

Applications of tree data structure

Difficulty Level: Rookie

Why Tree?

Unlike Array and Linked List, which are linear data structures, tree is hierarchical (or non-linear) data structure.

1) One reason to use trees might be because you want to store information that naturally forms a hierarchy. For example, the file system on a computer:

file system

```

      /   <-- root
     /    \
    ...    home
         /   \
        /      \
       /         \
      /           \
     /             \
    ...            cs101 cs112 cs113
  
```

2) If we organize keys in form of a tree (with some ordering e.g., BST), we can search for a given key in moderate time (quicker than Linked List and slower than arrays). **Self-balancing search trees** like **AVL** and **Red-Black trees** guarantee an upper bound of $O(\log n)$ for search.

3) We can insert/delete keys in moderate time (quicker than Arrays and slower than Unordered Linked Lists). **Self-balancing search trees** like **AVL** and **Red-Black trees** guarantee an upper bound of $O(\log n)$ for insertion/deletion.

4) Like Linked Lists and unlike Arrays, Pointer implementation of trees don't have an upper limit on number of nodes as nodes are linked using pointers.

As per Wikipedia, following are the common uses of tree.

1. Manipulate hierarchical data.
2. Make information easy to search (see tree traversal).
3. Manipulate sorted lists of data.
4. As a workflow for compositing digital images for visual effects.

5. Router algorithms

References:

<http://www.cs.bu.edu/teaching/c/tree/binary/>

http://en.wikipedia.org/wiki/Tree_%28data_structure%29#Common_uses

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.



15 Comments Category: Trees

Related Posts:

- [Check if removing an edge can divide a Binary Tree in two halves](#)
- [Check sum of Covered and Uncovered nodes of Binary Tree](#)
- [Lowest Common Ancestor in a Binary Tree | Set 2 \(Using Parent Pointer\)](#)
- [Construct a Binary Search Tree from given postorder](#)
- [BFS vs DFS for Binary Tree](#)
- [Maximum difference between node and its ancestor in Binary Tree](#)
- [Inorder Non-threaded Binary Tree Traversal without Recursion or Stack](#)
- [Check if leaf traversal of two Binary Trees is same?](#)

Like Share 20 people like this.

Writing code in comment? Please use code.geeksforgeeks.org, generate link and share the link here.
candycrush
how can these trees be implemented in a file system..

Nishant Mishra

Isn't the Expression Parser of our C compiler uses Binary tree for Parsing...??

```
/* Paste your code here (You may delete these lines if not writing code) */
```

sandyg

Binary Search Tree – Used in many search applications where data is constantly entering/leaving, such as the map and set objects in many languages' libraries.

Binary Space Partition – Used in almost every 3D video game to determine what objects need to be rendered.

Binary Tries – Used in almost every high-bandwidth router for storing router-tables.

Hash Trees – used in p2p programs and specialized image-signatures in which a hash needs to be verified, but the whole file is not available.

Heaps – Used in heap-sort; fast implementations of Dijkstra's algorithm; and in implementing efficient priority-queues, which are used in scheduling processes in many operating systems, Quality-of-Service in routers, and A* (path-finding algorithm used in AI applications, including video games).

Huffman Coding Tree (Chip Uni) – used in compression algorithms, such as those used by the .jpeg and .mp3 file-formats.

GGM Trees – Used in cryptographic applications to generate a tree of pseudo-random numbers.

Syntax Tree – Constructed by compilers and (implicitly) calculators to parse expressions.

Treap – Randomized data structure used in wireless networking and memory allocation.

T-tree – Though most databases use some form of B-tree to store data on the drive, databases which keep all (most) their data in memory often use T-trees to do so

Chan

very useful information.

@geeksforgeeks, Some rights reserved

[Contact Us!](#)

[About Us!](#)

[Advertise with us!](#)