Graph theory

| Algorithm | Time complexity | Space complexity | Constraints | | Optimal representation |
| --- | --- | --- | --- | --- | --- |
| Directed graph | Undirected graph |
| Breadth first search (BFS) | O(V + E) | O(V) | Yes | Yes | Adjacency list |
| Depth first search (DFS) | O(V + E) | O(V) | Yes | Yes | Adjacency list |
| Detect cycle in an undirected graph (DFS) | O(V + E) | O(V) | No | Yes | Adjacency list |
| Detect cycle in a directed graph (DFS) | O(V + E) | O(V) | Yes | No | Adjacency list |
| Detect cycle using colors | O(V + E) | O(V) | Yes | No | Adjacency list |
| Detect cycle using disjoint set or union-find – naive approach | O(V \* E) | O(E) | No | Yes | Edge list |
| Detect cycle using disjoint set or union-find – union by rank & path compression | O(E \* log(V)) | O(E) | No | Yes | Edge list |
| Topological sorting | O(V + E) | O(V) | DAG only | No | Adjacency list |
| Longest path in a directed acyclic graph | O(V + E) | O(V) | DAG only | No | Adjacency list |
| Kruskal’s minimum spanning tree algorithm – sparse graph | O(E \* log(V)) | O(E) | No | Yes | Edge list |
| Prim’s minimum spanning tree algorithm – dense graph | O(E \* log(V)) | O(V) | No | Yes | Adjacency list |
| Find Hamiltonian path – NP hard | O(V!) | O(V) | Yes | Yes | Adjacency list |
| Find Hamiltonian cycle – NP hard | O(V!) | O(V) | Yes | Yes | Adjacency list |
| Dijkstra’s shortest path algorithm | O(E \* log(V)) | O(V) | Yes | Yes | Adjacency list |
| Bellman – Ford algorithm | O(E \* V) | O(E) | Yes | No | Edge list |
| Floyd – Warshall algorithm – dense graph | O(V^3) | O(V^2) | Yes, but no negative weight cycles | Yes, but no negative weight edges | Adjacency matrix |
| Johnson’s algorithm – sparse graph | O(E \* V \* log(V)) | O(V + E) | Yes | No | Edge List + Adjacency list |
| Kosaraju’s strongly connected component algorithm | O(V + E) | O(V) | Yes | No | Adjacency list |
| Tarjan’s strongly connected component algorithm | O(V + E) | O(V) | Yes | No | Adjacency list |