

Graphs Cheat Sheet

Algorithms to solve common graph problems and associated complexities

Problem	Algorithms	Time	Space
Find the path to an element in an undirected graph	DFS	$E + V$	V
Find an Eulerian cycle in a digraph (if it exists)	DFS	$E + V$	V
Determine if a given undirected graph has a cycle and find such a cycle if it exists	DFS	$E + V$	V
Determine reachability in a digraph from a given vertex using depth-first search	DFS	$E + V$	V
Find an Eulerian cycle in a graph (if it exists)	DFS	$E + V$	$E + V$
Find a directed cycle in a digraph	DFS	$E + V$	V
Compute topological ordering of a DAG or edge-weighted DAG.	DFS	$E + V$	V
Given a graph, find either a bipartition or an odd-length cycle.	DFS	$E + V$	V
Compute connected components using depth-first search.	DFS	$E + V$	V
Compute transitive closure of a digraph and support reachability queries	DFS	$V(E + V)$	V^2
Breadth-first search on an undirected graph (shortest path/fewest edges)	BFS	$E + V$	V
Compute the shortest directed path (with fewest edges)	BFS	$E + V$	V
Compute the shortest path (assuming all weights are non-negative)	Dijkstra	$E \log V$	V
Compute the shortest path (when there are no negative cycles)	Bellman–Ford	$V(V + E)$	V
Compute shortest paths in an edge-weighted acyclic digraph (no cycles)	Topological Sort	$V + E$	V
Compute the all-pairs shortest path	Floyd–Warshall	V^3	V^2
Compute the strongly-connected components of a digraph	Tarjan/Gabow/Kosaraju–Sharir	$E + V$	V
Compute a minimum spanning tree/forest	Kruskal	$E \log E$	$E + V$
Compute a minimum spanning tree/forest	Prim/Boruvka	$E \log V$	V
Compute a max-flow and a min-cut	Ford–Fulkerson	$E V(E + V)$	V
Find max cardinality matching and min cardinality vertex cover in a bipartite graph	Hopcroft–Karp	$V^{1/2}(E + V)$	V
Solve an n-by-n assignment problem	Successive Shortest Path	$n^3 \log n$	n^2