

201201871 서현택

1. 목표

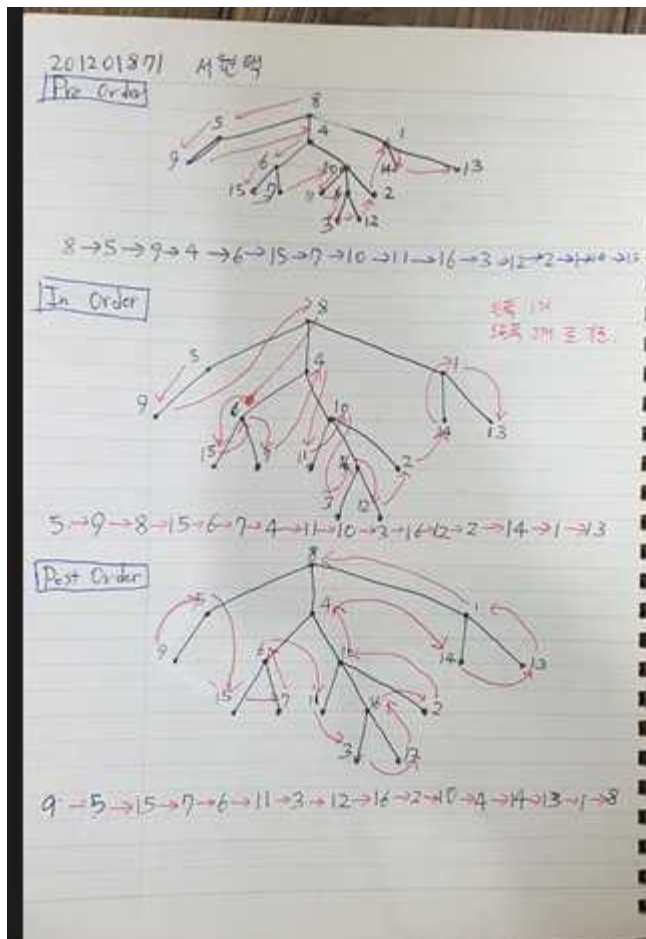
재귀를 사용하지 않고 반복문을 활용하여 preorder, inorder, postorder 구현

2. 과제를 해결하는 방법

1. 각 순회 알고리즘의 이해
2. 반복문의 이해

3. 과제를 해결한 방법

- 1) 그림을 통한 이해



- 2) PreOrder

```

def print_pre2(node):
    s = []
    result = []
    s.append(node)
    while len(s) > 0:
        node = s.pop()
        result.append(node.get_key())
        if node.get_right() is not None:
            s.append(node.get_right())
        if node.get_left() is not None:
            s.append(node.get_left())
    return result

```

스택과 결과를 저장할 s, result를 선언하고 s에 node를 넣는다. s의 길이가 0보다 크다면 node에 s.pop의 결과를 저장하고 result에 node의 키 값을 저장한다. 노드의 오른쪽과 왼쪽이 있다면 스택에 노드의 오른쪽부터 저장한다.

3) InOrder

```

def print_in2(node):
    s = []
    result = []
    while len(s) > 0 or node is not None:
        if node is not None:
            s.append(node)
            node = node.get_left()
        else:
            node = s.pop()
            result.append(node.get_key())
            node = node.get_right()
    return result

```

s의 길이가 0보다 크지와 노드의 None여부를 확인한 뒤 s에 노드 값을 저장하고, 노드의 왼쪽 노드 값을 저장한다. 그 외의 경우에는 s의 pop값을 노드에 저장하고 결과에 노드의 키 값을 저장한 후 node의 오른쪽 노드 값을 저장한다. 이를 통하여 반복을 통해 inorder가 돌아갈수 있게한다.

4) PostOrder

```

def print_post2(node):
    s = []
    result = []
    last = []
    while len(s) > 0 or node is not None:
        if node is not None:
            s.append(node)
            node = node.get_left()
        else:
            peek = s.pop()
            s.append(peek)
            if peek.get_right() is not None and last != peek.get_right():
                node = peek.get_right()
            else:
                result.append(peek.get_key())
                last = s.pop()
    return result

```

스택 s, 결과를 저장할 result, 마지막 방문한 노드인 last를 선언한 뒤, s의 길이가 0보다 크거나 node가 None이 아닐 때 node가 None이 아니라면 s에 node를 저장하고, node의 왼쪽 값을 node에 저장한다. node가 None일 경우에는 peek에 s.pop값을 저장하고 s.peek 값을 저장한다. peek의 오른쪽 노드가 None이 아니고 last가 peek의 오른쪽 노드가 아니라면 node에 peek의 오른쪽 노드를 저장하고 아니라면 result에 peek의 key값을 저장하고 last에는 s.pop 값을 저장한다.

4.결과화면

