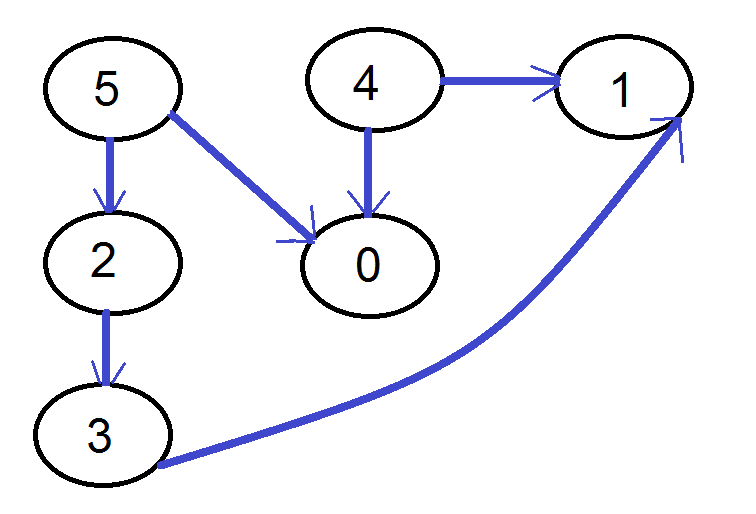
**Topological Sort**

**Document Id 43**

* Requires a Directed Graph with no Cycles (DAG)

https://en.wikipedia.org/wiki/Directed\_acyclic\_graph

* Any DAG has at least one topological ordering
* The first vertex is always a vertex with degree 0
* A common topological sort is based on incoming degree
* https://en.wikipedia.org/wiki/Topological\_sorting



**Questions**

Q1 What is the essential attribute for a graph to have a “Path”

Q2 give one application of a directed graph

Q3 define a “directed cycle”

Q4 when is a directed graph “strongly connected”

Q5 What is the attribute for a directed graph to have a topological ordering

Q6 For an objects transitive dependency graph a cyclic dependency is a ?

Q8 to optimise the potential for an object to be garbage collected its dependency graph must

be a ?

**Practical Task Shortest Path Optimisation**

As shown a topological sort depends on the Graph being a DAG, one approach to applying a topological sort on non acyclic data involves finding the cycle and temporary removing it (surgery), another approach is to ‘pinch’ that is disconnect the cycle(s)

Implement code that finds the shortest directed cycle in a directed graph

https://en.wikipedia.org/wiki/Shortest\_path\_problem

https://en.wikipedia.org/wiki/Shortest\_path\_problem#Road\_networks

Hint BFS

Given a directed graph, design an algorithm to find the minimum directed cycle

BFS → search for the shortest path from A to all other nodes

BFS → queue to enqueue and dequeue → traverse nodes in order of depth.