# Graph Data Structure And Algorithms

* [Introduction, DFS and BFS](https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/#introDFSnBFS)
* [Graph Cycle](https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/#graphCycle)
* [Topological Sorting](https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/#topologicalSorting)
* [Minimum Spanning Tree](https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/#minimumSpanningTree)
* [BackTracking](https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/#backtracking)
* [Shortest Paths](https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/#shortestPath)
* [Connectivity](https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/#connectivity)
* [Hard Problems](https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/#hardProblems)
* [Maximum Flow](https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/#maximumFlow)
* [STL Implementation of Algorithms](https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/#STLimplementation)
* [Misc](https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/#misc)

**Introduction, DFS and BFS:**

1. [Graph and its representations](https://www.geeksforgeeks.org/graph-and-its-representations/)
2. [Breadth First Traversal for a Graph](https://www.geeksforgeeks.org/breadth-first-traversal-for-a-graph/)
3. [Depth First Traversal for a Graph](https://www.geeksforgeeks.org/depth-first-traversal-for-a-graph/)
4. [Applications of Depth First Search](https://www.geeksforgeeks.org/applications-of-depth-first-search/)
5. [Applications of Breadth First Traversal](https://www.geeksforgeeks.org/applications-of-breadth-first-traversal/)
6. [Longest Path in a Directed Acyclic Graph](https://www.geeksforgeeks.org/find-longest-path-directed-acyclic-graph/)
7. [Find Mother Vertex in a Graph](https://www.geeksforgeeks.org/find-a-mother-vertex-in-a-graph/)
8. [Transitive Closure of a Graph using DFS](https://www.geeksforgeeks.org/transitive-closure-of-a-graph-using-dfs/)
9. [Find K cores of an undirected Graph](https://www.geeksforgeeks.org/find-k-cores-graph/)
10. [Iterative Depth First Search](https://www.geeksforgeeks.org/iterative-depth-first-traversal/)
11. [Count the number of nodes at given level in a tree using BFS](https://www.geeksforgeeks.org/count-number-nodes-given-level-using-bfs/)
12. [Count all possible paths between two vertices](https://www.geeksforgeeks.org/count-possible-paths-two-vertices/)
13. [Minimum initial vertices to traverse whole matrix with given conditions](https://www.geeksforgeeks.org/minimum-initial-vertices-traverse-whole-matrix-given-conditions/)
14. [Shortest path to reach one prime to other by changing single digit at a time](https://www.geeksforgeeks.org/shortest-path-reach-one-prime-changing-single-digit-time/)
15. [Water Jug problem using BFS](https://www.geeksforgeeks.org/water-jug-problem-using-bfs/)
16. [Magical Indices in an array](https://www.geeksforgeeks.org/magical-indices-array/)
17. [Count number of trees in a forest](https://www.geeksforgeeks.org/count-number-trees-forest/)
18. [BFS using vectors & queue as per the algorithm of CLRS](https://www.geeksforgeeks.org/bfs-using-vectors-queue-per-algorithm-clrs/)
19. [Level of Each node in a Tree from source node](https://www.geeksforgeeks.org/level-node-tree-source-node-using-bfs/)
20. [Construct binary palindrome by repeated appending and trimming](https://www.geeksforgeeks.org/construct-binary-palindrome-by-repeated-appending-and-trimming/)
21. [Iterative Deepening Search(IDS) or Iterative Deepening Depth First Search(IDDFS)](https://www.geeksforgeeks.org/iterative-deepening-searchids-iterative-deepening-depth-first-searchiddfs/)

**Graph Cycle:**

1. [Detect Cycle in a Directed Graph](https://www.geeksforgeeks.org/detect-cycle-in-a-graph/)
2. [Detect Cycle in a an Undirected Graph](https://www.geeksforgeeks.org/union-find/)
3. [Detect cycle in an undirected graph](https://www.geeksforgeeks.org/detect-cycle-undirected-graph/)
4. [Detect cycle in a direct graph using colors](https://www.geeksforgeeks.org/detect-cycle-direct-graph-using-colors/)
5. [Assign directions to edges so that the directed graph remains acyclic](https://www.geeksforgeeks.org/assign-directions-to-edges-so-that-the-directed-graph-remains-acyclic/)
6. [Detect a negative cycle in a Graph | (Bellman Ford)](https://www.geeksforgeeks.org/detect-negative-cycle-graph-bellman-ford/)

**Topological Sorting:**

1. [Topological Sorting](https://www.geeksforgeeks.org/topological-sorting/)
2. [All topological sorts of a Directed Acyclic Graph](https://www.geeksforgeeks.org/all-topological-sorts-of-a-directed-acyclic-graph/)
3. [Kahn’s Algorithm for Topological Sorting](https://www.geeksforgeeks.org/topological-sorting-indegree-based-solution/)

**Minimum Spanning Tree:**

1. [Prim’s Minimum Spanning Tree (MST))](https://www.geeksforgeeks.org/greedy-algorithms-set-5-prims-minimum-spanning-tree-mst-2/)
2. [Applications of Minimum Spanning Tree Problem](https://www.geeksforgeeks.org/applications-of-minimum-spanning-tree/)
3. [Prim’s MST for Adjacency List Representation](https://www.geeksforgeeks.org/greedy-algorithms-set-5-prims-mst-for-adjacency-list-representation/)
4. [Kruskal’s Minimum Spanning Tree Algorithm](https://www.geeksforgeeks.org/greedy-algorithms-set-2-kruskals-minimum-spanning-tree-mst/)
5. [Boruvka’s algorithm for Minimum Spanning Tree](https://www.geeksforgeeks.org/greedy-algorithms-set-9-boruvkas-algorithm/)
6. [Minimum cost to connect all cities](https://www.geeksforgeeks.org/minimum-cost-connect-cities/)
7. [Steiner Tree](https://www.geeksforgeeks.org/steiner-tree/)

***BackTracking***

1. [Find if there is a path of more than k length from a source](https://www.geeksforgeeks.org/find-if-there-is-a-path-of-more-than-k-length-from-a-source/)
2. [Tug of War](https://www.geeksforgeeks.org/tug-of-war/)
3. [The Knight-Tour Problem](https://www.geeksforgeeks.org/backtracking-set-1-the-knights-tour-problem/)
4. [Rat in a Maze](https://www.geeksforgeeks.org/backttracking-set-2-rat-in-a-maze/)
5. [n-Queen’s Problem](https://www.geeksforgeeks.org/backtracking-set-3-n-queen-problem/)
6. [m Coloring Problem](https://www.geeksforgeeks.org/backttracking-set-5-m-coloring-problem/)
7. [Hamiltonian Cycle](https://www.geeksforgeeks.org/backtracking-set-7-hamiltonian-cycle/)

***Shortest Paths:***

1. [Dijkstra’s shortest path algorithm](https://www.geeksforgeeks.org/greedy-algorithms-set-6-dijkstras-shortest-path-algorithm/)
2. [Dijkstra’s Algorithm for Adjacency List Representation](https://www.geeksforgeeks.org/greedy-algorithms-set-7-dijkstras-algorithm-for-adjacency-list-representation/)
3. [Bellman–Ford Algorithm](https://www.geeksforgeeks.org/dynamic-programming-set-23-bellman-ford-algorithm/)
4. [Floyd Warshall Algorithm](https://www.geeksforgeeks.org/dynamic-programming-set-16-floyd-warshall-algorithm/)
5. [Johnson’s algorithm for All-pairs shortest paths](https://www.geeksforgeeks.org/johnsons-algorithm/)
6. [Shortest Path in Directed Acyclic Graph](https://www.geeksforgeeks.org/shortest-path-for-directed-acyclic-graphs/)
7. [Some interesting shortest path questions,](https://www.geeksforgeeks.org/interesting-shortest-path-questions-set-1/)
8. [Shortest path with exactly k edges in a directed and weighted graph](https://www.geeksforgeeks.org/shortest-path-exactly-k-edges-directed-weighted-graph/)
9. [Dial’s Algorithm](https://www.geeksforgeeks.org/dials-algorithm-optimized-dijkstra-for-small-range-weights/)
10. [Printing paths in Dijsktra’s Algorithm](https://www.geeksforgeeks.org/printing-paths-dijkstras-shortest-path-algorithm/)
11. [Shortest path of a weighted graph where weight is 1 or 2](https://www.geeksforgeeks.org/shortest-path-weighted-graph-weight-edge-1-2/)

***Connectivity:***

1. [Find if there is a path between two vertices in a directed graph](https://www.geeksforgeeks.org/find-if-there-is-a-path-between-two-vertices-in-a-given-graph/)
2. [Connectivity in a directed graph](https://www.geeksforgeeks.org/connectivity-in-a-directed-graph/)
3. [Articulation Points (or Cut Vertices) in a Graph](https://www.geeksforgeeks.org/articulation-points-or-cut-vertices-in-a-graph/)
4. [Biconnected graph](https://www.geeksforgeeks.org/biconnectivity-in-a-graph/)
5. [Bridges in a graph](https://www.geeksforgeeks.org/bridge-in-a-graph/)
6. [Eulerian path and circuit](https://www.geeksforgeeks.org/eulerian-path-and-circuit/)
7. [Fleury’s Algorithm for printing Eulerian Path or Circuit](https://www.geeksforgeeks.org/fleurys-algorithm-for-printing-eulerian-path/)
8. [Strongly Connected Components](https://www.geeksforgeeks.org/strongly-connected-components/)
9. [Transitive closure of a graph](https://www.geeksforgeeks.org/transitive-closure-of-a-graph/)
10. [Find the number of islands](https://www.geeksforgeeks.org/find-number-of-islands/)
11. [Count all possible walks from a source to a destination with exactly k edges](https://www.geeksforgeeks.org/count-possible-paths-source-destination-exactly-k-edges/)
12. [Euler Circuit in a Directed Graph](https://www.geeksforgeeks.org/euler-circuit-directed-graph/)
13. [Biconnected Components](https://www.geeksforgeeks.org/biconnected-components/)
14. [Count the number of non-reachable nodes](https://www.geeksforgeeks.org/count-number-non-reachable-nodes/)
15. [Find the Degree of a Particular vertex in a Graph](https://www.geeksforgeeks.org/find-degree-particular-vertex-graph/)
16. [Check if a given graph is tree or not](http://geeksquiz.com/check-given-graph-tree/)
17. [Karger’s algorithm for Minimum Cut](https://www.geeksforgeeks.org/kargers-algorithm-for-minimum-cut-set-1-introduction-and-implementation/)
18. [Eulerian Path in undirected graph](https://www.geeksforgeeks.org/eulerian-path-undirected-graph/)
19. [Find if there is a path of more than k length](https://www.geeksforgeeks.org/find-if-there-is-a-path-of-more-than-k-length-from-a-source/)
20. [Length of shortest chain to reach the target word](https://www.geeksforgeeks.org/length-of-shortest-chain-to-reach-a-target-word/)
21. [Print all paths from a given source to destination](https://www.geeksforgeeks.org/find-paths-given-source-destination/)
22. [Find minimum cost to reach destination using train](https://www.geeksforgeeks.org/find-the-minimum-cost-to-reach-a-destination-where-every-station-is-connected-in-one-direction/)
23. [Tarjan’s Algorithm to find strongly connected Components](https://www.geeksforgeeks.org/tarjan-algorithm-find-strongly-connected-components/)

***Hard Problems:***

1. [Graph Coloring (Introduction and Applications)](https://www.geeksforgeeks.org/graph-coloring-applications/)
2. [Greedy Algorithm for Graph Coloring](https://www.geeksforgeeks.org/graph-coloring-set-2-greedy-algorithm/)
3. [Travelling Salesman Problem (Naive and Dynamic Programming)](https://www.geeksforgeeks.org/travelling-salesman-problem-set-1/)
4. [Travelling Salesman Problem (Approximate using MST)](https://www.geeksforgeeks.org/travelling-salesman-problem-set-2-approximate-using-mst/)
5. [Hamiltonian Cycle](https://www.geeksforgeeks.org/backtracking-set-7-hamiltonian-cycle/)
6. [Vertex Cover Problem | Set 1 (Introduction and Approximate Algorithm)](https://www.geeksforgeeks.org/vertex-cover-problem-set-1-introduction-approximate-algorithm-2/)
7. [K Centers Problem | Set 1 (Greedy Approximate Algorithm)](https://www.geeksforgeeks.org/k-centers-problem-set-1-greedy-approximate-algorithm/)

***Maximum Flow:***

1. [Ford-Fulkerson Algorithm for Maximum Flow Problem](https://www.geeksforgeeks.org/ford-fulkerson-algorithm-for-maximum-flow-problem/)
2. [Find maximum number of edge disjoint paths between two vertices](https://www.geeksforgeeks.org/find-edge-disjoint-paths-two-vertices/)
3. [Find minimum s-t cut in a flow network](https://www.geeksforgeeks.org/minimum-cut-in-a-directed-graph/)
4. [Maximum Bipartite Matching](https://www.geeksforgeeks.org/maximum-bipartite-matching/)
5. [Channel Assignment Problem](https://www.geeksforgeeks.org/channel-assignment-problem/)
6. [Push Relabel- Set 1-Introduction](https://www.geeksforgeeks.org/push-relabel-algorithm-set-1-introduction-and-illustration/)
7. [Push Relabel- Set 2- Implementation](https://www.geeksforgeeks.org/push-relabel-algorithm-set-2-implementation/)
8. [Karger’s Algorithm- Set 1- Introduction and Implementation](https://www.geeksforgeeks.org/kargers-algorithm-for-minimum-cut-set-1-introduction-and-implementation/)
9. [Karger’s Algorithm- Set 2 – Analysis and Applications](https://www.geeksforgeeks.org/kargers-algorithm-for-minimum-cut-set-2-analysis-and-applications/)

***STL Implementation of Algorithms***

1. [Kruskal’s Minimum Spanning Tree using STL in C++](https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-using-stl-in-c/)
2. [Prim’s Algorithm using Priority Queue STL](https://www.geeksforgeeks.org/prims-algorithm-using-priority_queue-stl/)
3. [Dijkstra’s Shortest Path Algorithm using STL](https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-using-priority_queue-stl/)
4. [Dijkstra’s Shortest Path Algorithm using set in STL](https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-using-set-in-stl/)

***Misc***

1. [Number of triangles in an undirected Graph](https://www.geeksforgeeks.org/number-of-triangles-in-a-undirected-graph/)
2. [Number of triangles in directed and undirected Graph](https://www.geeksforgeeks.org/number-of-triangles-in-directed-and-undirected-graphs/)
3. [Check whether a given graph is Bipartite or not](https://www.geeksforgeeks.org/bipartite-graph/)
4. [Snake and Ladder Problem](https://www.geeksforgeeks.org/snake-ladder-problem-2/)
5. [Minimize Cash Flow among a given set of friends who have borrowed money from each other](https://www.geeksforgeeks.org/bipartite-graph/)
6. [Boggle (Find all possible words in a board of characters)](https://www.geeksforgeeks.org/boggle-find-possible-words-board-characters/)
7. [Hopcroft Karp Algorithm for Maximum Matching-Introduction](https://www.geeksforgeeks.org/hopcroft-karp-algorithm-for-maximum-matching-set-1-introduction/)
8. [Hopcroft Karp Algorithm for Maximum Matching-Implementation](https://www.geeksforgeeks.org/hopcroft-karp-algorithm-for-maximum-matching-set-2-implementation/)
9. [Minimum Time to rot all oranges](https://www.geeksforgeeks.org/minimum-time-required-so-that-all-oranges-become-rotten/)
10. [Construct binary palindrome by repeated appending and trimming](https://www.geeksforgeeks.org/construct-binary-palindrome-by-repeated-appending-and-trimming/)
11. [Find same contents in a list of contacts](https://www.geeksforgeeks.org/find-same-contacts-in-a-list-of-contacts/)
12. [Hypercube Graph](https://www.geeksforgeeks.org/hypercube-graph/)
13. [Check for star graph](https://www.geeksforgeeks.org/check-star-graph/)
14. [Optimal read list for a given number of days](https://www.geeksforgeeks.org/optimal-read-list-given-number-days/)
15. [Print all jumping numbers smaller than or equal to a given value](https://www.geeksforgeeks.org/print-all-jumping-numbers-smaller-than-or-equal-to-a-given-value/)