Introductory Tutorials For Competitive Programming

Pre-requisite: Knowledge and understanding of basic constructs of a programming language

Mathematics

1. Basic Number theory
   - Motivation and concept of modulo(mod)
   - Addition, subtraction and multiplication under modulo
   - Prime numbers(Some important properties)
   - Euclid's GCD algorithm
   - Appendix: Division under modulo:
2. Sets
   - Venn diagrams
   - Set builder notations
   - subsets
   - Union, intersections, difference, membership
   - Set equality
   - Empty, Universal sets
   - Types of numbers and their symbols
3. Relations and Functions
   - Relations: Intuitive idea - member of set A is related to member of set B
   - Relations: Bigger picture - subsets of cartesian products
   - Concept of domain, co-domain of a relation
   - Direction in a relation
   - No pre-image can have more than one image
   - Concept of functions
   - Concept of domain, range of a function
   - Functions: Dependent and Independent variables
4. Intro to combinatorics
   - Principle of counting
   - Selection
   - Permutation
   - Factorials
   - Combinations
5. Logarithms
   - Idea of what is exactly logarithm of a number.
   - Properties.
   - Examples
6. Complexity Analysis
   - Notions of efficiency
   - How to measure efficiency - express runtime as a function of input size

- Comparison of functions
- Big-O notation

Language Constructs - C++ - 1

1. Introduction to C++
   - What is C++?
   - Why do we need another language?
   - Basic Structure of a C++ program

2. Data types
   - Recap of data types in Python
   - Declaring and initializing variables of different data types in C++.
   - Primitives datatypes in C++
     a. int
     b. long long
     c. float
     d. double
     e. char
     f. Bool
   - Overflow of values
   - Arrays
     a. Motivation
     b. Arrays in C++ including implementation
     c. Difference between python lists and C++ array

3. Control flow
   - Decision Making statements
     a. if
     b. else if
     c. else
   - and, or, not operators
   - Loops
     a. for
     b. while

4. I / O
   - Introduction(Motivation)
   - Concept of Streams
   - Input using cin
   - Input using getline
   - Output using cout

5. Functions
   - return types
   - Parameters
     a. By value
     b. By reference

6. Recursion

7. Notion of Structs
   - Basic Idea of an user defined data type.
   - Motivational Problem
   - Implementation in C++

1. Searching
    - Motivational Problem
    - Linear Search
    - Binary Search
        a. Algorithm
        b. Complexity
        c. Code
    - Think about implementing your own lower_bound().

2. Sorting
    - Insertion sort
        a. Algorithm
        b. Visualization
        c. Code
        d. Complexity Analysis
    - Merge sort
        a. Merging two small sorted arrays to get another sorted array(Algorithm).
        b. The idea of Divide and Conquer
        c. Algorithm
        d. Visualization
        e. Code
        f. Complexity Analysis

Language Constructs - C++ 2

1. STL - Containers(stack + queue + priority queue)
    - Stack(Motivational Problem, some important functions, LIFO)
    - Queues(Motivational Problem, some important functions, FIFO)
    - Priority Queues(Motivational Problem, some important functions)

2. STL - Containers(string + vector + pair)
    - Strings(Basic idea, some important functions)
    - Vector(Basic idea, some important functions)
    - Pair(Basic idea, some important functions)
    - Appendix: Iterators

3. STL - Algorithms + Set + Map
    - sort
    - upper_bound
    - lower_bound
    - set(Basic idea, some important functions of set container)
    - map(Basic idea, some important functions of map container)

4. Structs - Using them as comparators
    - Basic Idea of comparators
    - Implementation in C++.

Algorithms And Data Structures - 2

1. Dynamic Programming(DP)
    - Concept of Subproblem of a problem and overlapping subproblems.
    - DP as Recursion+Memoization
    - Linear DPs
    - Multi-dimensional DP

2. Graphs and Graph Algorithms
    - Modelling problems as graphs

   a. Undirected Graphs
   b. Directed Graphs

- Representation of graphs
   a. Adjacency Matrix
   b. Adjacency list
- Depth First Search(Algorithm, Code, Visualization)
- Breadth First Search(Algorithm, Code, Visualization)
- (Optional) Shortest Path Algorithms(Algorithms, Codes, Visualizations)

3. Sieves
   - Sieve of Eratosthenes

Credits: We would like to thank Swetanjal Dutta, Parth Mittal and Kushagra Arora from the community for preparing these tutorials. In case you would like to contribute in any kind of content, or can provide improvements in the existing one, you are most welcome. Feel free to get in touch with us on schools@codechef.com