

제7장 트리

1. 트리

- 트리의 의미
 - 검색의 편리함
 - 논리적 계층
 - 계급적 특성

2. 용어와 논리적 표현 방법

- 노드: 트리의 항목 / 트리에 저장되는 데이터의 묶음
- 부모노드-자식노드: 상하 계층구조가 있고 직접적으로 연결된 노드로서 상위 계층의 부모노드와 하위계층의 자식노드
- 루트노드: 트리의 최상위 노드(부모가 없는 노드)
- 서브트리: 부모노드를 삭제하면 생기는 트리들
- 리프노드: 트리의 맨 끝에 있으면서, 자신의 서브트리를 갖지 않는 노드
- 진입/진출 차수
 - 루트노드: 진입차수=0
 - 루트를 제외한 모든 노드의 진입차수: 1
 - 리프노드: 진출차수=0
- 노드의 레벨: 루트로부터 그 노드까지 이어진 선(경로)의 길이
- 트리의 높이: 루트로부터 가장 멀리 있는 노드까지 이어진 선(경로)의 길이에 1을 더한 값

3. 추상 자료형

4. 이진 트리

- 모든 노드의 차수가 2이하인 트리
- 수학적으로 이진트리의 구성에 관한 이론을 정리하기 쉽고, 컴퓨터 내부에서 구현하기도 효율적임
- 모든 노드가 2개 이하의 자식노드를 가지므로 일반성을 잃지 않고 '오른쪽', '왼쪽'이라는 방향 개념을 부여할 수 도 있음
 - 오른쪽 노드, 왼쪽 노드의 개념적 접근도 있음
- 포화 이진 트리
 - 이진 트리의 각 레벨에서 허용되는 최대 개수 노드를 가지는 트리
- 배열을 이용한 이진 트리의 구현

- 트리가 완전 이진 트리 또는 포화 이진 트리인 경우 낭비되는 공간이 없어 효율적임
- 트리가 깊어질수록 기억장소 낭비가 2의 거듭제곱에 비례하여 심해짐

○ 포인터를 이용한 이진 트리의 구현

```
struct node {
    struct node *left ;
    struct node *right ;
    int info ;
}
```

*left	info	*right
-------	------	--------

5. 이진 트리 연산

- 이진 트리의 순회
 - 이진 트리의 각 노드를(빠짐없이 그리고 중복없이) 한 번씩 방문하는 것
- 이진 트리의 전위 순회
 - 루트노드 → 왼쪽 자식노드 → 오른쪽 자식노드
- 이진 트리의 후위 순회
 - 왼쪽 자식노드 → 오른쪽 자식노드 → 루트노드
- 이진 트리의 중위 순회
 - 왼쪽 자식노드 → 루트노드 → 오른쪽 자식노드
- 이진 트리의 생성/삽입/삭제
 - 일반적인 이진 트리를 생성하는 것은 연결 리스트 연산을 사용함
 - 첫 노드를 생성하면 루트 노드가 되고, 새로운 노드를 추가하려면 연결 리스트의 삽입 연산을 사용함
 - 노드를 삭제할 때, 삭제하려는 노드가 리프노드인 경우는 해당 노드를 가리키는 포인터를 null로 지정하면 됨
 - 리프노드가 아닌 경우에는 삭제하려는 노드의 자식노드에 대한 처리를 추가로 해주어야 함

6. 일반 트리를 이진 트리로 변환

- 자식 노드는 왼쪽 자식 노드에 형제 노드는 오른쪽 자식노드에 배치