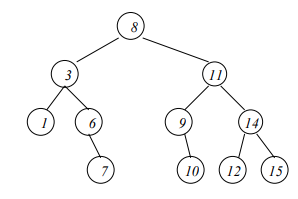
**트리(Trees)**

트리는 연결된 그래프에서 노드의 순서가 정해진 집합으로 구성되며, 각 노드는 최대 하나의 부모 노드와 특정 순서로 0개 이상의 자식 노드를 가집니다.

일반적인 트리의 사양(General specification of trees)

일반적으로 트리는 노드(nodes, vertex 또는 point라고도 함)와 간선(edges, line 또는 arc라고도 함)으로 구성된 트리 같은 구조로 지정할 수 있습니다.



트리는 빈 트리 또는 후속 트리의 목록을 가진 노드로 정의될 수 있습니다. 노드는 일반적으로 숫자나 검색 키와 같은 데이터 항목으로 레이블이 지정되지만 항상 그런 것은 아닙니다. 노드는 일반적으로 숫자나 검색 키와 같은 데이터 항목으로 레이블이 지정되지만 항상 그런 것은 아닙니다. 노드의 레이블을 값(value)이라고 부를 것입니다. 우리의 예에서는 일반적으로 정수로 레이블이 지정된 노드를 사용할 것이지만, 문자열과 같은 다른 것을 선택할 수도 있습니다.

항상 '최상위(top level)' 노드로 알려진 고유한 루트(root)가 있어야 합니다. 컴퓨터 과학에서 트리는 일반적으로 거꾸로 표시되며, 루트가 최상위를 형성한다는 점에 유의해야 합니다.

그림 에서 이것은 레이블이 8인 노드입니다.

그런 다음, 주어진 노드의 경우, 다음 레벨 '아래'의 모든 노드는 분기점을 통해 주어진 노드에 연결되어 있으며, 해당 노드의 자식(child)입니다.

그림 에서 노드 8의 자식은 노드 3과 11입니다.(the children of node 8 are nodes 3 and 11)

주어진 노드에 연결된 노드(최대 하나)는 위의 레벨에 있으며, 부모(parent)입니다.

그림 에서 노드 11은 노드 9(및 노드 14)의 부모입니다.( node 11 is the parent of node 9 (and of node 14 as well)

같은 부모를 가진 노드는 형제(siblings)로 알려져 있습니다. 형제는 정의에 따라 항상 같은 레벨에 있습니다.

한 노드가 다른 노드의 자식의 자식의 ... 자식이라면 첫 번째 노드는 두 번째 노드의 자손(descendent) 이라고 합니다. 반대로, 두 번째 노드는 첫 번째 노드의 조상(ancestor)입니다. 자식이 없는 노드를 잎(leaves)이라고 합니다

그림에서 레이블이 1, 7, 10, 12, 15인 노드를 잎이라고 한다.

경로는 한 노드에서 다른 노드로 연결된 간선의 시퀀스입니다. 트리는 모든 노드에 대해 루트와 연결하는 고유한 경로가 있다는 속성을 가지고 있습니다. 사실, 그것은 트리의 또 다른 가능한 정의입니다. 노드의 깊이 또는 레벨은 이 경로의 길이에 의해 주어집니다. 따라서 루트는 레벨 0이고, 자식은 레벨 1이고, 그렇게 계속됩니다. 트리에서 경로의 최대 길이를 트리의 높이라고도 합니다. 최대 길이의 경로는 항상 루트에서 잎까지 이어집니다. 트리의 크기는 포함된 노드의 수로 주어집니다. 우리는 일반적으로 모든 트리가 유한하다고 가정하지만, 일반적으로 그럴 필요는 없습니다.

그림 6.1의 트리는 높이가 3이고 크기가 11입니다. 단 하나의 노드로 구성된 트리는 높이가 0이고 크기가 1입니다. 빈 트리는 크기가 0이고 편리하게도 높이 -1을 갖도록 정의되어 있습니다.

대부분의 데이터 구조와 마찬가지로, 트리를 구축하고 조작하기 위해 일련의 기본 연산자(생성자, 선택기, 조건)가 필요합니다. 이들의 세부 사항은 트리의 유형과 목적에 따라 다릅니다. 이제 우리는 특히 유용한 몇 가지 트리 유형을 살펴볼 것입니다.