

ASSIGNMENT-01

Topological Sorting:

```
from collections import defaultdict, deque
```

```
class Graph:
    def __init__(self):
        self.graph = defaultdict(list)
        self.in_degree = defaultdict(int)

    def add_edge(self, u, v):
        self.graph[u].append(v)
        self.in_degree[v] += 1

    def topological_sort(self):
        queue = deque()
        result = []

        for node in self.graph:
            if self.in_degree[node] == 0:
                queue.append(node)

        while queue:
            node = queue.popleft()
            result.append(node)

            for neighbor in self.graph[node]:
                self.in_degree[neighbor] -= 1
                if self.in_degree[neighbor] == 0:
                    queue.append(neighbor)

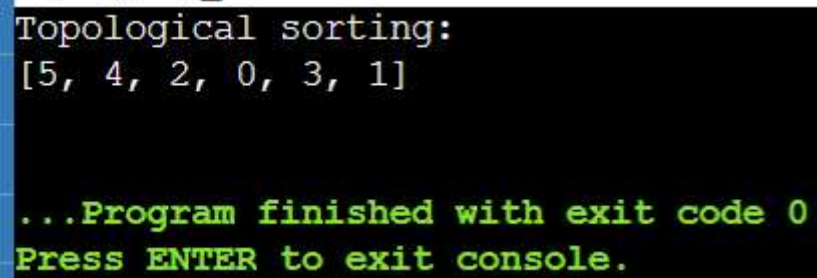
        return result

# Create a directed acyclic graph
graph = Graph()
graph.add_edge(5, 2)
graph.add_edge(5, 0)
```

```
graph.add_edge(4, 0)
graph.add_edge(4, 1)
graph.add_edge(2, 3)
graph.add_edge(3, 1)

print("Topological sorting:")
topo_sort = graph.topological_sort()
print(topo_sort)
```

Output:

A screenshot of a terminal window with a black background and white and green text. The output shows the result of a topological sort on a graph with 6 nodes (0-5). The sorted order is [5, 4, 2, 0, 3, 1]. The terminal also shows a green message indicating the program finished with exit code 0 and a prompt to press ENTER to exit the console.

```
Topological sorting:
[5, 4, 2, 0, 3, 1]

...Program finished with exit code 0
Press ENTER to exit console.
```