

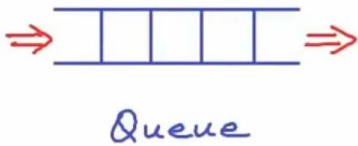
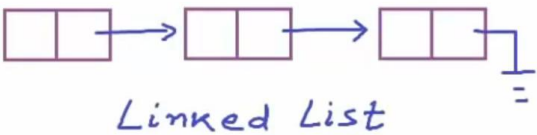
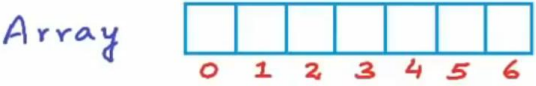
## 4 Tree（树）

- 4.1 树：基本介绍
- 4.2 二叉树
- 4.3 二叉搜索树
- 4.4 二叉搜索树（C/C++实现）
- 4.5 二叉搜索树（C/C++实现）——内存中的栈与堆详解
- 4.6 二叉搜索树——查找最小值和最大值
- 4.7 二叉树的高度
- 4.8 二叉树的遍历——广度优先 vs 深度优先
- 4.9 二叉树的层次遍历
- 4.10 二叉树的前序、中序、后序遍历
- 4.11 判断是否为二叉搜索树
- 4.12 二叉搜索树中删除一个节点
- 4.13 二叉搜索树的中序后继节点

4.1 树：基本介绍

Introduction to Trees

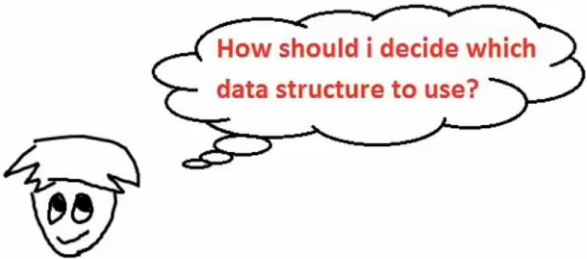
Linear data structures:



比如数组, 链表, 栈和队列, 这些都是线性数据结构

mycodeschool.com

Introduction to Trees



How should i decide which data structure to use?



- What needs to be stored?
- Cost of operations
- Memory usage
- Ease of implementation



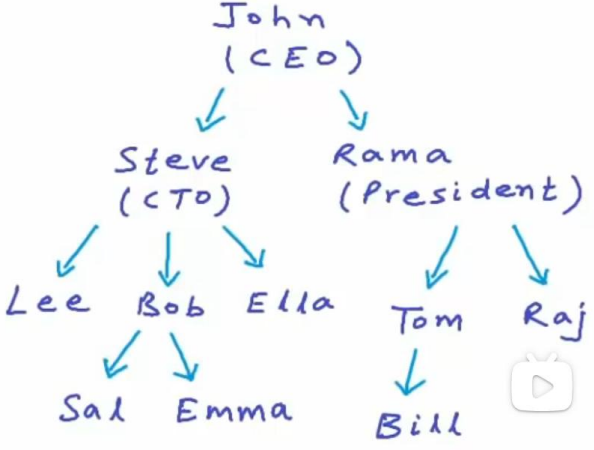
最后, 我们可能想要选择一个更加容易实现的数据结构

mycodeschool.com

Introduction to Trees



Ok! So, what do we use a tree for?



假设这是某公司的组织层级

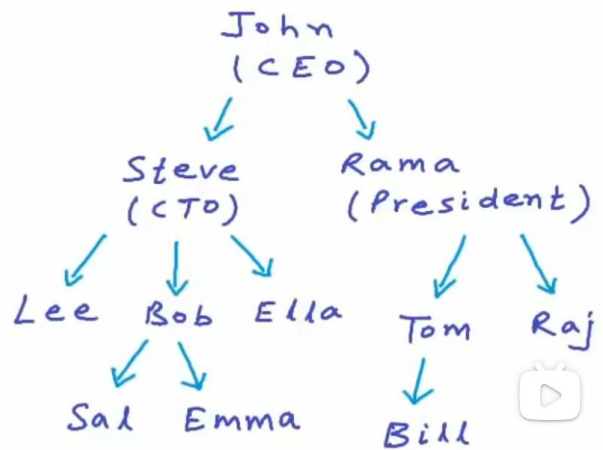
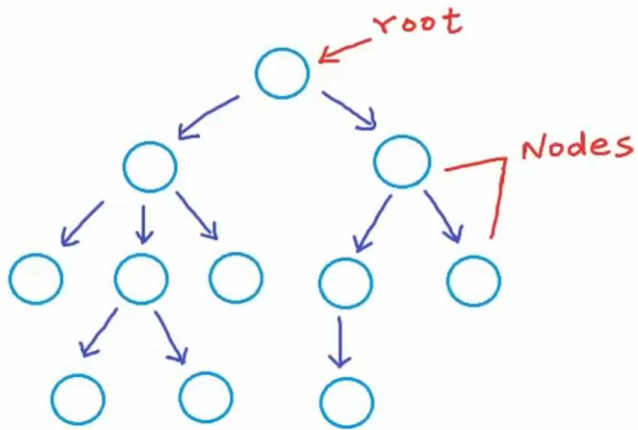
mycodeschool.com

树, 是经常用来表示层次数据的一种数据结构, 它是一种用来存储和组织天然地具有层级结构的数据的有效方式。但这不是树在计算机科学中的唯一应用。

总结:

## 2.1 基本介绍

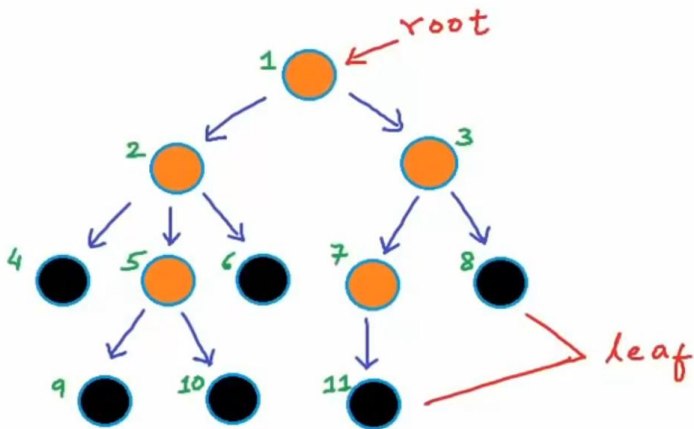
### Introduction to Trees



好的, 每个节点包含一些数据, 可能还包含一个(指向其他节点)的链接或者引用 [mycodeschool.com](https://www.mycodeschool.com)

树是一种非线性的数据结构, 它是一种层级结构。一个节点可能有 (指向其它节点的) 链接或者引用。子节点。

### Introduction to Trees

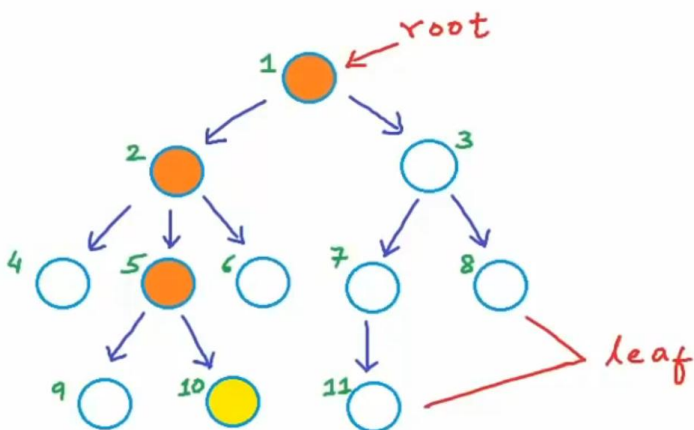


root  
children  
Parent  
Sibling → have same parent  
leaf → has no child



[mycodeschool.com](https://www.mycodeschool.com)

### Introduction to Trees



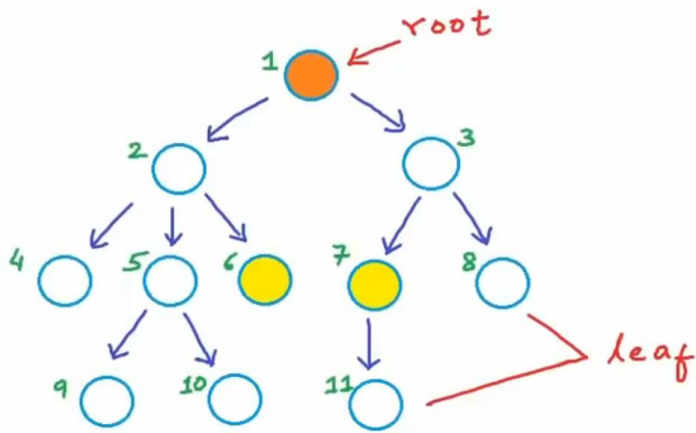
root  
children  
Parent  
Sibling → have same parent  
leaf → has no child  
If we can go from A to B  
A is ancestor of B  
B is descendent of A

[mycodeschool.com](https://www.mycodeschool.com)

当我们遍历树的时候, 只能从一个方向进行。

总结:

## Introduction to Trees



root  
children  
Parent  
Sibling → have same parent  
leaf → has no child

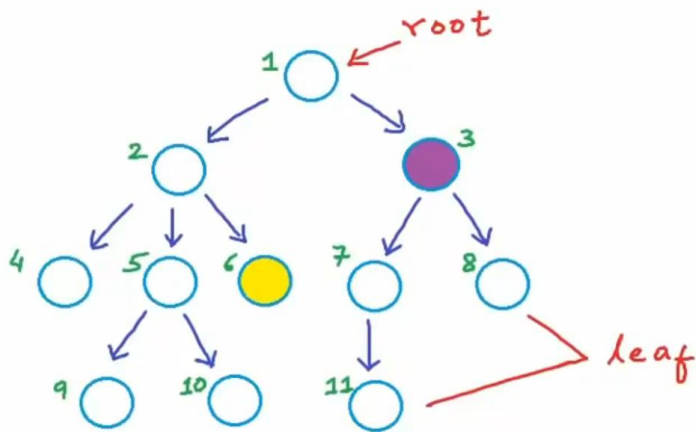
If we can go from A to B  
A is ancestor of B  
B is descendent of A

cousins

父母不同但是祖父母相同的节点被称为堂兄弟

mycodeschool.com

## Introduction to Trees



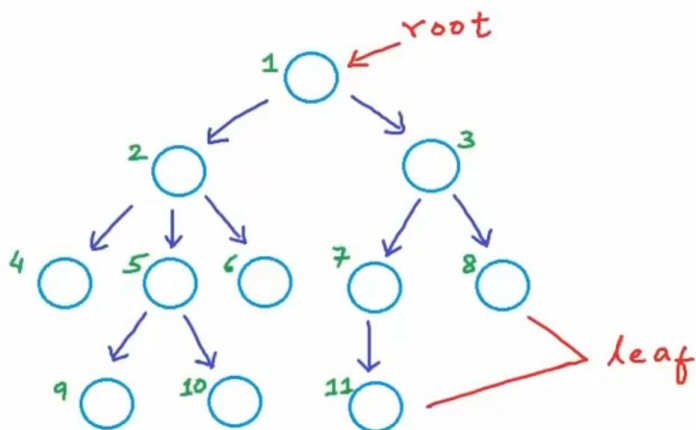
root  
children  
Parent  
Sibling → have same parent  
leaf → has no child

If we can go from A to B  
A is ancestor of B  
B is descendent of A

我们也可以说节点 3 是 6 的叔叔

mycodeschool.com

## Introduction to Trees



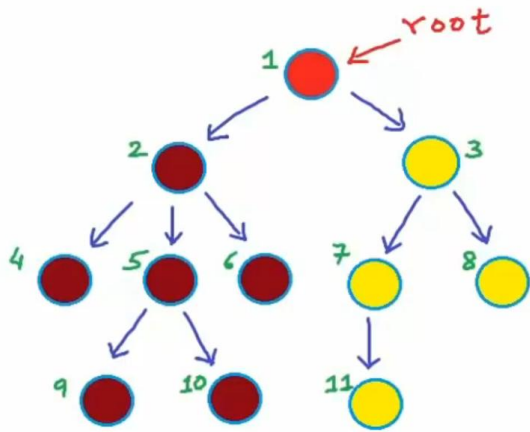
root  
children  
Parent  
Sibling  
leaf  
ancestor  
descendent  
cousin

好的, 现在我们来讨论一下树的一些属性

mycodeschool.com



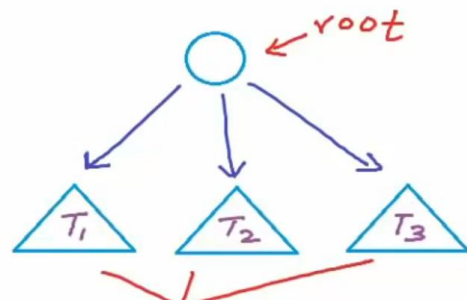
## Introduction to Trees



Tree

树可以被称为一个递归的数据结构

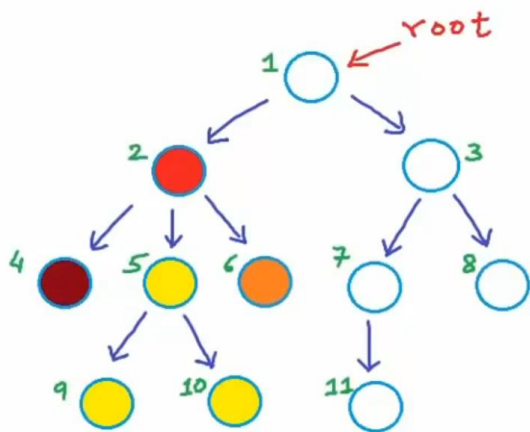
↳ recursive data structure



Sub-trees

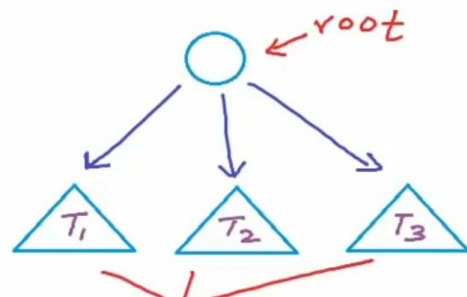
我用红色来表示根节点, 左边的子树用棕色表示, 右边的子树用黄色表示

## Introduction to Trees



Tree

↳ recursive data structure



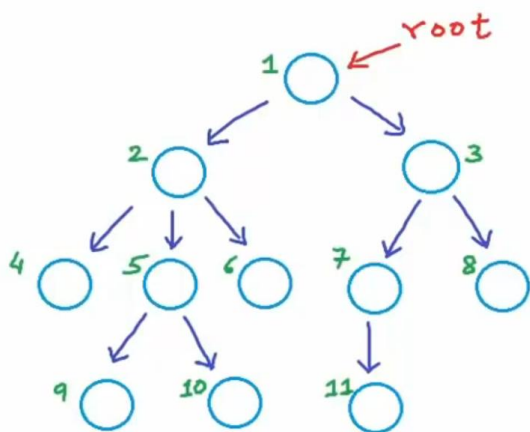
Sub-trees

这棵以节点 2 为根的特定树有 3 棵子树

mycodeschool.com

递归基本上就是用调用自身的方式来解决。在具体实现和使用树的过程中, 我们将会到处使用树的这种递归属性。

## Introduction to Trees



Tree

↳ N nodes

N-1 edges

这个图中的每个箭头可以被称为一个链接或者边

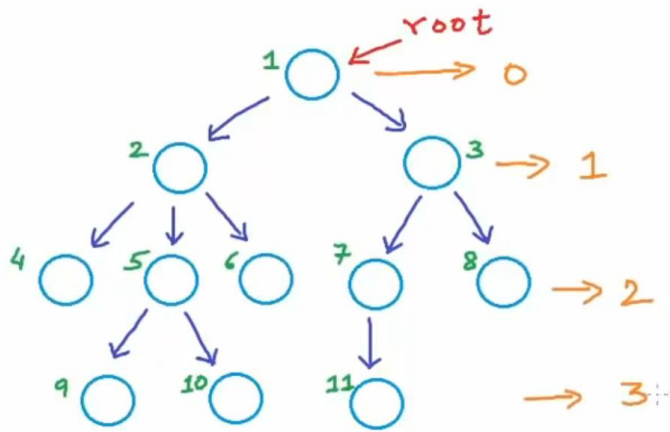
mycodeschool.com

树的下一个属性是: 如果一棵树有N个节点, 那么恰好会有 N-1 个链接或者边。除了根节点之外的所有节点刚好有一个传入的边。

总结:

## Introduction to Trees

### Depth and Height



Depth of  $x$  =

length of path from  
root to  $x$

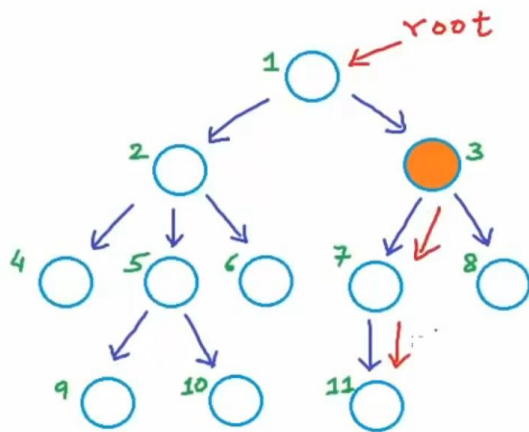
OR

No. of edges in path  
from root to  $x$



## Introduction to Trees

### Depth and Height



Depth of  $x$  =

No. of edges in path  
from root to  $x$

Height of  $x$  =

No. of edges in longest  
path from  $x$  to a leaf

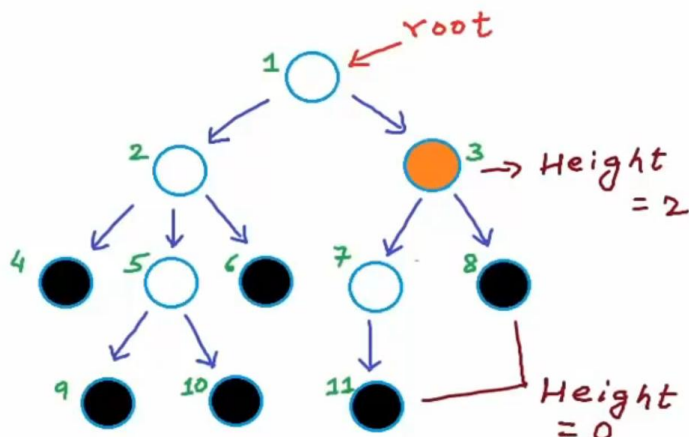


因此, 节点3的高度是2

mycodeschool.com

## Introduction to Trees

### Depth and Height



Depth of  $x$  =

No. of edges in path  
from root to  $x$

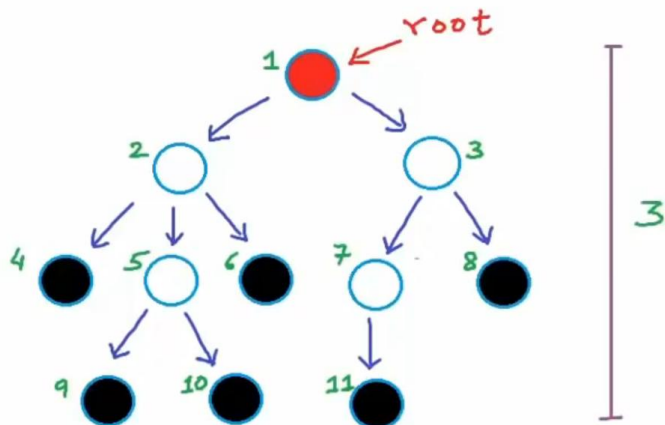
Height of  $x$  =

No. of edges in longest  
path from  $x$  to a leaf



mycodeschool.com

## Introduction to Trees



### Depth and Height

Depth of  $x$  =

No. of edges in path  
from root to  $x$

Height of  $x$  =

No. of edges in longest  
path from  $x$  to a leaf

Height of tree =

Height of root node

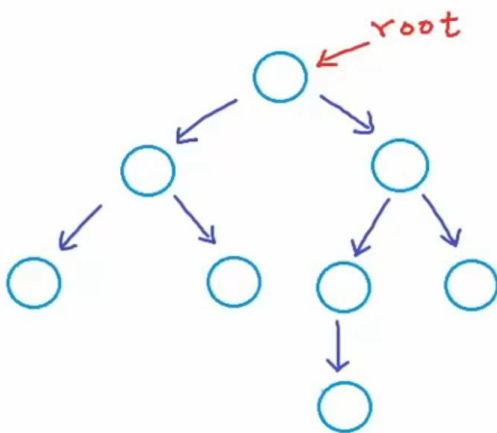
mycodeschool.com

高度和深度是不同的属性，一个节点的高度和深度可能相等，也可能不等，我们经常会把这两者搞混淆。

根据属性，树可以被分为不同的类型。不同的场景中使用不同类型的树。

最简单的也是最常见的类型是具备这种属性的：即任何节点最多包含两个子节点，叫**二叉树 (binary tree)**。

## Introduction to Trees



### Binary Tree

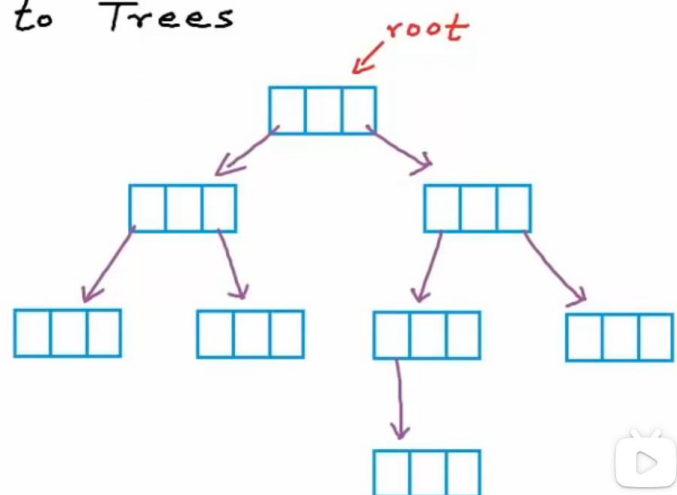
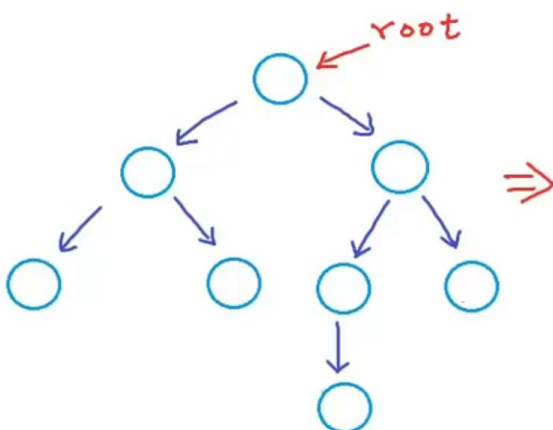
↓  
a tree in which each  
node can have at most  
2 children



树的最常见实现方式就是动态创建的节点，用指针或者引用把它们链接起来

mycodeschool.com

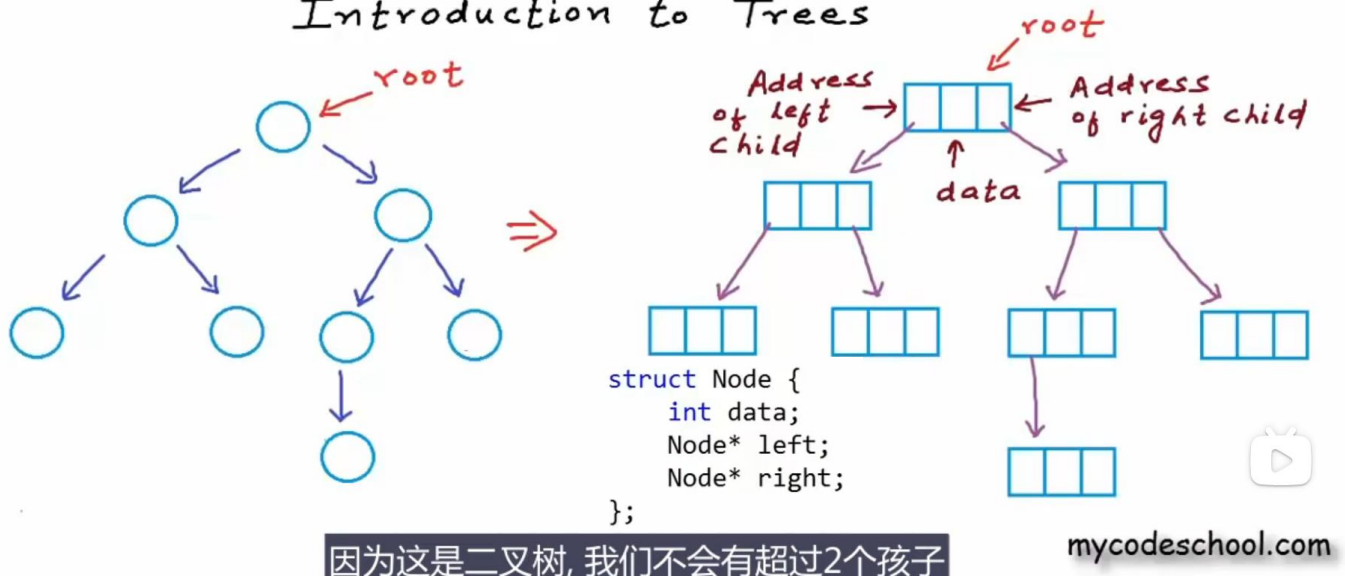
## Introduction to Trees



在我右边画的这个结构中，节点有3个域

mycodeschool.com

## Introduction to Trees



对于可以有任意多个孩子的通用树, 我们使用其它的结构。

开头我们也讲了, 存储天然具备层级结构的数据不是树的唯一应用。让我们来快速地看一下计算机科学中对于树的应用。

- 比如磁盘驱动器上的文件系统, 文件和文件夹就是天然的层次化的数据。它是按照树的形式存储。
- 组织数据, 以便让该集合能够做到快速查找、插入和删除。比如我们接下来要讲的二叉搜索树, 它查找的时间复杂度为  $O(\log N)$ 。
- Trie 树, 被用来存储字典。它非常快速和高效, 可以被用来做动态的拼写检查。
- 树这种数据结构还可以被用作网络路由算法。
- .....

## Introduction to Trees

### Applications:

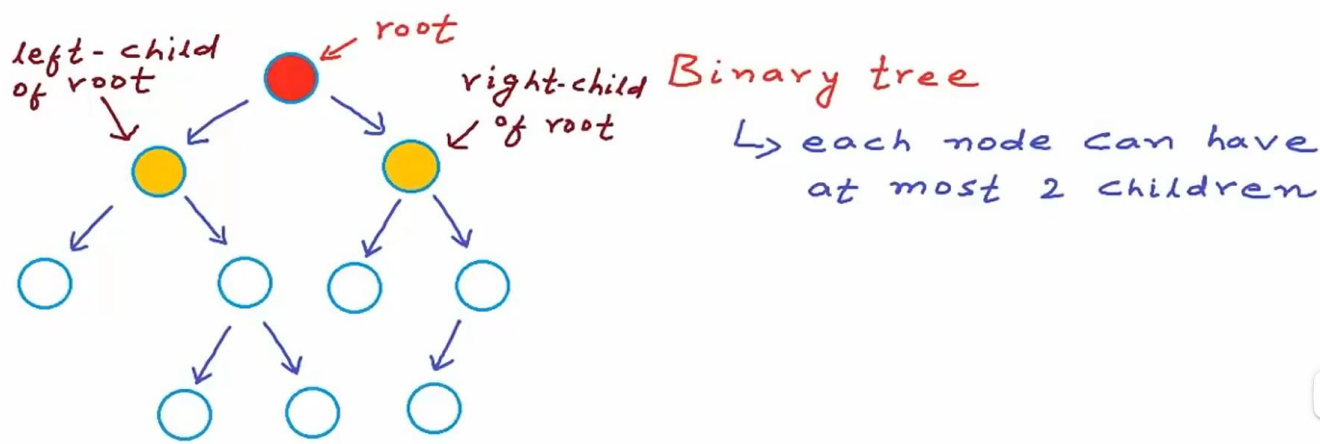
- 1) Storing naturally hierarchical data → eg:- file system
- 2) Organize data for quick search, insertion, deletion → eg:- Binary Search trees
- 3) Trie → dictionary
- 4) Network Routing algorithm

mycodeschool.com



4.2 二叉树

Binary Tree



ering



mycodeschool.com The read

and modify data.  
Based on the author entity example, we created the table using the entity name author and the entity attributes as the columns of the table. Rows were added to the author table to populate the table.

## 2.1 基本介绍

Hello and welcome to the UPDATE Statement and the DELETE Statement. In this video, we will learn about altering and deleting data in a relational database table. At the end of this lesson, you will be able to

- identify the syntax of the UPDATE statement and DELETE statement.
- explain the importance of the WHERE clause in these statements.

After a table is created and populated with data, the data in a table can be altered with the UPDATE statement. The UPDATE statement is one of the data manipulation language or DML statements. DML statements are used to read and modify data.

Based on the author entity example, we created the table using the entity name author and the entity attributes as the columns of the table. Rows were added to the author table to populate the table.

## 2.1 基本介绍

Hello and welcome to the UPDATE Statement and the DELETE Statement. In this video, we will learn about altering and deleting data in a relational database table. At the end of this lesson, you will be able to

- identify the syntax of the UPDATE statement and DELETE statement.
- explain the importance of the WHERE clause in these statements.

After a table is created and populated with data, the data in a table can be altered with the UPDATE statement. The UPDATE statement is one of the data manipulation language or DML statements. DML statements are used to read and modify data.

Based on the author entity example, we created the table using the entity name author and the entity attributes as the columns of the table. Rows were added to the author table to populate the table.

## 2.1 基本介绍

Hello and welcome to the UPDATE Statement and the DELETE Statement. In this video, we will learn about altering and deleting data in a relational database table. At the end of this lesson, you will be able to

- identify the syntax of the UPDATE statement and DELETE statement.
- explain the importance of the WHERE clause in these statements.

After a table is created and populated with data, the data in a table can be altered with the UPDATE statement. The UPDATE statement is one of the data manipulation language or DML statements. DML statements are used to read and modify data.

Based on the author entity example, we created the table using the entity name author and the entity attributes as the columns of the table. Rows were added to the author table to populate the table.



## 2.1 基本介绍

Hello and welcome to the UPDATE Statement and the DELETE Statement. In this video, we will learn about altering and deleting data in a relational database table. At the end of this lesson, you will be able to

- identify the syntax of the UPDATE statement and DELETE statement.
- explain the importance of the WHERE clause in these statements.

After a table is created and populated with data, the data in a table can be altered with the UPDATE statement. The UPDATE statement is one of the data manipulation language or DML statements. DML statements are used to read and modify data.

Based on the author entity example, we created the table using the entity name author and the entity attributes as the columns of the table. Rows were added to the author table to populate the table.

## 2.1 基本介绍

Hello and welcome to the UPDATE Statement and the DELETE Statement. In this video, we will learn about altering and deleting data in a relational database table. At the end of this lesson, you will be able to

- identify the syntax of the UPDATE statement and DELETE statement.
- explain the importance of the WHERE clause in these statements.

After a table is created and populated with data, the data in a table can be altered with the UPDATE statement. The UPDATE statement is one of the data manipulation language or DML statements. DML statements are used to read and modify data.

Based on the author entity example, we created the table using the entity name author and the entity attributes as the columns of the table. Rows were added to the author table to populate the table.