A Survey on Applications of Nonlinear Data Structures

C. Padmavathy[a], S.S.Suganthа Mallika[b]
[a]Assistant Professor(Sl.Gr)/CSE, [b]Assistant Professor(Sr.Gr)/IT
[a,b] Sri Ramakrishna Engineering College,Coimbatore-641022

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: In this work, Non-linear data structures and their applications are discussed. Each section in this work provides insights of each non-linear data structure. The types of each non-linear data structure are described along with their applications. This paper provides knowledge about the advantages of employing non-linear data structures in real-time applications.

Keywords: Non-linear, data structures, tree, graph

1. Introduction

Data structure is the collection and organization of data items efficiently in the system's memory in order to make operations such as insertion, deletion, searching, and retrieval of data items easier. The data structure where the data elements are stored in continuous memory location (sequential order) is linear data structure. Array, Stack, Queue, and Linked List are linear data structure models. Nonlinear data structure models store data elements in random memory locations. Tree and Graph are the nonlinear data structure models. In nonlinear data structure, the data items could be connected to more than one item to show the relationship among them. All data items cannot be traversed in a single run in nonlinear data items. In this paper, the Nonlinear data structures and their application are discussed.

2. Tree

Tree is an abstract data type (ADT). It is a nonlinear hierarchical data structure with a root node and subtrees from the parent nodes. It is a collection of connected nodes. It is used to store simple and complex data in many applications. Tree data structure is a collection of nodes connected by edges. The tree starts with a root node (topmost node) followed by the child node. Each node is a data structure with values along with the reference values of the children. The reference value can neither be duplicated nor can point to the root node. The root node being the topmost node does not have a parent node.

The nodes traversing down from the root node are the Child nodes. The nodes from which the child node is traversed is called the Parent node. The group of nodes having the same parent is called Siblings. Fig.1 is an example of a Tree data structure. Node A is the root node, nodes E and F are child nodes of node B, node B is the parent node of node E and F, nodes E and F are siblings because they have the same parent.

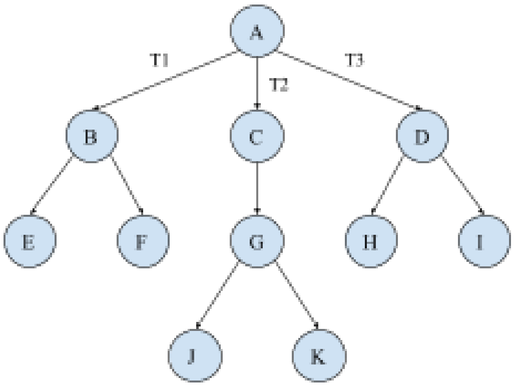


Fig 1. Tree Example

The reader should have knowledge about tree terminologies as a prerequisite.

Types of trees and its applications :

General tree :

A tree in which each node can have either zero or many child nodes is a general tree. There is no limitation on the number of child nodes. It has many subtrees.

Application of General Tree :

The General Tree data structure is used to model the application of file systems in the OS. All the data is organized as files in the operating system. The files are further arranged into directories. These directories are systematized into a tree-like structure called the file system. [2]

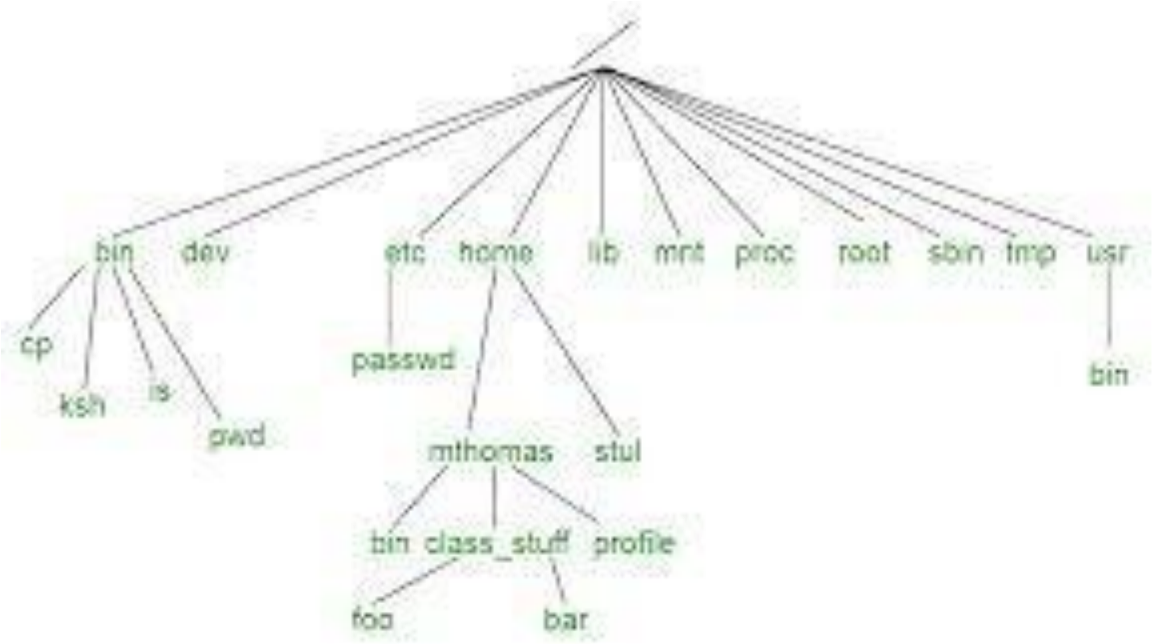


Fig 2. General Tree

General trees have a few drawbacks. The problem occurs when a subtree is further added to the node with the most number of subtrees. The algorithms used for traversing, searching, adding, and deleting the nodes are more complex. To overcome these complexities, the general tree may be converted into a graph data structure, but it loses many advantages of tree data structure. Therefore, it is best to convert the general tree to a binary tree. [3]

1. Binary Tree :

A binary tree is a tree with nodes having a maximum of two child nodes only, i.e., each node cannot have more than two children.

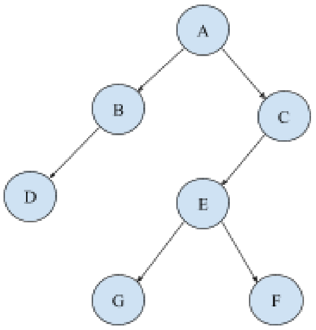


Fig 3. Binary Tree

Applications of Binary Tree :

1. Execution of routing tables in routers.
2. Used in data compression code.
3. Compilation of expression parsers and expression solvers.
4. Construction of symbol table in the compiler.
5. To check the syntax of the statement in the program while compilation using syntax tree.

6. Resolving the problem of indexing in the database.

7. Evaluation of an expression using the expression tree.

8. Used when a two-way decision has to
be made at each point of access.

9. To find duplicates.

10. Used to represent the decision-making process (yes/no decision).

Types of Binary Tree :

a. Full Binary Tree or Strictly Binary Tree

b. Complete Binary Tree

c. Skewed Binary Tree

d. Extended Binary Tree

a. Full Binary Tree :

Full Binary Tree also known as **Strictly Binary Tree** is a binary tree in which each node has either two or no child nodes. [4]

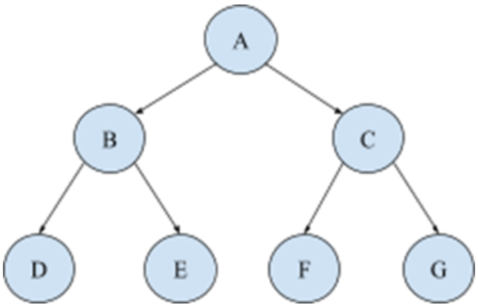


Fig 4. Full Binary Tree

b. Complete Binary Tree :

Complete Binary Tree conjointly referred to as **Perfect Binary Tree** is a binary tree in which all levels of the tree except the last level must be completely filled. In this tree, all nodes have exactly two child nodes and all leaf nodes are at the same level. For instance, if level 2 has 4 nodes (2^2 nodes) then level 3 must have 8 nodes (2^3 nodes).

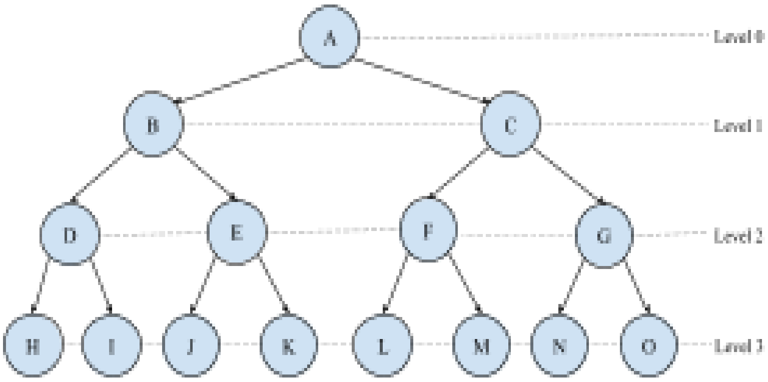


Fig 5. Complete Binary Tree

c. Skewed Binary Tree :

Skewed Binary Tree is a binary tree dominated by the right child node or left child node. All the nodes have one child except the leaf node which has no child node.

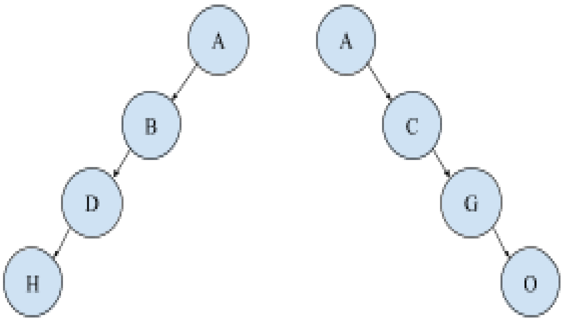


Fig 6. Left Skewed and Right Skewed Binary Tree

d. Extended Binary Tree :

Extended Binary Tree is a binary tree that can be remodeled into a full binary tree by adding special nodes called external nodes to existing nodes(internal nodes) whenever required. Each internal node in an extended tree has two child nodes and every external node is the leafnode of the tree. In an extended tree, the internal nodes are represented by circles and external nodes are represented by boxes.

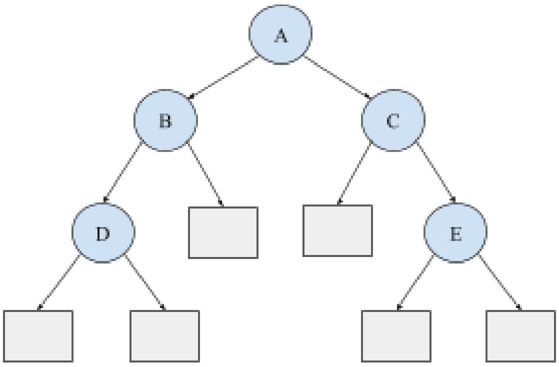


Fig 7. Extended Binary Tree

3. Binary Search Tree (BST) :

The Binary Search Tree is also referred to as **Ordered** or **Sorted Binary Tree**. The properties of BST are :

- The value of the left child of each node in the tree is less than its parent node value at each level.
- The value of the right node of each node subtree is greater than its parent node value at each level.

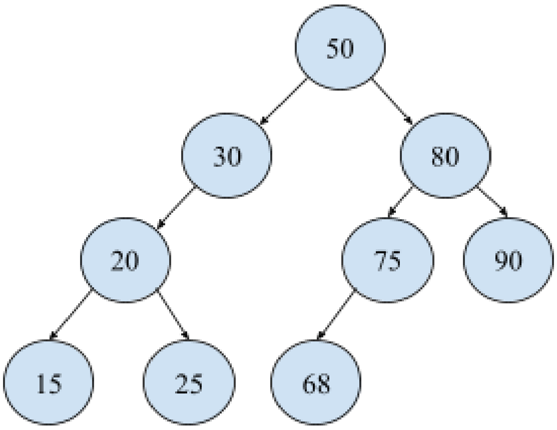


Fig 8. Binary Search Tree

Application of Binary Search Tree :

1. Used to effectively organize data in the sorted form to make quick access to data and to search the stored data elements.

2. Used in internet routers, to store a collection of names and to find the name, based on prefix.

3. Used in Traveling Salesman problem, to store the path in the graph and to reverse any subsection of the path.

4. To find the square root of the number.

5. Used for efficient search in range.

4. Expression tree :

Expression Tree is used to represent mathematical expressions. The expression tree can represent boolean and algebraic expressions. This tree can represent expressions having both unary and binary operators. Fig 9. illustrates the expression tree for the expression 3+((4*5)-(9+6)).

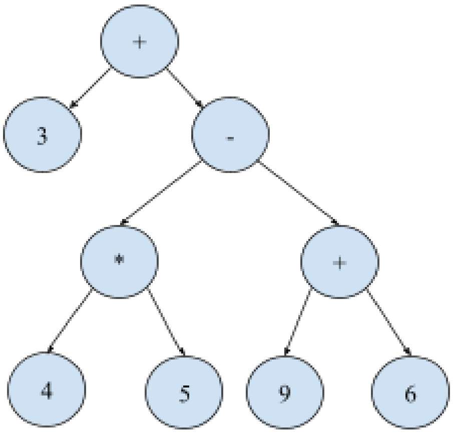


Fig 9. Expression Tree

Application of Expression Tree:

1. To parse the expressions, i.e., to verify the syntax of the expression in the compiler and calculator.

2. To represent numerical and logical expressions in programs.

3. Symbolic manipulation of equations.

5. AVL Tree :

AVL Tree is a height-balanced binary search tree which is named after the inventors of this data structure, G.M. Adelson, Velskey, and E.M. Landis in 1962. It is known as a height-balanced tree since each node in this tree is associated with a balance factor which is obtained by the difference between the height of its right and left subtree. AVL tree is also referred to as the self-balancing tree. [3]

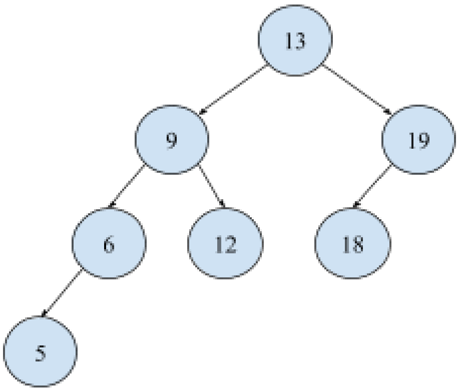


Fig 10. AVL Tree

Applications of AVL Tree :

1. AVL trees are used for frequent insertion. The Memory management subsystem of the Linux kernel uses the AVL tree during preemption to search memory regions of the processes.

2. AVL tree is also used to sort data that can also be used in the database. [7] The examples are :

- Dictionary
- Google Search Engine
- Some sites to take data faster
- Sites that use large amounts of data like Facebook.

6. Red-black tree :

Red-Black Tree is a binary search tree with nodes colored in red or black. The black nodes are NULL (leaf nodes) and if a node is red, then both its child nodes are black. [3]

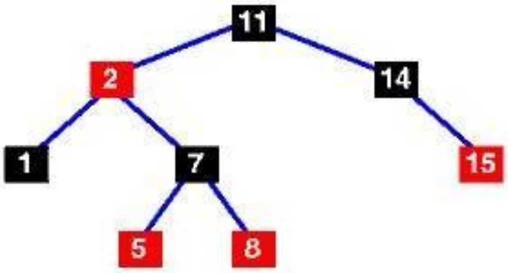


Fig 11. Red-Black Tree

Applications of Red-Black Tree :

1. Linux Completely Fair Scheduler
2. Java Hashmap
3. Process Abstraction(for file description management)
4. Virtual Memory System
5. Finding items in Frequent Pattern Mining

7. Splay tree :

Splay Tree, which is a self-balancing binary search tree has an additional property of quickly re-accessing the recently accessed data items. Operations such as insertion, deletion, and search are executed in O(log(n)) amortized time which makes the splay tree an efficient binary tree. The execution of operations with many non-uniform sequences, performed by the splay tree is better than other binary search trees, even when the sequence pattern is unknown. [3]

Applications of Splay Tree :

1. A set of strings can be represented by the lexicographic search tree. [5]

2. Used in network routers to quickly decide on which outgoing connection, each network packet has to be sent based on the IP address.

3. Used to implement memory allocations and caches in data compression. It is also utilized by Windows NT in virtual memory, networking, and filesystem.

4. Intruder Detection System (IDS) being a part of security infrastructure to detect, identify and stop intruders, uses the splay tree (decision tree) data structure to increase the probability of detecting suspect threats while the risk of false positives is reduced.

5. Splay trees are also utilized in software such as GCC compiler, GNU C++ library, Unix malloc, the sed string editor, Linux loadable kernel modules, and other software. [13]

8. B-Tree :

B-Tree is a self-balancing tree data structure that organizes data elements in sorted order and allows operations such as insertion, deletion, searches, and sequential access in logarithmic amortized time. [8]

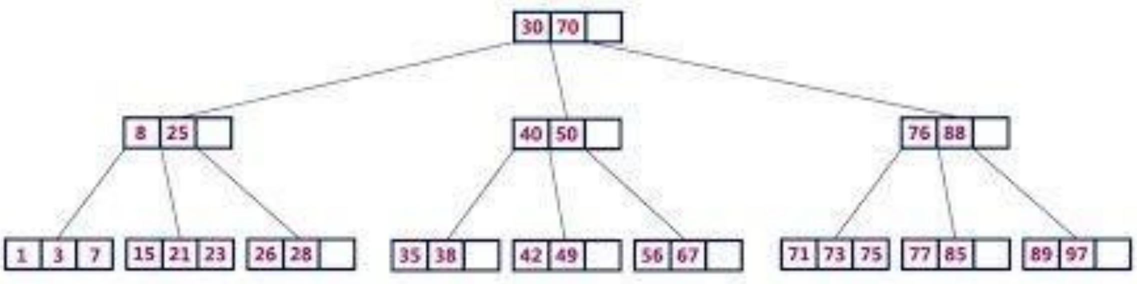


Fig 12. B-Tree of order 4

Applications of B-Tree :

1. B-Tree is commonly used for indexing in database and file systems to allow quick access to data stored in disks and to improve the search operations.

2. The database uses the B-Tree data structure for secondary indexing to facilitate the select operation in range queries. [3]

3. Used to store blocks of data (secondary storage media).

4. Utilized in multi-level indexing.

9. B+ TREE :

The B+ tree serves as the extension of the B-tree which provides efficient and effective insertion, retrieval, and deletion operations of the data items. The difference between B-tree and B+ tree is that both keys and records(data) are stored in interior nodes in B-tree while in the B+tree, the interior nodes store only the key value and the leaf nodes store the records. The interior nodes in the B+tree are referred to as index nodes or i-nodes as they store the index values. [3]

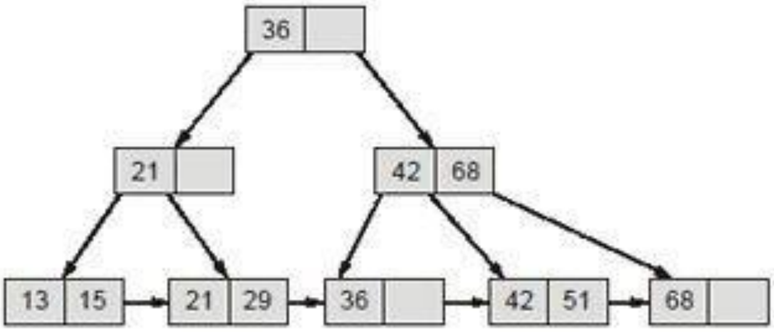


Fig 13. B+ Tree

Application of B+ Tree :

1. File organization in the database

10. 2-3 TREE :

The 2-3 Tree is a form of B-tree and each node can have 2 or 3 child nodes. It can never be a binary tree because it can have more than two children i.e., 3 nodes.

This tree is a search tree but is different from the binary search tree. The 2-node is either a leaf node with one non-null value or has exactly 2 child nodes with values. The 3-node is either a leaf node with two values or has exactly 3 child nodes with non-null values. The 2-3 tree has all leaf nodes at the same level. [3]

The 2-3 trees are efficient for inserting, searching, and deleting data elements as the tree is self-balanced. They can store and retrieve data in a nonlinear order.

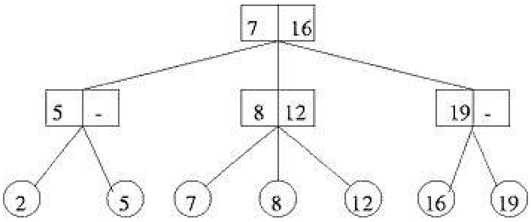


Fig 14. 2-3 Tree

11. TRIE :

Trie is an efficient information re***trie***val data structure and hence the name "trie". This tree stores the dynamic array of strings as key values. The trie also known as **Prefix Tree** since it retrieves a word from the dictionary if the prefix of the word is provided. The number of pointers in each node is equal to the number of characters. For instance, if the strings stored in nodes consist of all the alphabets from 'a' to 'z', then there are 26 pointers in each node of the trie.[10]

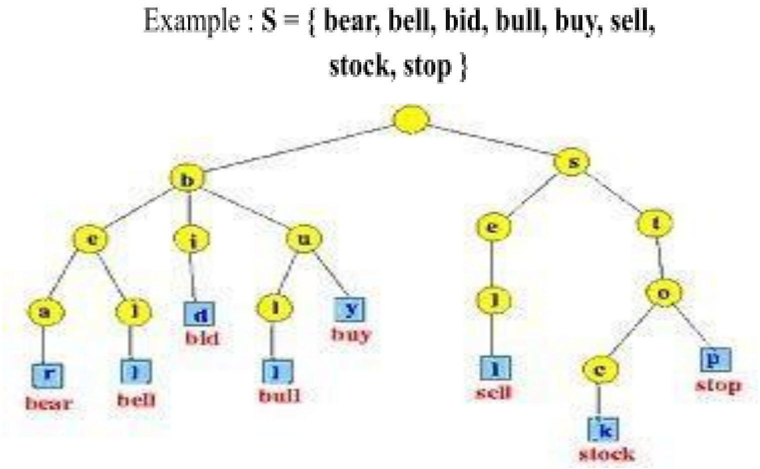

Example : S = { bear, bell, bid, bull, buy, sell, stock, stop }
b
s
e
i
u
e
t
a
l
d
l
y
l
o
r
l
l
l
c
p
bear
bell
bid
bull
buy
sell
stop
k
stock

Fig 15. Trie

Applications of Trie :

1. **Spell Checkers:** Spell checking is done in three steps. Find the word in the dictionary. If the word is found, generate the potential suggestions. Finally, sort the word suggestions with the intended word on top.

2. **AutoComplete:** The autocomplete function uses trie as this tree provides an easy search operation to find possible words from the dictionary to complete the word. The Autocomplete function is used mostly in text editors and mobile applications.

3. **Longest Prefix Matching:** In routers, the trie is used in longest prefix matching to select the entry from the routing table for Internet Protocol (IP)networking.

4. **Automatic Command Completion:** Used to complete the commands automatically while typing system commands in the operating system such as UNIX.

5. **Network browser history:** Previously visited sites' URLs are recorded on the history page of the network browser. The trie organizes this history and provides the complete URL once the prefix of the previously visited URL is typed.

12. HEAP :

Heap is a complete binary tree data structure with two properties, structure and heap order property. Heap is a binary tree by structure and min or max heap by order. [12]

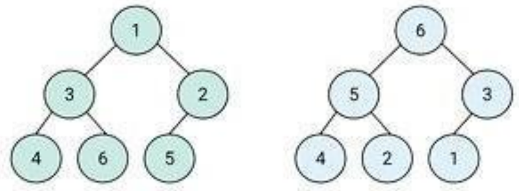


Fig 16. a) Min Heap b) Max Heap

Applications of Heap :

1. To implement a priority queue in Operating System Design.

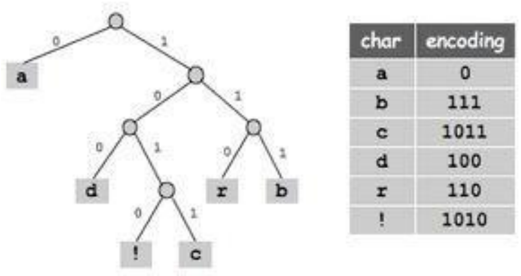
2. Used in the heap sort algorithm, one of the fast and space-saving algorithms.

3. QoS in routers.

4. AI applications such as robotics and video games use the heap data structure for the path-finding algorithm.

13. Huffman tree :

Huffman Tree is a full binary tree, in which the given alphabet corresponds to each leaf node of the tree. The data in each leaf node is unique, and only the leaf nodes have data. [14]



char	encoding
a	0
b	111
c	1011
d	100
r	110
!	1010

Fig 17. a) Huffman Tree b) Character code

Applications of Huffman Tree :

1. Used for lossless data (text/image/audio) compression formats like GZIP, PKZIP (winzip) and BZIP2.

2. Multimedia codecs like JPEG, PNG, and MP3. [15]

3. Optimization based on frequency.

4. Used for text and fax transmission.

14. Segment Tree :

The Segment Tree is a binary tree used to store the intervals or segments [16] of linear data structure (array). It is also known as the **Static Tree**. This tree allows the processing of range queries effectively and also allows modification of array. Each node in the tree consists of the

interval of the array. The segment tree can be used to process multiple queries. The leaf nodes store the array while the internal nodes store segment information of the array represented by its child nodes. The internal nodes merge the information from its child nodes and hence the tree is modeled in a bottom-up manner. [17]

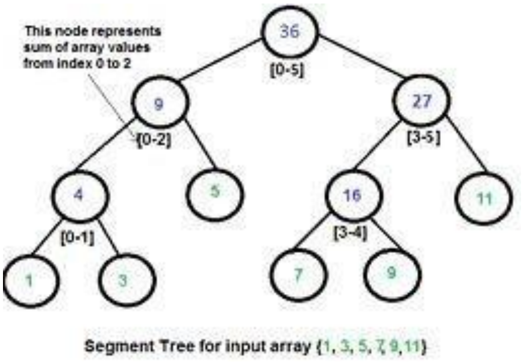

This node represents sum of array values from index 0 to 2
36
[0-5]
9
[0-2]
27
[3-5]
4
5
16
11
[0-1]
[3-4]
1
3
7
9
Segment Tree for input array {1, 3, 5, 7, 9, 11}

Fig 18. Segment Tree

Applications of Segment Tree :

1.This tree is used when a search-heavy application performs a lot of specific range queries on a data set (e.g., sum, min, and max queries).

2.It can be used in the front end application if the needs of the application calls for it. [18]

15. Fenwick tree :

The Fenwick Tree is known as the **Binary Indexed Tree** is used to process interval/range-based queries, compute prefix sums, and update data elements effectively. It is a multi-dimensional data structure. Fenwick tree utilizes less space when compared to the segment tree and is easy to implement. It can be used with an operation that is invertible. Fig 19. shows each index from 1 to 8 and each node is liable to its own index and the index of the children.

Applications of Fenwick Tree :

1.It is used to implement the arithmetic coding algorithm.It is used to count inversions in an array in O(NlogN) time.

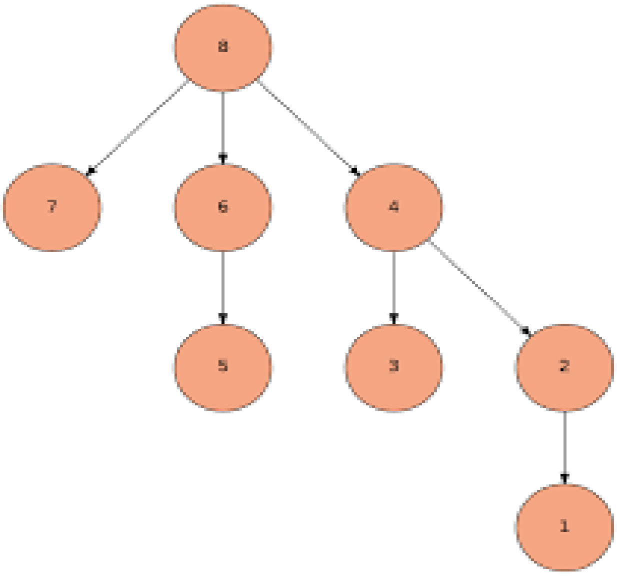


Fig 19. Fenwick Tree

16. Suffix tree :

The Suffix Tree is a tree data structure constructed from the suffixes of a string. It is a compressed trie.

Consider a set of strings:

{“banana”,”nabd”,”bcdef”,”bcfeg”,”aaaaaa”,”aaabaa”}. The standard trie and compressed trie for the above set are shown below.

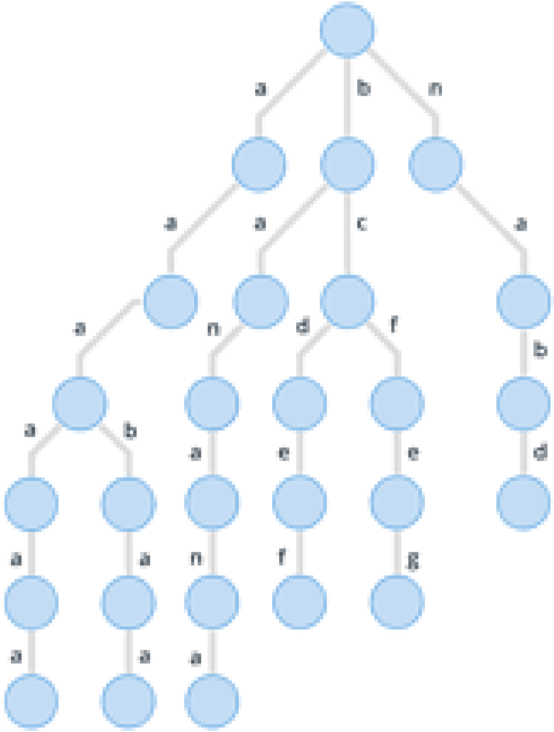


Fig 20. a) Standard Trie

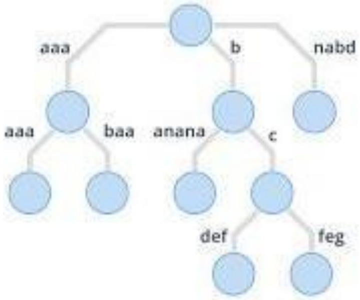


Fig 20. b) Compressed Trie

Applications of Suffix Tree :

1. Helps in solving a lot of string-related problems like pattern matching.
2. Finding distinct substrings in the given string.
3. Finding the longest palindrome.

17. GGM TREE :

GGM Tree is used to model a tree data structure of pseudo-random numbers in cryptographic applications.

18. TREAP :

Treap is the combination of both tree and heap. Each node of a treap consists of two values - key and priority. The key is ordered according to the BST property (the left child is smaller and the right child is greater than the parent node). Priority is a random value ordered according to max-heap property.

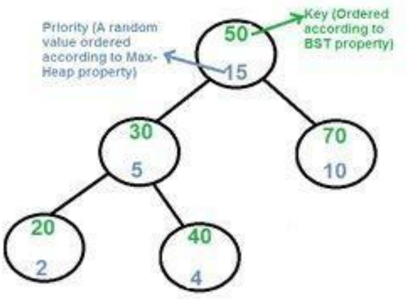

Priority (A random value ordered according to Max-Heap property)
50
15
Key (Ordered according to BST property)
30
5
70
10
20
2
40
4

Fig 21. Treap

Application of Treap :

1. Used in memory allocation and wireless networking.

19. T-TREE :

Though most databases generally use B-trees to store data on drives, some databases that store every data in memory utilizes the T-trees.

T-trees are used in main-memory databases such as ExtremeDB, Datablitz, MySQL Cluster, MobileLite, and Oracle TimesTen.

20. HASH TREE / MERKLE TREE:

Hash Tree is a tree data structure in which each internal node is a hash of its child node. It is used in specialized image-signatures and in p2p programs in which the hash is verified. These trees are used in :

- Hash-based cryptography
- IPFS, BTRFS, and ZFS filesystems
- DAT Protocol
- Apache Wave Protocol
- Zeronet
- Tahoe-LAFS backup system
- NoSQL systems such as Cassandra, Apache, Dynamo, and Riak
- Bitcoin and Ethereum peer-peer networks

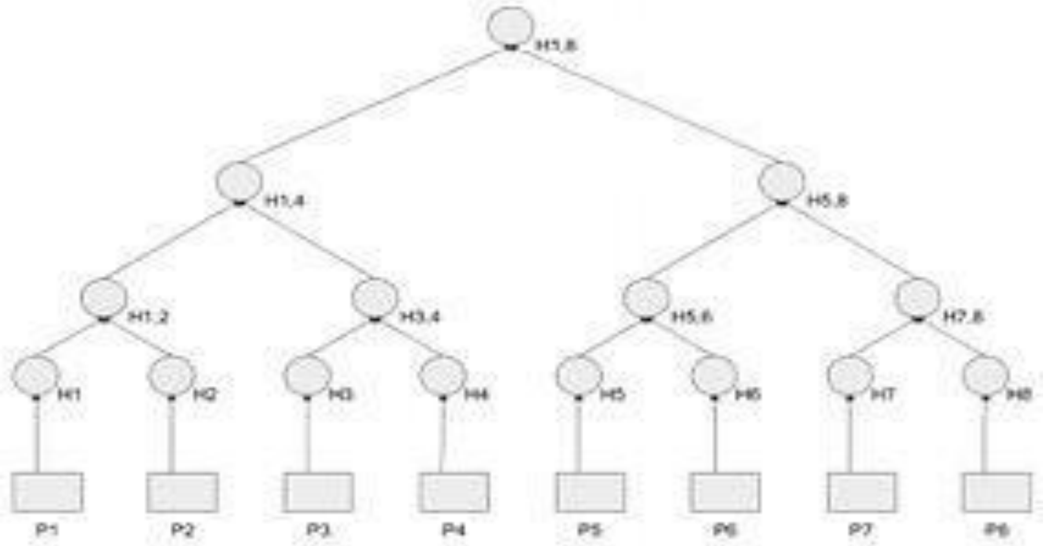


Fig 22. Hash Tree

3. GRAPH :

The graph is used to model nonlinear data elements. It is a nonlinear data structure that consists of a number of nodes called vertices and these vertices are connected by the edges. The graph data structure represents paired relationships between the objects. For example, graphs are used for transportation networks in which the ports, airports, intersections, etc are represented as nodes while shipping routes, airline flights, one-way roads, etc are represented by edges respectively.[3]

Types of graphs :

1. Directed graph :

In a directed graph, the edges from one vertex to another are directed. This graph can be a weighted or unweighted graph. In Fig.23, A, B, C, D, and E are the vertices of the graph, connected by the edges. The adjacent pairs of vertices in the graph are (A,B), (A,D), (B,D), (C,B), (D,E), and (E,C).

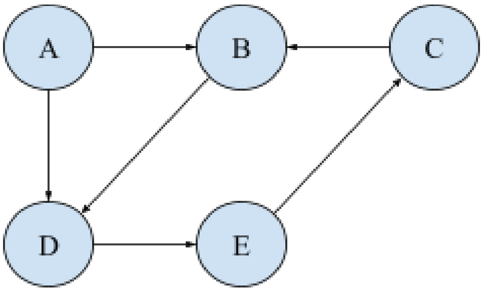


Fig 23. Directed Graph

2. Undirected graph :

In an undirected graph, the edges are not directed. In Fig.24, the edges between nodes A and B forms two adjacent pairs (A,B) and (B,A), and the remaining pairs are formed similarly for the other edges.

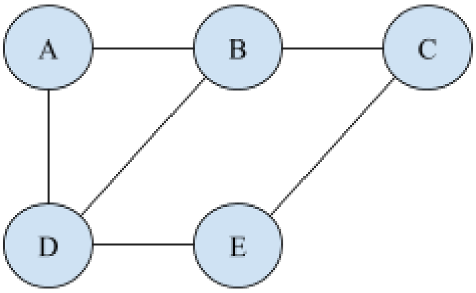


Fig 24. Undirected Graph

3. Mixed graph :

A mixed graph is the combination of a directed and undirected graph i.e., it consists of both directed and undirected edges.

Applications of Graph :

1. Google Maps use the graph data structure as the routing system algorithm to find the shortest path between the places. The vertices are the interconnection of the roads while the edges are the roads connecting the vertices. [21] The graph theory is now used by Apple IOS 6 maps and Bing maps. [23]

2. Friend Suggestion algorithm in Facebook uses graph theory. The undirected graph is used, by which the users (vertices) are connected to their friends through the edges. [21]

3. Directed graphs are used in the World Wide Web to link the web pages. The web pages are the vertices and there is an edge between the web pages if there's a link among the pages. Google Ranking Algorithm works based on the graph. [21]

4. In the Operating System, graph theory is used to allocate resources in which each process and resource is represented as a vertex, and the connection from resources to the allocated process or connection from requesting process to requesting resource is drawn by the edges. [21]

5. Graph theory is used in e-commerce websites in the “Recommendation for you” section to suggest items of similarity based on the user’s choice. [22]

6. Graph theory is used in the study of molecules in physics and chemistry. [22]

7. Social Network Graph : These graphs are used to model data like who follows whom, like in the Twitter graph, and other social structure relationships like who influences whom, who communicates with whom, etc. They are used to determine how communication develops, information flows, or even who it is. [23]

8. Utility Graph : The water network, power grid, the internet use examples of the utility graph. The connection points are represented by vertices while the edges represent the pipes or wires between them. This graph is important to analyze the failure, attack, or minimizing the cost of infrastructure based on the demands. [23]

9. Protein-protein Intersection Graph :This graph is used in the study of molecular chain interactions of proteins in the human cellular process. While the vertices in the graph represent the proteins, the edges represent the biological interactions between the proteins to perform the function in the cell. [23]

10. Network Packet Traffic Graph :The vertices in this graph are the IP addresses and the edges are the packets flowing between the IP addresses. These graphs are used to analyze network security and to track criminal and non-criminal activity. [23]

11. Scene Graph : These graphs are of major importance in the computer graphics and gaming industry. It is used to represent the relationships between the objects in the game or scenes. [23]

12. Finite Element Meshes : Engineering simulations of physical systems uses finite element meshes. The elements from the simulations of physical systems such as the spread of earthquakes through the ground, the flow of air over the car, vibrations of building, etc along with the connection between their adjacent elements, form the finite element mesh. [23]

13. Robot planning : The vertices represent the states of the robot and edges represent the possible transition between the states. Such graphs are used in autonomous vehicles in planning paths. [23]

14. Neural networks : Neural networks are used to understand brain connections and their work. The vertices represent the neurons and edges represent the synapses between them. [23]

15. Quantum field theory graph : The vertices represent the states of the quantum system and edges represent the transition between them. This graph analyzes the path integrals and generates the quantum amplitude by summing the path integrals. [23]

16. Semantic Networks : Semantic network graphs represent words as the vertices and their relationship as edges. It is used to model how humans and machines organize their knowledge and organization respectively. [23]

17. Graphs used in epidemiology represent individual persons as vertices and infection transfer from one person to another as directed edges. This graph helps us to understand and control the spread of diseases. [23]

18. Compilers use graphs for data flow analysis, type interference, register allocation, etc. Database languages use the graph for query optimization.[23]

19. Constraint Graph : This graph represents the constraints among the items. In the GSM network of mobile phones, the constraints are modeled as a graph in which vertices represent the overlapped cells connected by edges. [23]

20. Dependence Graph : It represents dependencies and priorities among data elements. Analyzing this graph minimizes the cost or time based on the dependencies. [23]

4. Conclusion :

Thus the nonlinear data structures play a vital role in developing many real-world applications. Trees are used for quick data access from databases, easy information retrieval, and efficient searching. It can be implemented in other data structures like sets, hash tables, and maps. Graphs are used to model the applications where numbers of items are used and to analyze the relationship among the items.

References

1) https://en.wikipedia.org/wiki/Tree_(data_structure)
2) https://www.geeksforgeeks.org/operating-system-unix-file-system/
3) ReemaThareja, “Data Structures Using C”, 2nd Edition, Oxford UniversityPress
4) https://www.tutorialride.com/data-structur

5) es/types-of-binary-tree.htm
6) Daniel Dominic Sleator, Robert EndreTarjan, “Self Adjusting BinarySearch trees”, Journal of the Association for Computing Machinery. Vol. 32, No. 3,
7) July 1985, pp. 652-686.
8) https://perptuallearning.wordpress.com/20 15/03/09/avl-trees-and-its-applications/
9) https://www.quora.com/What-are-some-practical-applications-of-AVL-trees-and-spl ay-trees
10) https://www.cpp.edu/~ftang/courses/CS241/notes/b-tree.htm
11) Tenenbaum, Langsam, and Augenstein, “Data Structures using C”, Prentice Hall India.
12) http://blog.xebia.in/index.php/2015/09/28/applications-of-trie-data-structure/
13) http://www.mathcs.emory.edu/~cheung/Courses/323/Syllabus/Text/trie01.html
14) Mark Allen Weiss, “Data Structuresand Algorithm Analysis in C”, 2nd Edition, Pearson Education.
15) https://www.quora.com/What-are-some-practical-applications-of-AVL-trees-and-spl ay-trees
16) http://www2.hawaii.edu/~janst/211/lecture/Huffman%20Trees.htm
17) https://www.studytonight.com/data-structures/huffman-coding
18) https://www.hackerearth.com/practice/data-structures/advanced-data-structures/segment-trees/tutorial/
19) https://iq.opengenus.org/segment-tree/
20) https://hackernoon.com/practical-data-stru ctures-for-frontend-applications-when-to-u se-segment-trees-9c7cdb7c2819
21) https://www.hackerearth.com/practice/data-structures/advanced-data-structures/suffix-trees/tutorial/
22) https://en.wikipedia.org/wiki/Merkle_tree
23) https://www.geeksforgeeks.org/applications-of-graph-data-structure/
24) https://www.quora.com/What-are-the-real-
25) life-applications-of-graph-data-structure
26) http://www.cs.cmu.edu/afs/cs/academic/class/15210-s14/www/lectures/graphs.pdf.