Applications of Depth First Search

Depth-first search (DFS) is an algorithm (or technique) for traversing a graph. DFS uses Following problems as a building block:

Detecting cycle in a graph

Topological Sorting

Finding Strongly Connected Components

Path Finding

To test a graph is bipartite

Solving puzzles with only one solution, such as mazes

**Detecting cycle in a graph**

A cycle of a [graph](https://mathworld.wolfram.com/Graph.html) G, also called a [circuit](https://mathworld.wolfram.com/Circuit.html) if the first vertex is not specified, is a subset of the [edge set](https://mathworld.wolfram.com/EdgeSet.html) of G that forms a [path](https://mathworld.wolfram.com/Path.html) such that the first node of the path corresponds to the last.

A graph has cycle if and only if we see a back edge during DFS. So we can run DFS for the graph and check for back edges.

**Topological Sorting**

Topological Sorting is mainly used for scheduling jobs from the given dependencies among jobs. In computer science, applications of this type arise in instruction scheduling, ordering of formula cell evaluation when recomputing formula values in spreadsheets, logic synthesis, determining the order of compilation tasks to perform in makefiles, data serialization, and resolving symbol dependencies in linkers.

For example, a topological sorting of the following graph is “5 4 2 3 1 0”. There can be more than one topological sorting for a graph. For example, another topological sorting of the following graph is “4 5 2 3 1 0”. The first vertex in topological sorting is always a vertex with in-degree as 0 (a vertex with no incoming edges).



**Finding Strongly Connected Components of a graph**

A directed graph is strongly connected if there is a path between all pairs of vertices. A strongly connected component (**SCC**) of a directed graph is a maximal strongly connected subgraph. For example, there are 3 SCCs in the following graph.

