



Last Updated : 10 Sep, 2024

Topics :

- [illegible]

Graph Algorithms

- <https://www.geeksforgeeks.org/top-algorithms-and-data-structures-for-competitive-programming/>

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 Fulkerson 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.