

Tree Traversal

트리 순회

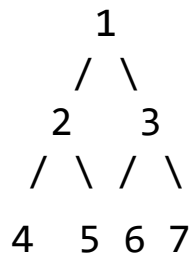
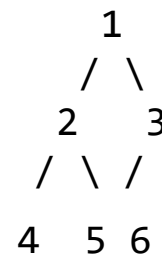
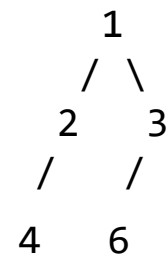
이영석

이진 트리

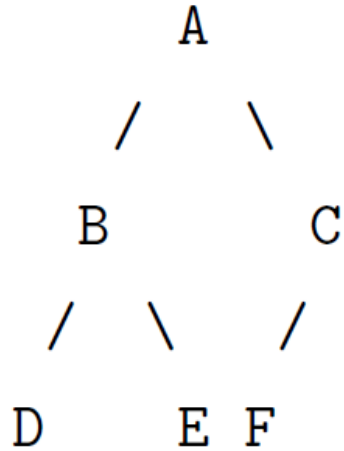
- 이진 트리 개념과 연산
 - 그래프의 일종 (연결이 없는)
 - parent node, left/right child node
 - 트리 순회: level-order, preorder, inorder, postorder
- 구현
 - 직접 구현: 재귀 vs 반복
 - 사전 또는 리스트 활용

이진트리 주요 용어

- 이진트리
 - 루트(최상위), 부모, 왼쪽 + 오른쪽 자식 노드
- 완전 이진트리(complete binary tree)
 - 레벨의 순서대로 노드가 있는 경우
- 포화 이진트리: 레벨의 모든 노드가 가득 차 있는 경우
- 단말(leaf): 자식 노드가 없는 노드
- 단말 노드까지 경로
 - 루트 노드에서 단말 노드까지 경유하는 모든 노드 순서: 1, 2, 4
- 높이(height)
 - 자식 노드 경로의 길이: (1, 2, 4) -> 3



순회(Traversal)



- postorder: LR V
- inorder: L V R
- preorder: V L R

```
PS C:\Users\yslee\Dropbox\python-algorithm\tree> cat .\tree_traverse_input.txt
6
A B C
B D E
D . .
C F .
E . .
F . .
```

```
PS C:\Users\yslee\Dropbox\python-algorithm\tree> cat .\tree_traverse_input.txt | python3 .\tree_traverse.py
D E B F C A
D B E A F C
A B D E C F
PS C:\Users\yslee\Dropbox\python-algorithm\tree>
```

이진 트리 순회 예제

```
15 def preorder(tree, node, visit_list):
16     visit_list.append(node)
17     if (tree[node][0] != '.'):
18         preorder(tree, tree[node][0], visit_list)
19     if (tree[node][1] != '.'):
20         preorder(tree, tree[node][1], visit_list)
21
22 if __name__ == "__main__":
23     n = int(input().strip())
24     tree = {}
25     visit_list = []
26     node = None
27     for _ in range(n):
28         # parent left_child right_child
29         node_list = input().strip().split(' ')
30         tree[node_list[0]] = [node_list[1], node_list[2]]
31
32     postorder(tree, 'A', visit_list)
33     print(*visit_list)
```