

[실습 및 과제 3]

Youn-Hee Han

LINK@KOREATECH

<http://link.koreatech.ac.kr>

제시 문제

◆ [문제 1] BST에 대한 Python 프로그램을 작성해보자.

- `binary_search_tree.py` 꼼꼼하게 분석하기
- 완성되지 않은 다음 두 함수를 작성한다.
 - 노드 삽입: `def tree_insert(self, key=None):`
 - 노드 삭제: `def _delete_node(self, node_to_be_deleted):`
- 해당 프로그램 수행 결과
 - 반드시 오른쪽 수행결과와 동일하게 나와야 함

```
1 3 4 6 7 8 10 14 : SIZE = 8

Search 10: [NODE - key: 10, parent key: 8]

Search 100: None

Delete 10
1 3 4 6 7 8 14 : SIZE = 7

Delete 10
1 3 4 6 7 8 14 : SIZE = 7

Delete 3
1 4 6 7 8 14 : SIZE = 6

Delete 4
1 6 7 8 14 : SIZE = 5

Delete 8
1 6 7 14 : SIZE = 4
```

제시 문제

◆ [문제 1] BST에 대한 Python 프로그램을 작성해보자.

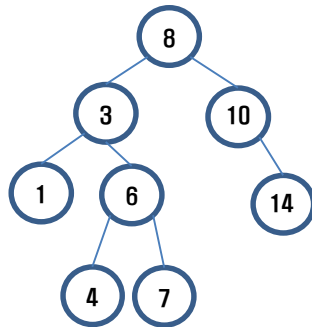
– 참고 사항

- bst.print_bst()에 대한 이해

메인파트 시작부분

```
bst = BST()
```

```
bst.tree_insert(8)
bst.tree_insert(3)
bst.tree_insert(1)
bst.tree_insert(6)
bst.tree_insert(7)
bst.tree_insert(10)
bst.tree_insert(14)
bst.tree_insert(4)
```



```
bst.print_bst()
```

```
>> 1 3 4 6 7 8 10 14 : SIZE = 8
```

```
# 중위 순회 (Inorder traversal)
def _inorder(self, node=None):
```

```
    if node is not None:
```

```
        # 왼쪽으로 순회
```

```
        self._inorder(node.left)
```

```
        # 중앙 노드 키 출력
```

```
        print(str(node.key), end=" ")
```

```
        # 오른쪽으로 순회
```

```
        self._inorder(node.right)
```

```
def print_bst(self):
```

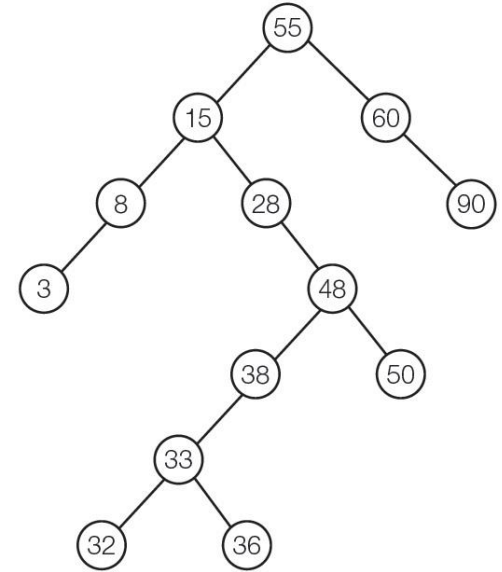
```
    self._inorder(self.root)
```

```
    print(": SIZE = {0}".format(self.size))
```

제시 문제

◆ [문제 2]

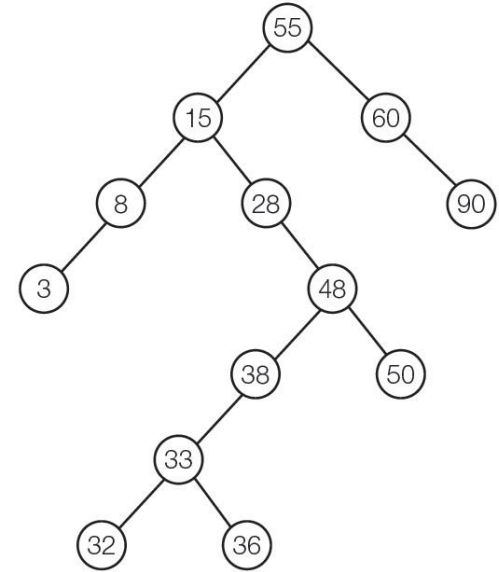
- [2-1] 오른쪽 BST에 대한 중위 순회 결과를 한 줄로 제시하시오.
- [2-2] 아무런 노드도 존재하지 않는 비어있는 BST에서 오른쪽 BST를 만들기 위하여 처음에 키 55 노드를 삽입하면서 구성한다고 가정하고, 순차적으로 삽입되는 노드의 키 값을 순서대로 나열하시오. (아래와 같이 키 값을 화살표와 함께 순차적 나열한 답안 제시)
 - 55 → ...
- [2-3] [문제 1]에서 작성한 코드의 메인 파트를 처음부터 다시 작성하여 (즉, `bst.tree_insert(55)` 등을 순차적으로 다시 작성) 위와 같은 BST를 구성하고 `bst.print_bst()`를 통해 BST를 출력한 결과가 [2-1]에서 제시한 중위순회 결과와 동일한지 비교하시오.
 - 메인파트 재작성 코드 내용 제시
 - 출력결과 캡처화면 제시



제시 문제

◆ [문제 3]

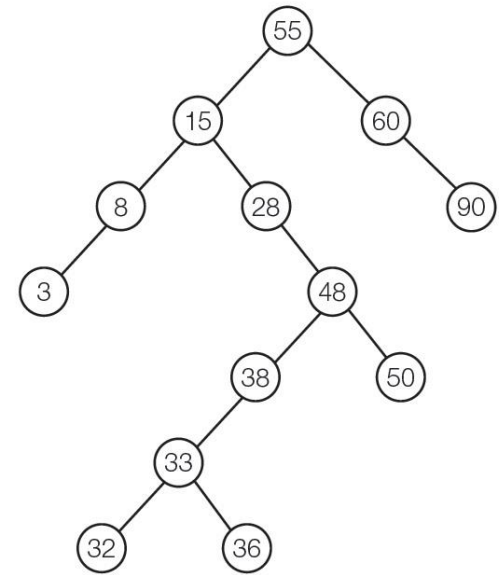
- [3-1] 앞선 [문제 2]에서 구성한 오른쪽 BST에서 **키 40, 키 39, 키 80**을 삽입하는 과정을 [알고리즘 6-4]에 입각하여 제시하시오.
 - 서술형으로 답안 작성
 - 각 키가 삽입된 그림(또는 삽입되는 과정)을 연습장 등에 그리고 사진을 찍어 보고서에 삽입 추천
- [3-2] [문제 3-1]에서 제시된 3개의 키 노드 대하여 순차적으로 삽입된 직후의 BST에 대한 중위 순회 결과를 한줄씩 차례로 제시하시오.
- [3-3] 앞선 [문제 1]에서 작성한 코드 및 [문제 2-3]에서 작성한 메인 파트 이후에, 직접 추가적으로 [문제 3-1]에서 제시된 3개의 키 노드에 대한 삽입 코드를 넣어서 위 [문제 3-1] 및 [문제 3-2]에서 제시한 답안과 일치하는지 조사하시오.
 - 각 키가 삽입될 때마다 출력된 중위 순회 결과 캡처화면 제시



제시 문제

◆ [문제 4]

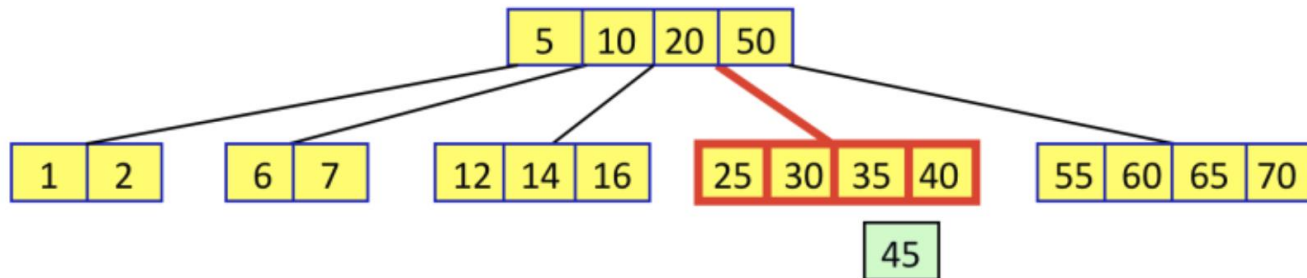
- [4-1] 앞선 [문제 3]에서 구성한 오른쪽 BST에서 키 40, 키 39, 키 80을 삽입한 이후의 BST에 대해 **키 15 노드**를 [알고리즘 6-6]에 입각하여 제시하시오.
 - 서술형으로 답안 작성
 - 해당 키가 삭제된 그림(또는 삭제되는 과정)을 연습장 등에 그리고 사진을 찍어 보고서에 삽입 추천
- [4-2] [문제 4-1]에서 제시된 BST에 대한 중위 순회 결과를 제시하시오.
- [4-3] 앞선 [문제 1]에서 작성한 코드 및 [문제 3-3]에서 작성한 메인 파트 이후에, 직접 추가적으로 [문제 4-1]에서 제시된 1개의 키 노드에 대한 삭제 코드를 넣어서 위 [문제 4-1] 및 [문제 4-2]에서 제시한 답안과 일치하는지 조사하시오.
 - 해당 키가 삭제된 이후 출력된 중위 순회 결과 캡처화면 제시



제시 문제

◆ [문제 5] B-Tree에 대한 다음 문제를 해결하시오.

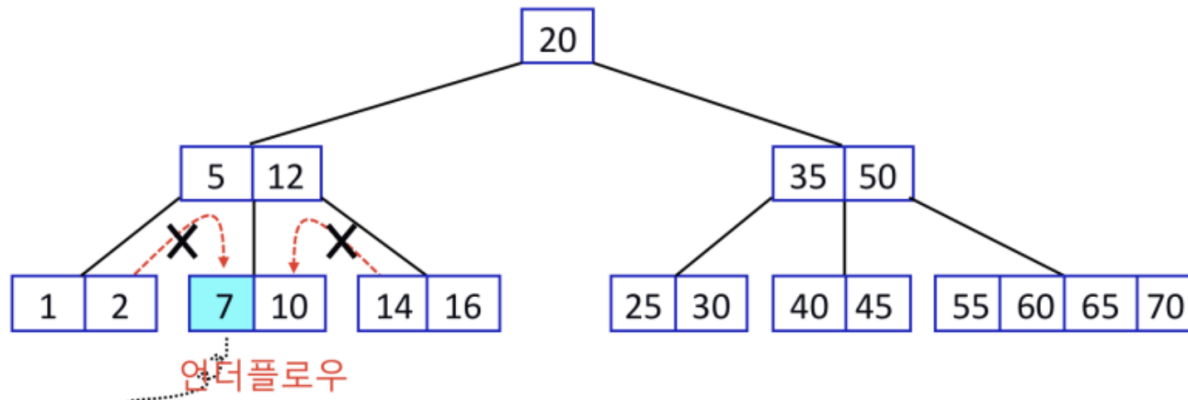
- [5-1] 다음 그림에 주어진 B-Tree($k=4$)에서 새롭게 키 45를 지닌 노드를 삽입하려고 한다. 교재에서 주어진 [알고리즘 6-7]에 입각하여 삽입하는 과정을 설명하고, 최종 결과 그림을 제시하시오.
 - 최종 결과 그림은 꼭 제시
 - 중간 과정은 말로 설명해도 되지만 가급적 그림 제시 추천



제시 문제

◆ [문제 5] B-Tree에 대한 다음 문제를 해결하시오.

- [5-2] 다음 그림에 주어진 B-Tree($k=4$)에서 키 7을 지닌 노드를 삭제하려고 한다. 교재에서 주어진 [알고리즘 6-8]에 입각하여 키 7을 지닌 노드가 Underflow가 발생하고 있다. 이후 부터 삭제하는 과정을 설명하고, 최종적으로 키 7을 지닌 노드를 삭제한 결과 그림을 제시하시오.
 - 최종 결과 그림은 꼭 제시
 - 중간 과정은 말로 설명해도 되지만 가급적 그림 제시 추천



제시 문제

◆ [문제 6]

– B-트리의

- 1) 한 블록(페이지) 크기가 35,448 바이트이고,
- 2) 키의 크기가 32바이트이고,
- 3) 페이지 번호 크기가 4 바이트일 때,

임의의 노드가 가질 수 있는 최대 키 개수와 최소 키 개수는 어떻게 되는지 풀이과정과 함께 제시하시오.

숙제 요구사항

◆ 보고서(Report) 요구사항

- 프로그램 코드를 정확하게 Report에 작성할 것
- 주석을 충분하게 넣을 것
- 프로그램 내에 작성한 특정한 부분 코드에 대한 설명이 구체적으로 필요한 경우는 프로그램 전체 코드 바로 다음에 해당 설명을 적을 것
- 프로그램 수행 화면을 캡처하여 넣을 것
- 캡처화면은 다양하게 많이 넣을 것
- 자신이 취득한 기술적 사항/고찰 내용을 생각한 바 그대로 적을 것

숙제 요구사항

◆ 제출형태 및 방법

– 보고서 파일 작성

- HWP 또는 MS-Word로 작성하여 **PDF 산출 및 제출**
- 서론, 본론, 결론의 구조를 가진 보고서
 - 표지
 - 서론: Homework의 내용 및 목적
 - 본론: 각각의 문제를 코딩한 이후 앞에 있는 요구사항에 따라 작성
 - 결론: 숙제를 한 이후의 느낀점, 하고싶은 말, 또는 불평~~
- 파일명: 알고리즘-3차-홍길동-2020111222.pdf

– 숙제 제출 방식

- <http://el.koreatech.ac.kr> 의 “과제 ” 메뉴
- 숙제 게시물 제목: 알고리즘-3차-홍길동-2020111222

– 숙제를 포함한 여러가지 질문 방법

- <http://el.koreatech.ac.kr> 의 “질의응답” 게시판 활용할 것