**The C++ Standard Template Library (STL)**

**Why is this important for competitive programmers?**

1. Competitive programming is a part of various environments, be it job interviews, coding contests and all, and if you’re in one of those environments, you’ll be given limited time to code your program.
2. So, suppose you want in your program, a resizable array, or sort an array or any other data structure. or search for some element in your container.
3. You will always try to code a function which will execute the above mentioned things, and end up losing a great amount of time. But here is when you will use STL.

* An STL is a library of generic functions and classes which saves you time and energy which you would have spent constructing for your use. This helps you reuse these well tested classes and functions umpteenth number of times according to your own convenience.
* STL is used because it is not a good idea to reinvent something which is already built and can be used to innovate things further. Suppose you go to a company who builds cars, they will not ask you to start from scratch, but to start from where it is left. This is the basic idea behind using STL.

**COMPONENTS OF STL:**

1. Containers
2. Algorithm
3. Iterators

1. **Containers:**

Container is an object which stores data. We have different containers having their own benefits. These are the implemented template classes for our use, which can be used just by including this library. You can even customise these template classes.

2. **Algorithms:**

Algorithms are a set of instructions which manipulates the input data to arrive at some desired result. In STL, we have already written algorithms, for example, to sort some data structure, or search some element in an array. These algorithms use template functions

**3.Iterators:**

Iterators are objects which refer to an element in a container. And we handle them very much container and play a vital role in manipulation of the data.

I’ll give you a quick illustration of how they work combinedly.

5

3

2

7

1

Container

Pointers, which are free to move along container

Iterators moves as instructed by algorithm