Coding Club
IIT Guwahati

CP.begin()

Coding Club, IIT Guwahati

Week - 2

STL :-

Expected Time: 1-2 days

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 generalised library of container **classes, algorithms,** and **iterators**. Basic knowledge of STL is quite handy, and in fact, necessary for competitive programming.

Resources:

- The C++ Standard Template Library (STL) - GeeksforGeeks
- Watch this video to learn the basics of STL
- Read Ch-4 of **CPH** (pg35-pg43)

Frequently used containers:

- Array: Vector used more frequently compared to array, since the former is dynamic in nature. However, higher dimensional arrays are simpler to declare.
- Vector: One of the most powerful and frequently used STL containers.

Study these functions only: push_back(), pop_back(), sort(), *upper_bound() , *lower_bound() ,begin(), end(), rbegin(), size(), resize(), accumulate(), *binary_search() and clear().

- Set: A container which keeps a **unique** copy of every element in **sorted order.**
  Study these functions only: insert(), erase(), begin(), end(), rbegin(), size(), lower_bound(), upper_bound(), find() and clear().
- Multiset: Similar to a set, but it can also store multiple copies of an element.
- Map: Study these functions only: insert(), erase(), begin(), end(), rbegin(), size(), lower_bound(), upper_bound(), find() and clear() and the operator [].

**Will be covered in week-3 in detail*

Not-so-frequently used containers:

- Queue
- Priority Queue
- Stack
- Deque

Following are some practice problems :-

- Problem 1
- Problem 2
- Problem 3
- Problem 4
- Problem 5
- Problem 6
- Problem 7
- Problem 8
- Problem 9

Try to solve the above listed problems using STL.

To strengthen your grasp on STL , you can solve some very easy problems here : https://www.hackerrank.com/domains/cpp/stl/page/1

Sorting:-

Expected Time: 1-2 days

Sorting means arranging elements (in a data structure) in a particular order.
There are several sorting algorithms. You are required to study the following:

1. Insertion Sort
    a. Article
    b. Video
2. Selection Sort
    a. Article
    b. Video
3. Bubble Sort
    a. Article
    b. Video
4. Merge Sort
    a. Article
    b. Video

But, why so many algorithms!!? Read here

While doing CP problems, we don't write any sorting algorithm. C++ STL provides a function to sort elements in any order in O(N log N) time.

Syntax:
For array: sort(arr, arr+arr.size());
For vector: sort(vect.begin(), vect.end());

Formally, sort(it1, it2) will sort elements in ascending order in the range [it1, it2)
Custom sort (Ignore qsort)

Further Read (Optional):

- Sorting a vector of pairs-1
- Sorting a vector of pairs-2

Questions:

- Problem 1
- Problem 2
- Problem 3
- Problem 4
- Problem 5

Additional Questions

- Problem 1
- Problem 2

Maths :-

Expected Time: 0-1 day

There are several questions in CP, which can be solved only by observations and logical reasoning skills. No algorithm is needed as such to solve these kinds of problems. Such questions test the intuitive/mathematical skills of the programmer, and not his/her coding skills.

Questions:

- Problem 1
- Problem 2
- Problem 3
- Problem 4
- Problem 5
- Problem 6
- Problem 7

Additional Questions:

- Problem 1
- Problem 2
- Problem 3

Greedy Algorithm :-

Expected Time: 0-1 day

Resources:

- Watch this lecture of Algomaniacs

Questions:

- Problem 1
- Problem 2
- Problem 3
- Problem 4

Additional Questions:

- Problems by Algomaniacs

Number Theory :-

Expected Time: 2-3 days

Another important part of CP is number theory, and recently questions based on maths and number theory have become quite frequent. A basic grasp on these topics will be very useful.

Topics:

- Calculating GCD
  - CPH, page 200-201
  - Euclidean algorithm (Ignore Binary GCD)
  - **Extended Euclidean Algorithm*
- Prime numbers and their tests
  - CPH, page 199
  - Primality test: Trial Division
  - **Primality Tests*
- Pre-processing primes till integer N
  - CPH, page 201
  - Sieve of Eratosthenes (Ignore Segmented Sieve)
- Integer Factorization
  - CPH, page 197-199
  - **Integer factorization*

** Optional topics. Study them only if you have time.*

Modular Arithmetic :-

An essential part of CP is working with remainders of integers with a particular number, instead of working with integers themselves. This is done to prevent integer overflows in built-in data types.

A complete introduction to modular arithmetic along with sufficient practice problems can be found here :

- CPH, page 201-203
- Modular Arithmetic
- Binary Exponentiation
- Modular Inverse

Some Practice Problems :-

- Problem 1
- Problem 2
- Problem 3
- Problem 4
- Problem 5
- Problem 6
- Problem 7
- Problem 8

Additional Questions:-

- Problem 1
- Problem 2
- Problem 3