 호출하고, 결과를 터미널에 출력하세요. 터미널 화면을 캡처하고 본 문서에 첨부하세요.

[Task 2] bst5 이진 탐색 트리를 생성하고, print() 메소드를 호출하세요. bst5.checkBST()를 호출하고 결과를 터미널에 출력하세요. 터미널 화면을 캡처하고 본 문서에 첨부하세요.

소스코드도 제출해야 합니다.

소스코드 : [Q4] check BST 소스코드.txt

답변 [Task 1]:

```
<terminated> checkBSTMain [Java Application] C:\WProgr  
10 25 30 50 60 75 80 >> 이진 탐색 트리인가? true
```

답변 [Task 2]:

```
10 25 30 50 80 75 60 >> 이진 탐색 트리인가? false
```

끝! 수고하셨습니다 ☺