

이진트리 보충

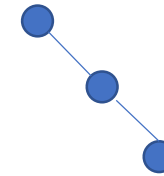
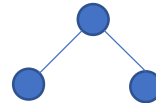
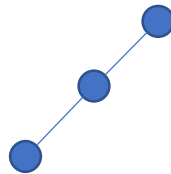
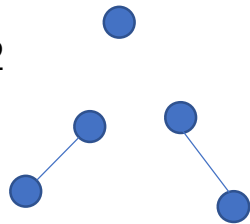
서로 다른 이진트리 개수

- $B(n)$: n 개의 노드를 가지는 서로 다른 이진트리의 개수

$$B(1) = 1$$

$$B(2) = 2$$

$$B(3) = 5$$



$$B(0) = B(1) = 1$$

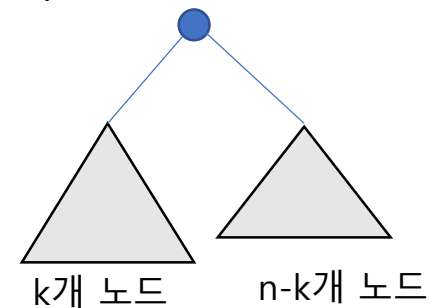
$$B(n) = B(n-1)B(0) + B(n-2)B(1) + \dots + B(k)B(n-k-1) + \dots + B(0)B(n-1), n > 1$$

$$= \sum_{k=0}^{n-1} B(k)B(n-k)$$

$$B(4) = B(0)B(3) + B(1)B(2) + B(2)B(1) + B(3)B(0)$$

$$5 + 2 + 2 + 5 = 14$$

$$B(5) \dots$$

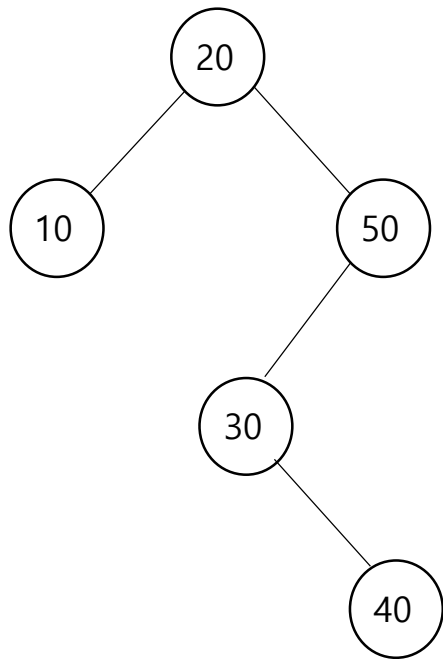


$B(n)$: Catalan Number

주어진 n개의 키를 가지는 이진탐색트리

- 주어진 n개 키 => 이진탐색트리

n = 5인 다음의 이진탐색트리가 만들어진 경우, 키들의 가능한 삽입순서



20, 10, 50, 30, 40
20, 50, 10, 30, 40
20, 50, 30, 10, 40
20, 50, 30, 40, 10

주어진 n 개의 키를 가지는 이진트리

- 주어진 n 개 키 => 이진탐색트리

n 개의 키들이 random하게 삽입될 때 높이 $O(\log n)$

=> n 개의 키들이 삽입될 때 평균적인 높이가 $O(\log n)$

평균적인 수행시간: 탐색 $O(\log n)$, 삽입 $O(\log n)$, 삭제 $O(\log n)$

이진트리 구성

Inorder traverse할 때 방문되는 노드들의 원소들:

5, 20, 40, 50, 60, 70

Preorder travers할 때 순서대로 노드들의 원소들:

40, 20, 5, 70, 60, 50

⇒이진트리

루트: preorder traverse의 첫번째 원소

left subtree: inorder traverse에서 루트 원소 이전에 나오는 원소들

right subtree: inorder traverse에서 루트 원소 다음에 나오는 원소들

이진트리 구성

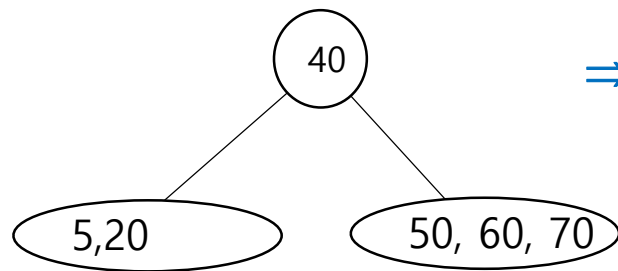
Inorder traverse할 때 방문되는 노드들의 원소들:

5, 20, 40, 50, 60, 70

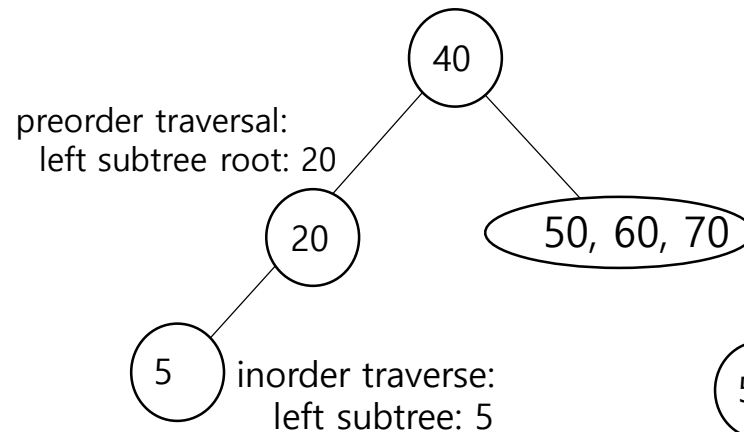
Preorder travers할 때 순서대로 노드들의 원소들:

40, 20, 5, 70, 60, 50

⇒ preorder traverse한 원소들 순서로 부터
root: 40

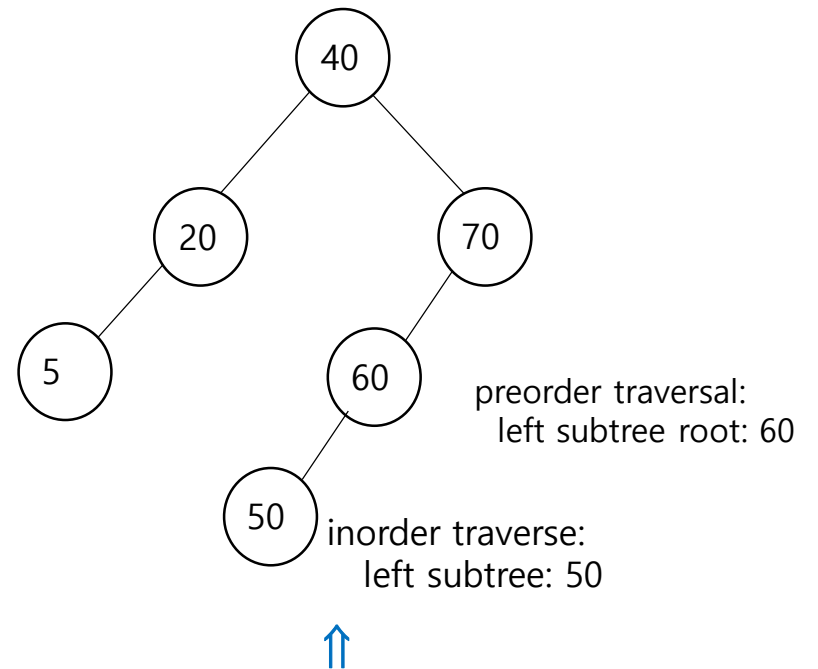


inorder traverse:
left subtree: 5, 20
right subtree: 50, 60, 70



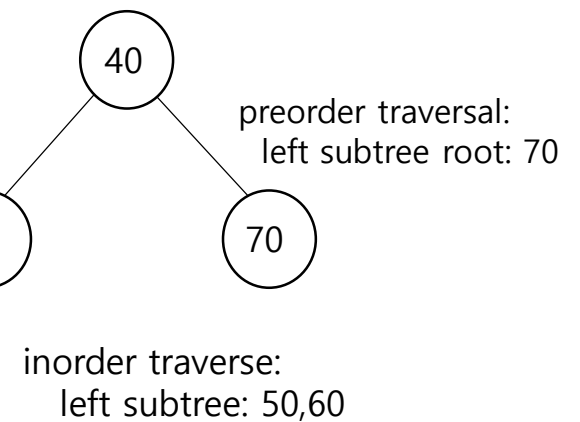
preorder traversal:
left subtree root: 20

inorder traverse:
left subtree: 5



preorder traversal:
left subtree root: 60

inorder traverse:
left subtree: 50



preorder traversal:
left subtree root: 70

inorder traverse:
left subtree: 50, 60

수식 트리 (expression tree)

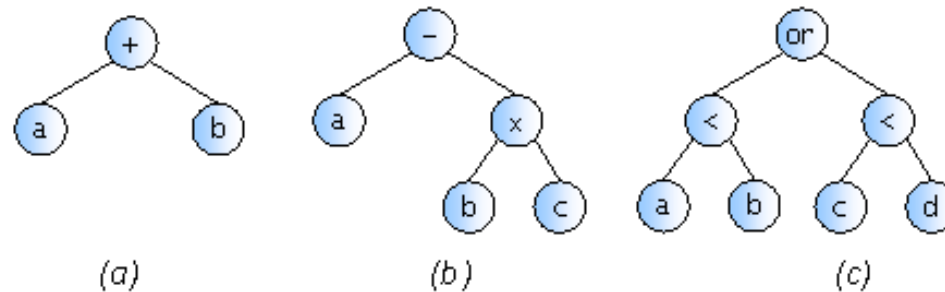
- 수식트리 : 수식을 이진트리형태로 표현한 것

op: 연산자, opr1: 왼쪽 피연산자, opr2: 오른쪽 피연산자

수식: opr1 op opr2 => 이진트리

루트: op, 왼쪽 자식 노드: opr1, 오른쪽 자식 노드: opr2

- 예)



수식	$a + b$	$a - (b \times c)$	$(a < b) \text{ or } (c < d)$
전위순회	$+ a b$	$- a \times b c$	$\text{or} < a b < c d$
중위순회	$a + b$	$a - b \times c$	$a < b \text{ or } c < d$
후위순회	$a b +$	$a b c \times -$	$a b < c d < \text{or}$

수식트리 결과값 계산

- 후위순회를 사용
 - 서브트리의 값을 순환호출로 계산
 - 비단말노드를 방문할때 양쪽서브트리의 값을 저장된 연산자를 이용하여 계산한다

evaluate(exp)

```
if exp = NULL
  then return 0;
else x←evaluate(LEFT(exp));
     y←evaluate(RIGHT(exp));
     op←DATA(exp);
     return (x op y);
```

