

13 강

알고리즘과 자료구조

트리1

서울과학기술대학교 신일훈 교수

학습목표

- 1 트리의 개념을 이해한다.
- 2 이진 트리의 개념 및 트리 순회 방법을 이해한다.
- 3 이진 탐색 트리의 개념 및 주요 연산을 이해한다.



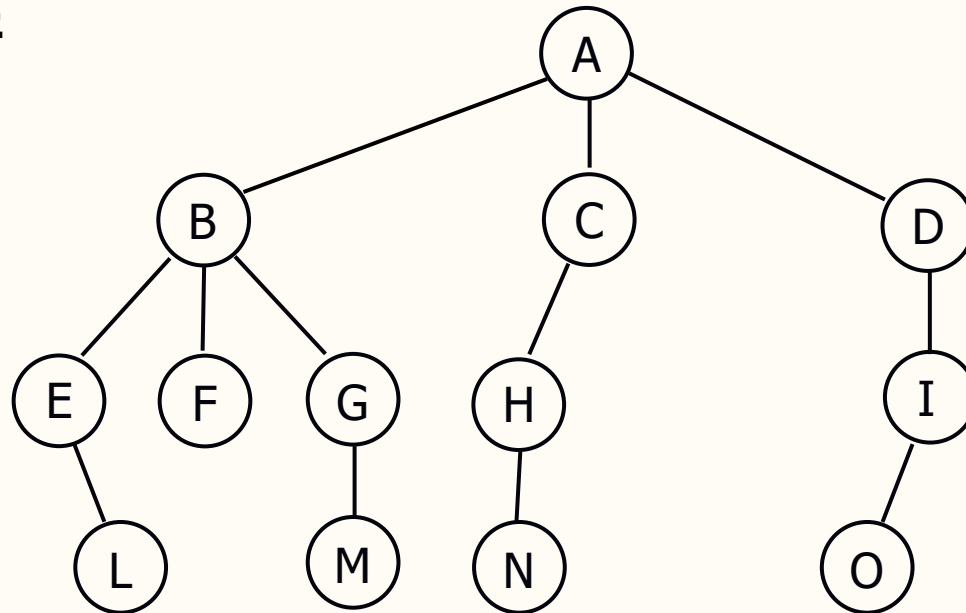


트리개념

1. 트리 개념

■ 트리

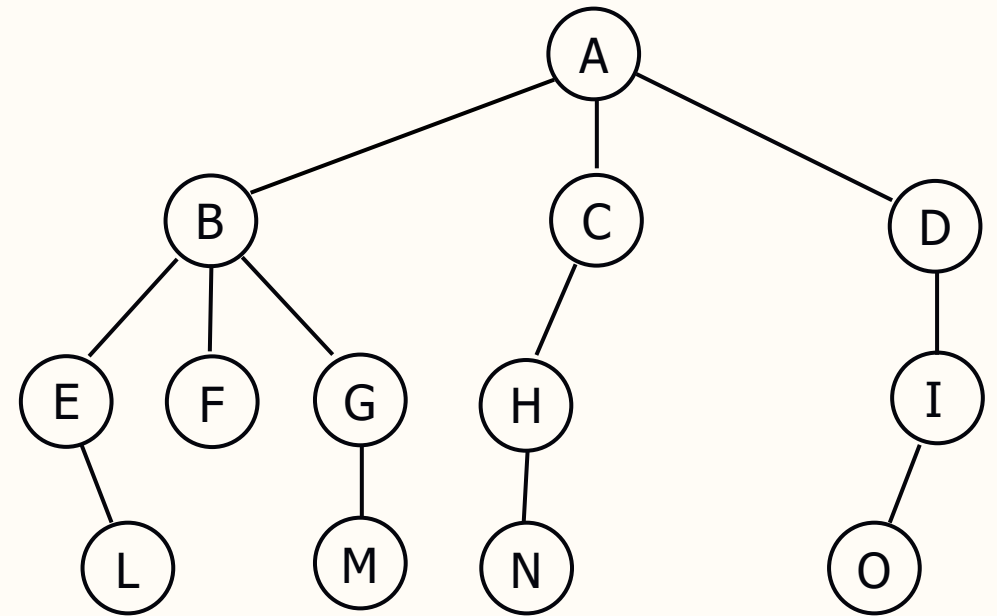
- 노드와 노드를 연결하는 링크로 구성됨
- 계층 구조 형태의 비선형 자료구조
- 사이클이 없음



1. 트리 개념

용어

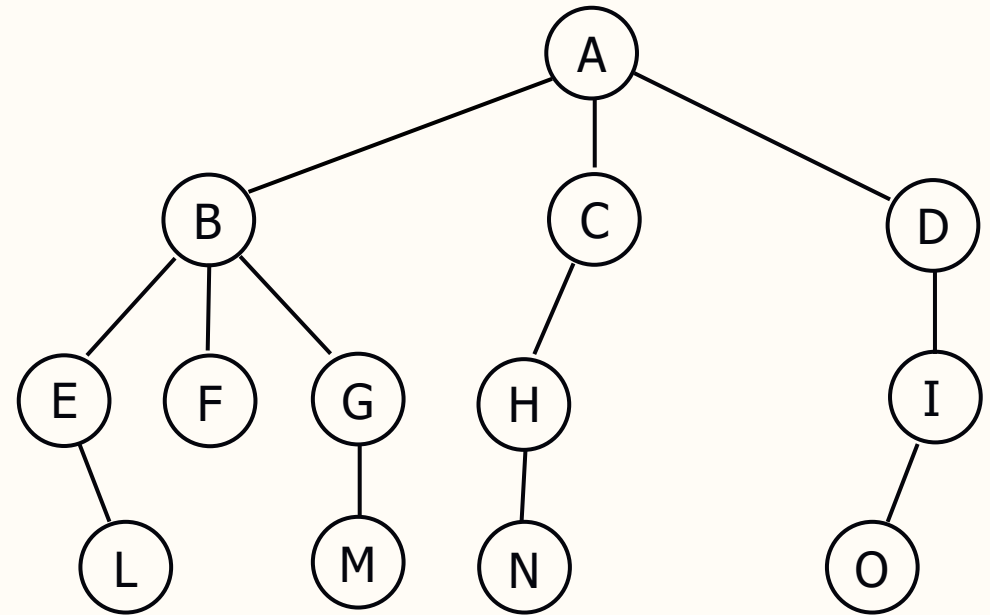
- 루트(Root) – 트리의 최상위 노드
- 자식(Child) – 노드의 하위에 연결된 노드
- 부모(Parent) – 노드의 상위에 연결된 노드
- 차수(Degree) – 자식의 수
- Leaf – 자식이 없는 노드



1. 트리 개념

용어

- 레벨(Level)
 - 루트는 레벨 0 또는 1.
 - 아래 층으로 내려가며 레벨이 1씩 증가
- 레벨은 깊이(Depth)와 동일
- 높이(Height)
 - 트리의 최대 레벨
- 키(Key)
 - 노드에 저장된 탐색에 사용되는 정보

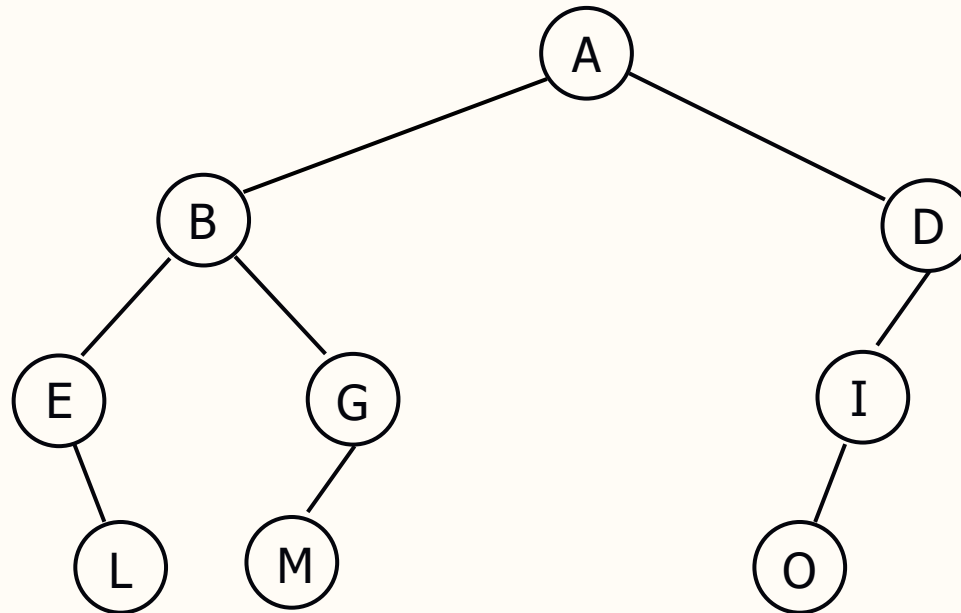




이진트리개념및트리순회

2. 이진 트리 개념 및 트리 순회

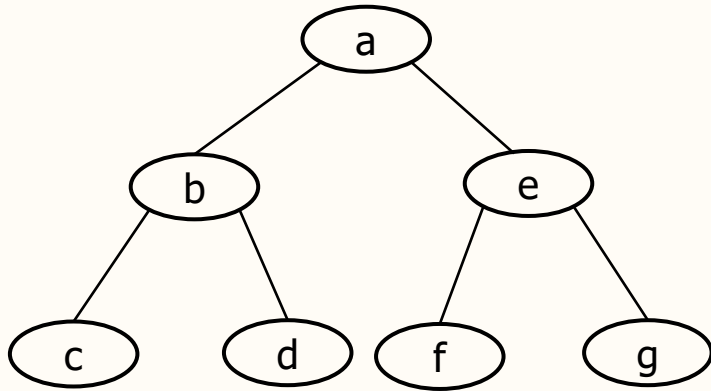
- 이진 트리(binary tree)
 - 각 노드의 자식 수가 2 이하인 트리



2. 이진 트리 개념 및 트리 순회

■ 이진 트리 종류

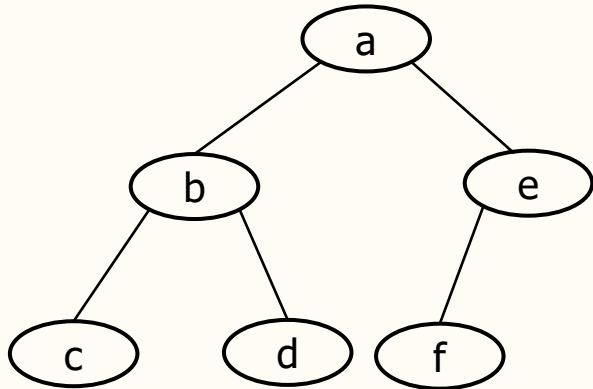
- 포화이진트리
 - 각 내부 노드(레벨이 높이보다 작은 노드)가 2개의 자식 노드를 가지는 트리



2. 이진 트리 개념 및 트리 순회

■ 이진 트리 종류

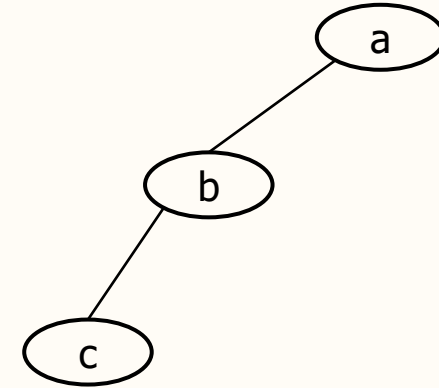
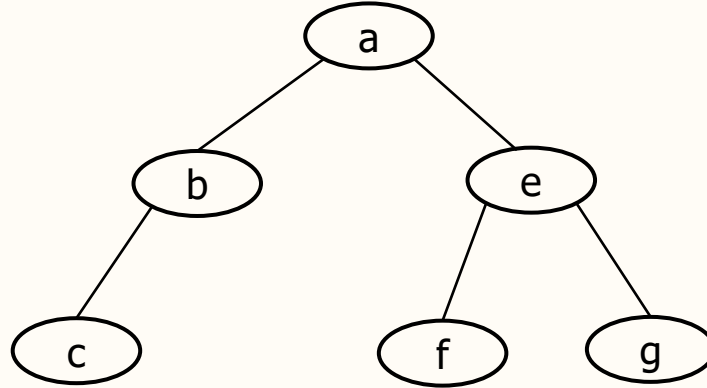
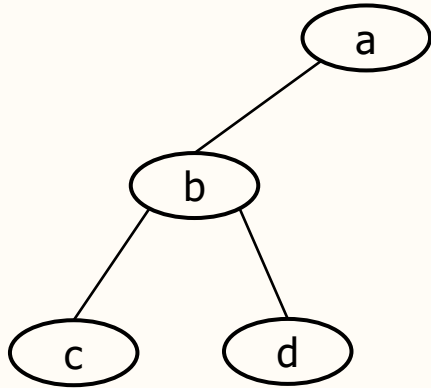
- 완전이진트리
 - 마지막 레벨을 제외한 트리가 포화이진트리이며,
마지막 레벨에는 노드들이 왼쪽부터 채워진 트리



2. 이진 트리 개념 및 트리 순회

■ 이진 트리 종류

- 불완전이진트리



2. 이진 트리 개념 및 트리 순회

- 이진 트리 순회(traversal)
 - 전위순회 (preorder traversal)
 - 중위순회 (inorder traversal)
 - 후위순회 (postorder traversal)
 - 레벨순회 (level-order traversal)

2. 이진 트리 개념 및 트리 순회

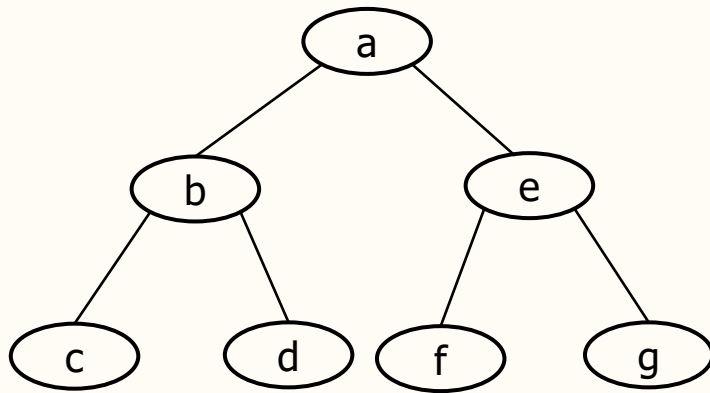
■ 전위순회

- 최상위 노드에서 순회 시작.
- 노드 n 에 도착하면 n 을 먼저 탐색
- 이후, n 의 왼쪽 서브 트리 순회.
- 왼쪽 서브 트리 순회 후에는 n 의 오른쪽 서브 트리 순회
- NLR 순서의 순회

2. 이진 트리 개념 및 트리 순회

■ 전위순회

- $a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow f \rightarrow g$



2. 이진 트리 개념 및 트리 순회

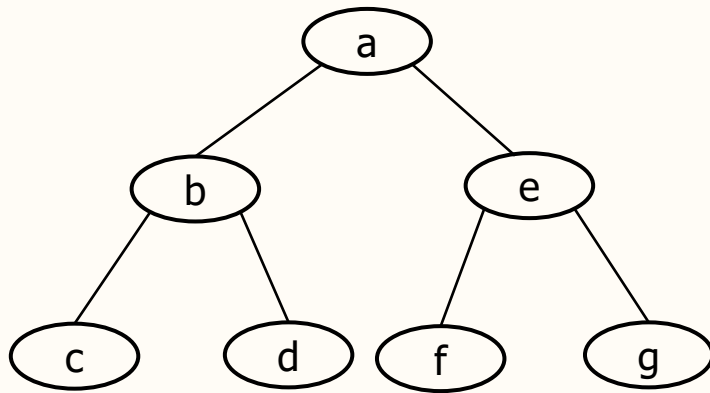
■ 중위순회

- 최상위 노드에서 순회 시작.
- 노드 n 에 도착하면 n 의 탐색을 보류하고 먼저 n 의 왼쪽 서브 트리 순회.
- 왼쪽 서브 트리의 모든 노드를 순회한 후에는 n 을 탐색.
- 이후 n 의 오른쪽 서브 트리 순회
- LNR 순서의 순회

2. 이진 트리 개념 및 트리 순회

■ 중위순회

• $c \rightarrow b \rightarrow d \rightarrow a \rightarrow f \rightarrow e \rightarrow g$



2. 이진 트리 개념 및 트리 순회

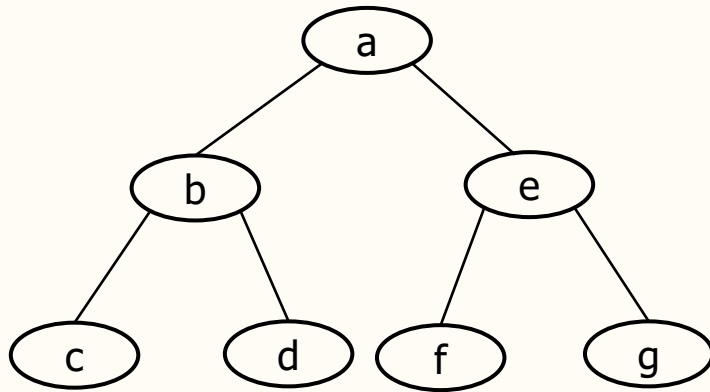
■ 후위순회

- 최상위 노드에서 순회 시작.
- 노드 n 에 도착하면 n 의 탐색을 보류하고 먼저 n 의 왼쪽 서브 트리 순회.
- 왼쪽 서브 트리의 모든 노드를 순회한 후에는 n 의 오른쪽 서브 트리 순회.
- 이후 n 을 탐색.
- LRN 순서의 순회

2. 이진 트리 개념 및 트리 순회

■ 후위순회

- $c \rightarrow d \rightarrow b \rightarrow f \rightarrow g \rightarrow e \rightarrow a$



2. 이진 트리 개념 및 트리 순회

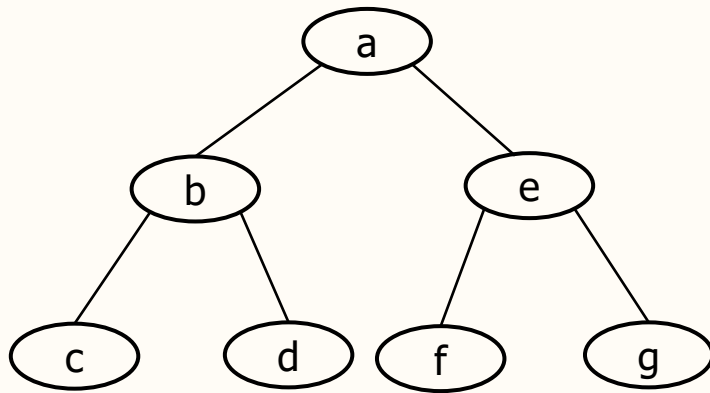
■ 레벨순회

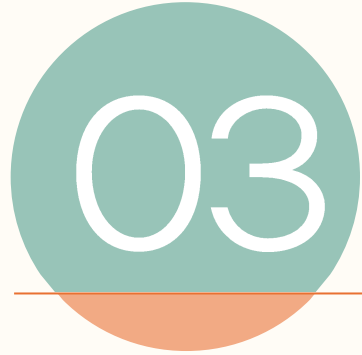
- 최상위 레벨의 노드부터 왼쪽에서 오른쪽 방향으로 순회
- 특정 레벨의 순회가 끝나면, 다음 하위 레벨의 노드들을 왼쪽부터 오른쪽 방향으로 순회
- 이를 마지막 레벨까지 반복

2. 이진 트리 개념 및 트리 순회

■ 레벨순회

- $a \rightarrow b \rightarrow e \rightarrow c \rightarrow d \rightarrow f \rightarrow g$





이진탐색트리개념

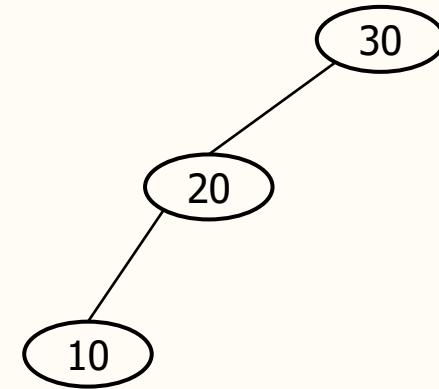
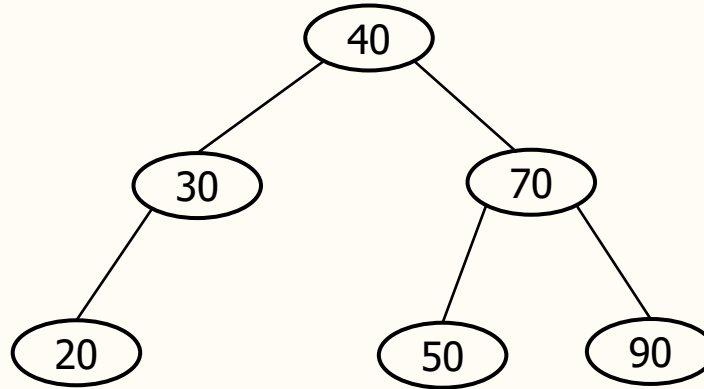
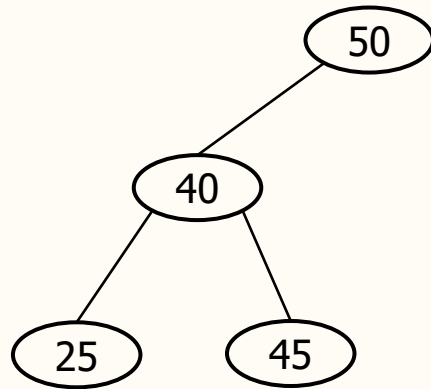
3. 이진 탐색 트리 개념

■ 이진 탐색 트리

- 이진 트리의 한 종류로서 다음의 조건을 만족하는 트리
 - 모든 노드에 대해, 노드 N 의 키가 N 의 왼쪽 서브 트리의 키들보다 크고, N 의 오른쪽 서브 트리의 키들보다 작아야 함.

3. 이진 탐색 트리 개념

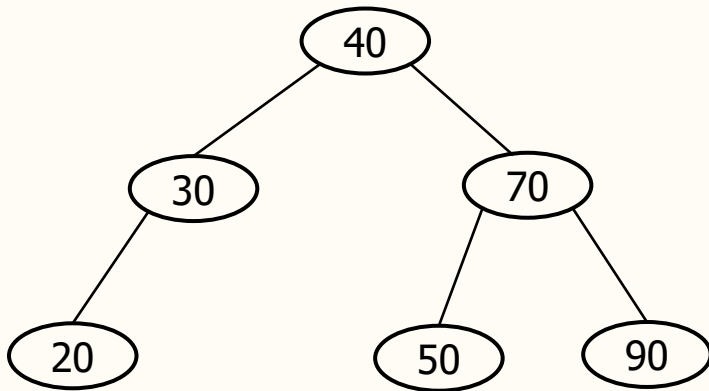
이진 탐색 트리



3. 이진 탐색 트리 개념

장점

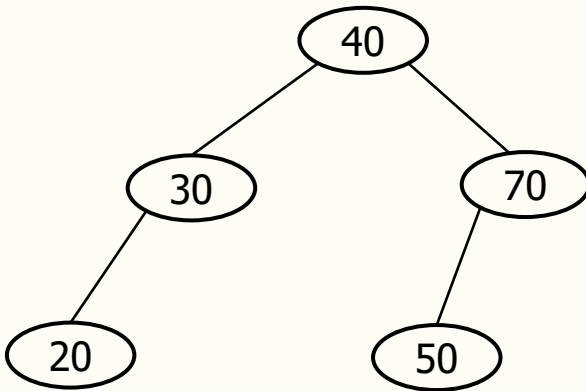
- 탐색의 효율성
 - 최대값 탐색: 오른쪽 자식을 계속 따라가는 것을 반복하고,
더 이상 오른쪽 자식이 없는 노드를 만나면, 해당 노드가 최대값을 가진다.



3. 이진 탐색 트리 개념

장점

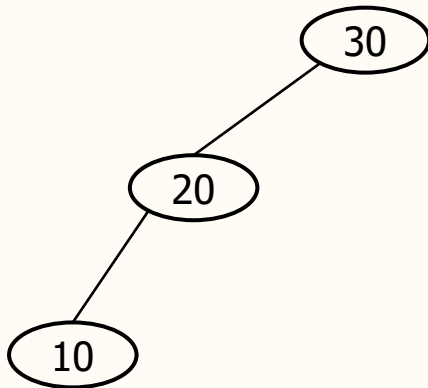
- 탐색의 효율성
 - 최대값 탐색: 오른쪽 자식을 계속 따라가는 것을 반복하고,
더 이상 오른쪽 자식이 없는 노드를 만나면, 해당 노드가 최대값을 가진다.



3. 이진 탐색 트리 개념

장점

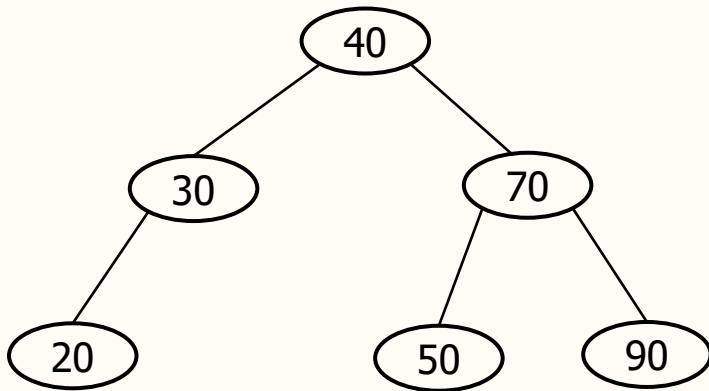
- 탐색의 효율성
 - 최대값 탐색: 오른쪽 자식을 계속 따라가는 것을 반복하고,
더 이상 오른쪽 자식이 없는 노드를 만나면, 해당 노드가 최대값을 가진다.



3. 이진 탐색 트리 개념

장점

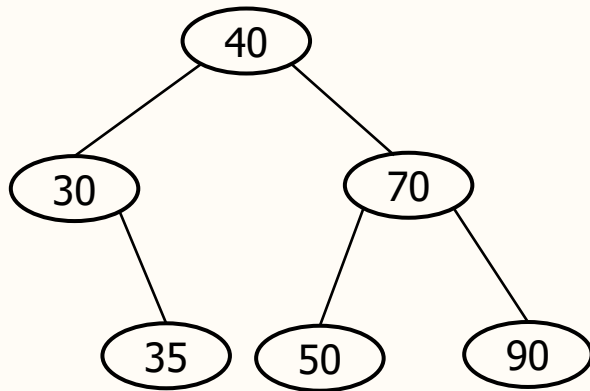
- 탐색의 효율성
 - 최소값 탐색: 왼쪽 자식을 계속 따라가는 것을 반복하고,
더 이상 왼쪽 자식이 없는 노드를 만나면, 해당 노드가 최소값을 가진다.



3. 이진 탐색 트리 개념

장점

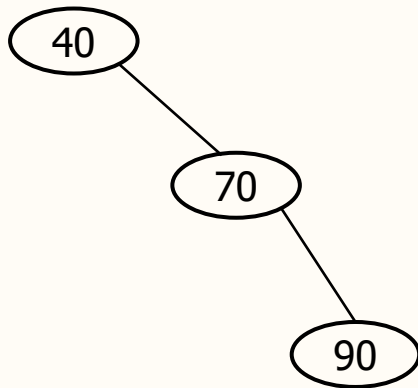
- 탐색의 효율성
 - 최소값 탐색: 왼쪽 자식을 계속 따라가는 것을 반복하고,
더 이상 왼쪽 자식이 없는 노드를 만나면, 해당 노드가 최소값을 가진다.



3. 이진 탐색 트리 개념

장점

- 탐색의 효율성
 - 최소값 탐색: 왼쪽 자식을 계속 따라가는 것을 반복하고,
더 이상 왼쪽 자식이 없는 노드를 만나면, 해당 노드가 최소값을 가진다.



3. 이진 탐색 트리 개념

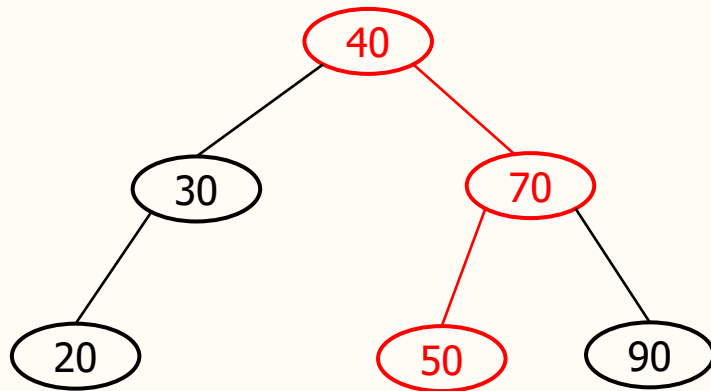
장점

- 탐색의 효율성
 - 특정값 탐색
 1. 최상위 노드부터 시작
 2. 노드의 키와 탐색 키를 비교
 - 탐색 키가 작으면 왼쪽 자식 노드에 대해 2를 반복
 - 왼쪽 자식 노드가 없으면 찾는 노드가 없음 & 종료
 - 탐색 키가 크면 오른쪽 자식 노드에 대해 2를 반복
 - 오른쪽 자식 노드가 없으면 찾는 노드가 없음 & 종료
 - 키가 동일하면 탐색을 종료함 (발견)

3. 이진 탐색 트리 개념

장점

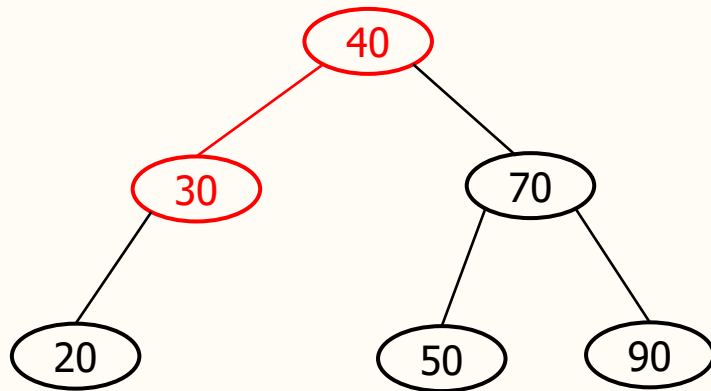
- 탐색의 효율성
 - 특정값 (50) 탐색 => 성공



3. 이진 탐색 트리 개념

장점

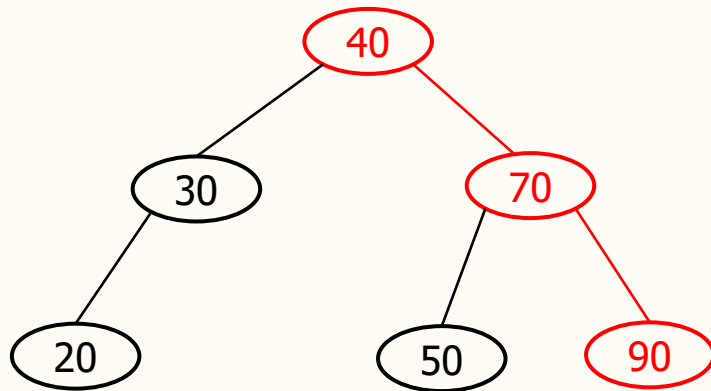
- 탐색의 효율성
 - 특정값 (35) 탐색 => 실패



3. 이진 탐색 트리 개념

장점

- 탐색의 효율성
 - 특정값 (80) 탐색 => 실패



3. 이진 탐색 트리 개념

장점

- 탐색의 효율성
 - 탐색의 최악 시간 복잡도
 - $O(h)$. h 는 트리의 높이
 - $O(n)$. n 은 노드의 개수



이진탐색트리 주요 연산

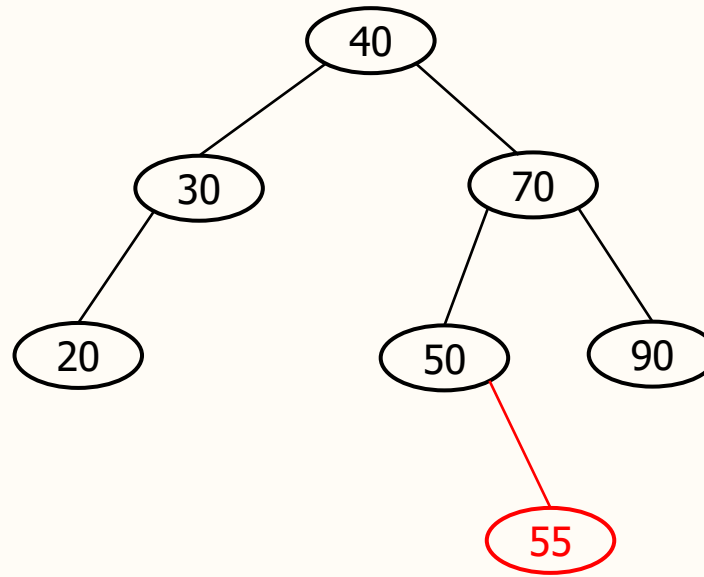
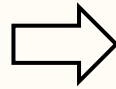
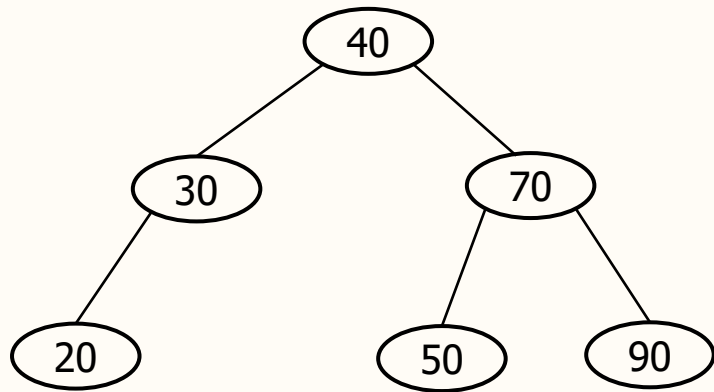
4. 이진 탐색 트리 주요 연산

■ 삽입

1. 최상위 노드가 None이면 (트리가 비어 있는 상태), 최상위 노드로 삽입하고 종료
2. 최상위 노드부터 시작하여 노드의 키와 탐색 키를 비교
 - 탐색 키가 더 작고 왼쪽 자식 노드가 존재하면 왼쪽 자식 노드에 대해 2번 작업 반복
 - 탐색 키가 더 크고 오른쪽 자식 노드가 존재하면 오른쪽 자식 노드에 대해 2번 작업 반복
 - 키가 동일하면 충돌 또는 update 케이스. 실패로 처리 또는 값 수정 후 종료
 - 키가 다른데 반복할 자식 노드가 존재하지 않으면,
키에 따라 왼쪽 자식 또는 오른쪽 자식 노드로 삽입

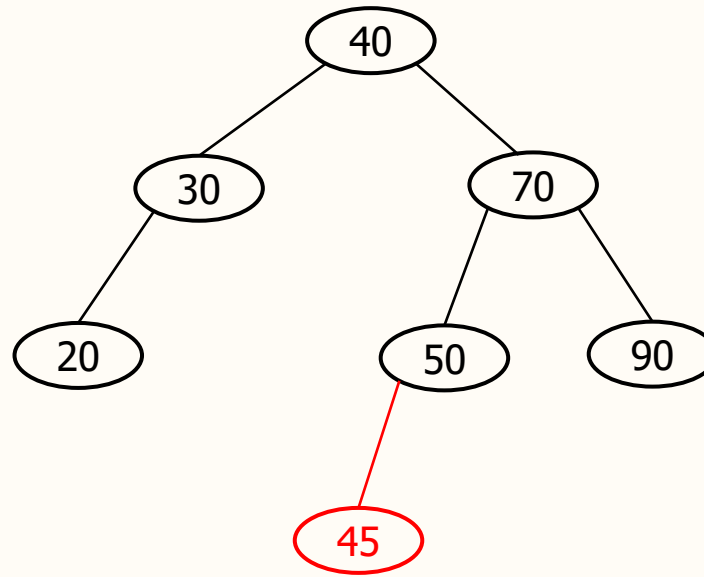
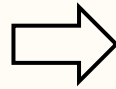
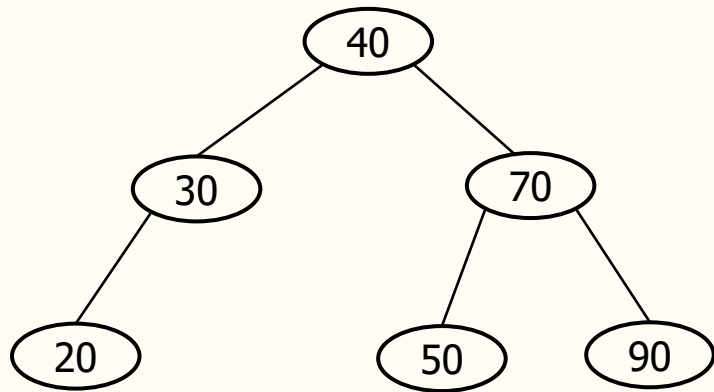
4. 이진 탐색 트리 주요 연산

삽입 (55)



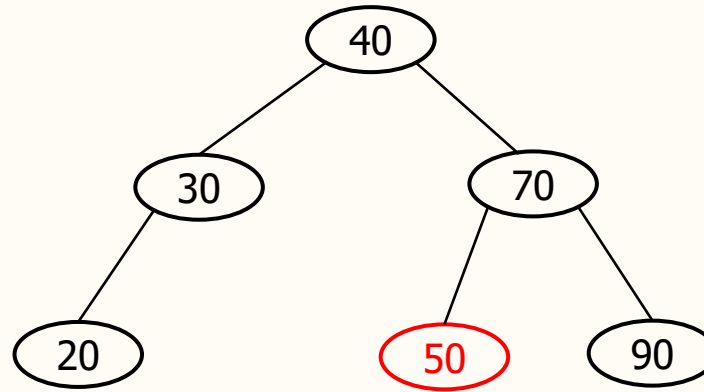
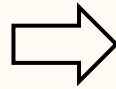
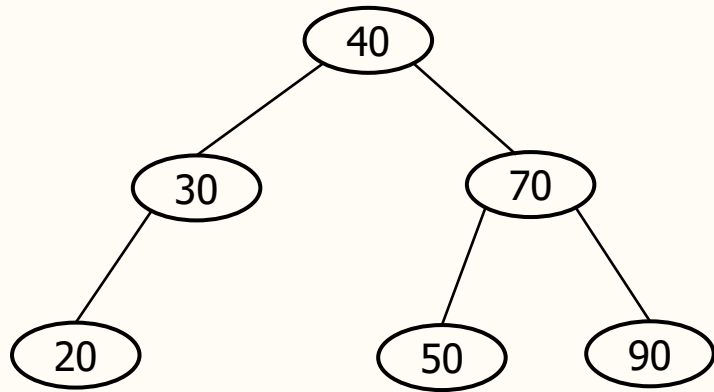
4. 이진 탐색 트리 주요 연산

삽입 (45)



4. 이진 탐색 트리 주요 연산

삽입 (50)



4. 이진 탐색 트리 주요 연산

■ 탐색

1. 최상위 노드부터 시작
2. 노드의 키와 탐색 키를 비교
 - 탐색 키가 작고 왼쪽 자식 노드가 존재하면 왼쪽 자식 노드에 대해 2를 반복
 - 탐색 키가 크고 오른쪽 자식 노드가 존재하면 오른쪽 자식 노드에 대해 2를 반복
 - 키가 동일하면 탐색을 종료함 (발견)
 - 키가 동일하지 않은데, 반복할 자식 노드가 없으면 찾는 노드가 없음 & 종료

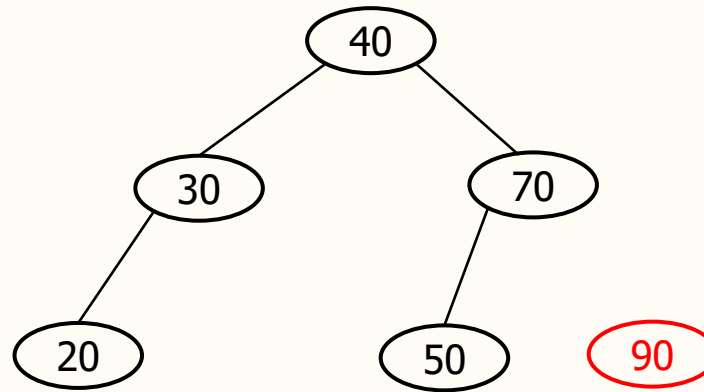
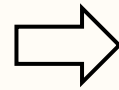
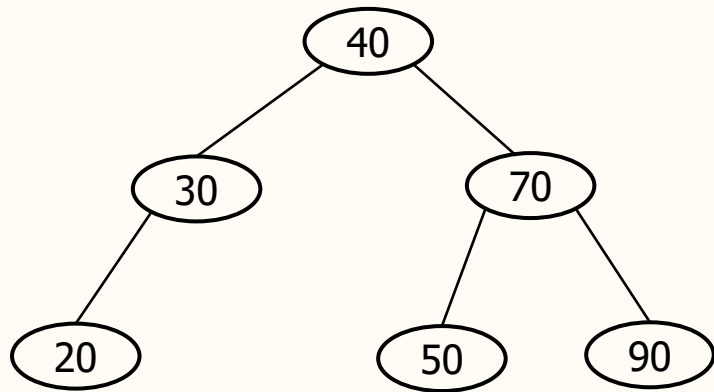
4. 이진 탐색 트리 주요 연산

■ 삭제

1. 타겟이 리프 노드인 경우 (자식이 없는 경우)
 - 부모 노드와 타겟 노드를 연결하는 링크 제거
2. 타겟이 하나의 자식 노드를 갖는 경우
 - 타겟의 부모와 타겟의 자식 노드를 직접 연결 (손자가 자식으로 바뀌는 격)
3. 타겟이 두 개의 자식 노드를 갖는 경우
 - 타겟의 자리를 다른 후계자로 대체
 - 후계자는 타겟의 왼쪽 서브 트리 중 키가 가장 큰 노드 또는 오른쪽 서브 트리 중 키가 가장 작은 노드
 - 후계자 노드 삭제 (후계자 노드는 1번 또는 2번 경우이므로 해당 작업 실행함)

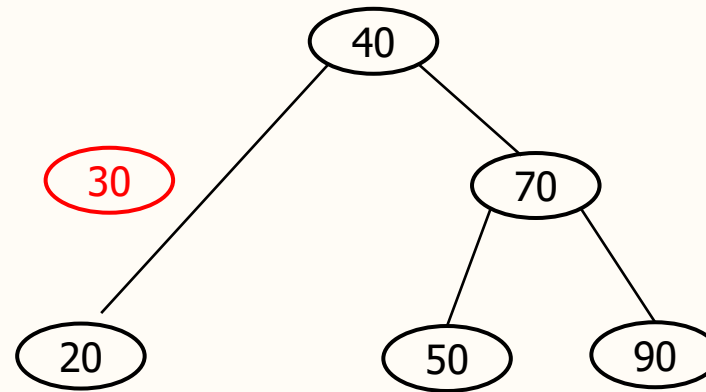
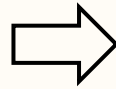
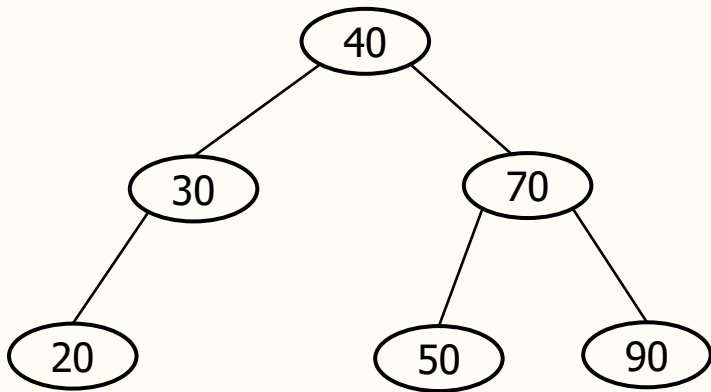
4. 이진 탐색 트리 주요 연산

■ 삭제 (90)



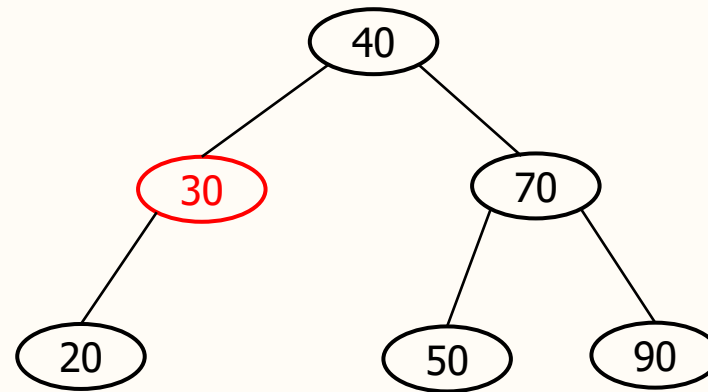
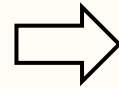
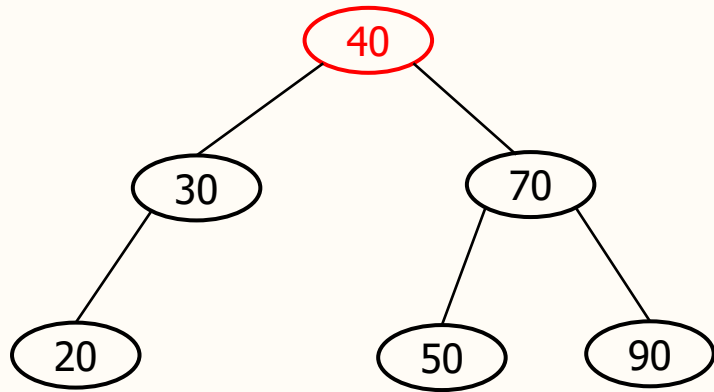
4. 이진 탐색 트리 주요 연산

❏ 삭제 (30)

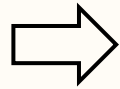


4. 이진 탐색 트리 주요 연산

❏ 삭제 (40)

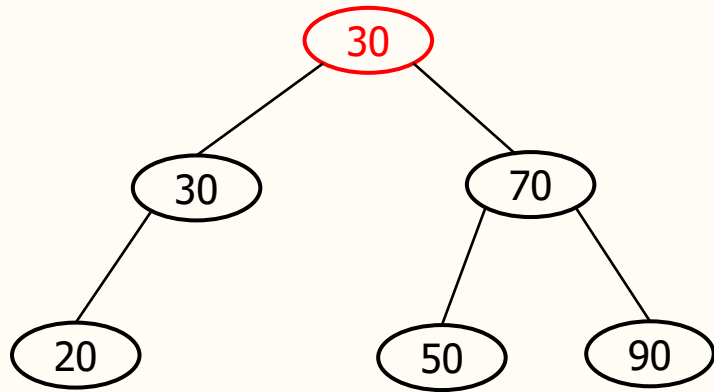


후계자 노드: 30

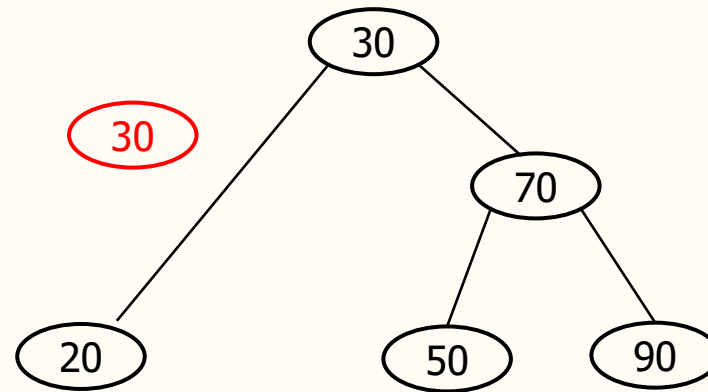
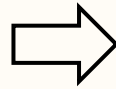


4. 이진 탐색 트리 주요 연산

❏ 삭제 (40)



후계자 노드를 타겟 노드로 복사



원 후계자 노드 제거 (2번 경우에 해당)

정리하기

- ✓ 트리의 개념
- ✓ 이진 트리의 개념 및 트리 순회
- ✓ 이진 탐색 트리 개념 및 주요 연산

14강

다음시간 안내▶▶▶

트리2