

List of data structures

This is a list of notable [data structures](#). For a wider list of terms, see [list of terms relating to algorithms and data structures](#). For a comparison of running time a subset of this list see [comparison of data structures](#).

Contents

[Data types](#)

- [Primitive types](#)

- [Composite types or non-primitive type](#)

- [Abstract data types](#)

[Linear data structures](#)

- [Arrays](#)

- [Lists](#)

[Trees](#)

- [Binary trees](#)

- [B-trees](#)

- [Heaps](#)

- [Trees](#)

- [Multi way trees](#)

- [Space-partitioning trees](#)

- [Application-specific trees](#)

[Hash-based structures](#)

[Graphs](#)

[Other](#)

[See also](#)

[External links](#)

Data types

[Primitive types](#)

- [Boolean](#), true or false.
- [Character](#)
- [Floating-point](#) numbers, limited precision approximations of [real number](#) values.
 - Including [Single precision](#) and [Double precision](#) [IEEE 754](#) Floats, among [others](#)
- [Fixed-point](#) numbers
- [Integer](#), integral or fixed-precision values.
- [Reference](#) (also called a pointer or handle), a small value referring to another object's address in memory, possibly a much larger one.

- Enumerated type, a small set of uniquely named values.
- Date Time, value referring to Date and Time

Composite types or non-primitive type

- Array (as an example String which is an array of characters)
- Record (also called Associative array, Map, or structure)
- Union (Tagged union is a subset, also called variant, variant record, discriminated union, or disjoint union)

Abstract data types

- Container
- List
- Tuple
- Multimap
- Set
- Multiset (bag)
- Stack
- Queue (example Priority queue)
- Double-ended queue
- Graph (example Tree, Heap)

Some properties of abstract data types:

Structure	Order	Unique
<u>List</u>	yes	no
<u>Associative array</u>	no	yes
<u>Set</u>	no	yes
<u>Stack</u>	yes	no
<u>Multimap</u>	no	no
<u>Multiset (bag)</u>	no	no
<u>Queue</u>	yes	no

Order means the insertion sequence counts. Unique means that duplicate elements are not allowed, based on some inbuilt or, alternatively, user-defined rule for comparing elements.

Linear data structures

A data structure is said to be linear if its elements form a sequence.

Arrays

- Array
- Bit array

- [Bit field](#)
- [Bitboard](#)
- [Bitmap](#)
- [Circular buffer](#)
- [Control table](#)
- [Image](#)
- [Dope vector](#)
- [Dynamic array](#)
- [Gap buffer](#)
- [Hashed array tree](#)
- [Lookup table](#)
- [Matrix](#)
- [Parallel array](#)
- [Sorted array](#)
- [Sparse matrix](#)
- [Iliffe vector](#)
- [Variable-length array](#)

Lists

- [Doubly linked list](#)
- [Array list](#)
- [Linked list](#)
- [Association list](#)
- [Self-organizing list](#)
- [Skip list](#)
- [Unrolled linked list](#)
- [VList](#)
- [Conc-tree list](#)
- [Xor linked list](#)
- [Zipper](#)
- [Doubly connected edge list](#) also known as half-edge
- [Difference list](#)
- [Free list](#)

Trees

Binary trees

- [AA tree](#)
- [AVL tree](#)
- [Binary search tree](#)
- [Binary tree](#)
- [Cartesian tree](#)
- [Conc-tree list](#)

- [Left-child right-sibling binary tree](#)
- [Order statistic tree](#)
- [Pagoda](#)
- [Randomized binary search tree](#)
- [Red-black tree](#)
- [Rope](#)
- [Scapegoat tree](#)
- [Self-balancing binary search tree](#)
- [Splay tree](#)
- [T-tree](#)
- [Tango tree](#)
- [Threaded binary tree](#)
- [Top tree](#)
- [Treap](#)
- [WAVL tree](#)
- [Weight-balanced tree](#)

B-trees

- [B-tree](#)
- [B+ tree](#)
- [B*-tree](#)
- [B sharp tree](#)
- [Dancing tree](#)
- [2-3 tree](#)
- [2-3-4 tree](#)
- [Queap](#)
- [Fusion tree](#)
- [Bx-tree](#)
- [AList](#)

Heaps

- [Heap](#)
- [Binary heap](#)
- [B-heap](#)
- [Weak heap](#)
- [Binomial heap](#)
- [Fibonacci heap](#)
- [AF-heap](#)
- [Leonardo Heap](#)
- [2-3 heap](#)
- [Soft heap](#)
- [Pairing heap](#)
- [Leftist heap](#)

- [Treap](#)
- [Beap](#)
- [Skew heap](#)
- [Ternary heap](#)
- [D-ary heap](#)
- [Brodal queue](#)

Trees

In these data structures each tree node compares a bit slice of key values.

- [Tree \(data structure\)](#)
- [Radix tree](#)
- [Suffix tree](#)
- [Suffix array](#)
- [Compressed suffix array](#)
- [FM-index](#)
- [Generalised suffix tree](#)
- [B-tree](#)
- [Judy array](#)
- [X-fast trie](#)
- [Y-fast trie](#)
- [Merkle tree](#)
- [C tree](#)

Multi way trees

- [Ternary tree](#)
- [K-ary tree](#)
- [And-or tree](#)
- [\(a,b\)-tree](#)
- [Link/cut tree](#)
- [SPQR-tree](#)
- [Spaghetti stack](#)
- [Disjoint-set data structure \(Union-find data structure\)](#)
- [Fusion tree](#)
- [Enfilade](#)
- [Exponential tree](#)
- [Fenwick tree](#)
- [Van Emde Boas tree](#)
- [Rose tree](#)

Space-partitioning trees

These are data structures used for [space partitioning](#) or [binary space partitioning](#).

- [Segment tree](#)
- [Interval tree](#)
- [Range tree](#)
- [Bin](#)
- [K-d tree](#)
- [Implicit k-d tree](#)
- [Min/max k-d tree](#)
- [Relaxed k-d tree](#)
- [Adaptive k-d tree](#)
- [Quadtree](#)
- [Octree](#)
- [Linear octree](#)
- [Z-order](#)
- [UB-tree](#)
- [R-tree](#)
- [R+ tree](#)
- [R* tree](#)
- [Hilbert R-tree](#)
- [X-tree](#)
- [Metric tree](#)
- [Cover tree](#)
- [M-tree](#)
- [VP-tree](#)
- [BK-tree](#)
- [Bounding interval hierarchy](#)
- [Bounding volume hierarchy](#)
- [BSP tree](#)
- [Rapidly exploring random tree](#)

Application-specific trees

- [Abstract syntax tree](#)
- [Parse tree](#)
- [Decision tree](#)
- [Alternating decision tree](#)
- [Minimax tree](#)
- [Expectiminimax tree](#)
- [Finger tree](#)
- [Expression tree](#)
- [Log-structured merge-tree](#)
- [Lexicographic Search Tree](#)

Hash-based structures

- [Bloom filter](#)

- [Count-Min sketch](#)
- [Distributed hash table](#)
- [Double hashing](#)
- [Dynamic perfect hash table](#)
- [Hash array mapped trie](#)
- [Hash list](#)
- [Hash table](#)
- [Hash tree](#)
- [Hash trie](#)
- [Koorde](#)
- [Prefix hash tree](#)
- [Rolling hash](#)
- [MinHash](#)
- [Quotient filter](#)
- [Ctrie](#)

Graphs

Many [graph](#)-based data structures are used in computer science and related fields:

- [Graph](#)
- [Adjacency list](#)
- [Adjacency matrix](#)
- [Graph-structured stack](#)
- [Scene graph](#)
- [Decision tree](#)
 - [Binary decision diagram](#)
- [Zero-suppressed decision diagram](#)
- [And-inverter graph](#)
- [Directed graph](#)
- [Directed acyclic graph](#)
- [Propositional directed acyclic graph](#)
- [Multigraph](#)
- [Hypergraph](#)

Other

- [Lightmap](#)
- [Winged edge](#)
- [Quad-edge](#)
- [Routing table](#)
- [Symbol table](#)

See also

- Purely functional data structure

External links

- Tommy Benchmarks (<http://tommyds.sourceforge.net/doc/benchmark.html>) Comparison of several data structures.
-

Retrieved from "https://en.wikipedia.org/w/index.php?title=List_of_data_structures&oldid=994736411"

This page was last edited on 17 December 2020, at 08:05 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.