Computing Foundation for Data Science

Project #3

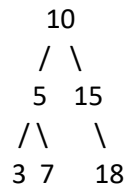
제출기한: 2021.10.26 10:59AM

문제 1. Sum of subtree in BST

Binary Search Tree 에서 주어진 값을 지닌 노드를 찾고, 해당 노드의 left subtree 와 right subtree 에 속하는 노드들의 모든 값들의 합을 구하는 P1 함수를 구현하여라.

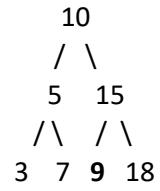
*BST 의 각 Node 는 TreeNode class 로 정의되며, 이 정의는 BST_Helper.py 파일을 확인하면 된다. printTree() Method 를 호출하면 tree 의 각 노드의 값을 리스트로 만들어 Return 해준다.

- P1 함수의 Input 으로는 root node 와 찾고자 하는 노드의 값 x 가 들어간다.
- 모든 node 의 값은 BST 안에서 유일하다.
- 코드 구현 확인을 위하여 BST_Helper.py 파일 내부에 아래 함수가 구현되어 있다.
    - create_linked_bst: 정수 List 를 입력받아 BST 를 구성하고 root node 를 반환. 이때 정수 List 는 첫 element 를 root node 의 값, 두번째는 root node 의 왼쪽 값, 세번째는 root node 의 오른쪽 값… 으로 Breadth first traversal 의 순서로 입력되어야 한다. 또한 가지에 해당하는 값이 없을 경우 None 을 입력한다.
    - 예를 들어, 다음 list 를 입력하면 [10,5,15,3,7, None, 18] 아래와 같은 BST 를 생성하고 root node 를 return 한다.



- BST 의 특성을 이용하여 함수를 구현하도록 한다. 이를 위해 BST 의 Child 의 조건이 위배된 형태로 Input 이 주어질 수 있다.
    - left node y of node x has value less than x's value
    - right node y of node x has value greater than x's value

- child node 의 경우는 해당 노드의 값보다 크거나 작아야 하지만 grand children 부터는 이 조건에 구애를 받지 않는다. 예를 들어 아래와 같은 형태의 BST 도 Input 으로 주어질 수 있다. 9 는 10 보다 작지만 오른쪽 Subtree 에 위치한다. 이 때의 계산 방법은 아래 예시 7 을 참고하라.



- subtree 가 없으면 0 이 반환되도록 한다. (예시 2 참고)
- root 가 None 일 때는 0 이 반환되도록 한다. (예시 3 참고)
- x 가 주어진 BST 에 없으면 0 이 반환되도록 한다. (예시 4 참고)

예시 1) 10 의 left, right subtree 에 속하는 모든 값들의 합은 5+15+3+7+9+18=57

```
>>> root = create_linked_bst([10,5,15,3,7, 9, 18])
>>> P1(root,10)
57
```

예시 2)

```
>>> root = create_linked_bst([10])
>>> P1(root,10)
0
```

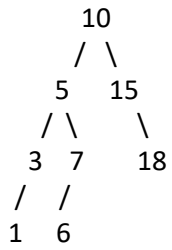
예시 3)

```
>>> root = None
>>> P1(root,15)
0
```

예시 4)

```
>>> root = create_linked_bst([10,5,15,3,7, 9, 18])
>>> P1(root,20)
0
```

예시 5) 5 의 left, right subtree 에 속하는 모든 값들의 합은 3+7+1+6=17

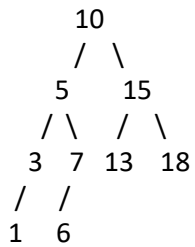


```
>>> root = create_linked_bst ([10,5,15,3,7,None, 18,1,None,6])

>>> P1(root,5)

17
```

예시 6) 15 의 left, right subtree 에 속하는 모든 값들의 합은 13+18=31

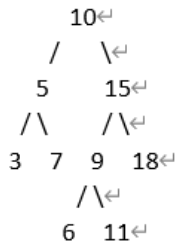


```
>>> root = create_linked_bst ([10,5,15,3,7,13,18,1,None,6])

>>> P1(root,15)

31
```

예시 7) 9 가 10 의 left subtree 가 아닌 right subtree 에 있기 때문에 검색을 거치지 못하여 17 이 아닌 0 이 결과값으로 반환되어야 한다.



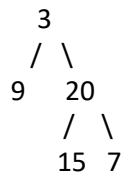
```
>>> root = create_linked_bst([10,5,15,3,7,9,18,None,None,None,None,6,11])
>>> P1(root,9)
0
```

문제 2. Binary Tree Level Order Traversal

Binary tree 에 대하여 Bottom-up level order traversal 을 하고 return 하는 함수를 작성하시오. 즉, Leaf 부터 Root 까지 각 level 에 대해, 왼쪽에서부터 오른쪽 순서로 저장되어야 한다.

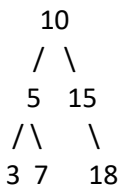
- **이중 리스트** 형태로 return 해야하며, 같은 Depth 인 Node 들의 값이 같은 List 에 저장되어야 한다.
- input 으로는 binary tree 의 root 가 주어진다.

예시 1)




```
>>> root = create_linked_bst ([3,9,20,None,None,15,7])
>>> P2(root)
[[15, 7], [9, 20], [3]]
```

예시 2)



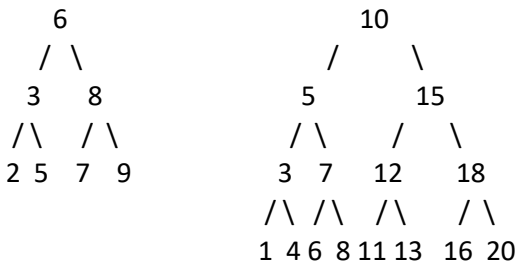

```
>>> root = create_linked_bst([10,5,15,3,7,None,18])
>>> P2(root)
[[3, 7, 18], [5, 15], [10]]
```

예시 3)

```
>>> root = create_linked_bst([5,3,6,2,4,None,7])
>>> P2(root)
[[2, 4, 7], [3, 6], [5]]
```

문제 3. Insert value into BST: make full BST

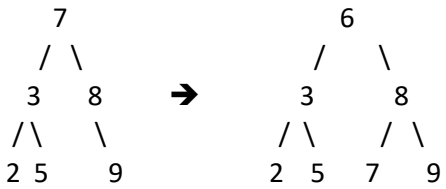
BST 에서 모든 Depth 의 모든 Node 에 값이 있는 경우 Full BST 라고 하자. 예를 들면 아래와 같은 BST 를 지칭한다.



Full BST 에서 1 개의 Node 만 비어있는 BST 에 1 개의 값을 추가하여 Full BST 로 만들고, **root node** 를 return 하는 함수를 작성하여라.

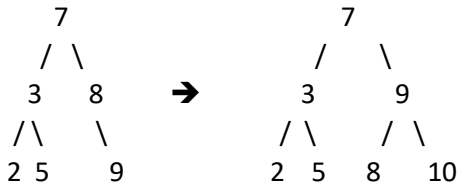
- Input 은 BST 의 root node 와 추가할 정수값이다.
- Node 의 모든 값은 정수라고 가정한다.
- 추가되는 값은 Input BST 에 존재하지 않는다고 가정한다.
- Tree 를 새로 생성하는 방식으로의 구현은 인정하지 않는다.

예시 1) 왼쪽의 BST 에 6 을 넣을 경우 오른쪽과 같은 BST 를 Return 해야 한다.



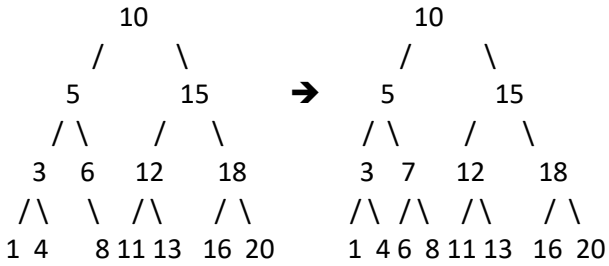

```
>>> root = create_linked_bst( [7,3,8,2,5,None,9])
>>> fullBST = P3(root, 6)
>>> print(fullBST .printTree())
[6, 3, 8, 2, 5, 7, 9]
```

예시 2) 왼쪽의 BST 에 10 을 넣을 경우 오른쪽과 같은 BST 를 Return 해야 한다.



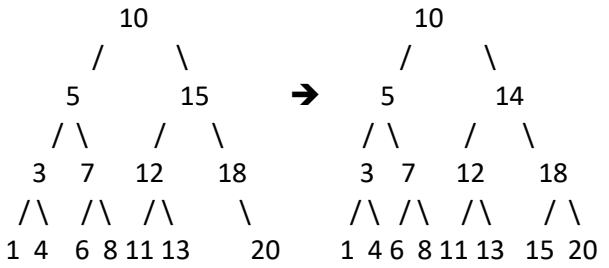
```
>>> root = create_linked_bst( [7,3,8,2,5,None,9])
>>> fullBST = P3(root, 10)
>>> print(fullBST .printTree())
[7, 3, 9, 2, 5, 8, 10]
```

예시 3) 왼쪽의 BST 에 7 을 넣을 경우 오른쪽과 같은 BST 를 Return 해야 한다.



```
>>> root = create_linked_bst( [10,5,15,3,6,12,18,1,4,None,8,11,13,16,20])
>>> fullBST = P3(root, 7)
>>> print(fullBST .printTree())
[10, 5, 15, 3, 7, 12, 18, 1, 4, 6, 8, 11, 13, 16, 20]
```

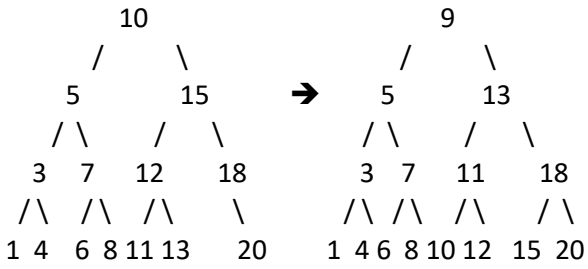
예시 4) 왼쪽의 BST 에 14 를 넣을 경우 오른쪽과 같은 BST 를 Return 해야 한다.



```
>>> root = create_linked_bst( [10,5,15,3,7,12,18,1,4,6,8,11,13,None,20])
>>> fullBST = P3(root, 14)
```

```
>>> print(fullBST .printTree())
[10, 5, 14, 3, 7, 12, 18, 1, 4, 6, 8, 11, 13, 15, 20]
```

예시 5) 왼쪽의 BST 에 9 를 넣을 경우 오른쪽과 같은 BST 를 Return 해야 한다.




```
>>> root = create_linked_bst( [10,5,15,3,7,12,18,1,4,6,8,11,13,None,20])
>>> fullBST = P3(root, 9)
>>> print(fullBST .printTree())
[9, 5, 13, 3, 7, 11, 18, 1, 4, 6, 8, 10, 12, 15, 20]
```

주의사항

- 코드를 Jupyter Notebook 에서 작성하였더라도 python 파일(.py)로 변환하여 제출할 것.
- 함수가 의도한 값을 Return 하는지를 확인. (Print 와 혼동하지 말 것)
- 파일명은 P1.py ~ P3.py 를 유지하고, 해당파일들을 PROJ3_학번_이름.zip 으로 압축하여 제출할 것. 예를 들면 학번이 2020-12345 이고, 이름이 Keondo Park 이라면 **PROJ3_2020_12345_KeondoPark.zip** 으로 압축하여 제출.
- 예시로 제시한 입력값 외에도 조교가 랜덤으로 생성한 입력값으로 코드가 잘 작성되었는지 테스트할 것이다.
- 채점은 프로그램에 의해 기계적으로 처리되므로 위 사항을 지키지 않은 경우 누락되거나 불이익을 받을 수 있음.
- 늦은 제출은 받지 않음.
- 표절검사를 수행하여 발각될 경우 성적 F 부여함.
- Python 의 기본 모듈은 사용해도 괜찮으나, 그 외 모듈은 사용을 금지한다. 애매한 경우 조교에게 질문하기 바람. (**Collections** 모듈 사용 가능)