Graphs and Sorting

Ishwar Suriyaprakash

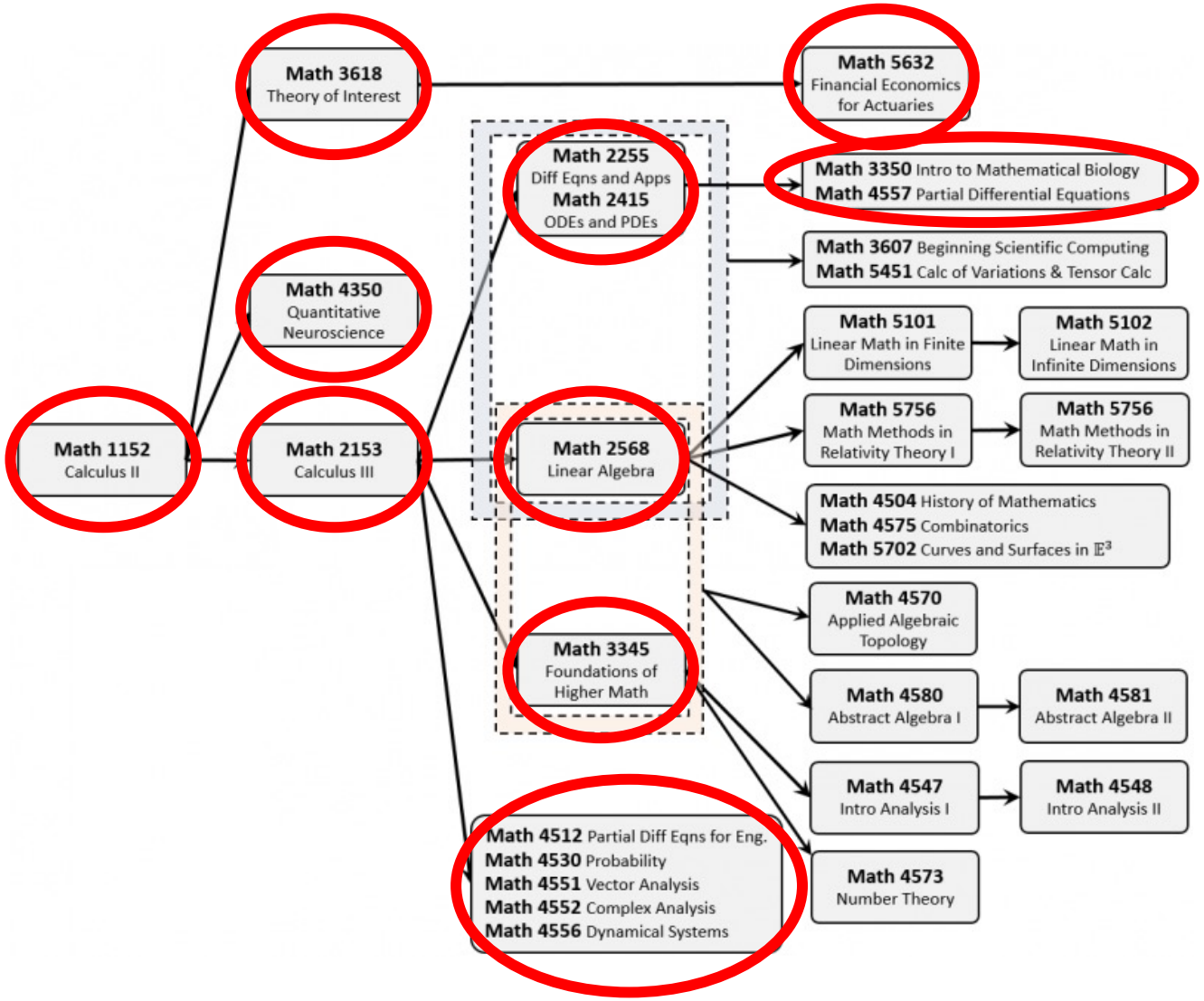
Scenario: Course Plan

Mathematics Core

- Math 1152 Calculus II
  - Math 2153 Calculus III
    - Math 2568 Linear Algebra
    - Math 3345 Foundations of Mathematics
    - Math 4530 Probability
    - Math 2255 Differential Equations and Their Applications
      - Math 4557 Partial Differential Equations
  - Math 3618 Theory of Interest
    - Math 5632 Financial Economics for Actuaries
  - Math 4350 Quantitative Neuroscience

Math 2153 and the courses that follow are the prerequisites for many upper division math courses. See Math Course page for more information.

Scenario: Course Plan

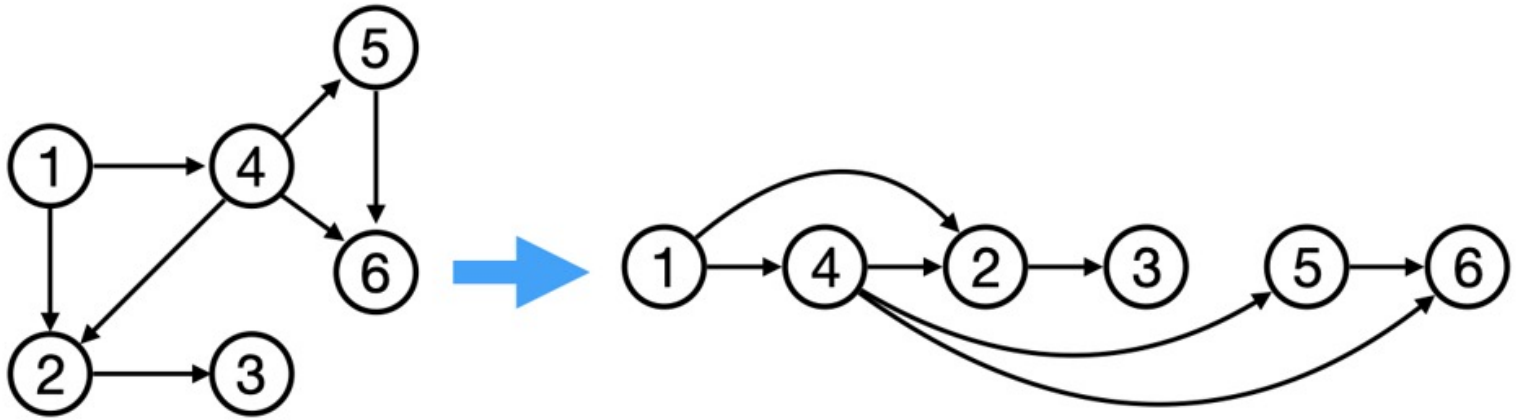


Scenario: Would it be so easy now?

- 2153 → 2568
- 2153 → 3345
- 1152 → 3618
- 2255 → 4557
- 2153 → 2255
- .
- .
- .

Topological Sort

- Creating ordering of vertices from a graph
- Key rule: NEIGHBORS of V are listed after V !!!!

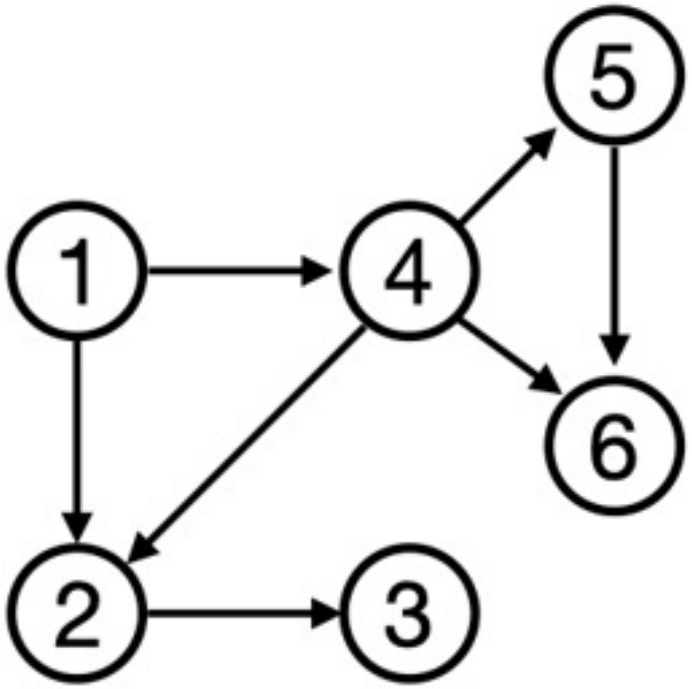


Topological Sort

- Depth-first search (DFS)
- Can also be implemented with Breadth-first search (BFS)

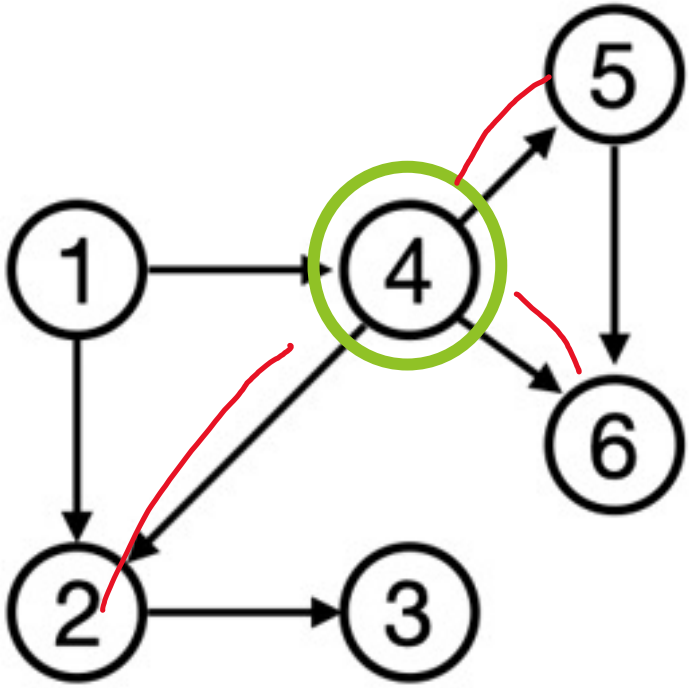
- Pick any vertex v
- DFS from v & append traversed vertices **(in reverse order)** to list
  - Done to order all vertices that “have to be done” after v
- Pick another vertex v that is **not visited**; Repeat

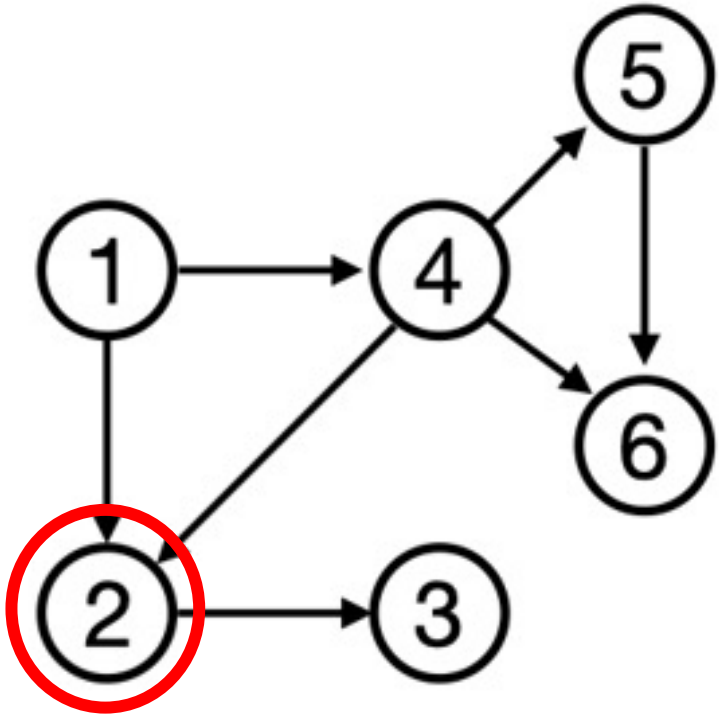
DFS Recursion



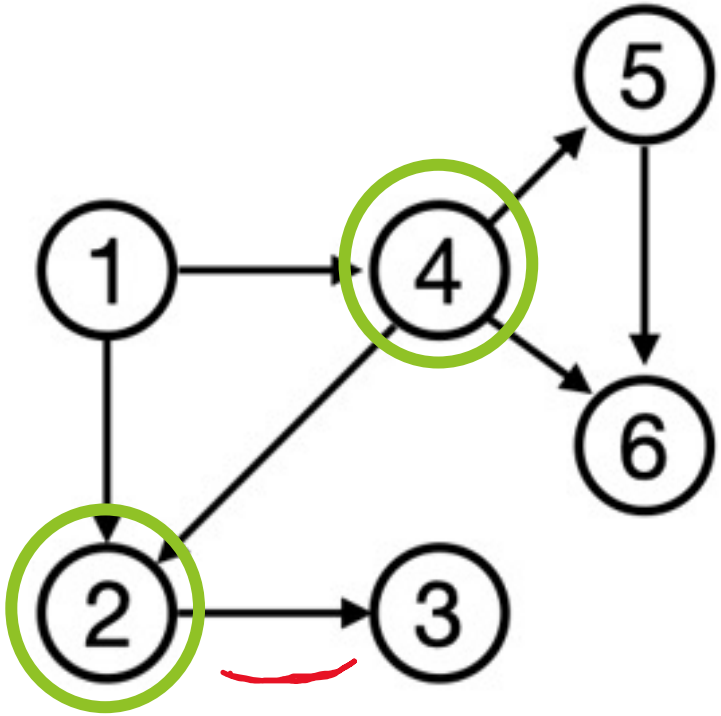
- Start at vertex 4
- All outgoing edges: {2, 5, 6}
- At each vertex, repeat from first step

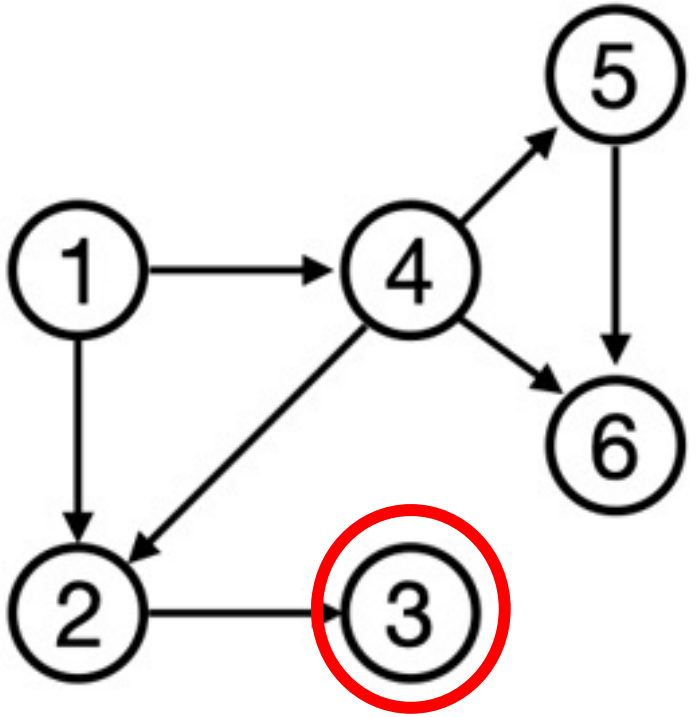
DFS: Example



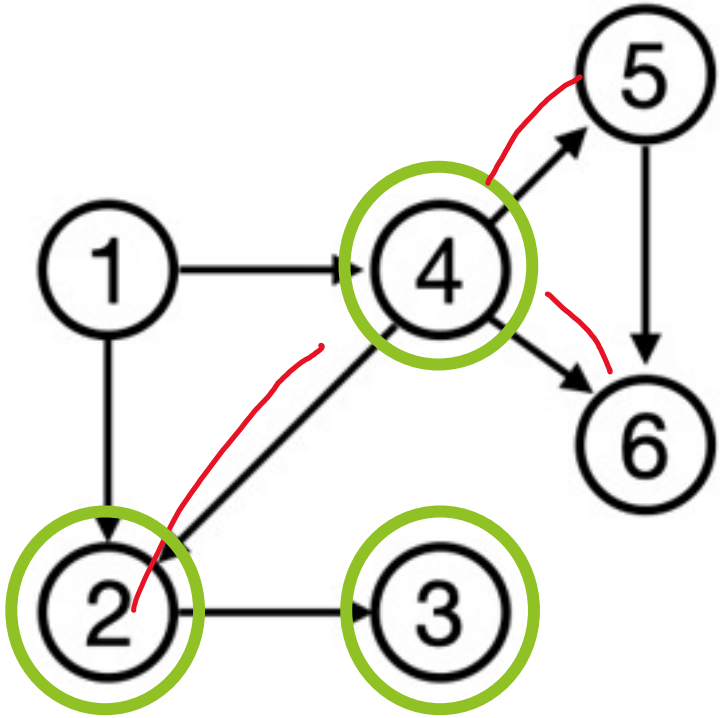


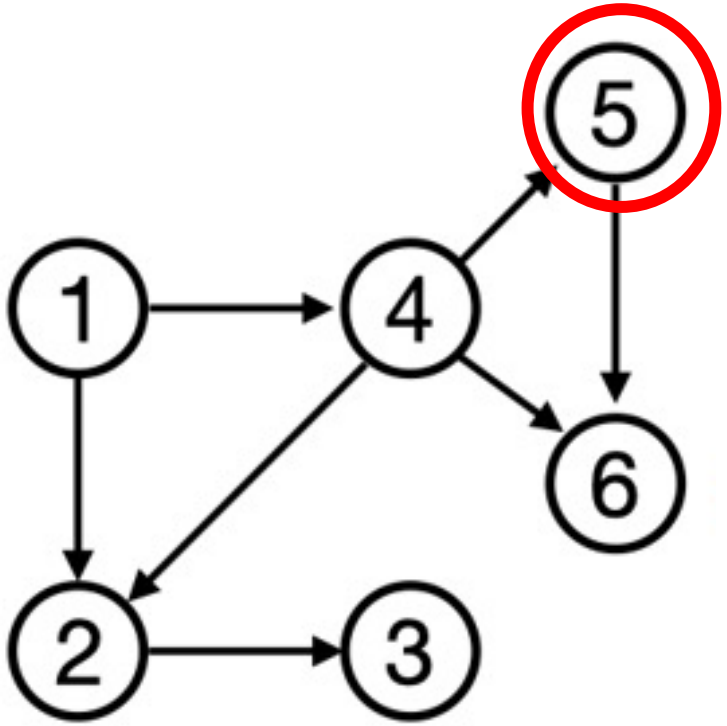
DFS: Example



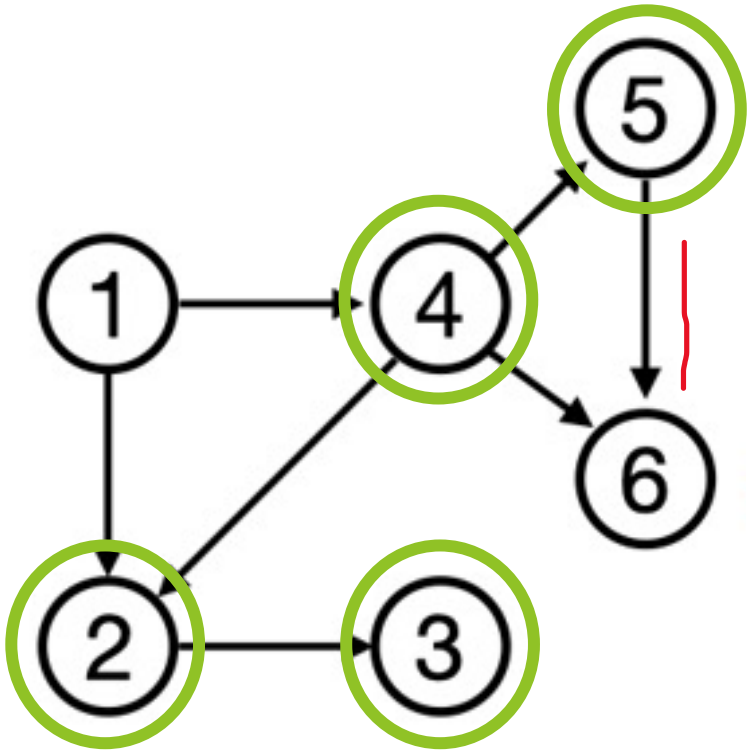


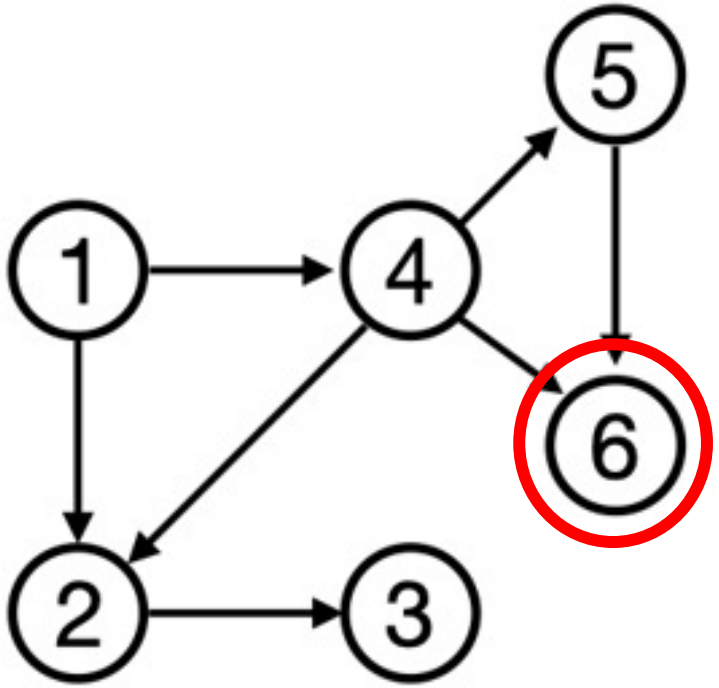
DFS: Example



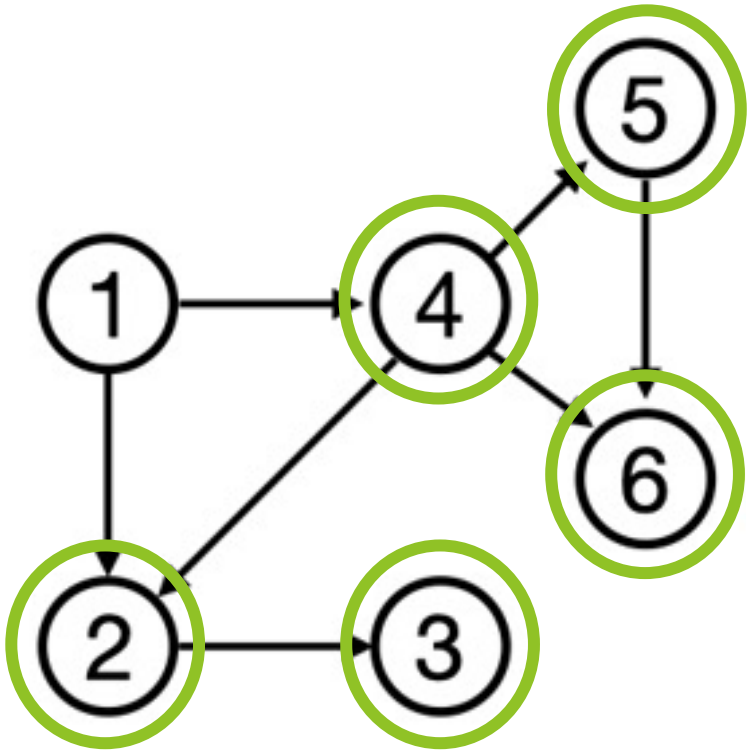


DFS: Example





DFS: Example



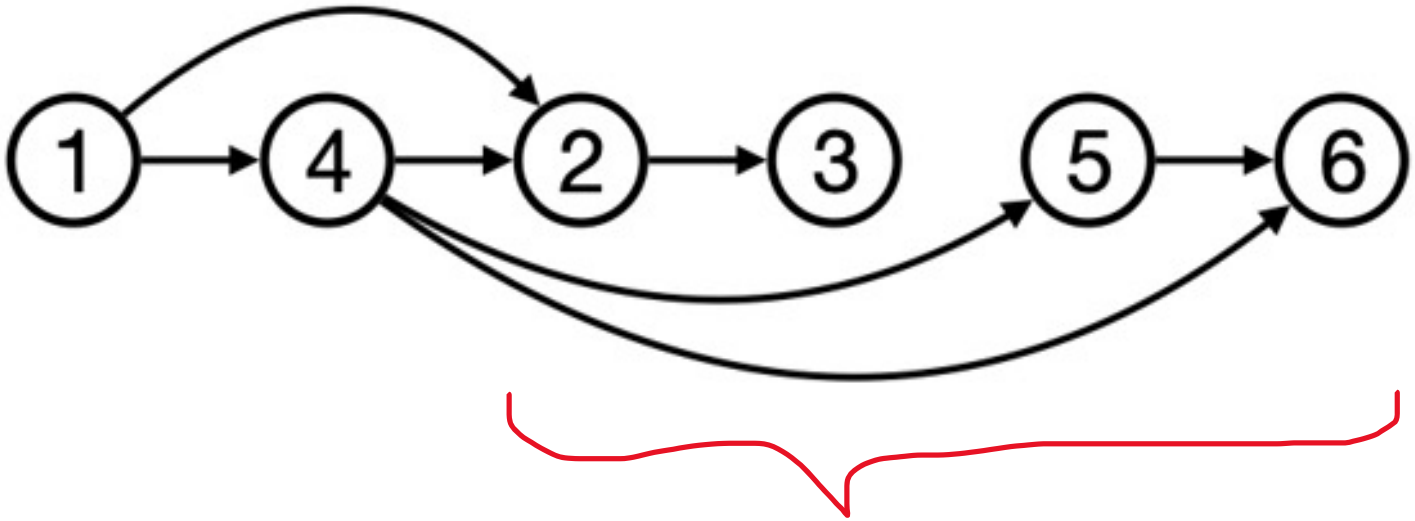
Topological Sort

- Time complexity: O(V+E)
- Cannot be done on graphs with cycles!! … only on **Directed Acyclical Graphs**

Topological Sort

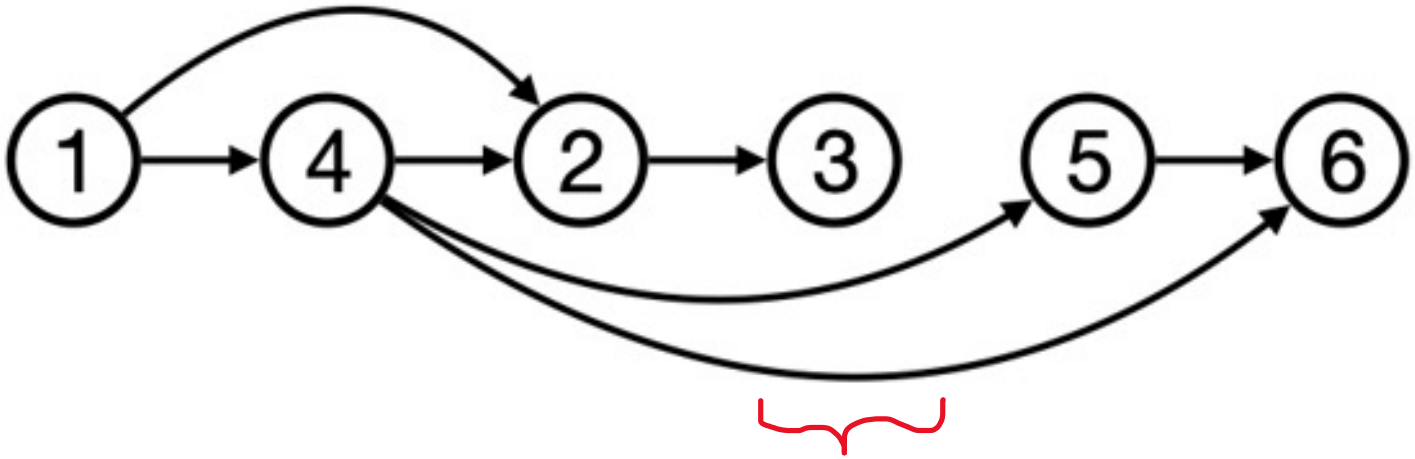
- Let’s go back to the DFS …..

Topological Sort



All vertices that require vertex 4 to precede them

Topological Sort



All vertices that require vertex 2 to precede them

Topological Sort: Implementation

https://replit.com/@IshwarSuriyapra/top-sort

Topological Sort: Implementation

```
# add the following line in dfs_helper where you see fit:
# reverse_sorted.append(v)
```

Scenario: Course Plan