**Graph Algorithms**

1. [Breadth First Search (BFS)](http://www.geeksforgeeks.org/breadth-first-traversal-for-a-graph/)
2. [Depth First Search (DFS)](http://www.geeksforgeeks.org/depth-first-traversal-for-a-graph/)
3. [Shortest Path from source to all vertices \*\*Dijkstra\*\*](http://www.geeksforgeeks.org/greedy-algorithms-set-6-dijkstras-shortest-path-algorithm/)
4. [Shortest Path from every vertex to every other vertex \*\*Floyd Warshall\*\*](http://www.geeksforgeeks.org/dynamic-programming-set-16-floyd-warshall-algorithm/)
5. [Minimum Spanning tree \*\*Prim\*\*](http://www.geeksforgeeks.org/greedy-algorithms-set-5-prims-minimum-spanning-tree-mst-2/)
6. [Minimum Spanning tree \*\*Kruskal\*\*](http://www.geeksforgeeks.org/greedy-algorithms-set-2-kruskals-minimum-spanning-tree-mst/)
7. [Topological Sort](http://www.geeksforgeeks.org/topological-sorting/)
8. [Johnson’s algorithm](http://www.geeksforgeeks.org/johnsons-algorithm/)
9. [Articulation Points (or Cut Vertices) in a Graph](http://www.geeksforgeeks.org/articulation-points-or-cut-vertices-in-a-graph/)
10. [Bridges in a graph](http://www.geeksforgeeks.org/bridge-in-a-graph/)

[All Graph Algorithms](http://www.geeksforgeeks.org/category/graph/)

**Dynamic Programming**

1. [Longest Common Subsequence](http://www.geeksforgeeks.org/dynamic-programming-set-4-longest-common-subsequence/)
2. [Longest Increasing Subsequence](http://www.geeksforgeeks.org/dynamic-programming-set-3-longest-increasing-subsequence/)
3. [Edit Distance](http://www.geeksforgeeks.org/dynamic-programming-set-5-edit-distance/)
4. [Minimum Partition](http://www.geeksforgeeks.org/partition-a-set-into-two-subsets-such-that-the-difference-of-subset-sums-is-minimum/)
5. [Ways to Cover a Distance](http://www.geeksforgeeks.org/count-number-of-ways-to-cover-a-distance/)
6. [Longest Path In Matrix](http://www.geeksforgeeks.org/find-the-longest-path-in-a-matrix-with-given-constraints/)
7. [Subset Sum Problem](http://www.geeksforgeeks.org/dynamic-programming-subset-sum-problem/)
8. [Optimal Strategy for a Game](http://www.geeksforgeeks.org/dynamic-programming-set-31-optimal-strategy-for-a-game/)
9. [0-1 Knapsack Problem](http://www.geeksforgeeks.org/dynamic-programming-set-10-0-1-knapsack-problem/)
10. [Assembly Line Scheduling](http://www.geeksforgeeks.org/dynamic-programming-set-34-assembly-line-scheduling/)

**Searching And Sorting**

1. [Binary Search](http://geeksquiz.com/binary-search/)
2. [Quick Sort](http://geeksquiz.com/quick-sort/)
3. [Merge Sort](http://geeksquiz.com/merge-sort/)
4. [Order Statistics](http://www.geeksforgeeks.org/kth-smallestlargest-element-unsorted-array-set-2-expected-linear-time/)
5. [KMP algorithm](http://www.geeksforgeeks.org/searching-for-patterns-set-2-kmp-algorithm/)
6. [Rabin karp](http://www.geeksforgeeks.org/searching-for-patterns-set-3-rabin-karp-algorithm/)
7. [Z’s algorithm](http://www.geeksforgeeks.org/z-algorithm-linear-time-pattern-searching-algorithm/)
8. [Aho Corasick String Matching](http://www.geeksforgeeks.org/aho-corasick-algorithm-pattern-searching/)
9. [Counting Sort](http://www.geeksforgeeks.org/counting-sort/)
10. Manacher’s algorithm: [Part 1](http://www.geeksforgeeks.org/manachers-algorithm-linear-time-longest-palindromic-substring-part-1/), [Part 2](http://www.geeksforgeeks.org/manachers-algorithm-linear-time-longest-palindromic-substring-part-2/) and [Part 3](http://www.geeksforgeeks.org/manachers-algorithm-linear-time-longest-palindromic-substring-part-3-2/)

**Number theory and Other Mathematical**

**Prime Numbers and Prime Factorization**

1. [Primality Test | Set 1 (Introduction and School Method)](http://www.geeksforgeeks.org/primality-test-set-1-introduction-and-school-method/)
2. [Primality Test | Set 2 (Fermat Method)](http://www.geeksforgeeks.org/primality-test-set-2-fermet-method/)
3. [Primality Test | Set 3 (Miller–Rabin)](http://www.geeksforgeeks.org/primality-test-set-3-miller-rabin/)
4. [Sieve of Eratosthenes](http://www.geeksforgeeks.org/sieve-of-eratosthenes/)
5. [Segmented Sieve](http://www.geeksforgeeks.org/segmented-sieve/)
6. [Wilson’s Theorem](http://www.geeksforgeeks.org/wilsons-theorem/)
7. [Prime Factorisation](http://www.geeksforgeeks.org/print-all-prime-factors-of-a-given-number/)
8. [Pollard’s rho algorithm](http://www.geeksforgeeks.org/pollards-rho-algorithm-prime-factorization/)

**Modulo Arithmetic Algorithms**

1. [Basic and Extended Euclidean algorithms](http://www.geeksforgeeks.org/basic-and-extended-euclidean-algorithms/)
2. [Euler’s Totient Function](http://www.geeksforgeeks.org/eulers-totient-function/)
3. [Modular Exponentiation](http://www.geeksforgeeks.org/modular-exponentiation-power-in-modular-arithmetic/)
4. [Modular Multiplicative Inverse](http://www.geeksforgeeks.org/multiplicative-inverse-under-modulo-m/)
5. [Chinese remainder theorem Introduction](http://www.geeksforgeeks.org/chinese-remainder-theorem-set-1-introduction/)
6. [Chinese remainder theorem and Modulo Inverse Implementation](http://www.geeksforgeeks.org/chinese-remainder-theorem-set-2-implementation/)
7. [nCr%m](http://www.geeksforgeeks.org/compute-ncr-p-set-2-lucas-theorem/)

**Miscellaneous:**

1. [Counting Inversions](http://www.geeksforgeeks.org/counting-inversions/)
2. [Counting Inversions using BIT](http://www.geeksforgeeks.org/count-inversions-array-set-3-using-bit/)
3. [logarithmic exponentiation](http://www.geeksforgeeks.org/write-a-c-program-to-calculate-powxn/)
4. [Square root of an integer](http://www.geeksforgeeks.org/square-root-of-an-integer/)
5. [Heavy light Decomposition](http://www.geeksforgeeks.org/heavy-light-decomposition-set-1-introduction/) , [this](http://e-maxx.ru/algo/heavy_light) and [this](http://blog.anudeep2011.com/heavy-light-decomposition/)
6. [Matrix Rank](http://www.geeksforgeeks.org/program-for-rank-of-matrix/)
7. [Gaussian Elimination to Solve Linear Equations](http://www.geeksforgeeks.org/gaussian-elimination/)
8. [Hungarian algorithm](https://en.wikipedia.org/wiki/Hungarian_algorithm)
9. [Link cut](http://www.cs.cmu.edu/~avrim/451f12/lectures/lect1009-linkcut.txt)
10. [Mo’s algorithm](http://www.geeksforgeeks.org/mos-algorithm-query-square-root-decomposition-set-1-introduction/) and [this](http://blog.anudeep2011.com/mos-algorithm/)
11. [Factorial of a large number in C++](http://www.geeksforgeeks.org/factorial-large-number/)
12. [Factorial of a large number in Java+](http://www.geeksforgeeks.org/biginteger-class-in-java/)
13. [Russian Peasant Multiplication](http://www.geeksforgeeks.org/fast-multiplication-method-without-using-multiplication-operator-russian-peasants-algorithm/)
14. [Catalan Number](http://www.geeksforgeeks.org/program-nth-catalan-number/)

**Geometrical and Network Flow Algorithms**

1. [Convex Hull](http://www.geeksforgeeks.org/convex-hull-set-1-jarviss-algorithm-or-wrapping/)
2. [Graham Scan](http://www.geeksforgeeks.org/convex-hull-set-2-graham-scan/)
3. [Line Intersection](http://www.geeksforgeeks.org/check-if-two-given-line-segments-intersect/)
4. [Interval Tree](http://www.geeksforgeeks.org/interval-tree/)
5. [Matrix Exponentiation](http://www.geeksforgeeks.org/matrix-exponentiation/) and [this](http://zobayer.blogspot.in/2010/11/matrix-exponentiation.html)
6. [Maxflow Ford Furkerson Algo and Edmond Karp Implementation](http://www.geeksforgeeks.org/ford-fulkerson-algorithm-for-maximum-flow-problem/)
7. [Min cut](http://www.geeksforgeeks.org/minimum-cut-in-a-directed-graph/)
8. [Stable Marriage Problem](http://www.geeksforgeeks.org/stable-marriage-problem/)
9. [Hopcroft–Karp Algorithm for Maximum Matching](http://www.geeksforgeeks.org/hopcroft-karp-algorithm-for-maximum-matching-set-1-introduction/)
10. [Dinic’s algo](http://www.geeksforgeeks.org/dinics-algorithm-maximum-flow/) and [e-maxx](http://e-maxx.ru/algo/dinic)

**Data Structures**

1. [Binary Indexed Tree or Fenwick tree](http://www.geeksforgeeks.org/binary-indexed-tree-or-fenwick-tree-2/)
2. [Segment Tree](http://www.geeksforgeeks.org/segment-tree-set-1-range-minimum-query/) ([RMQ](http://www.geeksforgeeks.org/segment-tree-set-1-range-minimum-query/), [Range Sum](http://www.geeksforgeeks.org/segment-tree-set-1-sum-of-given-range/) and [Lazy Propagation](http://www.geeksforgeeks.org/lazy-propagation-in-segment-tree/))
3. [K-D tree](http://www.geeksforgeeks.org/k-dimensional-tree/) (See [insert](http://www.geeksforgeeks.org/k-dimensional-tree/), [minimum](http://www.geeksforgeeks.org/k-dimensional-tree-set-2-find-minimum/) and [delete](http://www.geeksforgeeks.org/k-dimensional-tree-set-3-delete/))
4. [Union Find Disjoint Set](http://www.geeksforgeeks.org/union-find/) ([Cycle Detection](http://www.geeksforgeeks.org/union-find-algorithm-set-2-union-by-rank/) and [By Rank and Path Compression](http://www.geeksforgeeks.org/union-find-algorithm-set-2-union-by-rank/))
5. [Tries](http://www.geeksforgeeks.org/trie-insert-and-search/)
6. Suffix array ([this](http://web.stanford.edu/class/cs97si/suffix-array.pdf), [this](http://www.geeksforgeeks.org/suffix-array-set-1-introduction/) and [this](http://www.geeksforgeeks.org/suffix-array-set-2-a-nlognlogn-algorithm/))
7. [Sparse table](http://www.geeksforgeeks.org/range-minimum-query-for-static-array/)
8. [Suffix automata](http://www.geeksforgeeks.org/searching-for-patterns-set-5-finite-automata/)
9. [Suffix automata II](http://www.geeksforgeeks.org/pattern-searching-set-5-efficient-constructtion-of-finite-automata/)
10. [LCA and RMQ](http://www.geeksforgeeks.org/find-lca-in-binary-tree-using-rmq/)