

**COURSES****HIRE WITH US** 

## How to prepare for ACM – ICPC?

ACM ICPC(**A**ssociation for **C**omputing **M**achinery – **I**nternational **C**ollegiate **P**rogramming **C**ontest) is a world-wide annual multi-tiered programming contest being organized for over thirteen years. The contest is sponsored by IBM.

This article focuses on what all topics that are important for the competitive programming and should especially be studied in order to train yourself for upcoming ACM-ICPC contest.

**Rules of the Contest** – World final Rules for 2017 Click [here](#)

**Indian Participants** – Codechef conducts all the Indian Regionals. Click [here](#) to know about team formation, reimbursements etc

**ICPC for Schools by CodeChef** – This competition serves as a gateway for the school students to participate in ACM ICPC contest along with ICPC college participants held across India. It is an idea conceived by CodeChef and supported by Amrita University.

**A sample ICPC Problem** : A usual ICPC problem has the following features:

1. **Problem statement:** describing the problem and what output is to be generated.
2. **Input:** Make sure that you read this section with complete attention as missing out any minor detail may land you in wrong answer zone.
3. **Output:** Just like above, this one also should be read carefully.
4. **Constraints:** These can include constraints on input, time, memory, code size, etc.
5. **Time limit:** See if your algorithm can work in this range. If not, time to change it!

6. **Memory limit:** If you are fond of allocating memory for every small thing, it's a good time that you changed it.

## Preparing for ACM-ICPC

**First and foremost Step: PRACTICE** – Following are the resources that can be referred for practicing the ACM-ICPC alike contests and problems. For all these OJs, begin with the problems with maximum submissions and check other solutions to check how you may improve. Do Participate in their monthly contests to remain up to the mark.

- ACM-ICPC Past Problems – [ICPC Archive](#), Practice at [Codechef](#)
- [TopCoder](#) – Proceed by increasing problem levels gradually
- [Codeforces](#) -List of [Problem Sets](#)
- [Codechef](#) – Beginners can start with [Codechef Beginners](#) and proceed further
- [SPOJ](#) – Move from easy to tough problems
- [USACO](#) – Excellent training resource
- [uvaOnline Judge](#) – Huge repository of problems
- [Hackerrank](#) – Practice problems topic wise and participate in preparatory series
- [Hackerearth](#) – Participate in preparatory series
- [Practice](#) – Good for beginners. Has problems ranging from difficulty level School to Hard.
- [List](#) of various Competitive Programming Contests available online all through the year

## What to study?

Knowing just the basics of programming won't be fruitful for aspirants of ACM ICPC. One needs to have a thorough knowledge of advanced algorithms used as well. Following Topics list out the necessary Topics and Algorithms that one must surely know to improve and stand a chance in the actual competition.



**Elementary data structures:** To begin with competitive programming, one must master the Data Structures. Following is the list of most commonly used data structures:

- [Array](#)
- [Stack](#)
- [Queue](#)
- [String](#)
- [Heap](#)

- Hash
- Extensive list of Data structures

### Advanced Data Structures

Priority queues, union-find sets, (augmented) interval trees, (augmented) balanced BSTs and binary indexed trees

- Binary Indexed Tree or Fenwick tree
- Segment Tree (RMQ, Range Sum and Lazy Propagation)
- K-D tree (See insert, minimum and delete)
- Union Find Disjoint Set (Cycle Detection and By Rank and Path Compression)
- Tries
- Interval Tree

More Advanced Data Structures.

**Sorting and Searching** : Concentrate to learn the basic concepts and also get familiar with all the library functions available.

- Binary Search
- Quick Sort
- Merge Sort
- Order Statistics

**String manipulation** : Strings make programming problems interesting and difficult too and probably that's the reason they are used extensively in such contests. Learning library functions for String actually proves very helpful (C++ : See [this](#) and [this](#), [String in Java](#)).

- KMP algorithm
- Rabin karp
- Z's algorithm
- Aho Corasick String Matching

**Choosing the right Language** : C++ is till date most preferred language followed by Java when it comes to programming contests but you should always choose a language you are comfortable with. Being CONFIDENT in any language is most important.

**Standard Template Library** : A quintessential especially for those using C++ as a language for coding

- Power up C++ STL by Topcoder – [Part 1](#), [Part 2](#)
- [C++ Magicians – STL Algorithms](#)

## Dynamic Programming

- [Longest Common Subsequence](#)
- [Longest Increasing Subsequence](#)
- [Edit Distance](#)
- [Minimum Partition](#)
- [Ways to Cover a Distance](#)
- [Longest Path In Matrix](#)
- [Subset Sum Problem](#)
- [Optimal Strategy for a Game](#)
- [0-1 Knapsack Problem](#)
- [Assembly Line Scheduling](#)
- [Optimal Binary Search Tree](#)

## All DP Algorithms

## BackTracking

- [Rat in a Maze](#)
- [N Queen Problem](#)
- [Subset Sum](#)
- [m Coloring Problem](#)
- [Hamiltonian Cycle](#)

## More articles on Backtracking

## Greedy Algorithms

- [Activity Selection Problem](#)
- [Kruskal's Minimum Spanning Tree Algorithm](#)
- [Huffman Coding](#)
- [Efficient Huffman Coding for Sorted Input](#)
- [Prim's Minimum Spanning Tree Algorithm](#)

## More articles on Greedy Algorithms

**Graph Algorithms** : One of the most important topic which you can not ignore if preparing for ACM – ICPC.

- Breadth First Search (BFS)
- Depth First Search (DFS)
- Shortest Path from source to all vertices **\*\*Dijkstra\*\***
- Shortest Path from every vertex to every other vertex **\*\*Floyd Warshall\*\***
- Minimum Spanning tree **\*\*Prim\*\***
- Minimum Spanning tree **\*\*Kruskal\*\***
- Topological Sort
- Johnson's algorithm
- Articulation Points (or Cut Vertices) in a Graph
- Bridges in a graph

## All Graph Algorithms

### Basic Mathematics

**Arithmetic** : Programmers must know how integers and real numbers are represented internally and should be able to code high-precision numbers. Bit manipulation tricks and knowing library functions for number basic arithmetic would be very helpful.

**Number theory** : Knowing some of these concepts would save a lot of time and efforts while programming in the contests.

- Modular Exponentiation
- Modular multiplicative inverse
- Primality Test | Set 2 (Fermat Method)
- Euler's Totient Function
- Sieve of Eratosthenes
- Convex Hull
- Basic and Extended Euclidean algorithms
- Segmented Sieve
- Chinese remainder theorem
- Lucas Theorem

**Combinatorics** : Although directly might not seem to be important, Combinatorics is important to estimate asymptotic complexity of algorithms.

- Analysis of Algorithms
- Combinatorial Game Theory | Set 1 (Introduction)

## Geometrical Algorithms

- Convex Hull
- Graham Scan
- Line Intersection
- Matrix Exponentiation and this
- Online construction of 3-D convex hull
- Bentley Ottmann algorithm to list all intersection points of n line segments
- Rotating Calipers Technique
- Area/Perimeter of Union of Rectangles
- Closest pair of points
- Area of Union of Circles
- Delaunay Triangulation of n points
- Voronoi Diagrams of n points using Fortune's algorithm
- Point in a polygon problem

## Network Flow Algorithms

- Maxflow Ford Fulkerson Algo and Edmond Karp Implementation
- Min cut
- Stable Marriage Problem
- Dinic's Algorithm for Maximum Flow and Wiki
- Minimum Cost Flow Problem
- Successive Shortest path Algorithm
- Cycle Cancelling algorithm
- Maximum weighted Bipartite Matching (Kuhn Munkres algorithm/Hungarian Method)
  - Hungarian Algorithm Wiki
  - Hungarian Algorithm for Assignment Problem
  - Maximum Bipartite Matching
- Stoer Wagner min-cut algorithm
- Maximum matching in general graph (Blossom Shrinking)
- Gomory-Hu Trees
- Chinese Postman problem(Please see this too)
- Hopcroft-Karp Algorithm for Maximum Matching

All Articles on Geometric Algorithms

## More Advanced Stuff

Bit Algorithms , Randomized Algorithms , Branch and Bound , Mathematical Algorithms , Heavy Light Decomposition, A\* Search

### Informative Articles that you may like to read

- [An Awesome list for Competitive Programming – Codeforces](#)
- [What are the algorithms required to solve all C++ problems in Contests ? – Quora](#)
- [Best books and sites to prepare for ACM-ICPC – Quora](#)
- [Introduction to Programming Contest – Stanford.edu](#)
- [World Final Problems of ACM-ICPC – icpc.kattis](#)

### References:

[Programming Camp Syllabus](#)

This article is contributed by **Vishwesh Shrimali in association with Team GeeksforGeeks**. If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to [contribute@geeksforgeeks.org](mailto:contribute@geeksforgeeks.org). See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, missing or you want to share more information about the topic discussed above.

### Recommended Posts:

[How to prepare for Facebook Hacker Cup?](#)

[How to prepare for Google Asia Pacific University \(APAC\) Test ?](#)

[Sort an array without changing position of negative numbers](#)

[Flipping Sign Problem | Lazy Propagation Segment Tree](#)

[Cartesian tree from inorder traversal | Segment Tree](#)

[Find two vertices of an isosceles triangle in which there is rectangle with opposite corners \(0, 0\) and \(X, Y\)](#)

[Sum of Semi-Prime Numbers less than or equal to N](#)

[Optimally accommodate 0s and 1s from a Binary String into K buckets](#)

[Find two disjoint good sets of vertices in a given graph](#)

[Rearrange the given string such that all prime multiple indexes have same character](#)

[Find a number containing N - 1 set bits at even positions from the right](#)

[Find the minimum number of elements that should be removed to make an array good](#)

Number of elements from the array which are reachable after performing given operations on D

Print first N Mosaic numbers

**Article Tags :** Competitive Programming ACM-ICPC Binary Indexed Tree number-theory

Segment-Tree sieve

**Practice Tags :** number-theory sieve Binary Indexed Tree Segment-Tree



22

4.5

☐ To-do ☐ Done

Based on 111 vote(s)

Please write to us at [contribute@geeksforgeeks.org](mailto:contribute@geeksforgeeks.org) to report any issue with the above content.

Writing code in comment? Please use [ide.geeksforgeeks.org](https://ide.geeksforgeeks.org), generate link and share the link here.

Load Comments



A computer science portal for geeks

5th Floor, A-118,  
Sector-136, Noida, Uttar Pradesh - 201305  
[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)

#### **COMPANY**

About Us  
Careers  
Privacy Policy  
Contact Us

#### **PRACTICE**

Courses  
Company-wise  
Topic-wise  
How to begin?

#### **LEARN**

Algorithms  
Data Structures  
Languages  
CS Subjects  
Video Tutorials

#### **CONTRIBUTE**

Write an Article  
Write Interview Experience  
Internships  
Videos

@geeksforgeeks, Some rights reserved