**Final Project: Analysis Document**

[Inserting into a BST versus a Hash Table](#_qe086yykgip6)

[Data Analysis Graphs](#_h8xxxq7s7wet)

[Analysis and Summation](#_xfnecar7brgw)

[Concrete Real-World Example](#_51ylfiz2zgzl)

[Data Structure Selection](#_xa9uqglya3ir)

[Removing from a Vector at the Front versus the Back](#_q6rs0djmqa76)

[Data Analysis Graphs](#_qrp2zab0yaep)

[Analysis and Summation](#_h32t5n1guute)

[Concrete Real-World Example](#_lsicjnxr0afs)

[Data Structure Selection](#_3nngyjvvey7l)

[Searching a Vector, DLL, SLL, BST, and Hash Table](#_vxtx8aatasrn)

[Data Analysis Graphs](#_gls65vv8vaij)

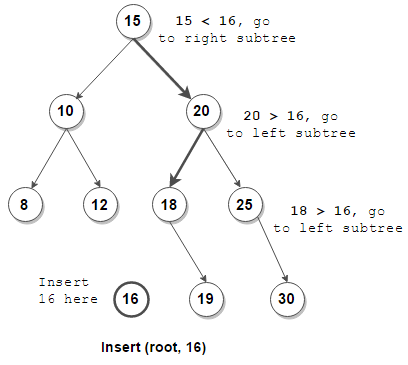
[Analysis and Summation](#_yibuntunok1)

[Concrete Real-World Example](#_it1hh5vzbv6f)

[Data Structure Selection](#_9vmof7irdv4)

# **Inserting into a BST versus a Hash Table**

## **Data Analysis Graphs**



## **Analysis and Summation**

In [computer science](https://en.wikipedia.org/wiki/Computer_science), a **binary tree** is a [tree data structure](https://en.wikipedia.org/wiki/Tree_(data_structure)) in which each node has at most two [children](https://en.wikipedia.org/wiki/Child_node), which are referred to as the *left child* and the *right child*. A [recursive definition](https://en.wikipedia.org/wiki/Recursive_definition) using just [set theory](https://en.wikipedia.org/wiki/Set_theory) notions is that a (non-empty) binary tree is a [tuple](https://en.wikipedia.org/wiki/Tuple) (*L*, *S*, *R*), where *L* and *R* are binary trees or the [empty set](https://en.wikipedia.org/wiki/Empty_set) and *S* is a [singleton set](https://en.wikipedia.org/wiki/Singleton_set) containing the root.[[1]](https://en.wikipedia.org/wiki/Binary_tree#cite_note-GarnierTaylor2009-1) Some authors allow the binary tree to be the empty set as well.[[2]](https://en.wikipedia.org/wiki/Binary_tree#cite_note-Skiena2009-2)

From a [graph theory](https://en.wikipedia.org/wiki/Graph_theory) perspective, binary (and K-ary) trees as defined here are [arborescences](https://en.wikipedia.org/wiki/Arborescence_(graph_theory)).A binary tree may thus be also called a **bifurcating arborescence**[[3]](https://en.wikipedia.org/wiki/Binary_tree#cite_note-Knuth1997-3)—a term which appears in some very old programming books,[[4]](https://en.wikipedia.org/wiki/Binary_tree#cite_note-Flores1971-4) before the modern computer science terminology prevailed. It is also possible to interpret a binary tree as an [undirected](https://en.wikipedia.org/wiki/Undirected_graph), rather than a [directed graph](https://en.wikipedia.org/wiki/Directed_graph), in which case a binary tree is an [ordered](https://en.wikipedia.org/wiki/Ordered_tree), [rooted tree](https://en.wikipedia.org/wiki/Rooted_tree).[[5]](https://en.wikipedia.org/wiki/Binary_tree#cite_note-5) Some authors use **rooted binary tree** instead of *binary tree* to emphasize the fact that the tree is rooted, but as defined above, a binary tree is always rooted.[[6]](https://en.wikipedia.org/wiki/Binary_tree#cite_note-Mazur2010-6) A binary tree is a special case of an ordered [K-ary tree](https://en.wikipedia.org/wiki/K-ary_tree), where *K* is 2.

In mathematics, what is termed *binary tree* can vary significantly from author to author. Some use the definition commonly used in computer science,[[7]](https://en.wikipedia.org/wiki/Binary_tree#cite_note-oem-7) but others define it as every non-leaf having exactly two children and don't necessarily order (as left/right) the children either.

## **Concrete Real-World Example**

* First, as a means of accessing nodes based on some value or label associated with each node.[[9]](https://en.wikipedia.org/wiki/Binary_tree#cite_note-Makinson2009b-9) Binary trees labelled this way are used to implement [binary search trees](https://en.wikipedia.org/wiki/Binary_search_tree) and [binary heaps](https://en.wikipedia.org/wiki/Binary_heap), and are used for efficient [searching](https://en.wikipedia.org/wiki/Search_algorithm) and [sorting](https://en.wikipedia.org/wiki/Sorting_algorithm). The designation of non-root nodes as left or right child even when there is only one child present matters in some of these applications, in particular, it is significant in binary search trees.[[10]](https://en.wikipedia.org/wiki/Binary_tree#cite_note-Gross2007-10) However, the arrangement of particular nodes into the tree is not part of the conceptual information. For example, in a normal binary search tree the placement of nodes depends almost entirely on the order in which they were added, and can be re-arranged (for example by [balancing](https://en.wikipedia.org/wiki/Self-balancing_binary_search_tree)) without changing the meaning.
* Second, as a representation of data with a relevant bifurcating structure. In such cases, the particular arrangement of nodes under and/or to the left or right of other nodes is part of the information (that is, changing it would change the meaning). Common examples occur with [Huffman coding](https://en.wikipedia.org/wiki/Huffman_coding) and [cladograms](https://en.wikipedia.org/wiki/Cladograms). The everyday division of documents into chapters, sections, paragraphs, and so on is an analogous example with n-ary rather than binary trees.

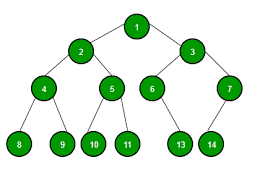
## **Data Structure Selection**

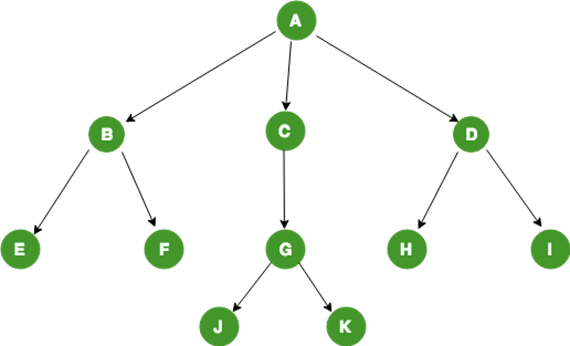
A binary tree is a **tree-type non-linear data structure** with a maximum of two children for each parent. Every node in a binary tree has a left and right reference along with the data element. The node at the top of the hierarchy of a tree is called the root node. The nodes that hold other sub-nodes are the parent nodes

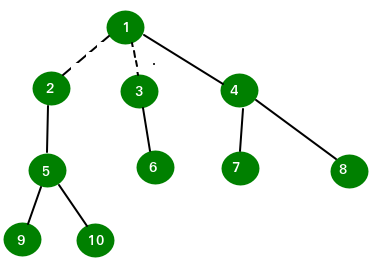
# 

# **Removing from a Vector at the Front versus the Back**

## **Data Analysis Graphs**

****

****

****

## Analysis and Summation

To actually define a binary tree in general, we must allow for the possibility that only one of the children may be empty. An artifact, which in some textbooks is called an *extended binary tree* is needed for that purpose. An extended binary tree is thus recursively defined as:[[11]](https://en.wikipedia.org/wiki/Binary_tree#cite_note-Rosen2011-11)

* the [empty set](https://en.wikipedia.org/wiki/Empty_set) is an extended binary tree
* if T1 and T2 are extended binary trees, then denote by T1 • T2 the extended binary tree obtained by adding a root *r* connected to the left to T1 and to the right to T2[[*clarification needed*](https://en.wikipedia.org/wiki/Wikipedia:Please_clarify)*where did the 'r' go in the 'T1• T2' symbol*] by adding edges when these sub-trees are non-empty.

Another way of imagining this construction (and understanding the terminology) is to consider instead of the empty set a different type of node—for instance square nodes if the regular ones are circles.

## **Concrete Real-World Example**

A binary tree is a [rooted tree](https://en.wikipedia.org/wiki/Rooted_tree) that is also an [ordered tree](https://en.wikipedia.org/wiki/Ordered_tree) (a.k.a. plane tree) in which every node has at most two children. A rooted tree naturally imparts a notion of levels (distance from the root), thus for every node a notion of children may be defined as the nodes connected to it a level below. Ordering of these children (e.g., by drawing them on a plane) makes it possible to distinguish a left child from a right child.[[13]](https://en.wikipedia.org/wiki/Binary_tree#cite_note-HsuLin2008-13) But this still doesn't distinguish between a node with left but not a right child from a one with right but no left child.

The necessary distinction can be made by first partitioning the edges, i.e., defining the binary tree as triplet (V, E1, E2), where (V, E1 ∪ E2) is a rooted tree (equivalently arborescence) and E1 ∩ E2 is empty, and also requiring that for all *j* ∈ { 1, 2 } every node has at most one E*j* child.[[14]](https://en.wikipedia.org/wiki/Binary_tree#cite_note-FlumGrohe2006-14) A more informal way of making the distinction is to say, quoting the [Encyclopedia of Mathematics](https://en.wikipedia.org/wiki/Encyclopedia_of_Mathematics), that "every node has a left child, a right child, neither, or both" and to specify that these "are all different" binary trees

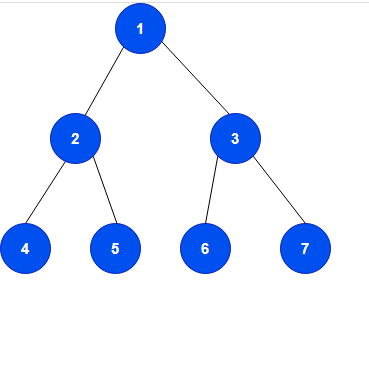
## **Data Structure Selection**

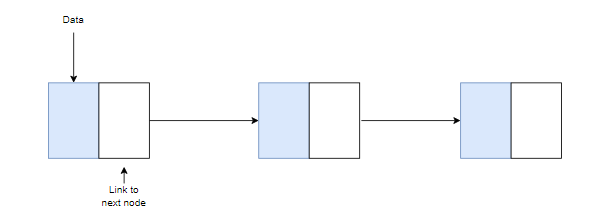
The **Tournament tree** is a complete binary tree with n external nodes and n – 1 internal nodes. The external nodes represent the players, and the internal nodes are representing the winner of the match between the two players. This tree is also known as Selection tree. There are some properties of Tournament trees

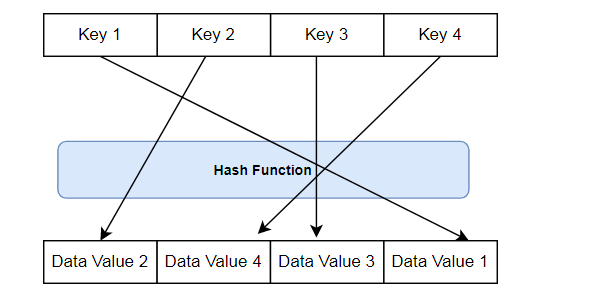
# 

# Searching a Vector, DLL, SLL, BST, and Hash Table

# **Data Analysis Graphs**







## **Analysis and Summation**

**A**[**binary tree**](https://www.baeldung.com/cs/binary-tree-intro#definition)**is a hierarchical tree-based data structure in which each node has at most two children**. The root node is the topmost node of a binary tree, while the left and right nodes are called left and right children, respectively. Furthermore, the links between nodes are known as branches, while a node without children is called a leaf node.

In a binary tree, a single node will contain a data value and a link to each of the child nodes. The following operations can be performed on binary trees: insertion, searching, and deletion. These operations can be executed in  time.

## Concrete Real-World Example

A few benefits to using linked lists are:

* Nodes can be easily deleted and inserted.
* A linked list can be easily implemented in most programming languages.

On the contrary, some limitations of linked lists are:

* Nodes must always be accessed sequentially, which is time consuming.
* The pointers used in linked lists require additional memory.

## Data Structure Selection

Consequently, some major benefits of using hash tables are:

* Insert, delete and search operations are very fast and can be done in  time.
* Hash tables can store large amounts of data.

Conversely, some limitations of using hash tables are:

* Hash functions tend to produce duplicate keys, which cause problems with storing data values, known as collisions.
* Good hash functions that produce distinct keys are expensive and difficult to implemen