

트리는 계층적인 구조를 가진 데이터를 표현하는 데 사용되는 비선형 자료구조이다. 가계도, 조직 우리 주변에서 쉽게 찾아볼 수 있는 계층적인 구조를 생각하면 이해가 쉽다.

트리 하위 설명

1. **Node** : 트리를 구성하는 기본 단위이다. 각 노드는 데이터를 저장하고 다른 노드와 연결한다.
2. **Root Node** : 트리의 가장 상위에 위치한 노드이다.
3. **Child Node** : 다른 노드의 하위에 위치한 노드이다.
4. **Parent Node** : 자식 노드의 상위에 위치한 노드이다.
5. **Sibling Node** : 같은 부모 노드를 가진 노드이다.
6. **Leaf Node** : 자식 노드가 없는 노드이다.
7. **Depth** : 루트 노드부터 노드까지 경로에 있는 간선의 수이다.
8. **Height** : 루트 노드에서 가장 먼 잎 노드까지 경로에 있는 간선의 수이다.

트리의 종류

1. **이진 트리** : 각 노드가 최대 두 개의 자식 노드를 가지는 트리이다. 이진 탐색 트리, 힙 등 다양한 변형이 있다.
2. **이진 탐색 트리** : 데이터를 정렬된 상태로 유지하기 위해 사용되는 트리이다. 최대 두 개의 자식 노드를 가질 수 있으며 왼쪽 자식 노드는 부모 노드보다 작은 값을 가지고 오른쪽 자식 노드는 부모 노드보다 큰 값을 가진다.
3. **완전 이진 트리** : 각 레벨이 왼쪽부터 차례대로 채워지고 마지막 레벨을 제외한 모든 레벨이 완전히 채워진 이진 트리이다.
4. **포화 이진 트리** : 모든 노드가 0개 또는 2개의 자식 노드를 가지는 이진 트리이다.
5. **N-ary 트리** : 각 노드가 최대 N개의 자식 노드를 가지는 트리이다.

트리의 활용

1. **파일 시스템** : 파일과 디렉토리를 계층적으로 관리한다.
2. **컴파일러** : 프로그램의 구조를 표현하고 분석한다.
3. **데이터베이스** : 데이터를 효율적으로 검색하고 관리한다.
4. **인공지능** : 의사 결정 트리, 신경망 등 다양한 알고리즘에 사용된다.

트리의 연산

1. **탐색** : 특정 노드를 찾는 연산
2. **삽입** : 새로운 노드를 삽입하는 연산

3. 삭제 : 특정 노드를 삭제하는 연산