**트리**

비선형 자료구조로 **계층적 관계를 표현하는 자료구조**이다. 데이터의 추가, 삭제, 검색 등이 주 이슈였던 선형 자료구조와는 다르게 **트리를 배울 때 초점**은 **트리를 통해 표현하는 자료구조가 트리의 특징 중 하나인 계층적 관계를 드러내는데 적합한가?**이다.

**알아야 할 용어에는 노드, 간선, 루트 노드, 단말 노드, 내부 노드 등이 있다.**

노드는 선형 자료구조의 노드와 같다. 데이터 값을 가지며 다음 노드를 가리키는 포인터 변수를 가질 수 있다. 간선은 노드와 노드를 잇는 선을 말한다. 단말 노드는 자신이 가리키는 노드가 없는 노드를 말하며 내부 노드는 간선 노드가 아닌 모든 노드를 의미한다.

트리는 큰 트리와 작은 트리로 구성된다. 트리가 되기 위해서는 다음과 같은 두 조건을 모두 만족시켜야 한다.

* 루트 노드를 중심으로 두 개의 서브 트리로 나누어진다.
* 나뉘어진 두 서브 트리도 모두 이진 트리여야 한다.

위와 같은 정의는 이진 트리의 재귀적 성질을 보여준다. 나누어진 서브 트리는 다시 스스로 이진 트리가 된다. 주의할 점은 포인터 변수가 널인 단말 노드 또한 이진 트리로 인정된다는 점인데 이는 노드가 없는 부분도 공집합 노드로 간주하여 이진 트리로 정의하기 때문이다. 따라서 한 노드에 연결된 노드의 개수가 2개를 초과하지 않고 계층적으로 이어져 있으면 이진 트리의 조건을 만족한다고 할 수 있다.

**여기서 추가적인 용어가 등장하는데 레벨과 높이이다.**

레벨은 트리의 계층 구조상 루트 노드를 0번으로 하여 1씩 증가하는 수로 트리의 위계를 표현한다. 높이는 트리에서 가장 높은 레벨의 수와 같으로 레벨 + 1은 트리의 총 계층 수와 같다.

**레벨과 높이를 사용해 포화 이진 트리와 완전 이진 트리를 구분할 수 있다.**

모든 레벨이 꽉 찬 이진 트리를 포화 이진 트리라고 하고 모든 레벨이 꽉 차 있는 상태는 아니지만 공집합 노드 없이 존재하는 이진 트리는 완전 이진 트리라고 한다.

**배열 기반의 이진 트리**

배열을 기반으로 이진트리를 구현하면 모든 노드에 고유의 노드 번호를 부여한다. 일반적으로 노드를 배열로 구현할 때는 0번 인덱스를 사용하지 않고 루트 노드를 1번 인덱스에 배정한다. 연결 리스트 기반의 이진트리에 비해 덜 직관적이지만 힙 자료구조를 구현할 때 힙이 요구하는 바를 충족시키기에 적합함으로 힙을 배울 때 자세히 배우게 된다.

**이진 트리의 순회(Traversal)**

순회는 전위, 중위, 후위 순회로 나뉜다. 말 그대로 각 노드를 방문하는 것을 말한다. 이진 트리는 한 노드를 삭제할 때 그 노드에 연계된 모든 섭트리를 함께 삭제해서 메모리 누수를 발생시키지 않게 해야 한다. 이때 순회가 필요하다.