

# Top 10 Algorithms and Data Structures for Competitive Programming

Difficulty Level : Medium • Last Updated : 20 Sep, 2021

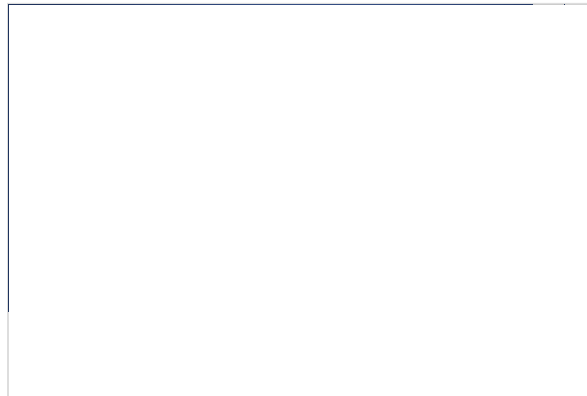
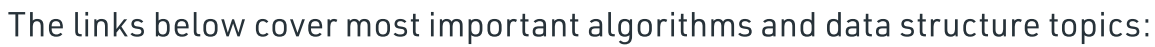
In this post, we will discuss Important top 10 algorithms and data structures for competitive coding.

## Topics :



1. [Graph algorithms](#)
2. [Dynamic programming](#)
3. [Searching and Sorting:](#)
4. [Number theory and Other Mathematical](#)
5. [Geometrical and Network Flow Algorithms](#)
6. [Data Structures](#)





1. Breadth First Search (BFS)
2. Depth First Search (DFS)
3. Shortest Path from source to all vertices \*\*Dijkstra\*\*
4. Shortest Path from every vertex to every other vertex \*\*Floyd Warshall\*\*
5. Minimum Spanning tree \*\*Prim\*\*
6. Minimum Spanning tree \*\*Kruskal\*\*
7. Topological Sort
8. Johnson's algorithm
9. Articulation Points (or Cut Vertices) in a Graph
10. Bridges in a graph

## [All Graph Algorithms](#)

### Dynamic Programming

1. [Longest Common Subsequence](#)
2. [Longest Increasing Subsequence](#)
3. [Edit Distance](#)
4. [Minimum Partition](#)
5. [Ways to Cover a Distance](#)
6. [Longest Path In Matrix](#)
7. [Subset Sum Problem](#)
8. [Optimal Strategy for a Game](#)
9. [0-1 Knapsack Problem](#)
10. [Assembly Line Scheduling](#)

## [All DP Algorithms](#)

### Searching And Sorting

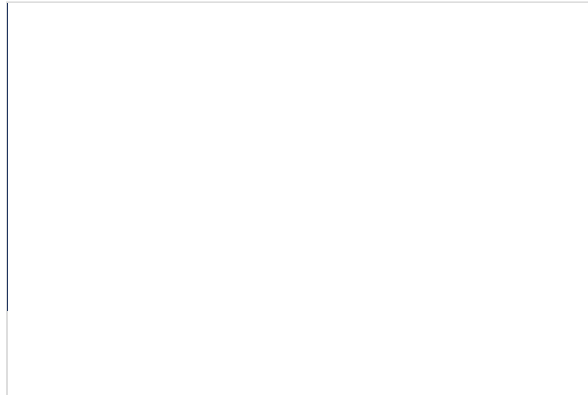
1. [Binary Search](#)
2. [Quick Sort](#)
3. [Merge Sort](#)
4. [Order Statistics](#)
5. [KMP algorithm](#)
6. [Rabin karp](#)
7. [Z's algorithm](#)
8. [Aho Corasick String Matching](#)
9. [Counting Sort](#)
10. Manacher's algorithm: [Part 1](#), [Part 2](#) and [Part 3](#)

All Articles on [Searching](#), [Sorting](#) and [Pattern Searching](#).

### Number theory and Other Mathematical

#### Prime Numbers and Prime Factorization





1. [Primality Test | Set 1 \(Introduction and School Method\)](#)
2. [Primality Test | Set 2 \(Fermat Method\)](#)
3. [Primality Test | Set 3 \(Miller-Rabin\)](#)
4. [Sieve of Eratosthenes](#)
5. [Segmented Sieve](#)
6. [Wilson's Theorem](#)
7. [Prime Factorization](#)
8. [Pollard's rho algorithm](#)

### Modulo Arithmetic Algorithms

1. [Basic and Extended Euclidean algorithms](#)
2. [Euler's Totient Function](#)
3. [Modular Exponentiation](#)
4. [Modular Multiplicative Inverse](#)
5. [Chinese remainder theorem Introduction](#)
6. [Chinese remainder theorem and Modulo Inverse Implementation](#)
7. [nCr%m](#) and [this](#).

### Miscellaneous:

1. [Counting Inversions](#)
2. [Counting Inversions using BIT](#)
3. [logarithmic exponentiation](#)
4. [Square root of an integer](#)
5. [Heavy light Decomposition](#), [this](#) and [this](#)
6. [Matrix Rank](#)
7. [Gaussian Elimination to Solve Linear Equations](#)
8. [Hungarian algorithm](#)



9. [Link cut](#)
10. [Mo's algorithm](#) and [this](#)
11. [Factorial of a large number in C++](#)
12. [Factorial of a large number in Java+](#)
13. [Russian Peasant Multiplication](#)
14. [Catalan Number](#)

### [All Articles on Mathematical Algorithms](#)

#### **Geometrical and Network Flow Algorithms**

1. [Convex Hull](#)
2. [Graham Scan](#)
3. [Line Intersection](#)
4. [Interval Tree](#)
5. [Matrix Exponentiation](#) and [this](#)
6. [Maxflow Ford Furkerson Algo and Edmond Karp Implementation](#)
7. [Min cut](#)
8. [Stable Marriage Problem](#)
9. [Hopcroft-Karp Algorithm for Maximum Matching](#)
10. [Dinic's algo](#) and [e-maxx](#)

### [All Articles on Geometric Algorithms](#)

#### **Data Structures**

1. [Binary Indexed Tree or Fenwick tree](#)
2. [Segment Tree](#) ([RMQ](#), [Range Sum](#) and [Lazy Propagation](#))
3. [K-D tree](#) (See [insert](#), [minimum](#) and [delete](#))
4. [Union Find Disjoint Set](#) ([Cycle Detection](#) and [By Rank and Path Compression](#))
5. [Tries](#)
6. [Suffix array](#) ([this](#), [this](#) and [this](#))
7. [Sparse table](#)
8. [Suffix automata](#)
9. [Suffix automata II](#)
10. [LCA and RMQ](#)

### [All Articles on Advanced Data Structures.](#)

#### **How to Begin?**

Please see [How to begin with Competitive Programming?](#)

## How to Practice?

Please see <https://practice.geeksforgeeks.org/>

## What are top algorithms in Interview Questions?

[Top 10 algorithms in Interview Questions](#)

## How to prepare for ACM – ICPC?

[How to prepare for ACM – ICPC?](#)

This is an initial draft. We will soon be adding more links and algorithms to this post. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Like 0

Previous

Next



RECOMMENDED ARTICLES

Page : 1 2 3

- 01

**Tips and Tricks for Competitive Programmers | Set 2 (Language to be used for Competitive Programming)**  
20, Mar 16
- 02

**How can one become good at Data structures and Algorithms easily?**  
13, Dec 18
- 03

**Input/Output from external file in C/C++, Java and Python for Competitive Programming | Set 2**  
19, May 17
- 04

**What is Competitive Programming and How to Prepare for It?**  
07, Oct 19
- 05

**Competitive Programming - Myths and Mind Blowing Facts**  
01, Nov 20
- 06

**Creation and Utilisation of Custom Snippets for Competitive programming in VScode**  
09, Mar 21
- 07

**Graph implementation using STL for competitive programming | Set 1 (DFS of Unweighted and Undirected)**  
19, Jan 17
- 08

**Input/Output from external file in C/C++, Java and Python for Competitive Programming**  
26, Feb 17

### Article Contributed By :



GeeksforGeeks

### Vote for difficulty

Current difficulty : [Medium](#)

Easy	Normal	Medium	Hard	Expert
------	--------	--------	------	--------

Improved By : [suhailmahmood](#), [chhabradhanvi](#)

Article Tags : [BFS](#), [Binary Indexed Tree](#), [BIT](#), [DFS](#), [Modular Arithmetic](#), [number-theory](#), [Segment-Tree](#), [sieve](#), [Topological Sorting](#), [Competitive Programming](#)

Practice Tags : [number-theory](#), [DFS](#), [sieve](#), [Binary Indexed Tree](#), [Modular Arithmetic](#), [Segment-Tree](#), [BFS](#)

[Improve Article](#)[Report Issue](#)

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

[Load Comments](#)

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](#)  
[Copyright Policy](#)

### Learn

[Algorithms](#)  
[Data Structures](#)  
[Languages](#)  
[CS Subjects](#)  
[Video Tutorials](#)

### Web Development

[Web Tutorials](#)  
[HTML](#)  
[CSS](#)  
[JavaScript](#)  
[Bootstrap](#)

### Contribute

[Write an Article](#)  
[Write Interview Experience](#)  
[Internships](#)  
[Videos](#)

@geeksforgeeks , Some rights reserved

