

# Mastering Algorithms Resources

## 1. Basic C++ STL and time complexity

- Resources

- [cppreference.com](http://cppreference.com)
- [STL Containers - C++ Reference](#)
- [STL for CP](#) 🧑🏫
- [Standard Template Library](#)
- [Power up C++ with the Standard Template Library: Part 1](#)
- [Power up C++ with the Standard Template Library: Part 2](#)
- [C++ intro](#) 🧑🏫
- [Intro to codeforces](#)
- [STL](#) 🧑🏫
- [Time complexity](#) 🧑🏫
- [Time complexity guide for contests](#)
- [Intro to CP](#) 🧑🏫

- Problems

- <https://codeforces.com/problemset/problem/855/B>
- <https://codeforces.com/problemset/problem/740/B>
- <https://leetcode.com/problems/maximal-rectangle/description/>

### Stacks

- <https://www.geeksforgeeks.org/problems/implement-two-stacks-in-an-array/1> -> two stacks implementation
- <https://leetcode.com/problems/valid-parentheses/description/>
- <https://leetcode.com/problems/largest-rectangle-in-histogram/description/>
- <https://codeforces.com/contest/548/problem/D>
- <https://leetcode.com/problems/sum-of-subarray-minimums/description/>
- <https://codeforces.com/contest/797/problem/C>

### QUEUES

- <https://leetcode.com/problems/implement-queue-using-stacks/description/>
- <https://www.geeksforgeeks.org/problems/queue-reversal/1>
- <https://leetcode.com/problems/sliding-window-maximum/description/>
- <https://leetcode.com/problems/shortest-subarray-with-sum-at-least-k/description/>
- <https://leetcode.com/problems/longest-continuous-subarray-with-absolute-diff-less-than-or-equal-to-limit/description/>

## 2. Mathematics for CP

**Lecture link** - <https://colab.research.google.com/drive/13dRKA2SYS89yRAQFMtzKICXvkYXgDANs?usp=sharing>

### Number Theory.

#### a. Modulus arithmetic - basic postulates

- Suggested reading

1. Chapter 1 from Number Theory for Computing by SY Yan [ Recommended ]
2. 31.1, 31.3, and 31.4 from Cormen [optional]
3. [www.topcoder.com/tc?module=Static&d1=tutorials&d2=primeNumbers](http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=primeNumbers)

- Problems

1. <http://projecteuler.net/index.php?section=problems&id=64>
2. <http://projecteuler.net/index.php?section=problems&id=65>

#### b. Fermat's theorem, Euler's Totient theorem ( totient function, order, primitive roots )

- Suggested Reading

1. 1.6, 2.2 from Number Theory by SY Yan
2. 31.6 , 31.7 from Cormen

- Problems

1. <http://projecteuler.net/index.php?section=problems&id=70>
2. <http://www.spoj.pl/problems/NDIVPHI/>

#### c. Chinese remainder theorem

- Suggested Reading

1. 1.6 from Number Theory by SY Yan
  - Problems
    1. Project Euler 271
    2. [http://www.topcoder.com/stat?c=problem\\_statement&pm=10551&rd=13903](http://www.topcoder.com/stat?c=problem_statement&pm=10551&rd=13903)
- d. Primality tests -
  - Deterministic  $O(\sqrt{n})$  approach
- e. Prime generation techniques - Sieve of Eratosthenes
  - Suggested Problems - PRIME1 on SPOJ
- f. Integer Factorization
  - Naive  $O(\sqrt{n})$  method
  - Pollard Rho factorization
  - Problems -
    1. [http://www.topcoder.com/stat?c=problem\\_statement&pm=2986&rd=5862](http://www.topcoder.com/stat?c=problem_statement&pm=2986&rd=5862)
    2. <http://www.spoj.pl/problems/DIVSUM2/>
    3. [http://www.topcoder.com/stat?c=problem\\_statement&pm=4481&rd=6538](http://www.topcoder.com/stat?c=problem_statement&pm=4481&rd=6538)
- g. Stirling numbers
- h. Wilson theorem
  - $nCr \% p$  in  $O(p)$  preprocess and  $O(\log n)$  query
- i. Lucas Theorem
- j. Suggested Reading for Number Theory -
  - Number Theory for Computing by Song Y Yan
  - Concepts are also superficially covered in Chapter 31 of Introduction to Algorithms by Cormen
  - <http://www.codechef.com/wiki/tutorial-number-theory>
  - [http://www.algorithmist.com/index.php/Category:Number\\_Theory](http://www.algorithmist.com/index.php/Category:Number_Theory)
- k. Problems on Number Theory -
  - [http://www.algorithmist.com/index.php/Category:Number\\_Theory](http://www.algorithmist.com/index.php/Category:Number_Theory)

## Bit manipulation, Combinatorics, and Game theory [optional]

- Resources

- [Bit manipulation](#) 👤
- [complete playlist for bit manipulation](#) 👤
- [Combinatorics](#)
- [Problems discussion](#) 👤
- [Intro to Game Theory](#) 👤

- Problems

### Bit manipulation

- <https://codeforces.com/problemset/problem/1567/B>
- <https://codeforces.com/problemset/problem/1514/B>
- <https://codeforces.com/contest/1879/problem/D>

### Combinatorics and Game Theory

- Basic principles - Pigeon hole principle, addition, multiplication rules
  1. Suggested problems
    - a. <http://acm.timus.ru/problem.aspx?space=1&num=1690>
    - b. [http://www.topcoder.com/stat?c=problem\\_statement&pm=10805](http://www.topcoder.com/stat?c=problem_statement&pm=10805)
  3. Suggested readings
    - a. [http://en.wikipedia.org/wiki/Combinatorial\\_principles](http://en.wikipedia.org/wiki/Combinatorial_principles)
    - b. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=combinatorics>
    - c. <http://www.maa.org/editorial/knot/pigeonhole.html>
- Inclusion-exclusion
  1. Suggested readings
    - a. [http://en.wikipedia.org/wiki/Inclusion-exclusion\\_principle](http://en.wikipedia.org/wiki/Inclusion-exclusion_principle)
  2. Suggested problems
    - a. [http://www.topcoder.com/stat?c=problem\\_statement&pm=4463&rd=6536](http://www.topcoder.com/stat?c=problem_statement&pm=4463&rd=6536)
    - b. [http://www.topcoder.com/stat?c=problem\\_statement&pm=10238](http://www.topcoder.com/stat?c=problem_statement&pm=10238)
- Basic Principles and Nim game[optional]
  1. Sprague grundy theorem, grundy numbers
  2. Suggested readings
    - a. [http://en.wikipedia.org/wiki/Sprague%E2%80%93Grundy\\_theorem](http://en.wikipedia.org/wiki/Sprague%E2%80%93Grundy_theorem)
    - b. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=algorithmGames>
    - c. <http://www.ams.org/samplings/feature-column/fcarc-games1>
    - d. <http://www.codechef.com/wiki/tutorial-game-theory>
  3. Suggested problems
    - a. <https://codeforces.com/contest/1965/problem/A>

- b. <https://cses.fi/problemset/task/2207>

### 3. Searching, Sorting, Divide and Conquer

- Problems

<https://codeforces.com/problemset/problem/1843/A>  
<https://codeforces.com/contest/456/problem/A>  
<https://codeforces.com/contest/492/problem/B>  
<https://codeforces.com/contest/755/problem/B>  
<https://codeforces.com/contest/1260/problem/B>  
<https://codeforces.com/problemset/problem/978/C>

[Rotated Array | Interviewbit](#)  
[Search for a Range | Interviewbit](#)  
[Allocate Books | Interviewbit](#)

[Median of two Sorted Arrays of Different Sizes - GeeksforGeeks](#)  
[Inversion count in an Array](#)  
[Binary Search Blog](#)

### 4.Binary Heaps

- References

<https://www.geeksforgeeks.org/priority-queue-in-cpp-stl/>

<https://www.geeksforgeeks.org/building-heap-from-array/>

<https://leetcode.com/discuss/general-discussion/1127238/master-heap-by-solving-23-questions-in-4-patterns-category>  
(optional)

- Questions

<https://leetcode.com/problems/kth-largest-element-in-an-array/description/>

<https://leetcode.com/problems/ugly-number-ii/description/> (can also be solved using dynamic programming once taught)

<https://leetcode.com/problems/design-twitter/description/>

<https://codeforces.com/contest/681/problem/C> (Hard to do without knowing greedy approach which will be taught later)