

CHAPTER.15

이진 트리의 기본 및 간단 구현





학습 내용

[1] 이진 트리의 기본

[2] 이진 트리의 간단 구현



학습 목표

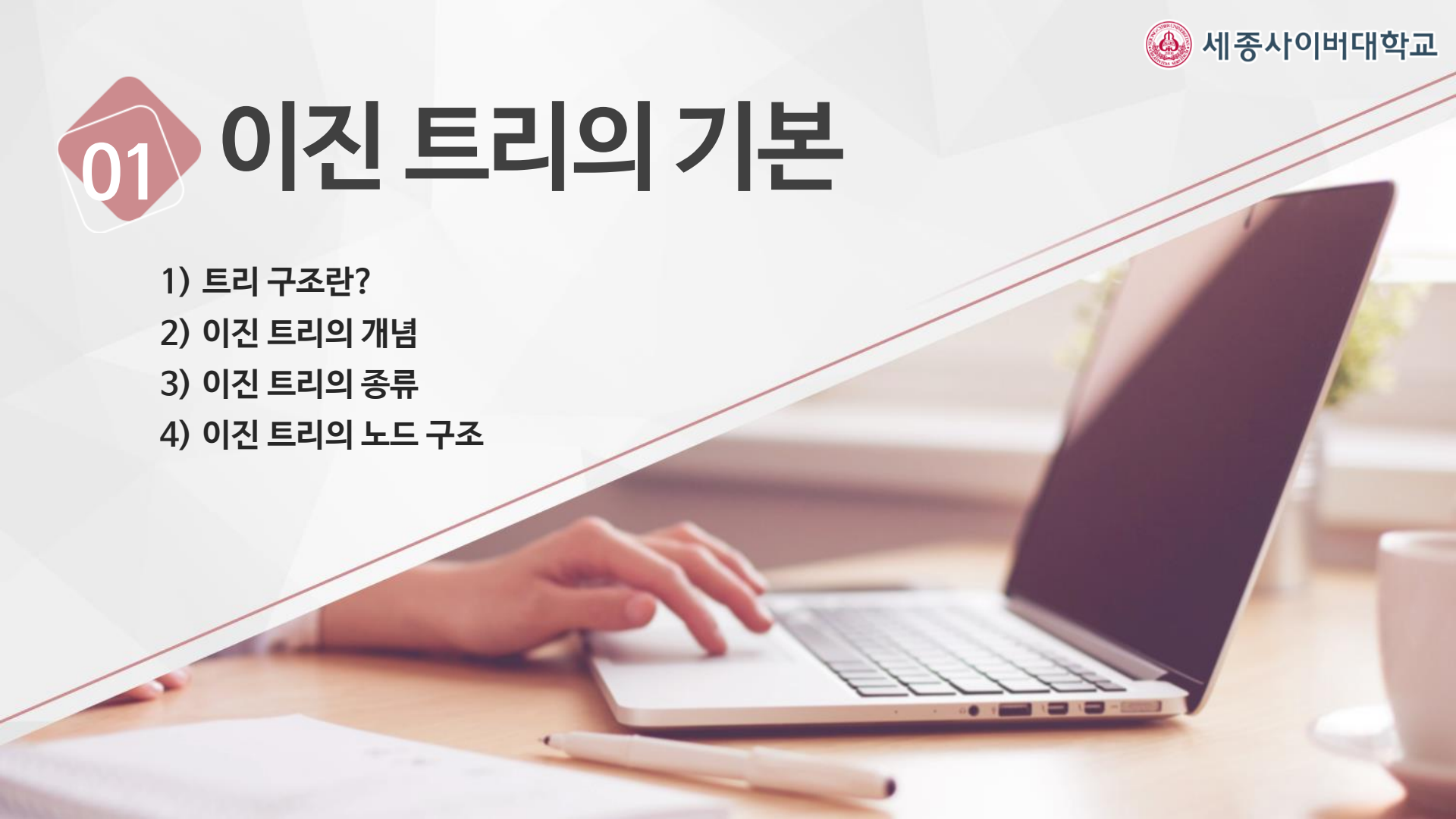
- ④ 이진 트리의 개념을 설명할 수 있다.
- ④ 파이썬으로 이진 트리를 구현하는 코드를 작성할 수 있다.



01

이진 트리의 기본

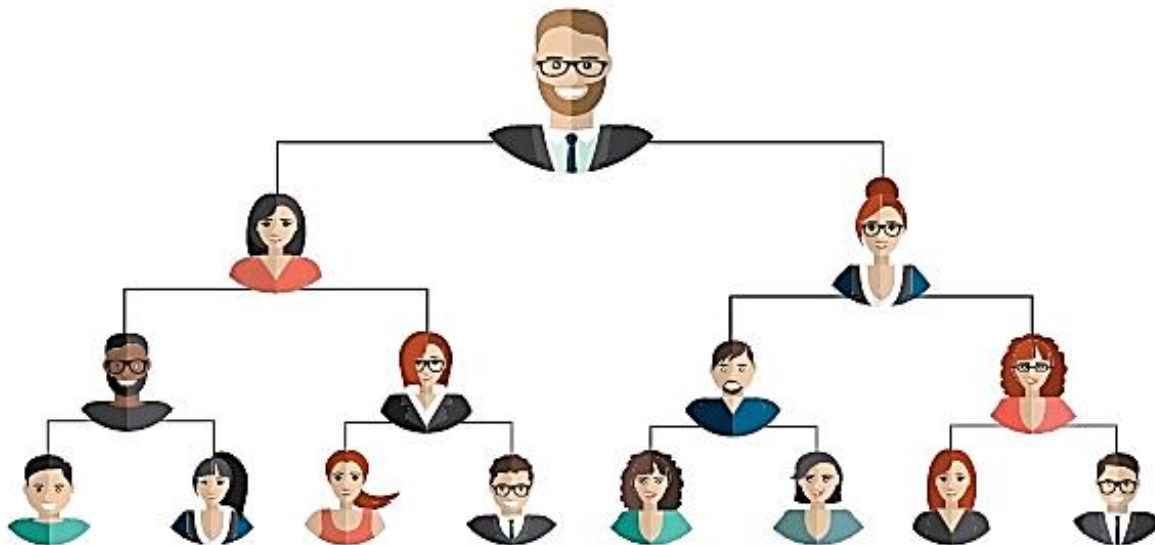
- 1) 트리 구조란?
- 2) 이진 트리의 개념
- 3) 이진 트리의 종류
- 4) 이진 트리의 노드 구조



1] 트리 구조란?



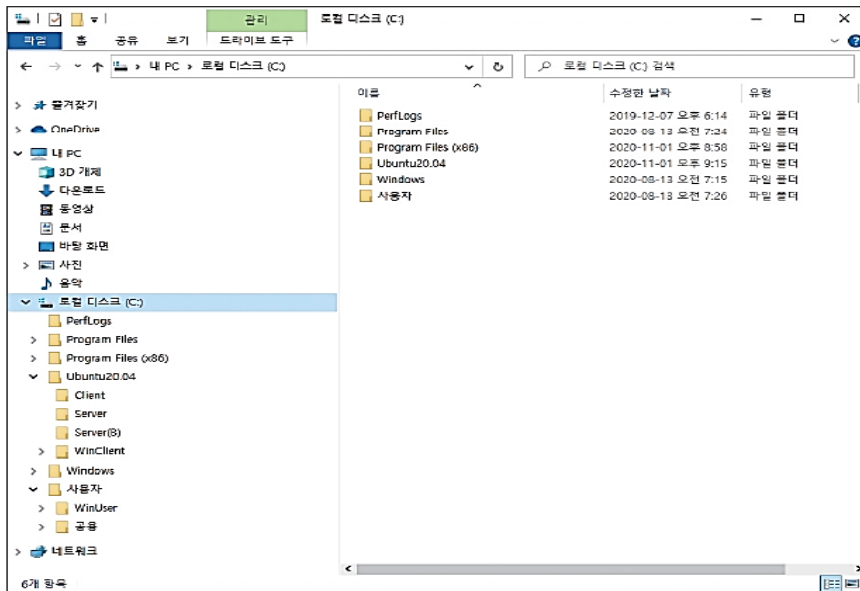
회사 사장을 필두로 그 아래 직책들이 구성되어 있는 조직표



1] 트리 구조란?



컴퓨터의 상위 폴더 안에 하위 폴더들이 계속 이어져 있는
구조와 같은 구성



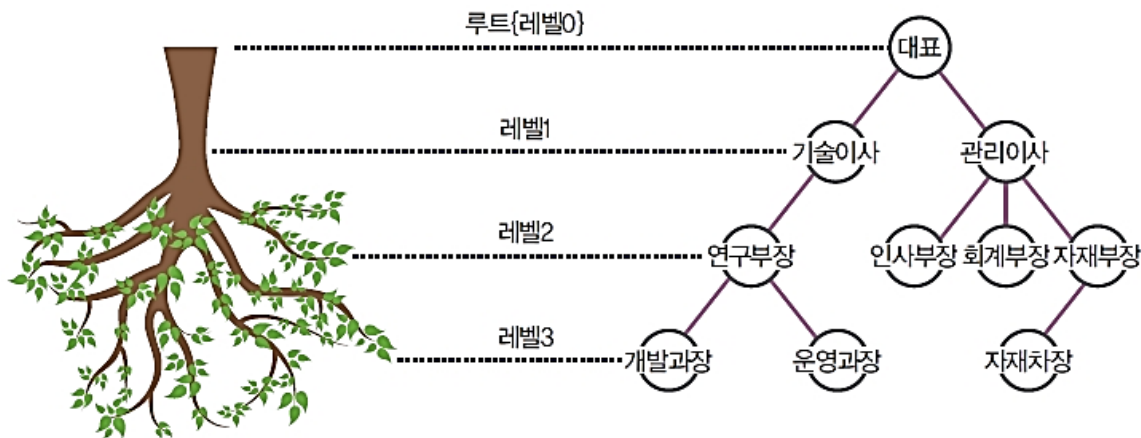
2] 이진 트리의 개념



트리(Tree) 자료구조

나무를 거꾸로 뒤집어 놓은 형태

■ 나무와 트리 자료구조 비교



나무를 거꾸로
뒤집은 형태

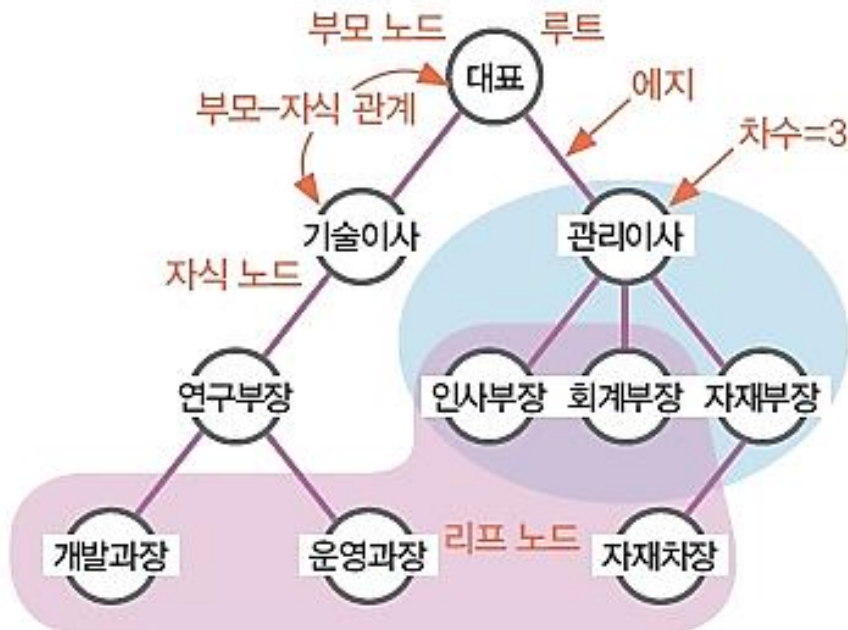
트리 자료 구조

2] 이진 트리의 개념



트리 자료구조 용어

트리 자료구조의 주요 용어



2] 이진 트리의 개념

이진 트리의 예

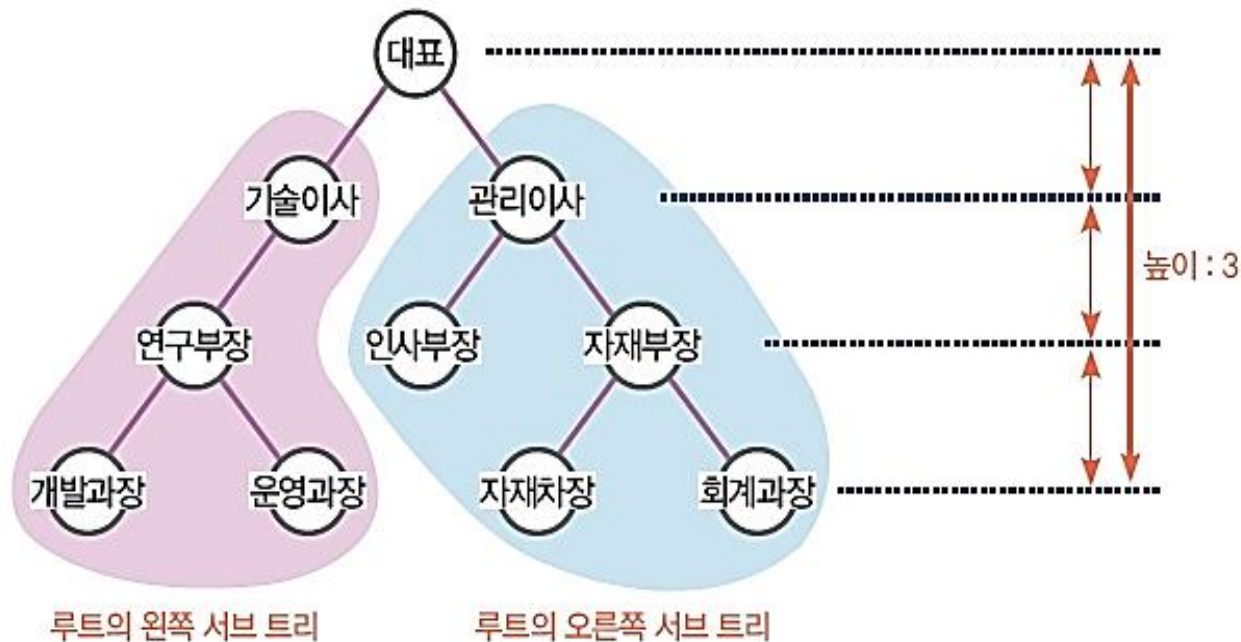
모든 노드의 자식이 최대 2개인 트리(자식이 2개 이하로 구성)



2] 이진 트리의 개념



전형적인 이진 트리

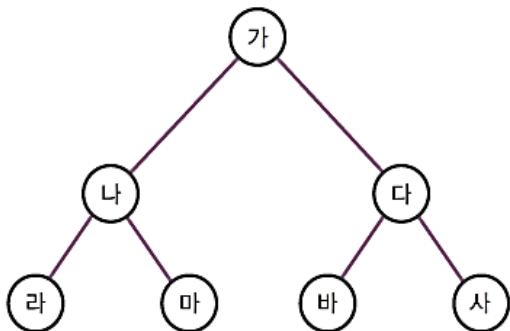


3] 이진 트리의 종류

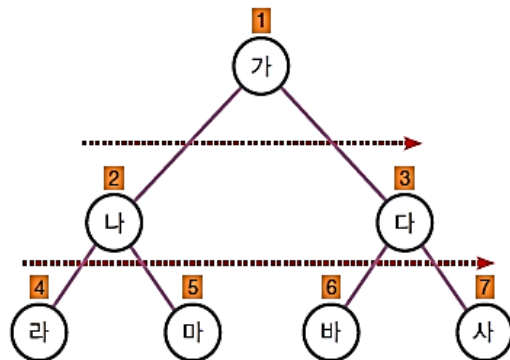


- 1 포화 이진 트리(Full Binary Tree)
- 2 완전 이진 트리(Complete Binary Tree)
- 3 일반 이진 트리(Binary Tree)
- 4 편향 이진 트리(Skewed Binary Tree)

1 포화 이진 트리(Full Binary Tree)

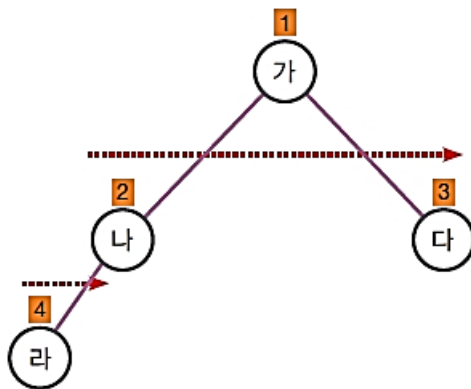
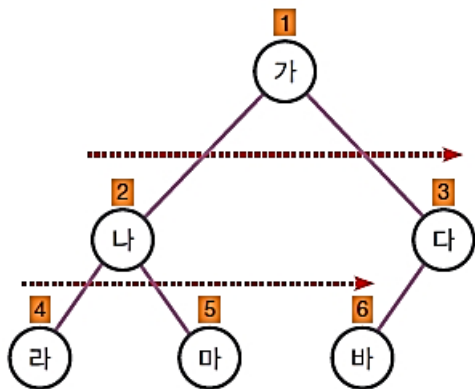


포화 이진 트리 예



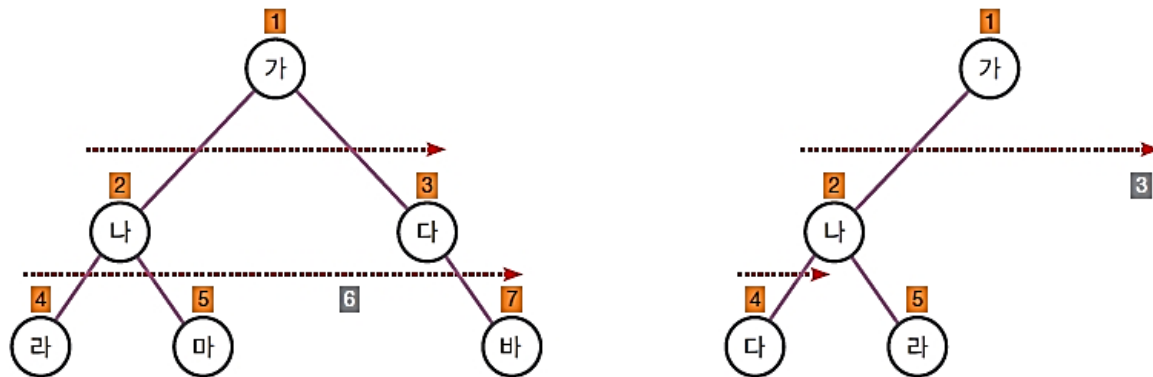
포화 이진 트리의
번호 부여 순서

2 완전 이진 트리(Complete Binary Tree)



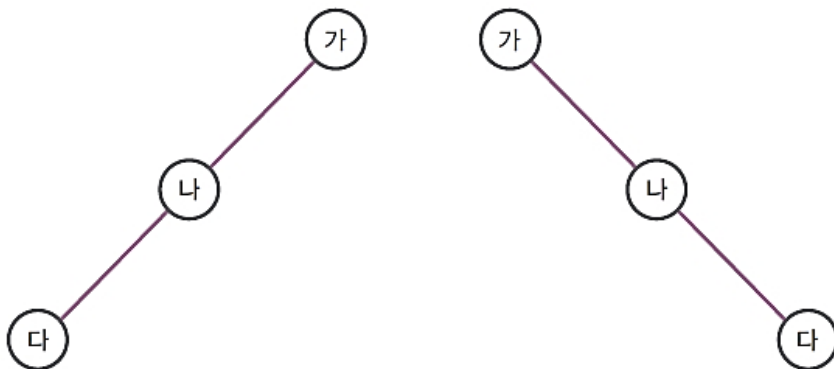
완전 이진 트리 예

3 일반 이진 트리(Binary Tree)



일반 이진 트리 예

3 편향 이진 트리(Skewed Binary Tree)

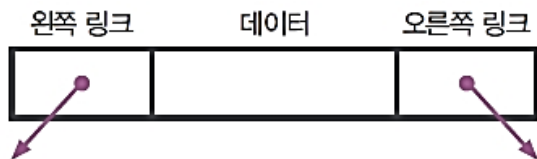


왼쪽 편향 이진 트리

오른쪽 편향 이진 트리

이중 연결 리스트를 이용한 트리 노드 표현

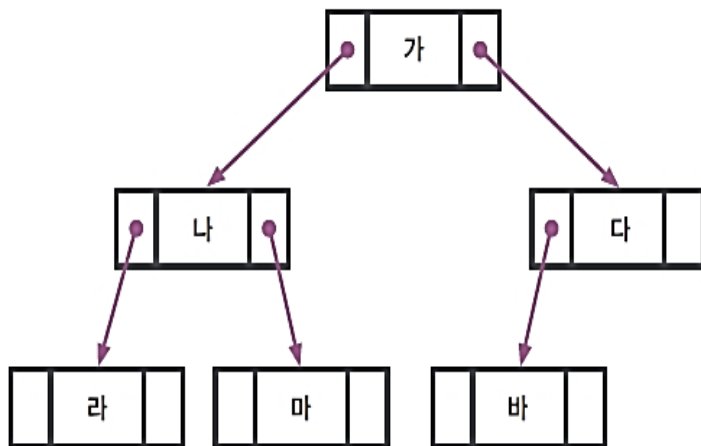
- 트리 노드의 구현을 위한 이중 연결 리스트



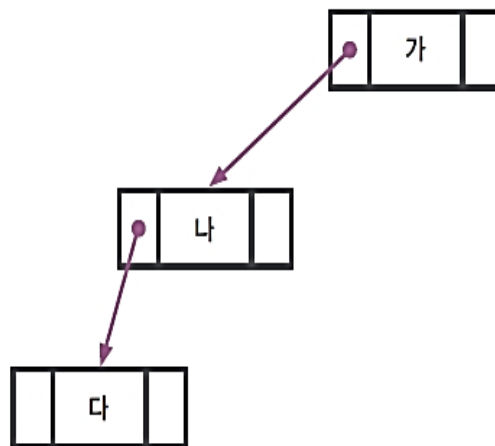
4] 이진 트리의 노드 구조



이중 연결 리스트를 이용한 트리 노드 표현



완전 이진 트리의 표현



편향 이진 트리의 표현



02

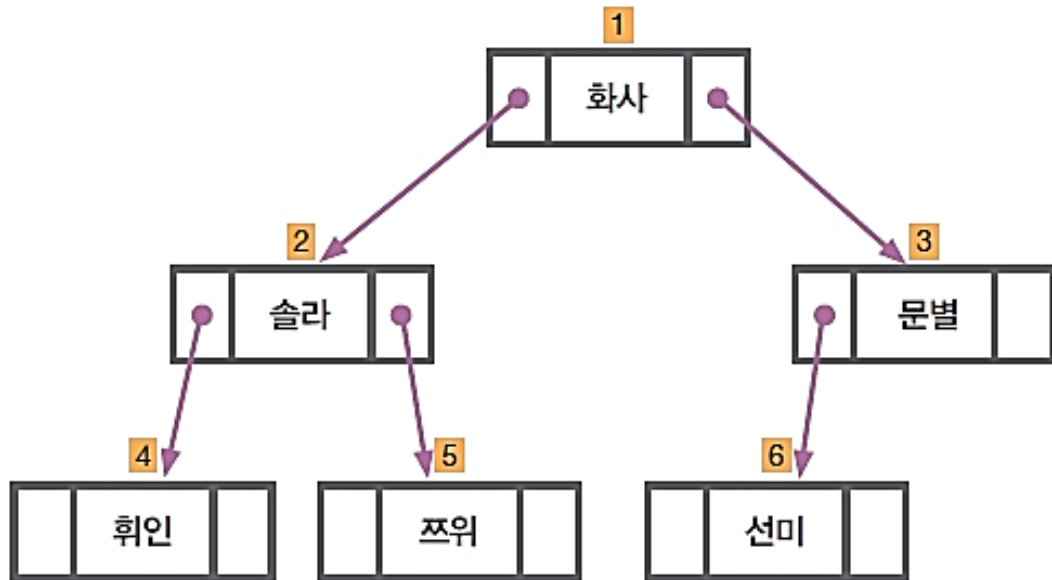
이진 트리의 간단 구현

- 1) 이진 트리의 생성
- 2) 이진 트리의 순회
- 3) 이진 트리의 간단 구현 실습



높이가 2고 데이터가 6개인 완전 이진 트리의 생성 예

■ 생성할 이진 트리 예



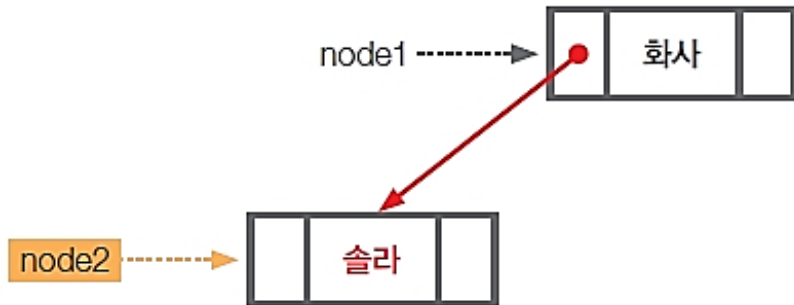
1] 이진 트리의 생성



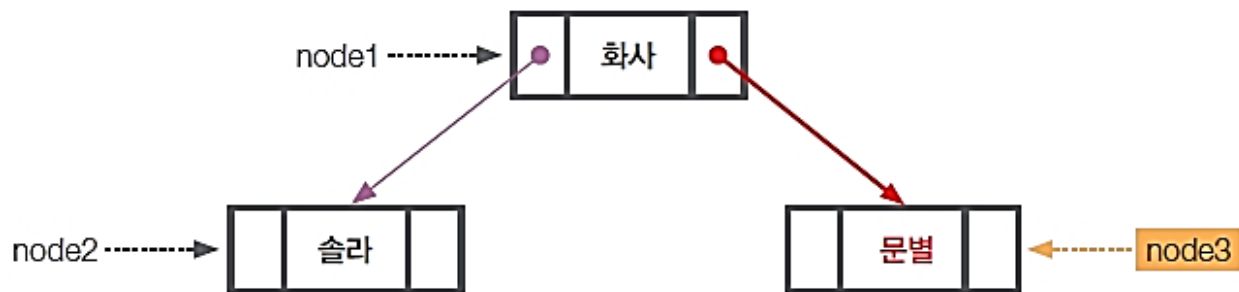
1 루트 노드(화사)를 생성



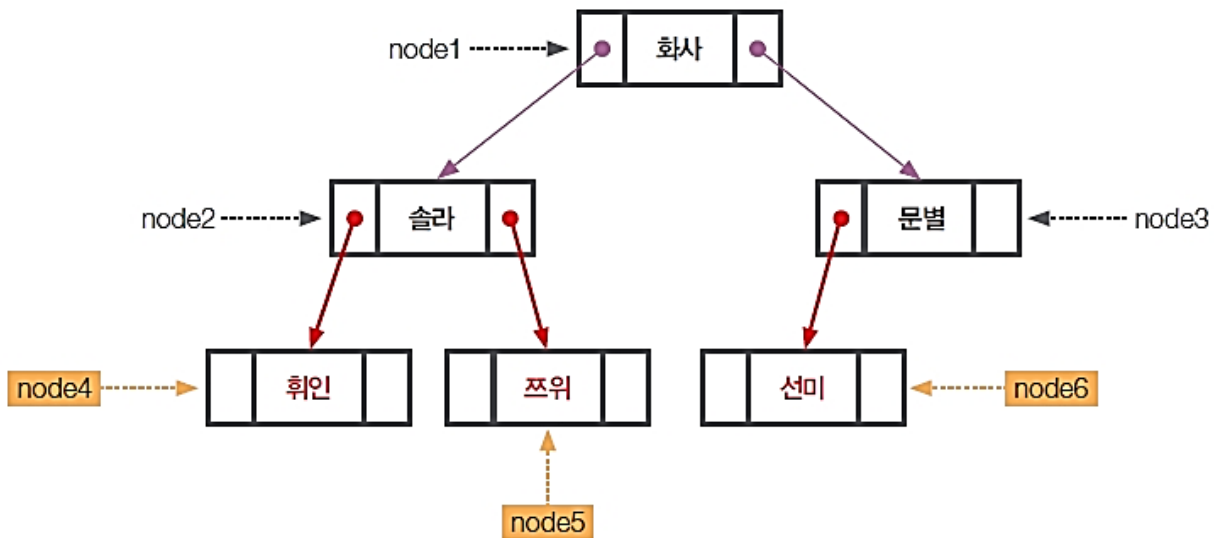
2 두 번째 노드(솔라)를 생성하고 루트 노드의 왼쪽 노드로 지정



3 세 번째 노드(문별)를 생성하고 루트 노드의 오른쪽 쪽 노드로 지정



4 네 번째부터 여섯 번째까지 노드를 생성하고 부모 노드와 연결



높이가 2인 완전 이진 트리의 생성

```
1 class TreeNode():
2     def __init__(self):
3         self.left = None
4         self.data = None
5         self.right = None
6
7     node1 = TreeNode() }
8     node1.data = '화사' } 1
9
10    node2 = TreeNode() }
11    node2.data = '솔라' } 2
12    node1.left = node2 }
```

이진 트리 노드 생성

높이가 2인 완전 이진 트리의 생성

```
13
14 node3 = TreeNode()
15 node3.data = '문별' } 3
16 node1.right = node3
17
18 node4 = TreeNode()
19 node4.data = '휘인'
20 node2.left = node4
21
22 node5 = TreeNode()
23 node5.data = '쯔위' } 4
24 node2.right = node5
25
```


높이가 2인 완전 이진 트리의 생성

```
26 node6 = TreeNode()
27 node6.data = '선미'
28 node3.left = node6
29
30 print(node1.data, end = ' ')
31 print()
32 print(node1.left.data, node1.right.data, end = ' ')
33 print()
34 print(node1.left.left.data, node1.left.right.data, node1.right.left.data, end = ' ')
```

높이가 2인 완전 이진 트리의 생성

실행 결과

화사

솔라 문별

휘인 쏘우 선미

순회 (Traversal)

이진 트리의 노드 전체를 한 번씩 방문하는 것

노드 데이터를 처리하는 순서에 따라

- 전위 순회 (Preorder Traversal)
- 중위 순회 (Inorder Traversal)
- 후위 순회 (Postorder Traversal)

2] 이진 트리의 순회



전위 순회 (Preorder Traversal)

- 1 현재 노드 데이터 처리
- 2 왼쪽 서브 트리로 이동
- 3 오른쪽 서브 트리로 이동

2] 이진 트리의 순회



중위 순회 (Inorder Traversal)

- 1 왼쪽 서브 트리로 이동
- 2 현재 노드 데이터 처리
- 3 오른쪽 서브 트리로 이동

2] 이진 트리의 순회



후위 순회 (Postorder Traversal)

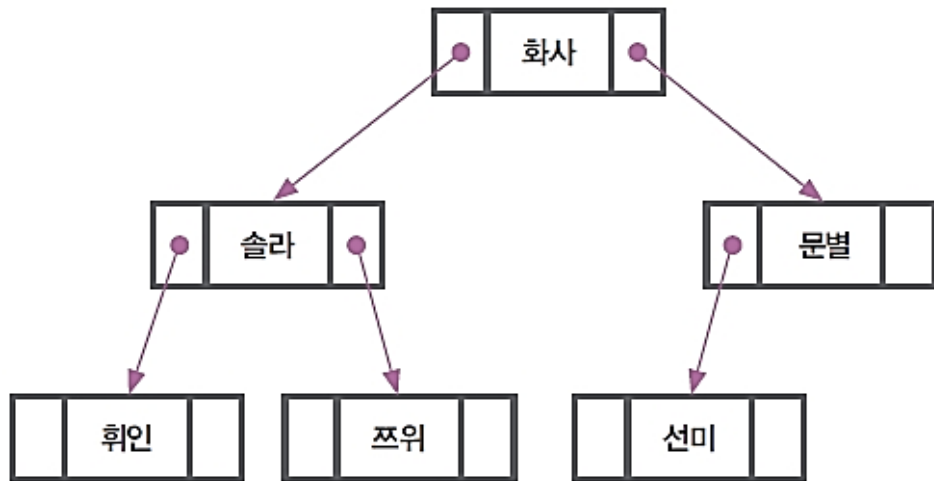
- 1 왼쪽 서브 트리로 이동
- 2 오른쪽 서브 트리로 이동
- 3 현재 노드 데이터 처리

2] 이진 트리의 순회



전위 순회 작동

0 전위 순회할 이진 트리

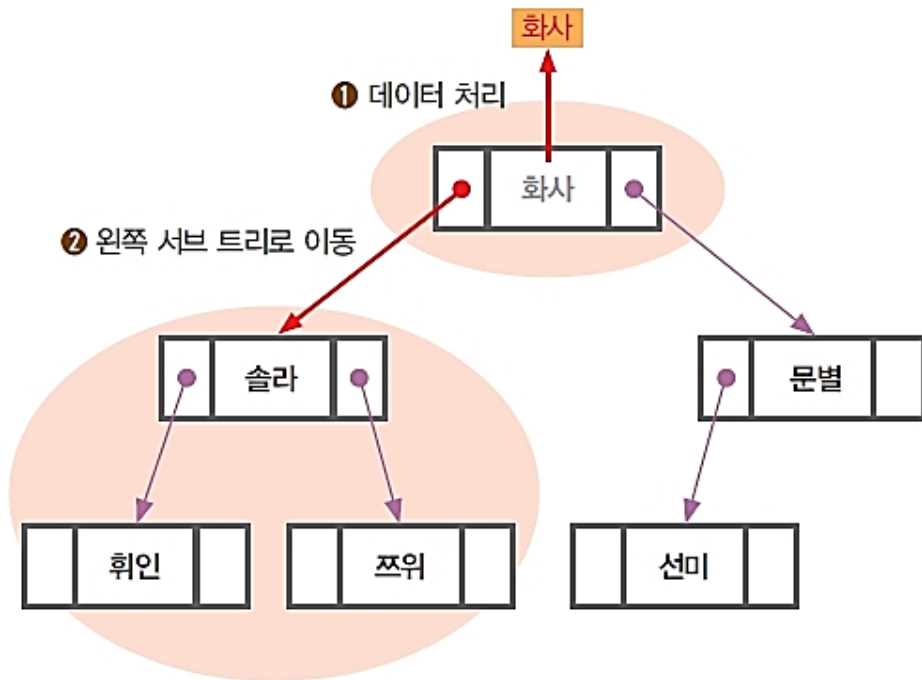


2] 이진 트리의 순회



전위 순회 작동

1 루트 노드(화사)의 데이터를 처리하고 왼쪽 서브트리로 이동

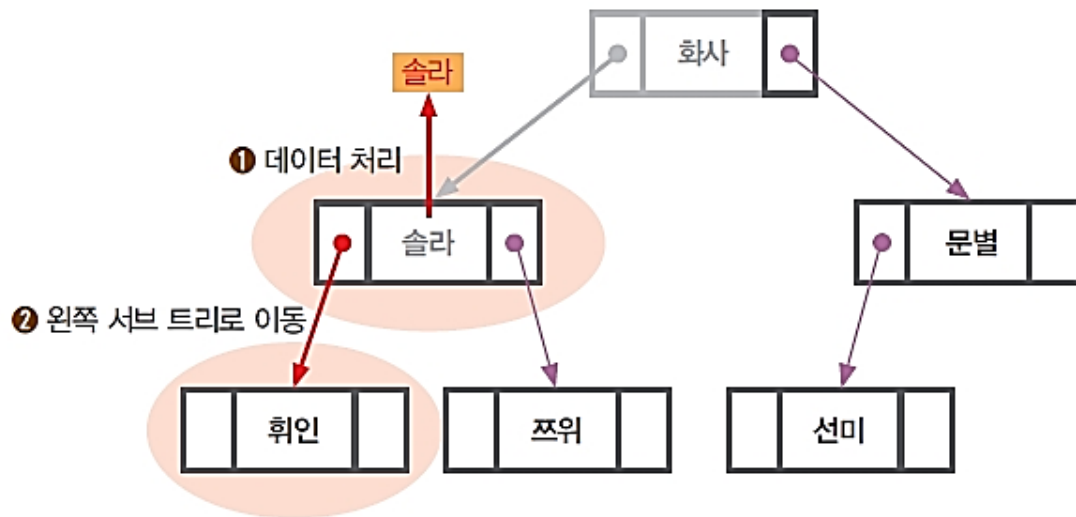


2] 이진 트리의 순회



전위 순회 작동

2 이동한 왼쪽 서브 트리의 '슬라' 데이터를 처리하고
다시 왼쪽 서브 트리로 이동



전위 순회 작동

3 이동한 왼쪽 서브 트리의 '휘인' 데이터를 먼저 처리하고,
다시 왼쪽 서브 트리를 처리

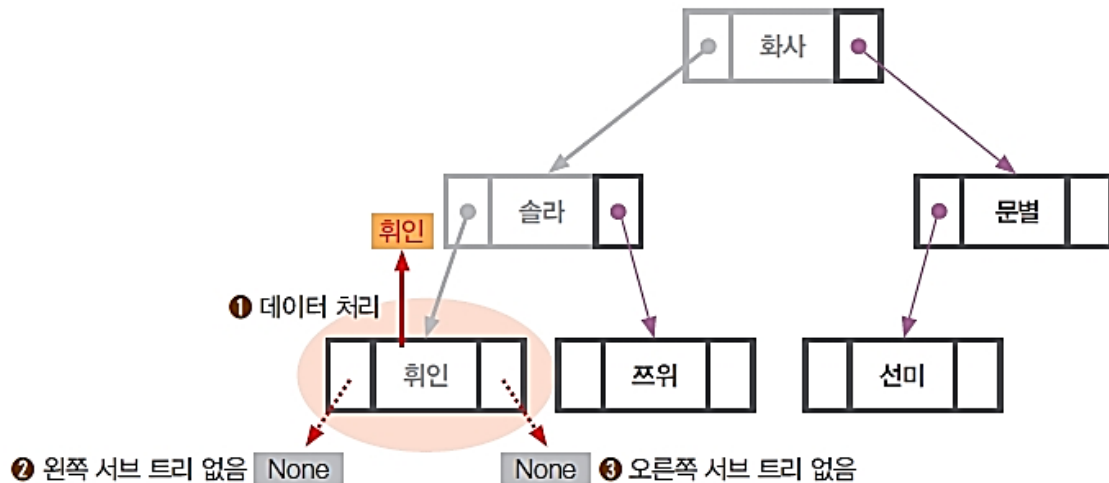
- 왼쪽 서브 트리가 없어 오른쪽 서브 트리를 처리하려고 함
- 그런데 오른쪽 서브 트리도 없으므로 '휘인' 노드는 처리가 완료

2] 이진 트리의 순회



전위 순회 작동

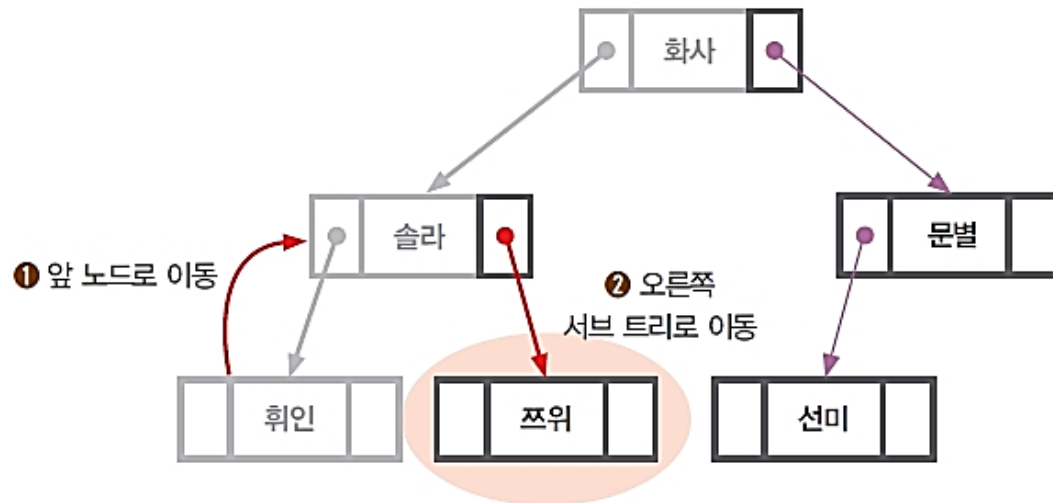
3 이동한 왼쪽 서브 트리의 '회인' 데이터를 먼저 처리하고,
다시 왼쪽 서브 트리를 처리



전위 순회 작동

4 현재 노드(회인 노드)는 더 이상 처리할 것이 없음

- 앞 노드로 올라가서 처리하지 않았던 오른쪽 서브 트리로 내려감



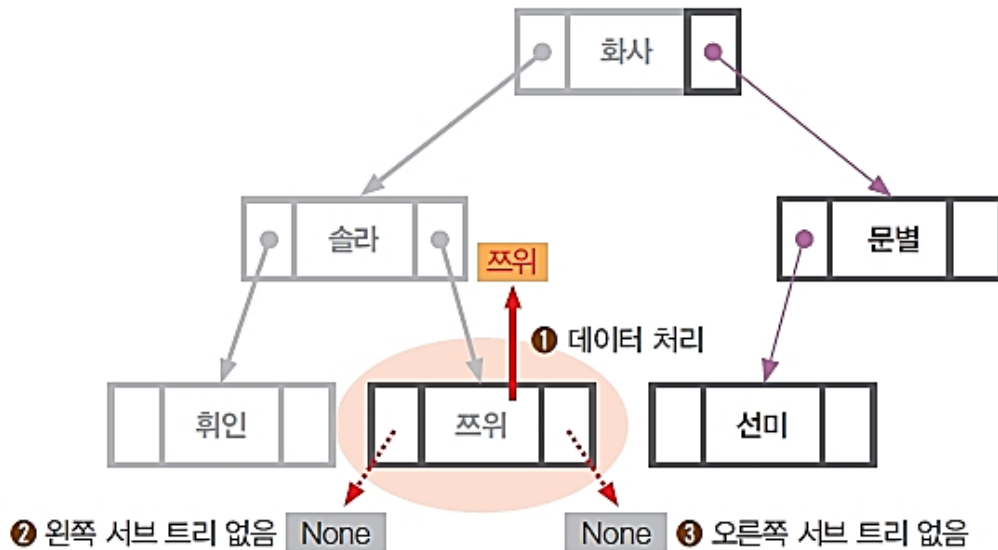
전위 순회 작동

5 이동한 오른쪽 서브 트리의 ‘찌위’ 데이터를 먼저 처리하고,
왼쪽 서브 트리를 처리

- 왼쪽 서브 트리가 없어 오른쪽 서브 트리를 처리하려고 함
- 그런데 오른쪽 서브 트리도 없으므로 ‘찌위’ 노드는
처리가 완료

전위 순회 작동

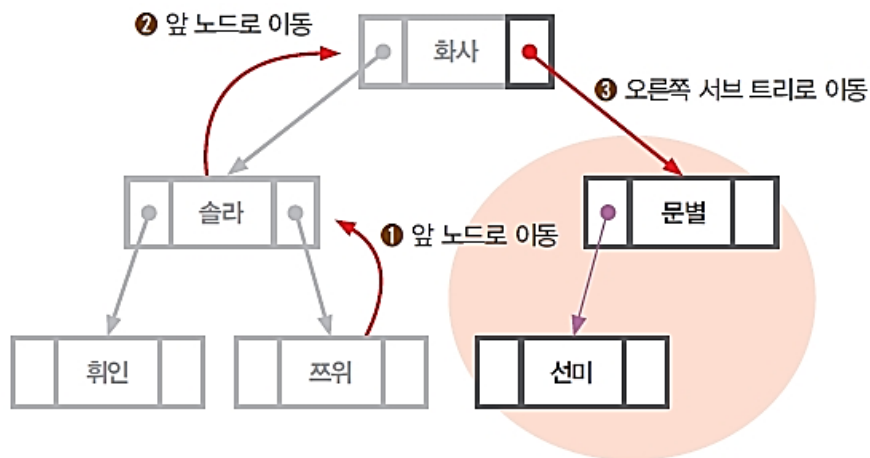
5 이동한 오른쪽 서브 트리의 '썬위' 데이터를 먼저 처리하고, 왼쪽 서브 트리를 처리



전위 순회 작동

6 현재 노드(쓰위 노드)는 더 이상 처리할 것이 없으므로
앞 노드(솔라 노드)로 올라감

- 앞 노드도 처리가 완료되었으므로
다시 앞 노드(화사 노드)로 올라가 오른쪽 서브 트리로 이동

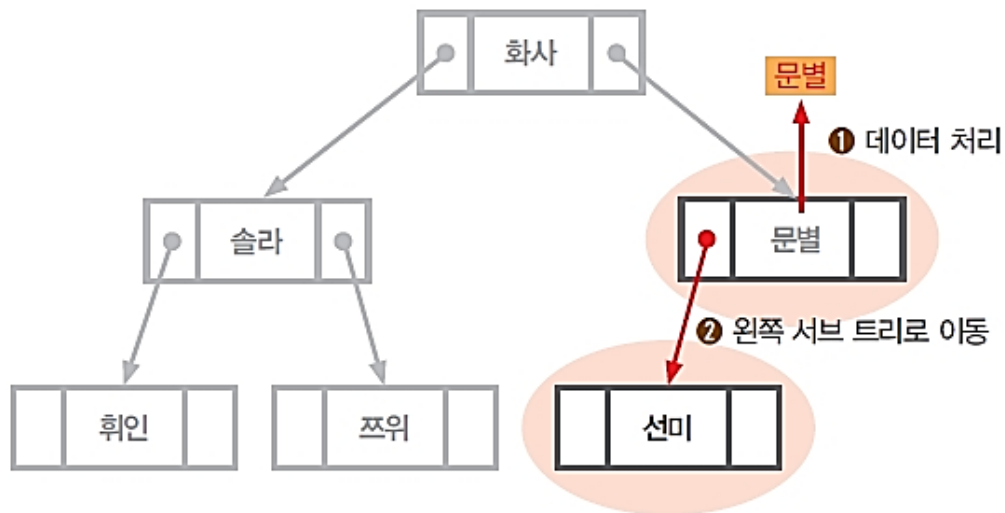


2] 이진 트리의 순회



전위 순회 작동

7 이동한 오른쪽 서브 트리의 문별 데이터를 먼저 처리하고
다시 왼쪽 서브 트리로 이동





전위 순회 작동

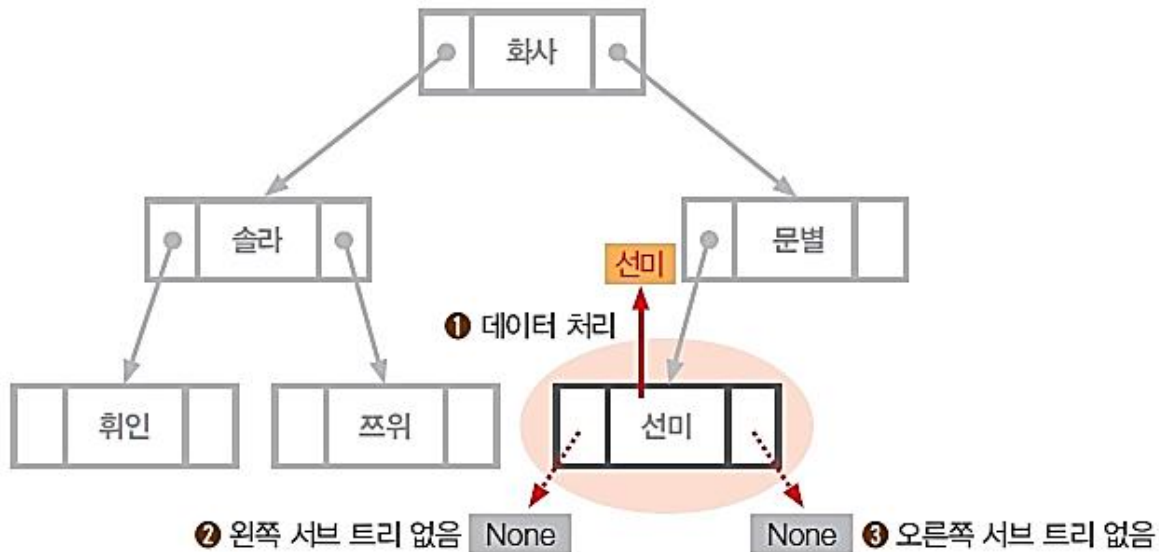
8

이동한 왼쪽 서브 트리의 '선미' 데이터를 먼저 처리하고,
다시 왼쪽 서브 트리를 처리

- 왼쪽 서브 트리가 없어 오른쪽 서브 트리를 처리하려고 함
- 그런데 오른쪽 서브 트리도 없으므로 '선미' 노드는 처리가 완료
- 트리의 모든 노드 순회가 완료됨

전위 순회 작동

8 이동한 왼쪽 서브 트리의 '선미' 데이터를 먼저 처리하고,
다시 왼쪽 서브 트리를 처리



2] 이진 트리의 순회



전위 순회 작동

①~⑧에 출력된 데이터를 확인하면
화사, 솔라, 휘인, 찰위, 문별, 선미 순

- 전위 순회인 현재 데이터 → 왼쪽 서브 트리 → 오른쪽 서브 트리 순서로 출력된 것을 확인할 수 있음

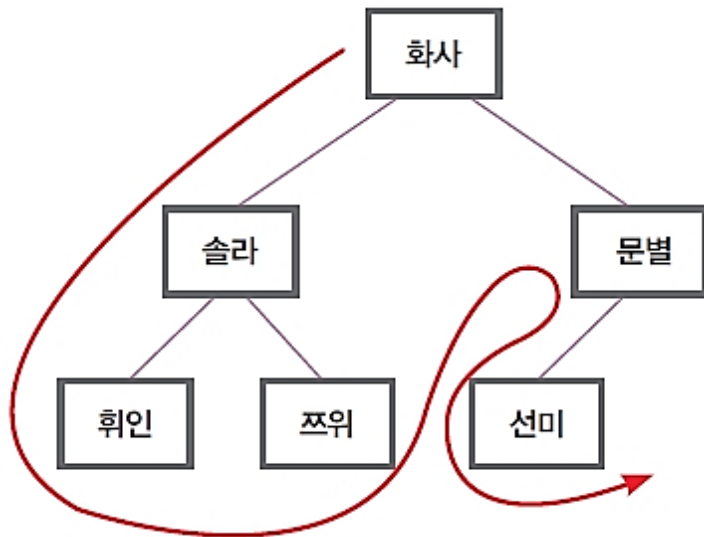
2] 이진 트리의 순회



좀 더 간단한 트리 순회

전위 순회

- 루트 → 왼쪽 → 오른쪽



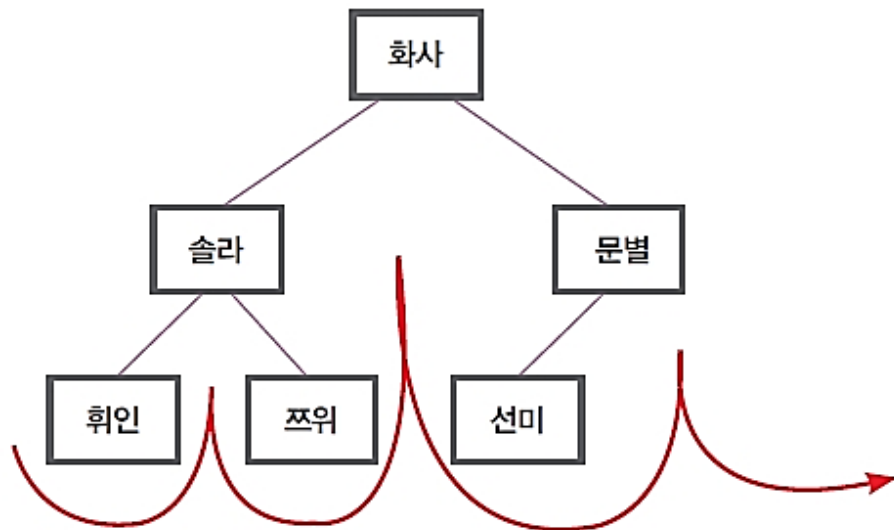
2] 이진 트리의 순회



좀 더 간단한 트리 순회

중위 순회

- 왼쪽 → 루트 → 오른쪽



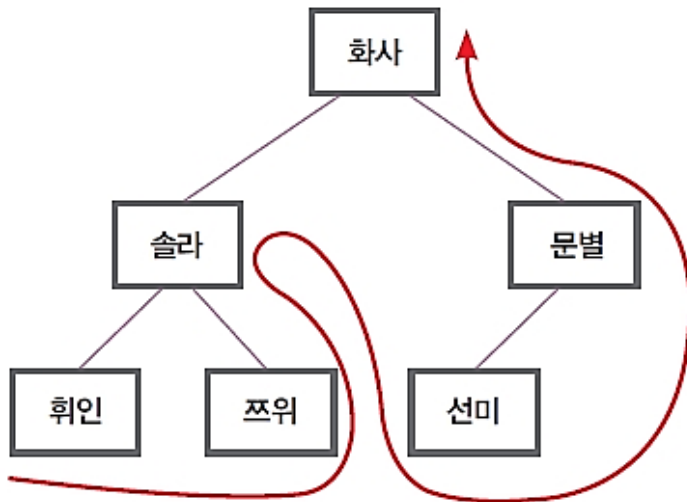
2] 이진 트리의 순회



좀 더 간단한 트리 순회

후위 순회

- 왼쪽 → 오른쪽 → 루트



이진 트리 순회 구현

■ 이진 트리의 순회(재귀 함수 사용)

```
... # 생략(    앞쪽의    1~28행과 동일)
29
30 def preorder(node) :
31     if node == None :
32         return
33     print(node.data, end = '->')
34     preorder(node.left)
35     preorder(node.right)
36
37 def inorder(node) :
38     if node == None :
39         return
40     inorder(node.left)
```

이진 트리 순회 구현

■ 이진 트리의 순회(재귀 함수 사용)

```
41     print(node.data, end = '->')
42     inorder(node.right)
43
44 def postorder(node) :
45     if node == None :
46         return
47     postorder(node.left)
48     postorder(node.right)
49     print(node.data, end = '->')
50
51 print('전위 순회 : ', end = ' ')
52 preorder(node1)
53 print('끝')
```


이진 트리 순회 구현

■ 이진 트리의 순회(재귀 함수 사용)

```
54
55 print('중위 순회 : ', end = ' ')
56 inorder(node1)
57 print('끝')
58
59 print('후위 순회 : ', end = ' ')
60 postorder(node1)
61 print('끝')
```

실행 결과

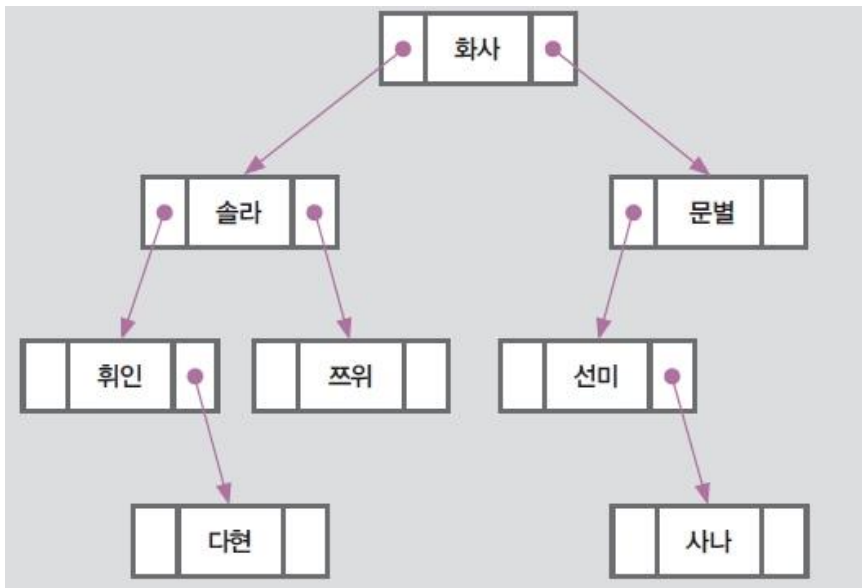
전위 순회 : 화사->솔라->휘인->쯔위->문별->선미->끝
중위 순회 : 휘인->솔라->쯔위->화사->선미->문별->끝
후위 순회 : 휘인->쯔위->솔라->선미->문별->화사->끝

3] 이진 트리의 간단 구현 실습



이진 트리의 간단 구현 실습

앞쪽의 소스를 수정해서 다음 그림과 같이 만들고,
전위/중위/후위 순회를 시켜 보자.



3] 이진 트리의 간단 구현 실습



이진 트리의 간단 구현 실습

앞쪽의 소스를 수정해서 다음 그림과 같이 만들고,
전위/중위/후위 순회를 시켜 보자.

실행 결과

전위 순회 : 화사->솔라->휘인->다현->쯔위->문별->선미->사나->끝

중위 순회 : 휘인->다현->솔라->쯔위->화사->선미->사나->문별->끝

후위 순회 : 다현->휘인->쯔위->솔라->사나->선미->문별->화사->끝

3] 이진 트리의 간단 구현 실습



세종사이버대학교

이진 트리의 간단 구현 실습



Q1

Q2

Q3

Q1

트리에 대한 설명 중에서 아래의 (1)~(4)에 알맞은 용어를 순서대로 넣으시오.

트리의 맨 위를 (1)라고 한다. (1)를 (2)
0으로 두고 나뭇잎에 해당하는 아래로 내려올수록
(2)이 1씩 증가한다. 트리에서 각 위치를 (3)라고
한다. 각 (3)는 (4)로 연결되어 있다.

(1)

(2)

(3)

(4)

Q1

Q2

Q3

Q1

트리에 대한 설명 중에서 아래의 (1)~(4)에 알맞은 용어를 순서대로 넣으시오.

트리의 맨 위를 (1)라고 한다. (1)를 (2)
0으로 두고 나뭇잎에 해당하는 아래로 내려올수록
(2)이 1씩 증가한다. 트리에서 각 위치를 (3)라고
한다. 각 (3)는 (4)로 연결되어 있다.

(1)

루트

(2)

레벨

(3)

노드

(4)

에지

정답

루트, 레벨, 노드, 에지

해설

트리는 루트부터 시작해서 리프 노드까지 에지로 연결되어 있습니다.

Q1

Q2

Q3

Q2

이진 트리의 노드 구조에 대한 설명 중 가장 거리가 먼 것은?

- 1 데이터와 링크 2개로 구성되어 있다.
- 2 링크는 왼쪽 링크와 오른쪽 링크 두 가지다.
- 3 필요한 경우 링크를 3개 이상 구성할 수도 있다.
- 4 보통 왼쪽 링크, 데이터, 오른쪽 링크의 순서로 구성한다.

Q1

Q2

Q3

Q2

이진 트리의 노드 구조에 대한 설명 중 가장 거리가 먼 것은?

- 1 데이터와 링크 2개로 구성되어 있다.
- 2 링크는 왼쪽 링크와 오른쪽 링크 두 가지다.
- ☒ 3 필요한 경우 링크를 3개 이상 구성할 수도 있다.
- 4 보통 왼쪽 링크, 데이터, 오른쪽 링크의 순서로 구성한다.

정답

3 필요한 경우 링크를 3개 이상 구성할 수도 있다.

해설

이진 트리의 노드는 링크 2개만 가집니다.

Q1

Q2

Q3

Q3

이진 트리의 순회는 전위, 중위, 후위 세 가지가 있다.
다음은 어떤 순회에 대한 설명인가?

1. 왼쪽 서브 트리로 이동
2. 오른쪽 서브 트리로 이동
3. 현재 노드 데이터 처리

Q1

Q2

Q3

Q3

이진 트리의 순회는 전위, 중위, 후위 세 가지가 있다.
다음은 어떤 순회에 대한 설명인가?

1. 왼쪽 서브 트리로 이동
2. 오른쪽 서브 트리로 이동
3. 현재 노드 데이터 처리

후위 순회

정답

후위 순회

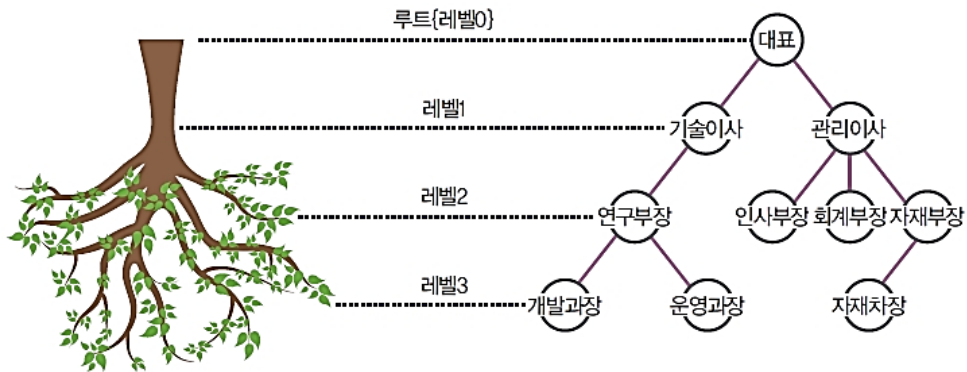
해설

후위 순회는 왼쪽 → 오른쪽 → 루트로 진행됩니다.

이진 트리의 기본

☑ 트리의 개념

- 트리(Tree) 자료구조는 나무를 거꾸로 뒤집어 놓은 형태
- 나무와 트리 자료구조의 비교



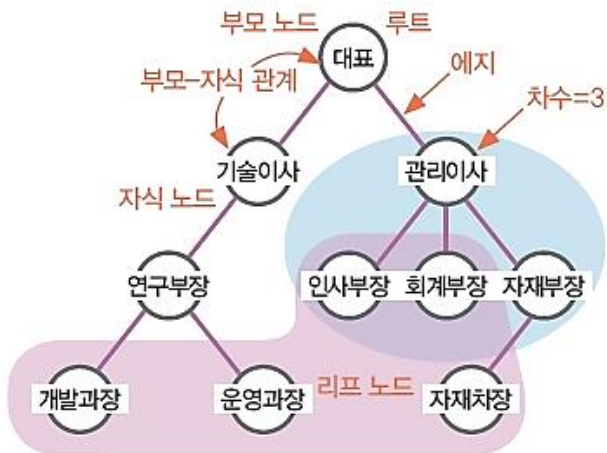
(a) 나무를 거꾸로 뒤집은 형태

(b) 트리 자료구조

이진 트리의 기본

☑ 트리의 개념

▪ 트리 자료 구조의 주요 용어



이진 트리의 기본

④ 이진 트리

- 모든 노드의 자식이 최대 2개인 트리(자식이 2개 이하로 구성)
- 이진 트리의 예



이진 트리의 기본

- ④ 이진 트리 종류
 - 포화 이진 트리
 - 완전 이진 트리
 - 편향 이진 트리

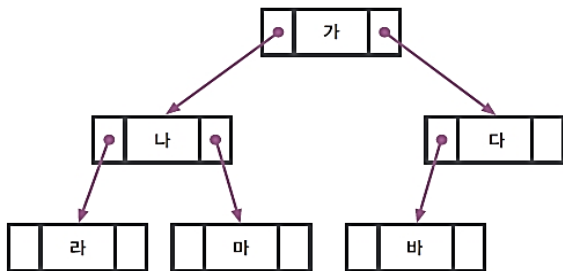
이진 트리의 기본

④ 이진 트리의 노드 구조

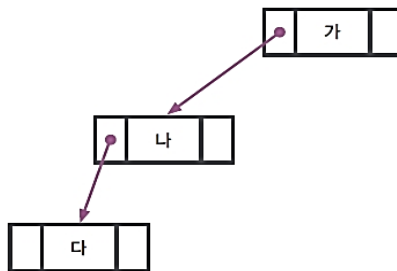
▪ 이중 연결 리스트를 이용한 트리 노드 표현



(a) 트리 노드의 구현을 위한 이중 연결 리스트



(b) 완전 이진 트리의 표현



(c) 편향 이진 트리의 표현

이진 트리의 간단 구현

④ 이진 트리의 순회

- 이진 트리의 노드 전체를 한 번씩 방문하는 것을 순회(Traversal)라고 함
- 노드 데이터를 처리하는 순서에 따라
전위 순회, 중위 순회, 후위 순회
 - ✓ 전위 순회: 루트 → 왼쪽 → 오른쪽
 - ✓ 중위 순회: 왼쪽 → 루트 → 오른쪽
 - ✓ 후위 순회: 왼쪽 → 오른쪽 → 루트