

### 1. Cycle Detection in Directed Graph :

Directed Acyclic Graph (DAG) → if the graph is DAG then, there will be no cycle, else there will be cycle in a graph.

BFS or DFS any traversal will work.

#### DFS Traversal :

While traversal in exploring time if we find any node which is paused, there definitely is exiting a cycle. That's how we can detect a cycle in a Directed Graph.

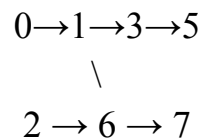
### 2. Cycle Detection in Directed Graph in Code : Implementation

From Cses.fi : [Round Trip II](https://cses.fi/problemset/task/1678) : <https://cses.fi/problemset/task/1678>

### 3. Topological Sorting :

For a dependency graph, FROM node will sort before the TO node. Graph must be Acyclic and Directed. BFS / DFS both will work.

E.g :



Topological Sort will be : 0, 2, 1, 6, 7, 3, 5

### 4. Topological Sorting in Code : Implementation

From Cses.fi : [Course Schedule](https://cses.fi/problemset/task/1679) : <https://cses.fi/problemset/task/1679>