The Standard Template Library (STL) is a set of C++ template classes to provide common programming data structures and functions such as lists, stacks, arrays, etc. It is a library of container classes, algorithms, and iterators. It is a generalized library and so, its components are parameterized. Working knowledge of [template classes](https://www.geeksforgeeks.org/templates-cpp/) is a prerequisite for working with STL.

**STL has 4 components:**

* **Algorithms**
* **Containers**
* **Functions**
* **Iterators**

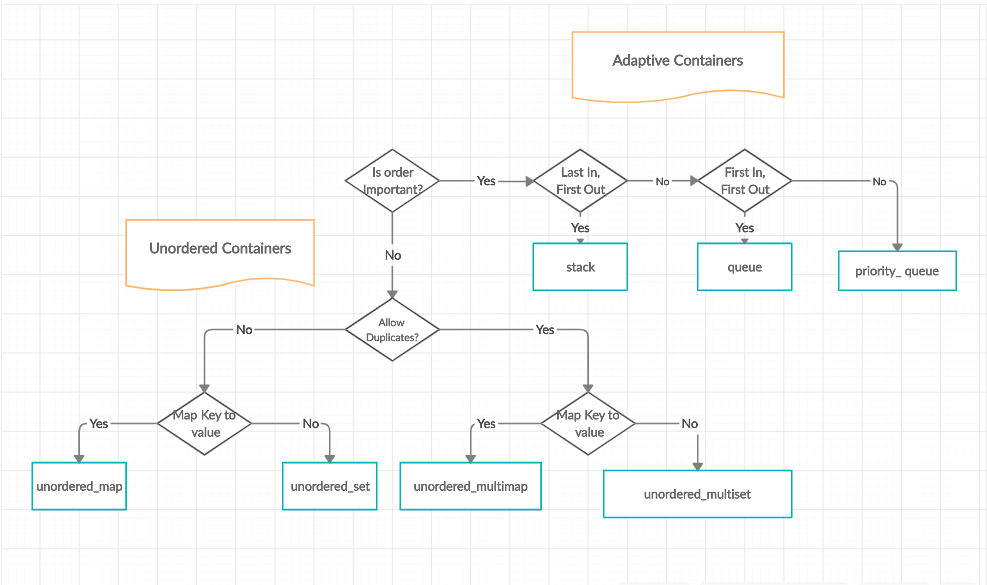
**Algorithms**

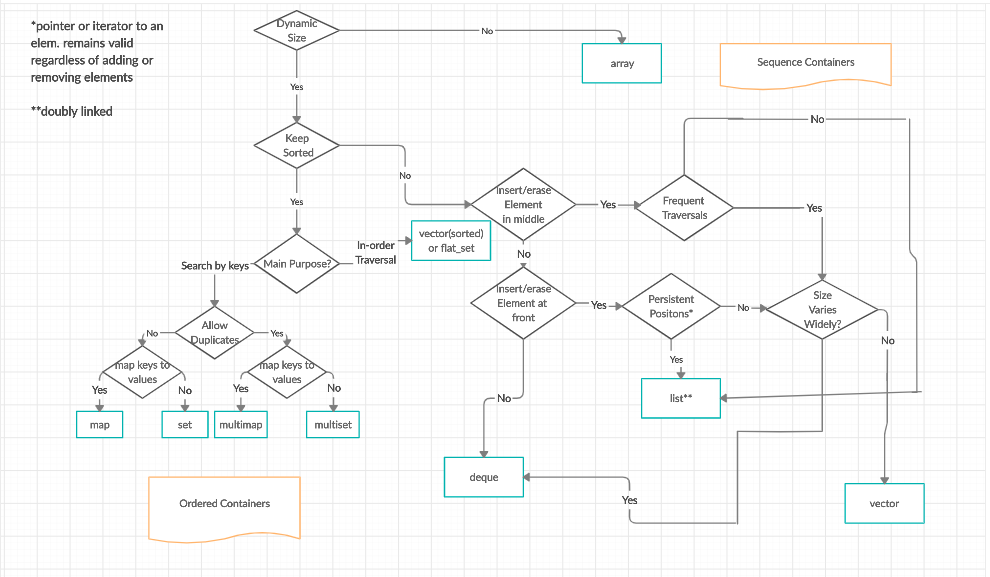
The header algorithm defines a collection of functions specially designed to be used on a range of elements. They act on containers and provide means for various operations for the contents of the containers.

* Algorithm
  + [Sorting](https://www.geeksforgeeks.org/sort-algorithms-the-c-standard-template-library-stl/)
  + [Searching](https://www.geeksforgeeks.org/binary-search-algorithms-the-c-standard-template-library-stl/)
  + [Important STL Algorithms](https://www.geeksforgeeks.org/c-magicians-stl-algorithms/)
  + [Useful Array algorithms](https://www.geeksforgeeks.org/useful-array-algorithms-in-c-stl/)
  + [Partition Operations](https://www.geeksforgeeks.org/stdpartition-in-c-stl/)
* Numeric
  + [valarray class](https://www.geeksforgeeks.org/std-valarray-class-c/)

**Containers**

Containers or container classes store objects and data. There are in total seven standards “first-class” container classes and three container adaptor classes and only seven header files that provide access to these containers or container adaptors.

* **Sequence Containers**: implement data structures that can be accessed in a sequential manner.
  + [vector](https://www.geeksforgeeks.org/vector-in-cpp-stl/)
  + [list](https://www.geeksforgeeks.org/list-cpp-stl/)
  + [deque](https://www.geeksforgeeks.org/deque-cpp-stl/)
  + [arrays](https://www.geeksforgeeks.org/array-class-c/)
  + [forward\_list](https://www.geeksforgeeks.org/forward-list-c-set-1-introduction-important-functions/)( Introduced in C++11)
* Container Adaptors: provide a different interface for sequential containers.
  + [queue](https://www.geeksforgeeks.org/queue-cpp-stl/)
  + [priority\_queue](https://www.geeksforgeeks.org/priority-queue-in-cpp-stl/)
  + [stack](https://www.geeksforgeeks.org/stack-in-cpp-stl/)
* **Associative Containers**: implement sorted data structures that can be quickly searched (O(log n) complexity).
  + [set](https://www.geeksforgeeks.org/set-in-cpp-stl/)
  + [multiset](https://www.geeksforgeeks.org/multiset-in-cpp-stl/)
  + [map](https://www.geeksforgeeks.org/map-associative-containers-the-c-standard-template-library-stl/)
  + [multimap](https://www.geeksforgeeks.org/multimap-associative-containers-the-c-standard-template-library-stl/)
* Unordered Associative Containers: implement unordered data structures that can be quickly searched
  + [unordered\_set](https://www.geeksforgeeks.org/unordered_set-in-cpp-stl/) (Introduced in C++11)
  + [unordered\_multiset](https://www.geeksforgeeks.org/unordered_multiset-and-its-uses/) (Introduced in C++11)
  + [unordered\_map](https://www.geeksforgeeks.org/unordered_map-in-cpp-stl/) (Introduced in C++11)
  + [unordered\_multimap](https://www.geeksforgeeks.org/unordered_multimap-and-its-application/) (Introduced in C++11)



STL=Container+ Algorithms+iterators

Container:

Sequence container

Associative container

Derived Container