

🕒 Cập nhật tháng 8 năm 2024

[Bài đọc] Tìm hiểu về Tree và BST

1. Cây (Tree) là gì?

- Định nghĩa:
 - Cây (Tree) là **cấu trúc dữ liệu phân cấp** gồm tập hợp các **nút (nodes)**, trong đó có một nút **gốc (root)** và các nút con (children)
 - **Không có chu trình (cycle)** – tức là không có vòng lặp trong cấu trúc
- Thành phần:
 - **Node**: phần tử cơ bản, chứa dữ liệu
 - **Root**: nút gốc, không có cha
 - **Leaf**: nút lá, không có con
 - **Parent - Child**: quan hệ cha-con
 - **Subtree**: cây con xuất phát từ một nút nào đó
- Tính chất:
 - Mỗi node chỉ có duy nhất **một cha**
 - Một node có thể có **nhiều con**
 - Tổng số cạnh trong cây = số node - 1

2. Cây Nhị Phân (Binary Tree)

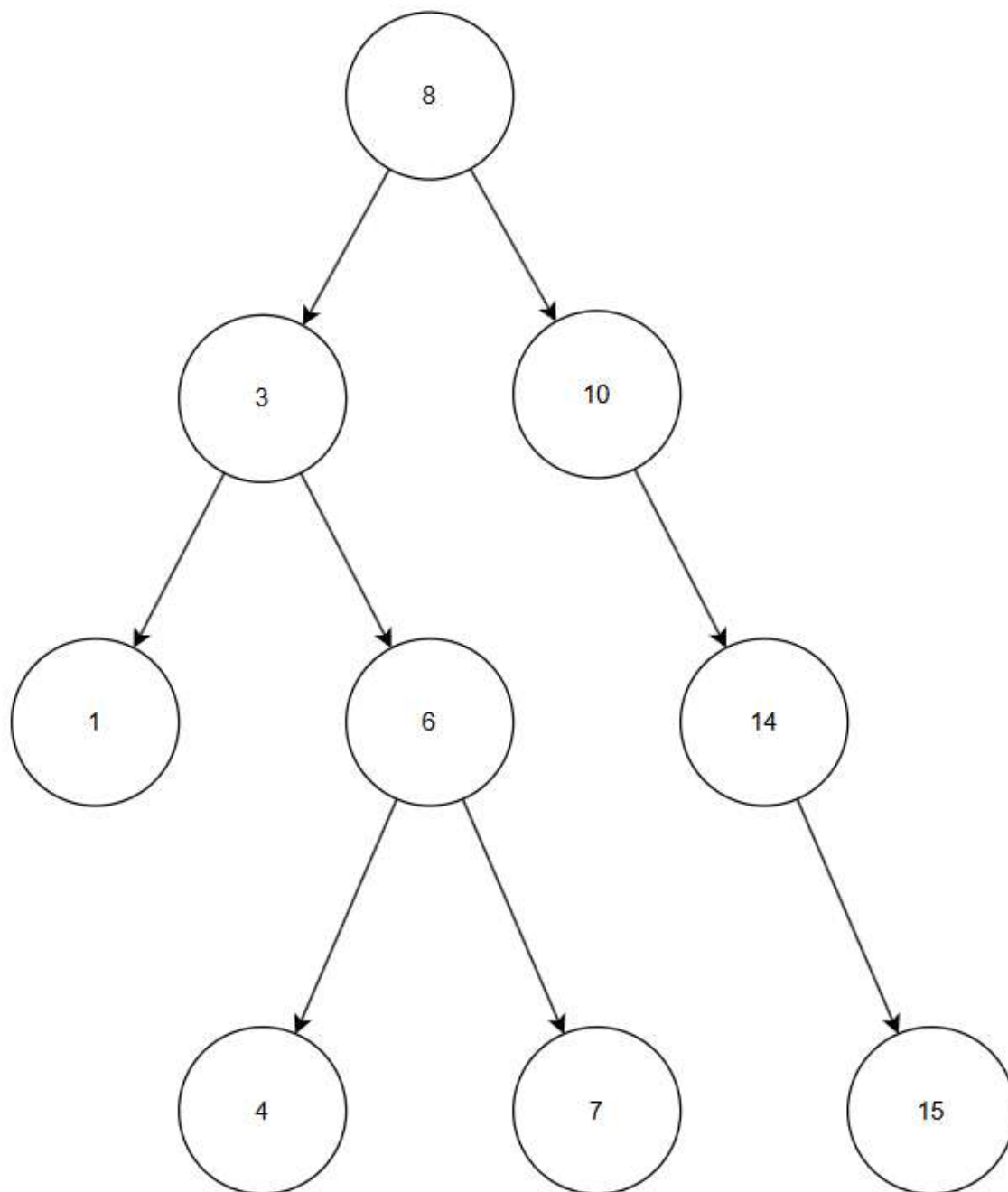
- Định nghĩa:
 - Là cây mà **mỗi node có tối đa 2 node con**, gọi là:
 - **Left child** (con trái)
 - **Right child** (con phải)
- Các loại cây nhị phân phổ biến:
 - **Full Binary Tree**: mỗi node có 0 hoặc 2 con
 - **Perfect Binary Tree**: mọi node lá đều ở cùng mức và mọi node đều có 2 con
 - **Complete Binary Tree**: mọi tầng đều đầy trừ tầng cuối, và các node ở tầng cuối nằm bên trái

3. Cây Tìm Kiếm Nhị Phân (Binary Search Tree - BST)

- Định nghĩa:
 - Là một cây nhị phân **thỏa mãn thuộc tính tìm kiếm**:

- Với mỗi node **N**, tất cả các node ở **trái** < **N**, và tất cả các node ở **phải** > **N**

- Ví dụ:



- Các thao tác cơ bản:

Thao tác	Ý nghĩa
Insert	Thêm node mới vào đúng vị trí
Search	Tìm một giá trị trong cây
Delete	Xóa node và cân bằng lại cây
Traversal	Duyệt cây: In-order, Pre-order, Post-order

- Duyệt cây (Traversal):
 - **In-order (Trái - Gốc - Phải)**: Trả về giá trị theo thứ tự tăng dần (với BST)
 - **Pre-order (Gốc - Trái - Phải)**: Dùng trong sao chép cây
 - **Post-order (Trái - Phải - Gốc)**: Dùng trong xóa cây

4. Ứng dụng của Tree & BST

- Tổ chức dữ liệu phân cấp: thư mục máy tính, DOM trong HTML...
- BST dùng để:
 - Cài đặt từ điển, danh bạ
 - Tìm kiếm nhanh $O(\log n)$
 - Biểu diễn biểu thức toán học
 - Tự động cân bằng dữ liệu (khi kết hợp với AVL Tree, Red-Black Tree)

5. Gợi ý học thêm:

- **AVL Tree, Red-Black Tree** - các biến thể của BST cân bằng
- **Heap, Trie, Segment Tree** - những cấu trúc nâng cao liên quan đến cây

Tài nguyên đọc thêm: <https://www.geeksforgeeks.org/binary-search-tree-data-structure>