[6.1 What is a Tree? 1](#_Toc151158977)

[6.2 Glossary 1](#_Toc151158978)

[Độ dài sâu của 1 cây nhị phân 2](#_Toc151158979)

[6.3 Cây nhị phân 2](#_Toc151158980)

[6.5 Thuộc tính của cây nhị phân 3](#_Toc151158981)

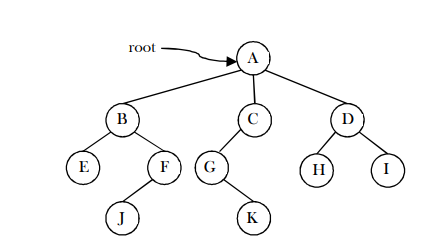
[6.4 Duyệt cây nhị phân 4](#_Toc151158982)

[Phân loại các đường truyền 5](#_Toc151158983)

## 6.1 What is a Tree?

Là một cấu trúc dữ liệu phi tuyến tính, tương tự như danh sách liên kết, nhưng khác biệt ở chỗ mỗi nút trong cây không chỉ trỏ đến một nút tiếp theo, mà có thể trỏ đến nhiều nút. Cây là một ví dụ điển hình của cấu trúc dữ liệu phi tuyến tính. Đồng thời, biểu đồ được mô tả như một cách để hình dung cấu trúc phân cấp của cây dưới dạng đồ họa. Trong Abstract Data Type (ADT) của cây, thứ tự của các phần tử không quan trọng. Nếu cần thông tin về thứ tự, ta có thể sử dụng các cấu trúc dữ liệu tuyến tính như danh sách liên kết, ngăn xếp, hàng đợi, và các cấu trúc tương tự.

## 6.2 Glossary



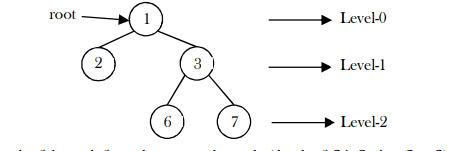
* Gốc của cây là nút không có nút cha. Có thể có nhiều nhất một nút gốc trong cây (nút A trong ví dụ trên).
* Một cạnh đề cập đến liên kết từ cha mẹ đến con cái (tất cả các liên kết trong hình).
* Một nút không có nút con được gọi là nút lá (E, J, K, H và I).
* Con cùng cha, mẹ được gọi là anh em ruột (B, C, D là anh em ruột của A, E, F là anh em ruột của B).
* Nút p là tổ tiên của nút q nếu tồn tại một đường dẫn từ gốc đến q và p xuất hiện trên đường dẫn đó. Nút q được gọi là
* hậu duệ của p. Ví dụ: A, C và G là tổ tiên của K.
* Tập hợp tất cả các nút ở một độ sâu nhất định được gọi là cấp độ của cây (B, C và D có cùng cấp độ). Nút gốc ở mức 0.

Độ dài sâu của 1 cây nhị phân

Cho [Cây nhị phân](https://www.geeksforgeeks.org/binary-tree-data-structure/) gồm **N** nút và số nguyên **K** , nhiệm vụ là tìm độ sâu và chiều cao của nút có giá trị **K** trong [Cây nhị phân](https://www.geeksforgeeks.org/binary-tree-data-structure/) .

Độ sâu của một nút là số cạnh có trong đường dẫn từ nút gốc của cây đến nút đó.

Chiều cao của một nút là số cạnh có trong đường đi dài nhất nối nút đó với nút lá.



Ví dụ như hình trên:

Đầu vào K = 6;

Nhìn hình ta có độ sâu của nút 6 = 2

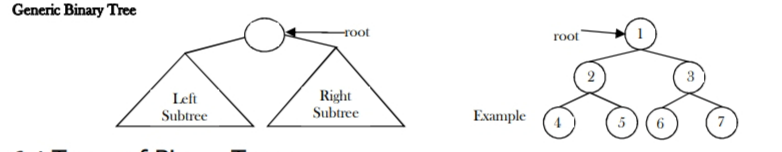
Vì Số cạnh trong đường đi từ nút gốc đến nút 6 là 2. Do đó, độ sâu của nút 25 là 2.

Chiều cao của nút 25 = 0

Số cạnh trong đường dẫn dài nhất nối nút 6 với bất kỳ nút lá nào là 0. Do đó, chiều cao của nút 6 là 0.

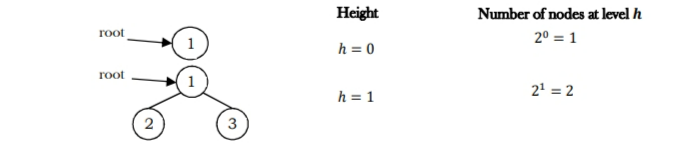
## 6.3 Cây nhị phân

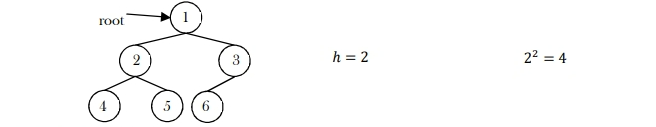
Cây được gọi là cây nhị phân nếu mỗi nút không có con, một hoặc hai con. Cây rỗng cũng là cây nhị phân hợp lệ. Chúng ta có thể hình dung một cây nhị phân bao gồm một gốc và hai cây nhị phân rời nhau, được gọi là cây con trái và cây con phải của gốc.



## 6.5 Thuộc tính của cây nhị phân

Với các tính chất sau, giả sử chiều cao của cây là h. Ngoài ra, giả sử rằng nút gốc có độ cao bằng 0.





Từ sơ đồ ta có thể suy ra các tính chất sau:

* Số nút n trong cây nhị phân đầy đủ là - 1. Vì có h cấp nên ta cần cộng tất cả các nút ở mỗi cấp [ +
* ++…. + = - 1].
* Các số nút n trong cây nhị phân hoàn chỉnh nằm trong khoảng từ (tối thiểu) đến - 1 (tối đa). Để biết thêm thông tin về

này, hãy tham khảo chương Hàng.

* đợi ưu tiên. Số nút lá trong cây nhị phân đầy đủ là .
* Số lượng liên kết None (con trỏ bị lãng phí) trong cây nhị phân hoàn chỉnh gồm n nút là n + 1.

## 6.4 Duyệt cây nhị phân

Để xử lý cây, chúng ta cần một cơ chế duyệt cây và cơ chế đó là chủ đề của phần này. Quá trình truy cập tất cả các nút của cây được gọi là duyệt cây. Mỗi nút chỉ được xử lý một lần nhưng nó có thể được truy cập nhiều lần. Như chúng ta đã thấy trong các cấu trúc dữ liệu tuyến tính (như danh sách liên kết, ngăn xếp, hàng đợi, v.v.), các phần tử được truy cập theo thứ tự tuần tự. Tuy nhiên, trong cấu trúc cây có nhiềucách khác nhau.

Duyệt cây cũng giống như tìm kiếm trên cây, ngoại trừ việc duyệt cây mục tiêu là di chuyển qua cây theo một thứ tự cụ thể. Ngoài ra, tất cả các nút đều được xử lý trong quá trình truyền tải nhưng việc tìm kiếm sẽ dừng lại khi tìm thấy nút yêu cầu.

Khả năng truyền tải

Bắt đầu từ gốc của cây nhị phân, có ba bước chính có thể được thực hiện và thứ tự thực hiện chúng sẽ xác định kiểu truyền tải. Các bước này là: thực hiện một hành động trên nút hiện tại (được gọi là "truy cập" nút và được ký hiệu là "D"), duyệt qua nút con bên trái (ký hiệu là "L") và duyệt qua nút con bên phải ( ký hiệu là “R”). Quá trình này có thể được mô tả dễ dàng thông qua đệ quy. Dựa vào định nghĩa trên có 6 khả năng:

1. LDR: Xử lý cây con bên trái, xử lý dữ liệu nút hiện tại rồi xử lý cây con bên phải
2. LRD: Xử lý cây con trái, xử lý cây con bên phải rồi xử lý dữ liệu nút hiện tại
3. DLR: Xử lý dữ liệu nút hiện tại, xử lý cây con bên trái và sau đó xử lý cây con bên phải
4. DRL: Xử lý dữ liệu nút hiện tại, xử lý cây con bên phải và sau đó xử lý cây con bên trái
5. RDL: Xử lý cây con bên phải, xử lý dữ liệu nút hiện tại rồi xử lý cây con bên trái
6. RLD: Xử lý cây con bên phải, xử lý cây con bên trái rồi xử lý dữ liệu nút hiện tại

### Phân loại các đường truyền

Trình tự xử lý các thực thể (nút) này xác định một phương thức truyền tải cụ thể. Việc phân loại dựa trên thứ tự nút hiện tại

được xử lý. Điều đó có nghĩa là, nếu chúng ta đang phân loại dựa trên nút hiện tại (D) và nếu D ở giữa thì việc L nằm ở bên trái

của D hay R nằm ở bên trái của D không thành vấn đề.

Tương tự, việc L nằm bên phải D hay R nằm bên phải D không quan trọng. Do đó, tổng 6 khả năng giảm xuống còn 3

và đó là:

Truyền tải đặt hàng trước (DLR)

Truyền tải theo thứ tự (LDR)

Truyền tải theo thứ tự sau (LRD)

Có một phương pháp duyệt khác không phụ thuộc vào các thứ tự trên và đó là:

• Truyền tải thứ tự cấp độ: Phương pháp này được lấy cảm hứng từ Traversal đầu tiên theo chiều rộng (BFS của thuật toán đồ thị).

Chúng ta hãy sử dụng sơ đồ dưới đây cho phần thảo luận còn lại.

