.

.

**Data structure through c++**

In [computer science](https://en.wikipedia.org/wiki/Computer_science), a **data structure** is a particular way of organizing [data](https://en.wikipedia.org/wiki/Data_(computing)) in a computer so that it can be used

## [https://upload.wikimedia.org/wikipedia/commons/thumb/7/7d/Hash_table_3_1_1_0_1_0_0_SP.svg/315px-Hash_table_3_1_1_0_1_0_0_SP.svg.png](https://en.wikipedia.org/wiki/File:Hash_table_3_1_1_0_1_0_0_SP.svg)

**Usage**

Data structures can implement one or more particular [abstract data types](https://en.wikipedia.org/wiki/Abstract_data_type) (ADT), which specify the operations that can be performed on a data structure and the [computational complexity](https://en.wikipedia.org/wiki/Computational_complexity_theory) of those operations. In comparison, a data structure is a concrete implementation of the specification provided by an ADT

Different kinds of data structures are suited to different kinds of applications, and some are highly specialized to specific tasks. For example, relational databases commonly use [B-tree](https://en.wikipedia.org/wiki/B-tree) indexes for data retrieval,[[3]](https://en.wikipedia.org/wiki/Data_structure" \l "cite_note-3) while [compiler](https://en.wikipedia.org/wiki/Compiler) implementations usually use [hash tables](https://en.wikipedia.org/wiki/Hash_table) to look up identifiers.

Data structures provide a means to manage large amounts of data efficiently for uses such as large [databases](https://en.wikipedia.org/wiki/Database) and [internet indexing services](https://en.wikipedia.org/wiki/Web_indexing). Usually, efficient data structures are key to designing efficient [algorithms](https://en.wikipedia.org/wiki/Algorithm). Some formal design methods and [programming languages](https://en.wikipedia.org/wiki/Programming_language) emphasize data structures, rather than algorithms, as the key organizing factor in software design. Data structures can be used to organize the storage and retrieval of information stored in both [main memory](https://en.wikipedia.org/wiki/Main_memory) and [secondary memory](https://en.wikipedia.org/wiki/Secondary_memory)

**Implementation**

Data structures are generally based on the ability of a computer to fetch and store data at any place in its memory, specified by a [pointer](https://en.wikipedia.org/wiki/Pointer_(computer_programming))—a bit string, representing a [memory address](https://en.wikipedia.org/wiki/Memory_address), that can be itself stored in memory and manipulated by the program. Thus, the [array](https://en.wikipedia.org/wiki/Array_data_structure) and [record](https://en.wikipedia.org/wiki/Record_(computer_science)) data structures are based on computing the addresses of data items with [arithmetic operations](https://en.wikipedia.org/wiki/Arithmetic_operations); while the [linked data structures](https://en.wikipedia.org/wiki/Linked_data_structure) are based on storing addresses of data items within the structure itself. Many data structures use both principles, sometimes combined in non-trivial ways (as in [XOR linking](https://en.wikipedia.org/wiki/XOR_linked_list)

The implementation of a data structure usually requires writing a set of [procedures](https://en.wikipedia.org/wiki/Subroutine) that create and manipulate instances of that structure. The efficiency of a data structure cannot be analyzed separately from those operations. This observation motivates the theoretical concept of an [abstract data type](https://en.wikipedia.org/wiki/Abstract_data_type), a data structure that is defined indirectly by the operations that may be performed on it, and the mathematical properties of those operations (including their space and time cost)

***Examples***

*Main article:*[*List of data structures*](https://en.wikipedia.org/wiki/List_of_data_structures)An [*array*](https://en.wikipedia.org/wiki/Array_data_structure) is a number of elements in a specific order, typically all of the same type. Elements are accessed using an integer index to specify which element is required (Depending on the language, individual elements may either all be forced to be the same type, or may be of almost any type). Typical implementations allocate contiguous memory words for the elements of arrays (but this is not always a necessity). Arrays may be fixed-length or resizable.

* A [*linked list*](https://en.wikipedia.org/wiki/Linked_list) (also just called *list*) is a linear collection of data elements of any type, called nodes, where each node has itself a value, and points to the next node in the linked list. The principal advantage of a linked list over an array, is that values can always be efficiently inserted and removed without relocating the rest of the list. Certain other operations, such as [random access](https://en.wikipedia.org/wiki/Random_access) to a certain element, are however slower on lists than on arrays.
* A [*record*](https://en.wikipedia.org/wiki/Record_(computer_science)) (also called *tuple* or *struct*) is an aggregate data structure. A record is a value that contains other values, typically in fixed number and sequence and typically indexed by names. The elements of records are usually called *fields* or *members*.
* A [*union*](https://en.wikipedia.org/wiki/Union_(computer_science)) is a data structure that specifies which of a number of permitted primitive types may be stored in its instances, e.g. *float* or *long integer*. Contrast with a [record](https://en.wikipedia.org/wiki/Record_(computer_science)), which could be defined to contain a float *and* an integer; whereas in a union, there is only one value at a time. Enough space is allocated to contain the widest member datatype.
* A [*tagged union*](https://en.wikipedia.org/wiki/Tagged_union) (also called [*variant*](https://en.wikipedia.org/wiki/Variant_type), *variant record*, *discriminated union*, or *disjoint union*) contains an additional field indicating its current type, for enhanced type safety.
* A [*class*](https://en.wikipedia.org/wiki/Class_(computer_programming)) is a data structure that contains data fields, like a record, as well as various [methods](https://en.wikipedia.org/wiki/Method_(computer_programming)) which operate on the contents of the record. In the context of [object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming), records are known as [plain old data structures](https://en.wikipedia.org/wiki/Plain_old_data_structures) to distinguish them from classes.[