

자료구조 10주차 과제

- 이름: 안찬웅
- 학번: 32162566
- 과제: 각 5점

1. P8.5

2. P8.6

** 한 문제라도 컴파일 에러를 해결하지 못하고 제출하는 경우, 전체 과제 0점.

** 풀지 못한 문제 - 만일 과제의 문제를 다 풀지 못한 경우, 여기에 풀지 못한 번호를 적으시오.

과제는 문제에 대한 코딩이 완성되고 테스트를 통해 적절성이 검증된 경우만 점수가 부여되며, 이외 사항에 대해서는 0점 처리. 코드에 에러가 있음에도 불구하고, 과제 앞 부분 미완성 부분에 적시하지 않은 경우 전체 과제를 0점 처리합니다.

더블클릭 또는 Enter 키를 눌러 수정

▼ 8.5

아래 코드셀에 함수 path_length 구현하라.

```
class TNode:                                # 이진트리를 위한 노드 클래스
    def __init__(self, data, left, right): # 생성자
        self.data = data                  # 노드의 데이터
        self.left = left                  # 왼쪽 자식을 위한 링크
        self.right = right                # 오른쪽 자식을 위한 링크

def count_node(n):                           # 순환을 이용해 트리의 노드 수 계산
    if n is None:                            # n이 None이면 공백트리 -> 0을 반환
        return 0
    else:                                    # 좌우 서브트리의 노드수의 합 + 1을 반환
        return 1+count_node(n.left) + count_node(n.right)

def path_length(root):
    if root is None:                        # root가 none이면 0
        return 0
    num_child = count_node(root) - 1
    res = path_length(root.left)
    res += path_length(root.right)          # res에 왼쪽이랑 오른쪽의 path_length들의 합을 더함
    res += num_child * 1
    return res
```

return res

아래의 코드를 이용하여 테스트를 수행하시오.

```
c = TNode('C', None, None)
d = TNode('D', None, None)
b = TNode('B', c, d)
f = TNode('F', None, None)
e = TNode('E', None, f)
root = TNode('A', b, e)

print("path length: ", path_length(root)) # should be 8

path length: 8
```

▼ 8.6

아래 코드셀에 함수 `reverse` 를 구현하라. 결과는 `inorder traversal` 결과이다. 결과 출력에는 교재 283 페이지의 `inorder` 함수를 이용.

```
class TNode:                                # 이진트리를 위한 노드 클래스
    def __init__(self, data, left, right): # 생성자
        self.data = data                  # 노드의 데이터
        self.left = left                  # 왼쪽 자식을 위한 링크
        self.right = right                # 오른쪽 자식을 위한 링크

def reverse(root):
    if root:
        root.left, root.right = reverse(root.right), reverse(root.left) # 교환
        return root
    return None

def inorder(n):                              # 전위 순회 함수
    if n is not None:
        inorder(n.left)                   # 왼쪽 서브트리 처리
        print(n.data, end=' ')            # 루트노드 처리(화면 출력)
        inorder(n.right)                  # 오른쪽 서브트리 처리
```

아래의 코드를 이용하여 테스트를 수행하시오.

```
reversed_tree = reverse(root)
inorder(reversed_tree) # F-E-A-D-B-C

F E A D B C
```

Colab 유료 제품 - [여기에서 계약 취소](#)

✓ 0초 오전 12:13에 완료됨

