1 트리의 순회

1 트리의 순회

1 트리의 순회(Traversal)

- 모든 노드의 데이터를 처리할 수 있도록
  한 번씩 방문하는 방법

1 트리의 순회

2 이진 트리 순회(Binary Tree Traversal)

- 일정한 규칙에 따라 이진 트리의 모든 정점을 꼭 한번씩 방문하는 것

- 전위순회(Preorder Traversal), 중위순회(Inorder Traversal), 후위순회(Postorder Traversal)가 있음

1 트리의 순회

2 이진 트리 순회(Binary Tree Traversal)

- 전위순회
  : 루트를 가장 먼저 순회한 후
  왼쪽 부분 트리와 오른쪽 부분 트리를 방문

- 중위순회
  : 왼쪽 부분 트리와 오른쪽 부분 트리를
  방문하는 중간에 루트를 방문

- 후위순회
  : 왼쪽 부분 트리와 오른쪽 부분 트리를
  방문한 후에 가장 마지막으로 루트를 방문

1 트리의 순회

2 이진 트리 순회(Binary Tree Traversal)

- 전위순회
  : 루트 → 왼쪽 부분 트리
  → 오른쪽 부분 트리순으로 전위순회함

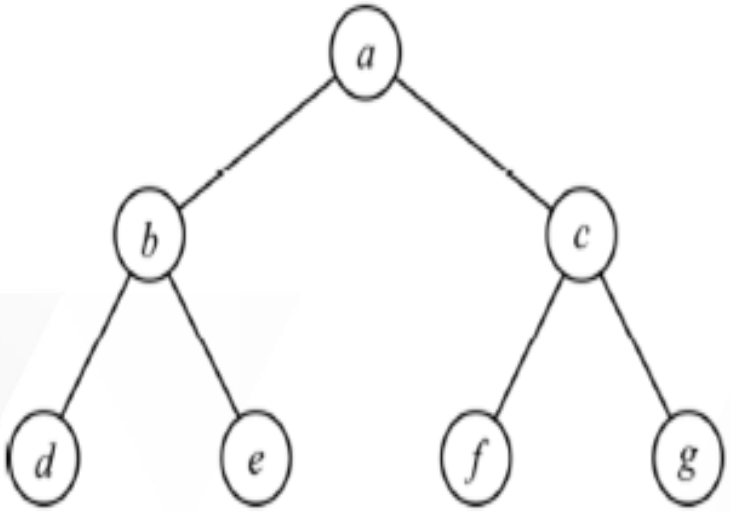
- 중위순회
  : 왼쪽 부분 트리 → 루트
  → 오른쪽 부분 트리순으로 중위순회함

- 후위순회
  : 왼쪽 부분 트리 → 오른쪽 부분 트리 → 루트를 방문

1 트리의 순회

3 전위순회(Preorder Traversal)

[이진 트리의 순회]

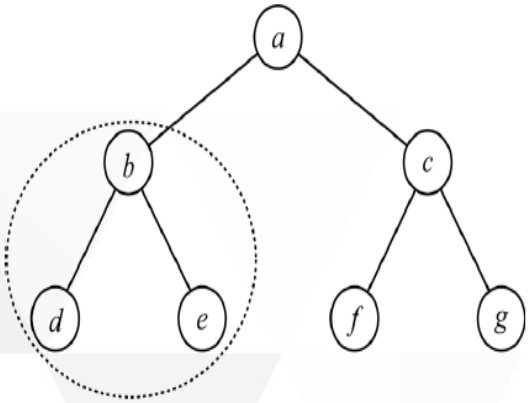


※출처: 이산수학, 류금한, 지식과미래

1 트리의 순회

3 전위순회(Preorder Traversal)

- 정점 a를 방문한 후 왼쪽 부분 트리를 방문
- 왼쪽 부분 트리를 다시 전위순회해야 하므로
  왼쪽 부분 트리의 루트인 b를 방문한 후
  왼쪽 자식인 d와 오른쪽 자식인 e를 순서대로 방문함



[왼쪽 부분 트리의 순회]

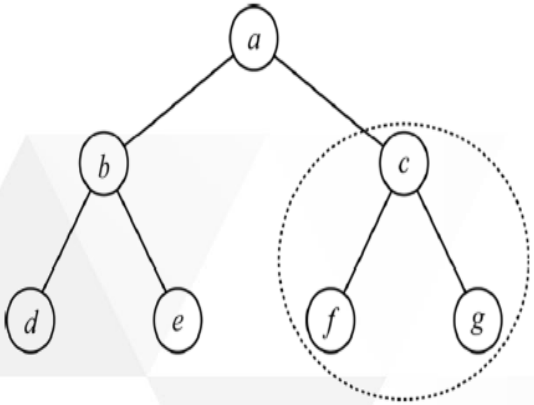
※출처: 이산수학, 류금한, 지식과미래

1 트리의 순회

3 전위순회(Preorder Traversal)

- 오른쪽 부분 트리를 방문할 차례이므로 오른쪽 부분 트리의 루트는 c이므로 c를 먼저 방문하고 왼쪽 자식인 f를 방문한 후 오른쪽 자식인 g를 방문함

[오른쪽 부분 트리의 순회]



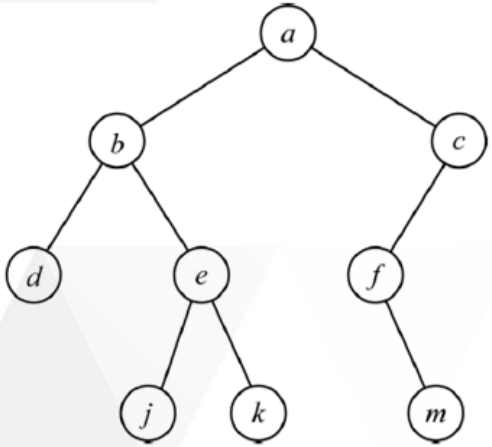
- 전위순회의 최종 방문 순서
  a-b-d-e-c-f-g
- 중위순회의 최종 방문 순서
  d-b-e-a-f-c-g
- 후위순회의 최종 방문 순서
  d-e-b-f-g-c-a

※출처: 이산수학, 류금한, 지식과미래

1 트리의 순회

3 전위순회(Preorder Traversal)

◆ 예) 다음 이진 트리를 전위순회, 중위순회, 후위순회 한 후의 결과를 구하시오



(풀이)
전위순회 : a-b-d-e-j-k-c-f-m
중위순회 : d-b-j-e-k-a-f-m-c
후위순회 : d-j-k-e-b-m-f-c-a

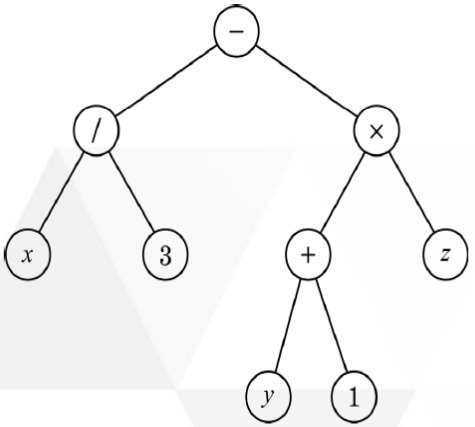
※출처: 이산수학, 류금한, 지식과미래

1 트리의 순회

4 이진 트리의 수식 표현

- 이진 트리는 수식을 표현하기 위한 방법으로 사용하기도 하는데 피연산자(Operand)와 연산자(Operator)로 구성하는 수식을 이진 트리로 표현

[이진 트리의 수식 표현]



$$(x / 3) - (y + 1) \times z$$

- 먼저 피연산자에 해당하는 x, 3, y, 1, z를 잎 노드로 표현함
- 그런 다음 왼쪽 부분 트리와 오른쪽 부분 트리의 수식을 수행하면 됨

※출처: 이산수학, 류금한, 지식과미래

4 이진 트리의 수식 표현

- 연산자와 피연산자가 표현된 수식은 전위표기법(Prefix Notation), 중위표기법(Infix Notation), 후위표기법(Postfix Notation)으로 나타낼 수 있음

  - 전위표기법 : 연산자가 연관된 2개의 피연산자 앞에 위치하도록 표기
  - 중위표기법 : 연산자가 연관된 2개의 피연산자 사이에 위치하게 표기
  - 후위표기법 : 연산자가 연관된 2개의 피연산자 뒤에 위치하도록 하는 표기법

1 트리의 순회

4 이진 트리의 수식 표현

- $(x / 3) - (y + 1) \times z$

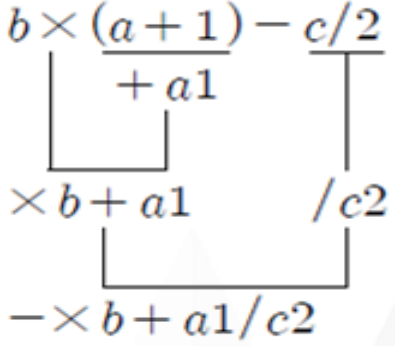
전위표기법	중위표기법	후위표기법
-/ x 3×+y1z	x /3-y+1×z	x 3/y1+z×-

4 이진 트리의 수식 표현

◆ 예) 다음 수식을 전위표기법과 후위표기법으로 나타내시오

$$b \times (a + 1) - c/2$$

(풀이)
전위표기법은 연산자를 연관된 2개의 피연산자 앞에 위치하도록 하면 되는데 다음과 같은 순서로 변환하면 됨



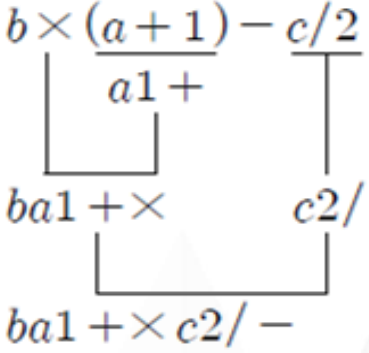
➡ 따라서 최종적으로 얻은 전위표기법은 **−×b+a1/c2**

4 이진 트리의 수식 표현

◆ 예) 다음 수식을 전위표기법과 후위표기법으로 나타내시오

$$b \times (a + 1) - c/2$$

(풀이)
후위표기법은 연산자가 2개의
피연산자 뒤에 위치하도록 표현하면
되고 다음과 같은 순서로 변환하면 됨



➡ 따라서 최종적인 후위표기법의 결과는 **ba1+×c2/−**

2 이진 탐색 트리

2 이진 탐색 트리

1 순서 트리

- 데이터가 왼쪽 자식이나 오른쪽 자식 위치 중 어디에 놓이느냐에 따라 다른 의미를 갖음
- 이 중 이진 트리는 순서가 있는 데이터들을 삽입, 삭제, 정렬, 탐색 등을 효율적으로 할 수 있음

2 이진 탐색 트리

2 이진 트리

- 순서가 있는 데이터들에서
  특정 값을 찾거나 저장하는데 많이 사용

3 이진 탐색 트리

- 임의의 정점의 좌측 부분 트리에는 해당 정점보다 작은 값의 데이터들이 놓이고, 우측 부분 트리에는 더 큰 값들이 오도록 구성
- 탐색은 항상 루트에서 시작하여 크기 관계에 따라 좌측 또는 우측 자식을 따라가면서 이루어짐
- 효율적이고 빠르게 데이터 검색과 삽입, 삭제, 정렬할 수 있도록 이진 트리에 몇가지 제약조건을 추가한 트리

3 이진 탐색 트리

- 데이터의 삽입, 삭제가 간단하고
  특정 키값을 빠르게 찾을 수 있음

- 데이터의 삽입, 삭제, 탐색 등이
  자주 발생하는 경우에 효율적인 구조

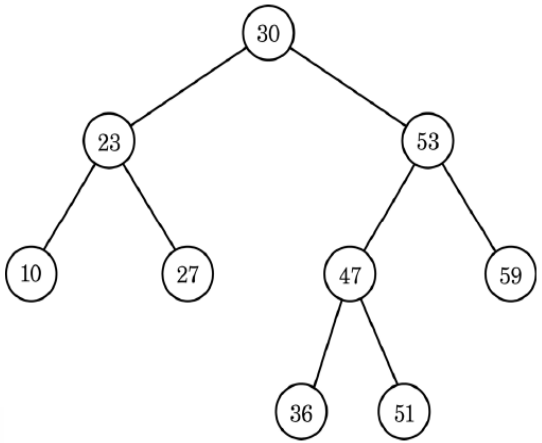
3 이진 탐색 트리

- 정의
  - 이진 트리에서 모든 노드가 키값을 가지고 있으며 키들이 다음의 속성을 만족할 때, 주어진 이진 트리를 이진 탐색 트리(Binary Search Tree)라 함

    1) 임의의 노드 R_i에 대해, R_i의 왼쪽 부분 트리의 키값들은 R_i의 키값 K_i보다 작아야 함
    2) 임의의 노드 R_i에 대해, R_i의 오른쪽 부분 트리의 키값들은 R_i의 키값 K_i보다 커야 함

2 이진 탐색 트리

4 이진 탐색 트리의 예

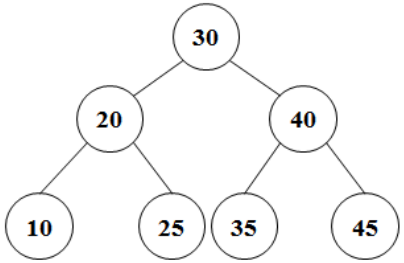


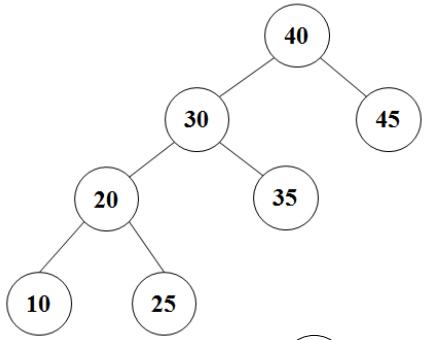
- 루트인 30을 기준으로 왼쪽 부분 트리에는 30보다 작은 값들이 위치하고 오른쪽 부분 트리에는 30보다 큰 값들이 위치함

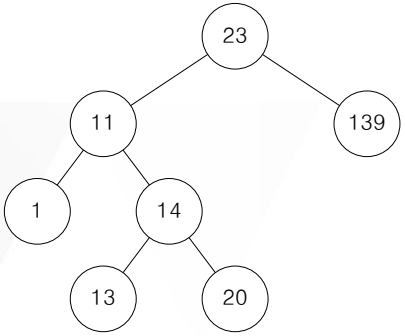
※출처: 이산수학, 류금한, 지식과미래

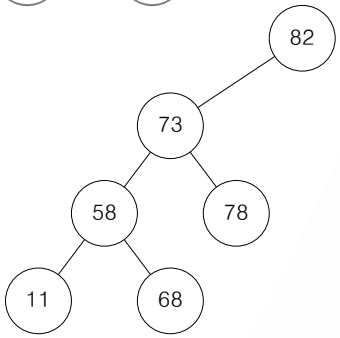
2 이진 탐색 트리

4 이진 탐색 트리의 예







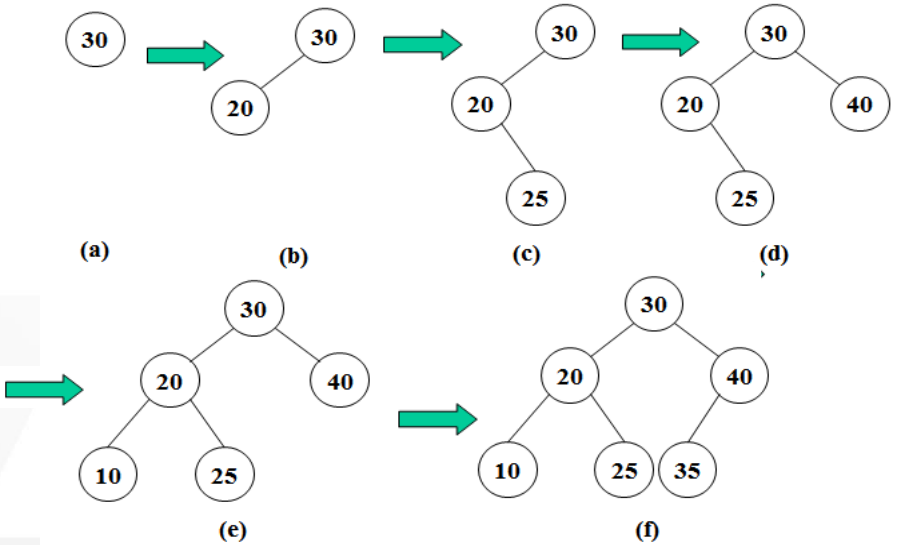


※출처: 이산수학, 류금한, 지식과미래

2 이진 탐색 트리

4 이진 탐색 트리의 예

- 예) 데이터가 30, 20, 25, 40, 10, 35의 순서로 원소가 삽입되는 경우 이진 탐색 트리가 만들어지는 과정



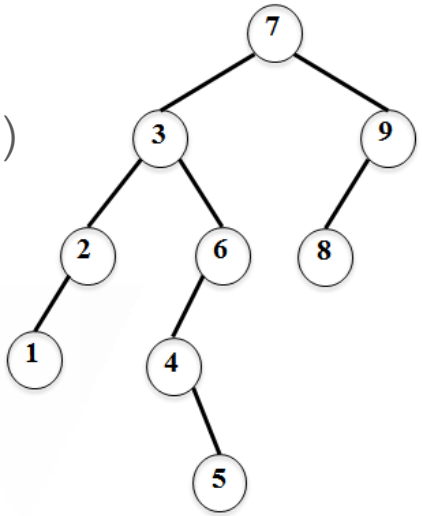
※출처: 쉽게 배우는 알고리즘, 문병로, 한빛미디어

2 이진 탐색 트리

4 이진 탐색 트리의 예

◆ 예) 데이터가 7, 3, 9, 6, 2, 4, 8, 1, 5 의 순서로 삽입될 때 이진 탐색 트리를 구성하시오

(풀이)



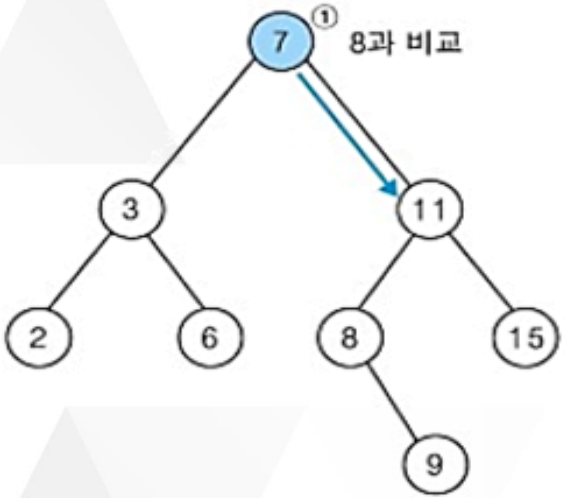
※출처: 쉽게 배우는 알고리즘, 문병로, 한빛미디어

5 이진 탐색 트리에서의 탐색

- 데이터 탐색은 루트에서부터 시작됨
- 루트 노드의 데이터와 찾으려는 데이터를 비교하여 루트 노드와 찾으려는 데이터가 같으면 탐색은 성공적으로 종료함
- 그렇지 않고 루트 노드의 데이터가 찾으려는 데이터보다 작으면 루트 노드의 오른쪽 부분 트리를 탐색해가고, 루트 노드의 데이터가 찾으려는 데이터보다 크면 루트 노드의 왼쪽 부분 트리를 탐색해 감

2 이진 탐색 트리

6 이진 탐색 트리에서의 탐색 예

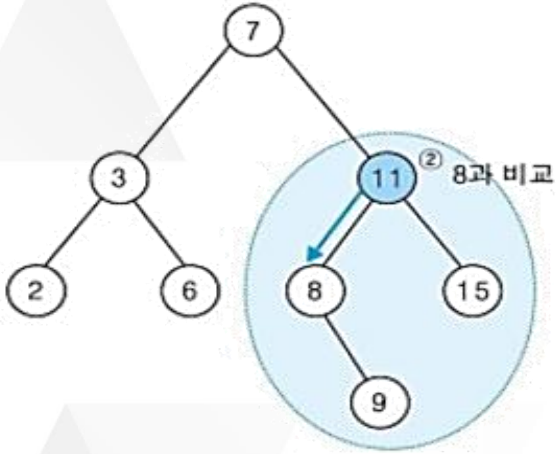


[데이터 8 탐색하기]

① 루트 노드의 데이터 7이 찾으려는 데이터 8보다 작으므로 오른쪽 부분 트리를 탐색함

※출처: 쉽게 배우는 알고리즘, 문병로, 한빛미디어

6 이진 탐색 트리에서의 탐색 예

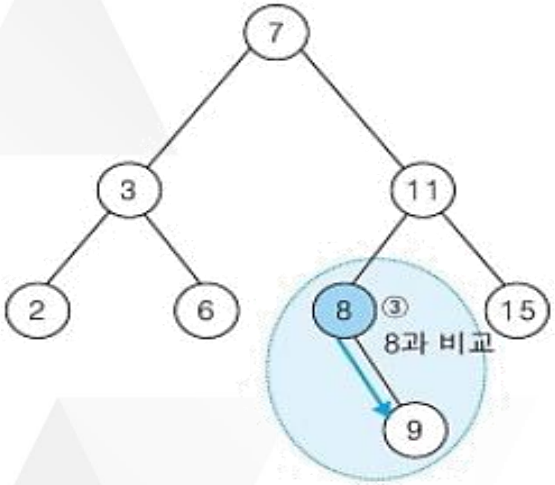


② 오른쪽 부분 트리의 루트 노드에 있는 데이터 11이 찾으려는 데이터 8보다 크므로 왼쪽 부분 트리를 탐색함

※출처: 쉽게 배우는 알고리즘, 문병로, 한빛미디어

2 이진 탐색 트리

6 이진 탐색 트리에서의 탐색 예



③ 왼쪽 부분 트리의
루트 노드가 8이므로 원하는
노드의 탐색이 성공하여
탐색 완료

※ 만약 단말 노드에 이를 때까지 같은 데이터를
찾지 못하면 탐색에 실패하는 것임

※출처: 쉽게 배우는 알고리즘, 문병로, 한빛미디어

3 신장 트리

3 신장 트리

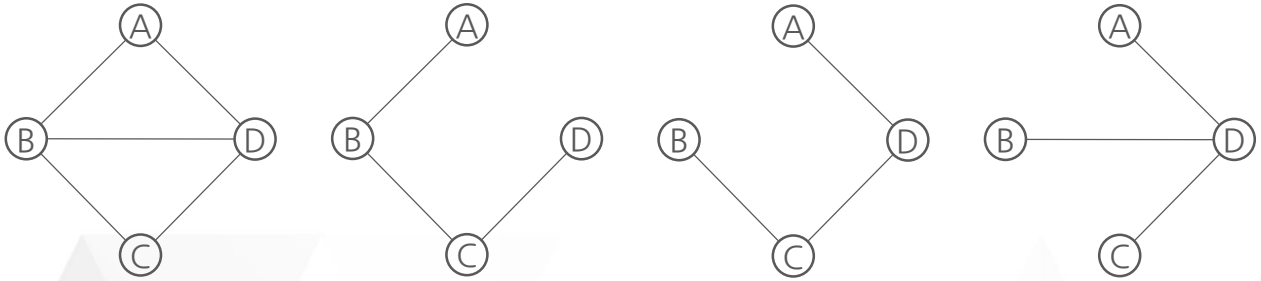
1 신장 트리(Spanning Tree)

- 그래프 내의 모든 정점들을 포함하는 트리
- 트리의 특수한 형태로 모든 정점들이 연결되어 있어야 하고 사이클을 포함해서는 안됨
- 임의의 그래프에서 만들 수 있는 신장 트리는 매우 다양함
- 최소 신장 트리는 가능한 신장 트리 중에 간선의 가중치의 합이 최소인 신장 트리

3 신장 트리

1 신장 트리(Spanning Tree)

◆ 그래프 G1과 신장 트리의 예



G1 G1의 신장 트리

※출처: 자바로 배우는 쉬운 자료구조, 이지영, 한빛미디어