

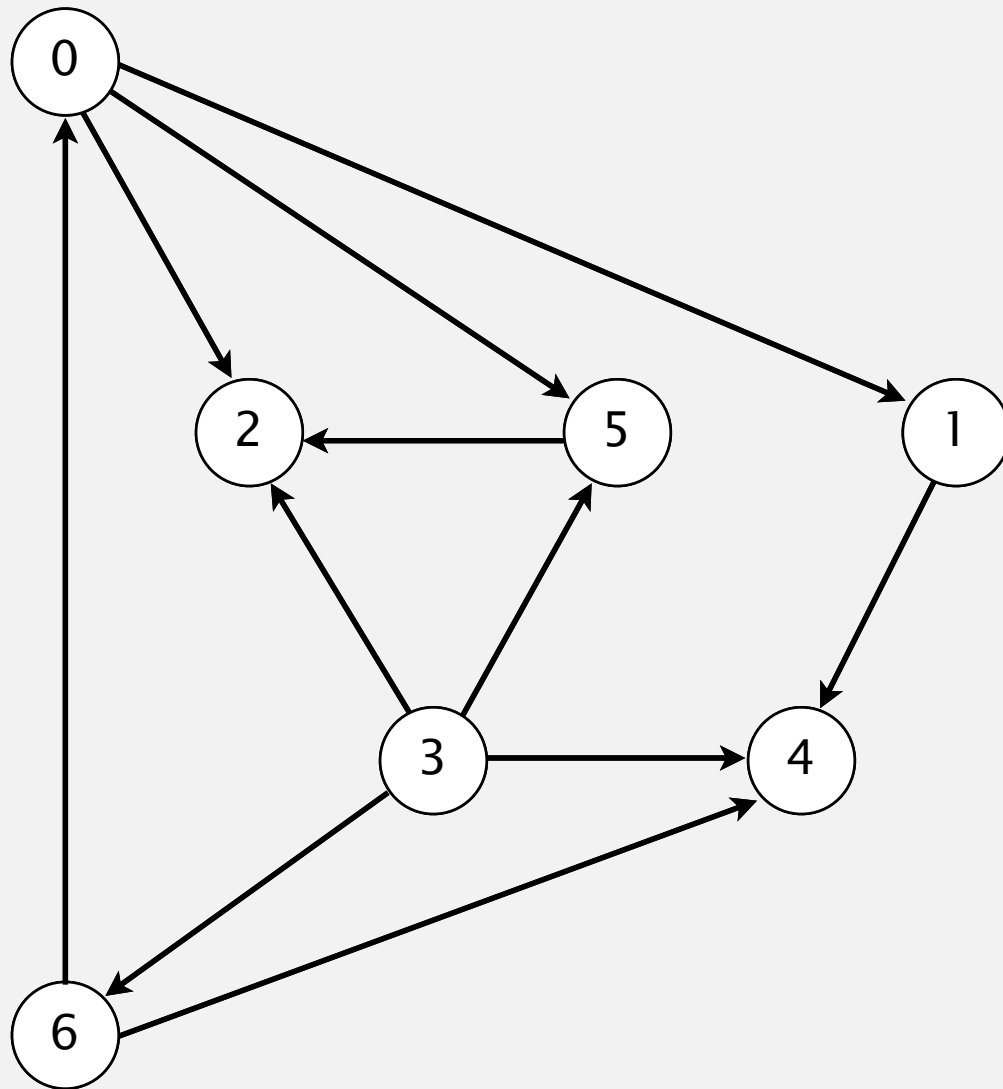


<http://algs4.cs.princeton.edu>

4.2 TOPOLOGICAL SORT DEMO

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



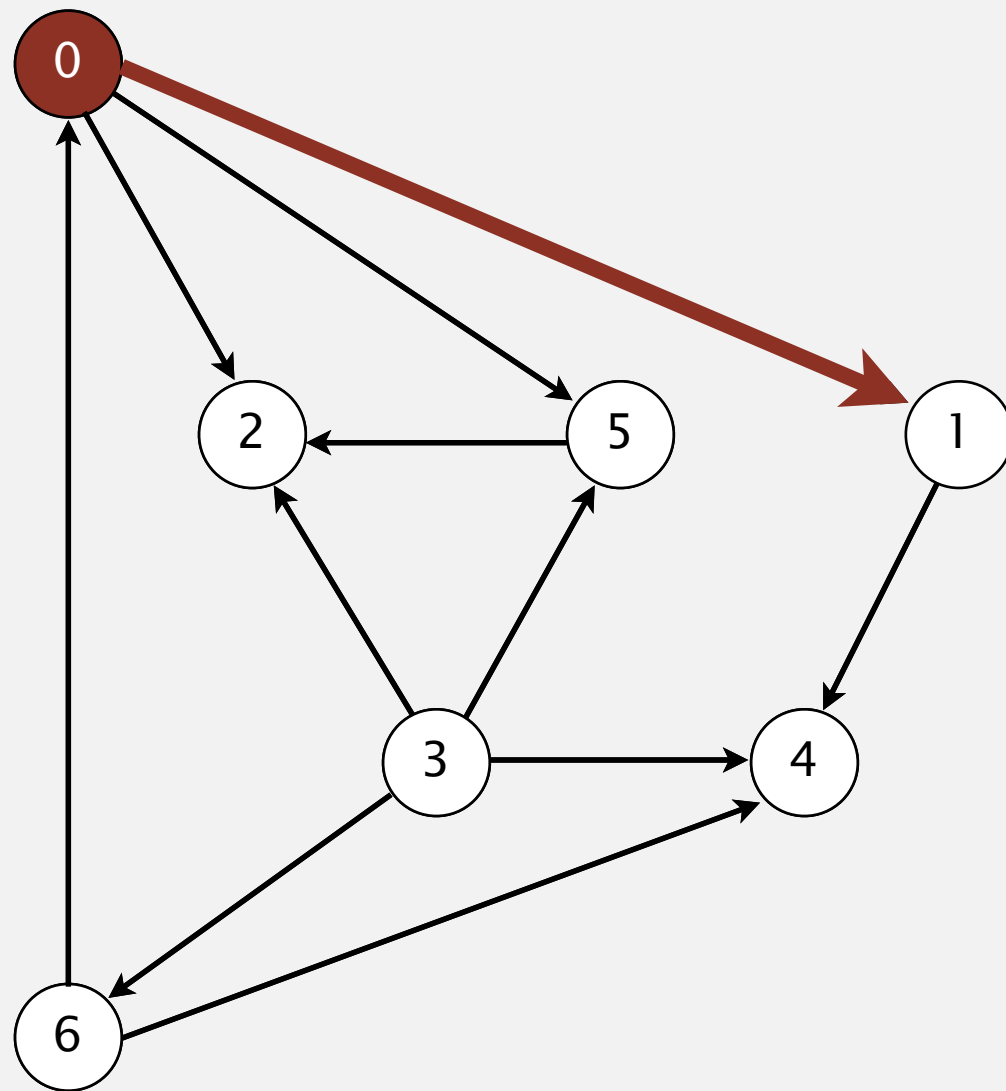
tinyDAG7.txt

```
7
11
0 5
0 2
0 1
3 6
3 5
3 4
5 2
6 4
6 0
3 2
```

a directed acyclic graph

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



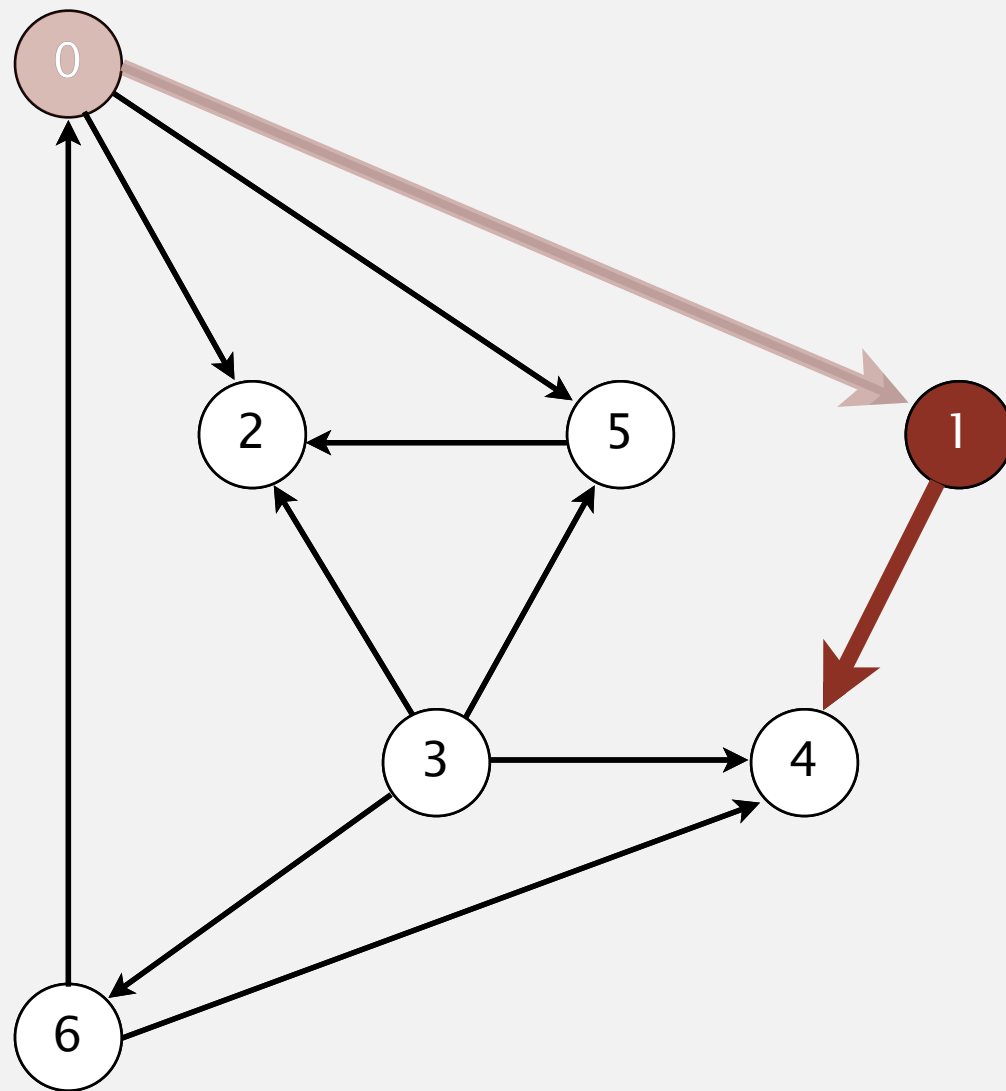
postorder

v	marked[]
0	T
1	F
2	F
3	F
4	F
5	F
6	F

visit 0: check 1, check 2, and check 5

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



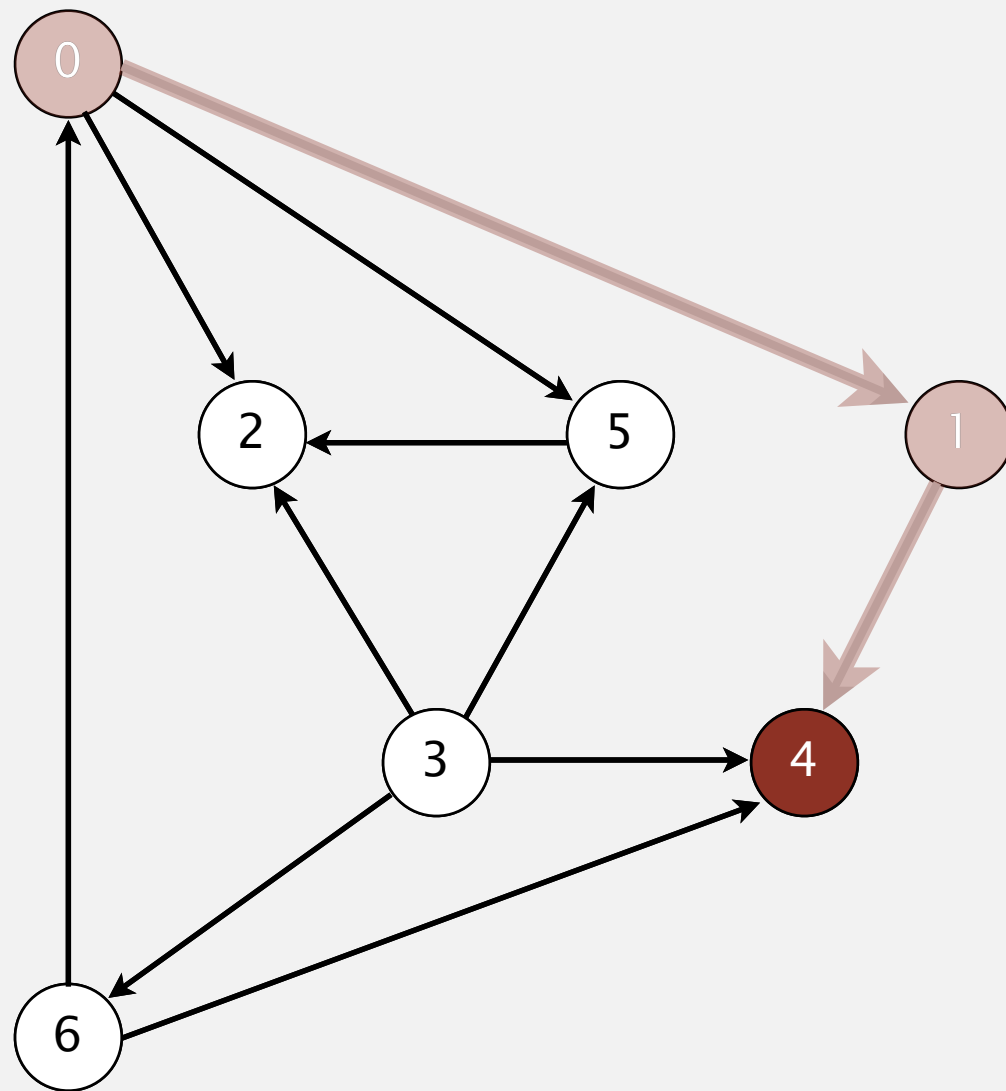
postorder

v	marked[]
0	T
1	T
2	F
3	F
4	F
5	F
6	F

visit 1: check 4

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



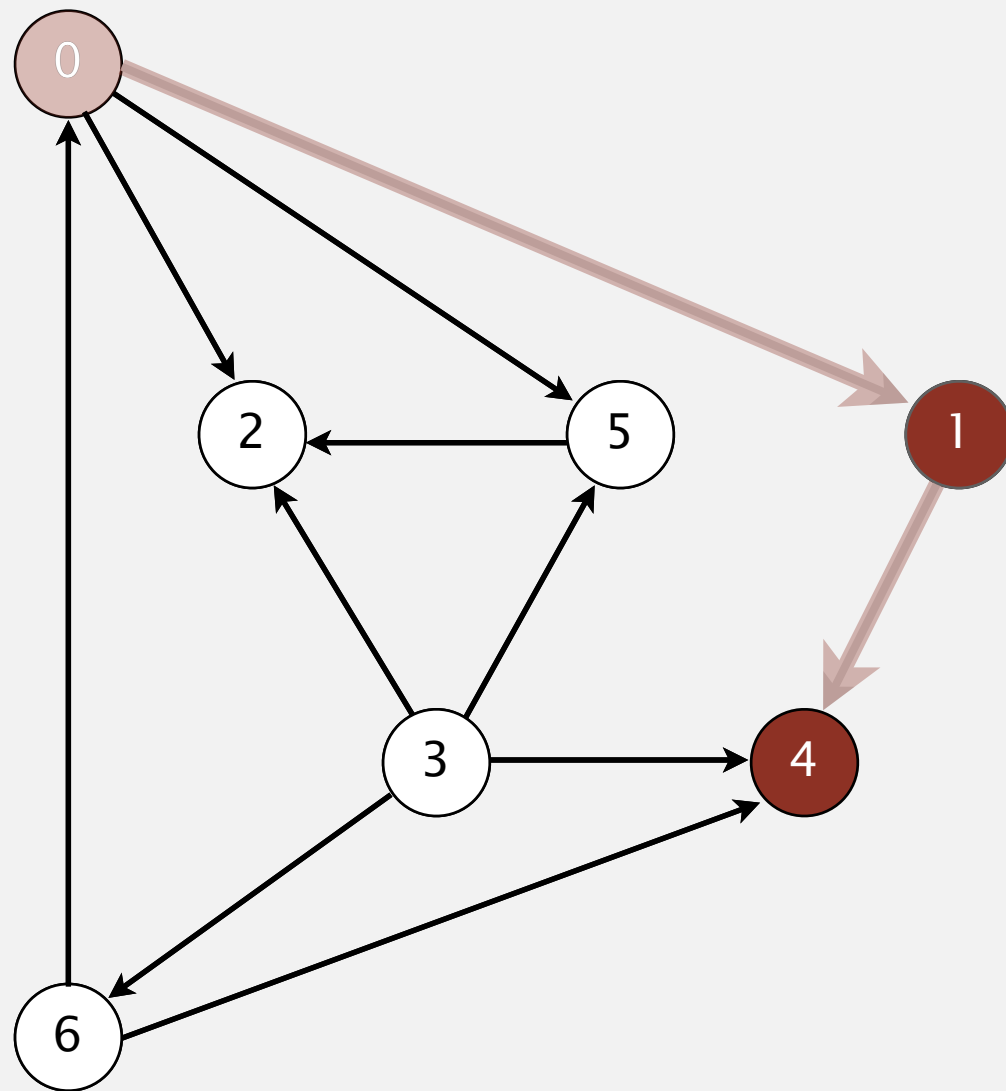
postorder

v	marked[]
0	T
1	T
2	F
3	F
4	T
5	F
6	F

visit 4

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

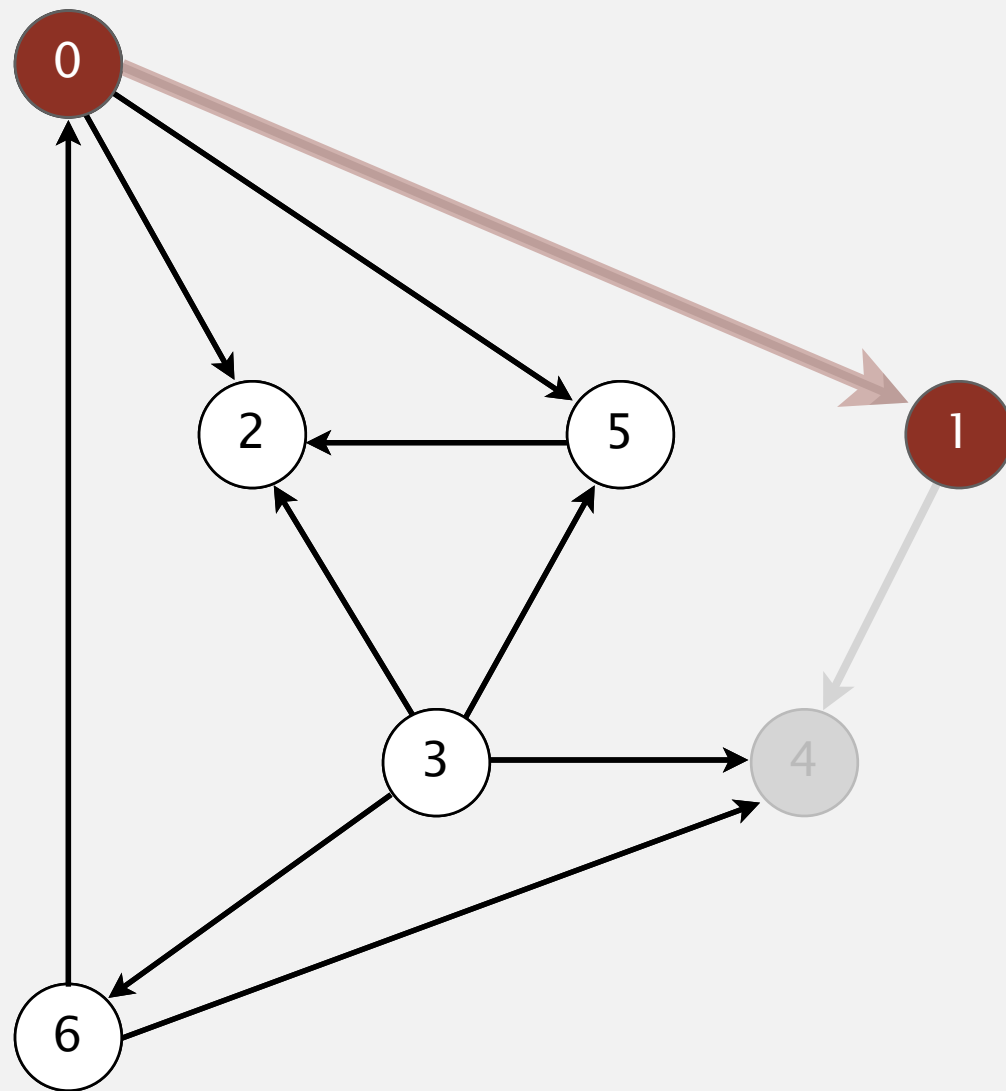
4

v	marked[]
0	T
1	T
2	F
3	F
4	T
5	F
6	F

4 done

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

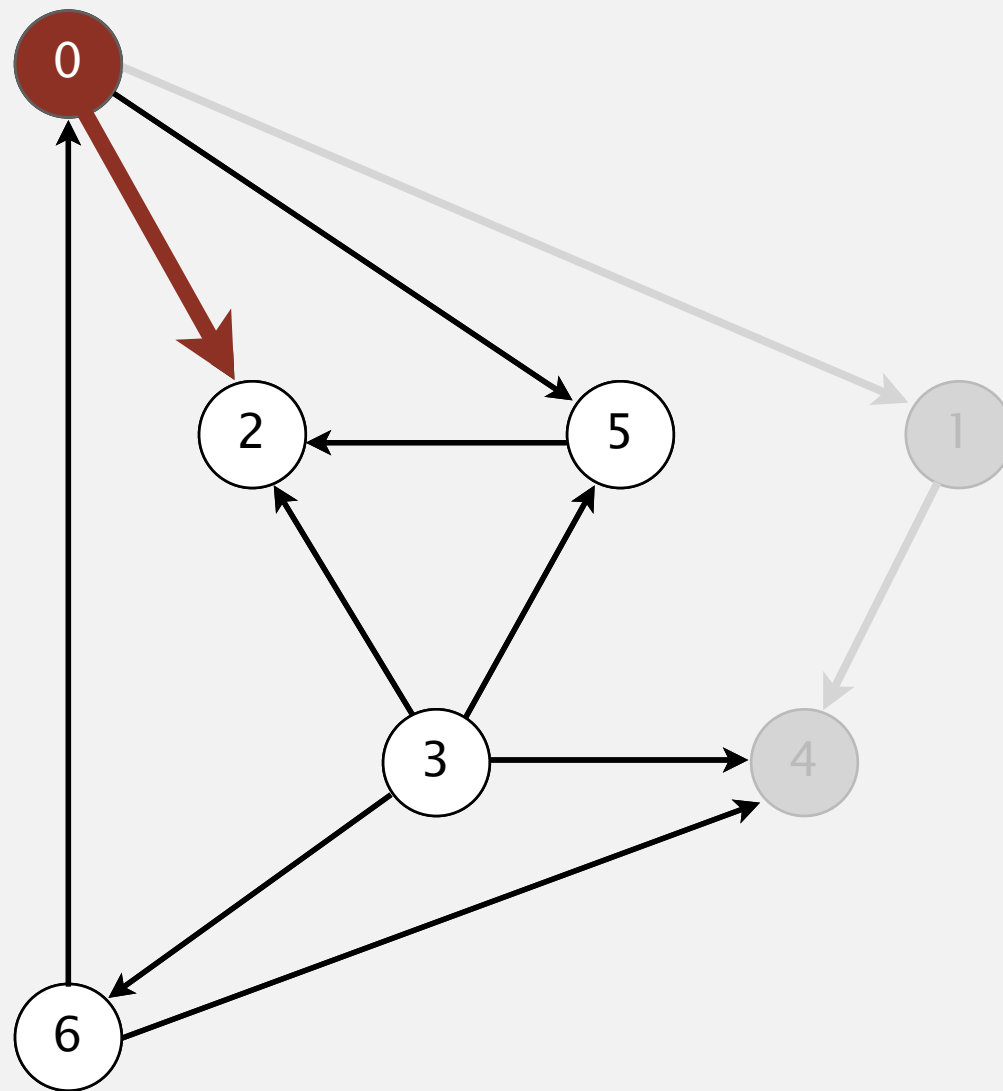
4 1

v	marked[]
0	T
1	T
2	F
3	F
4	T
5	F
6	F

1 done

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

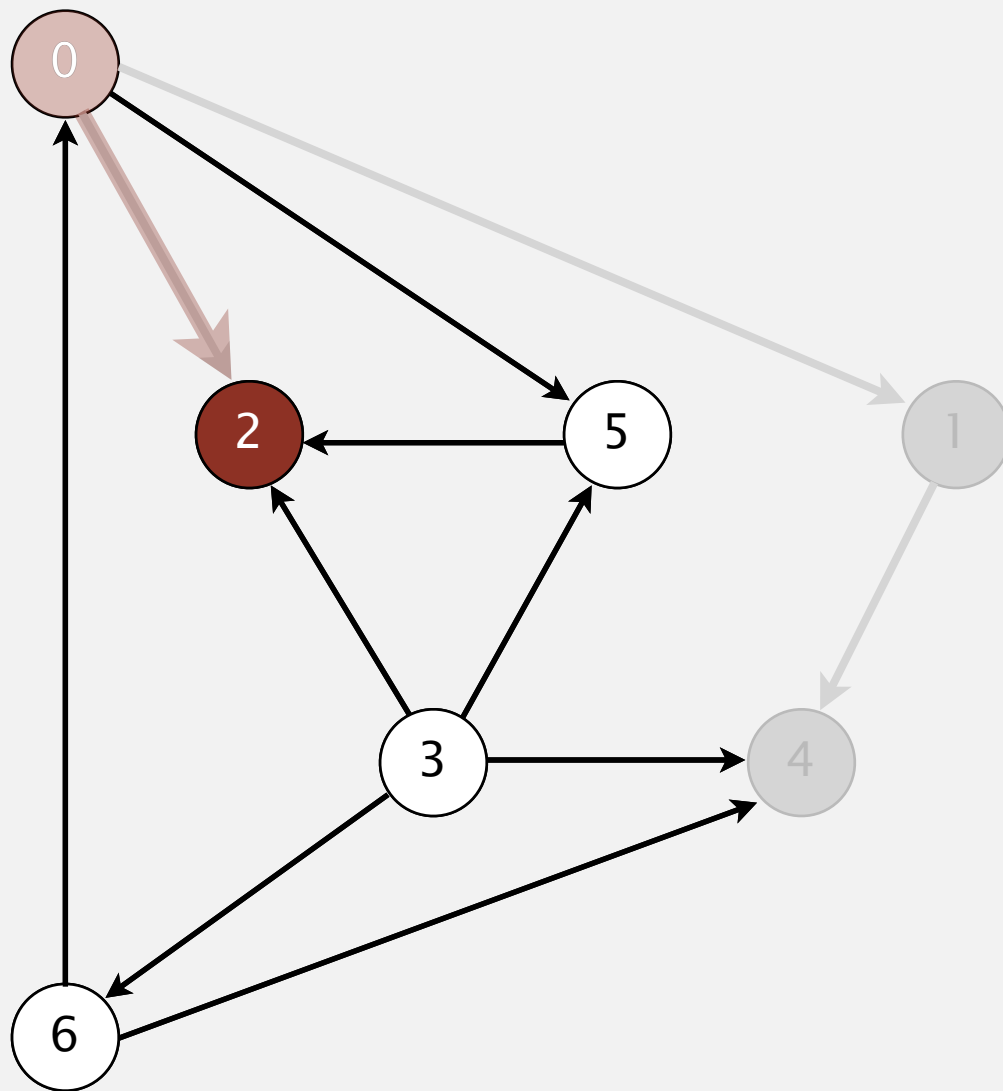
4 1

v	marked[]
0	T
1	T
2	F
3	F
4	T
5	F
6	F

visit 0: check 1, check 2, and check 5

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

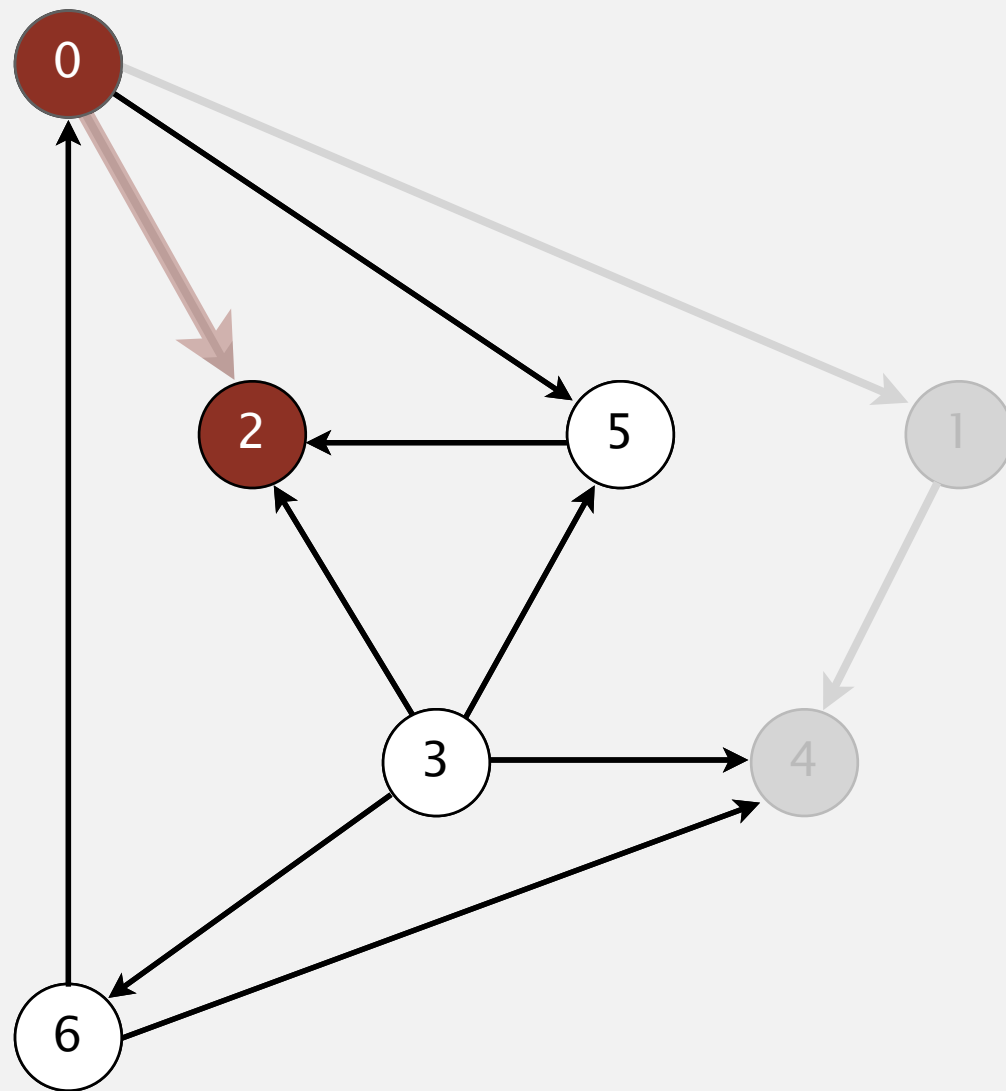
4 1

v	marked[]
0	T
1	T
2	T
3	F
4	T
5	F
6	F

visit 2

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

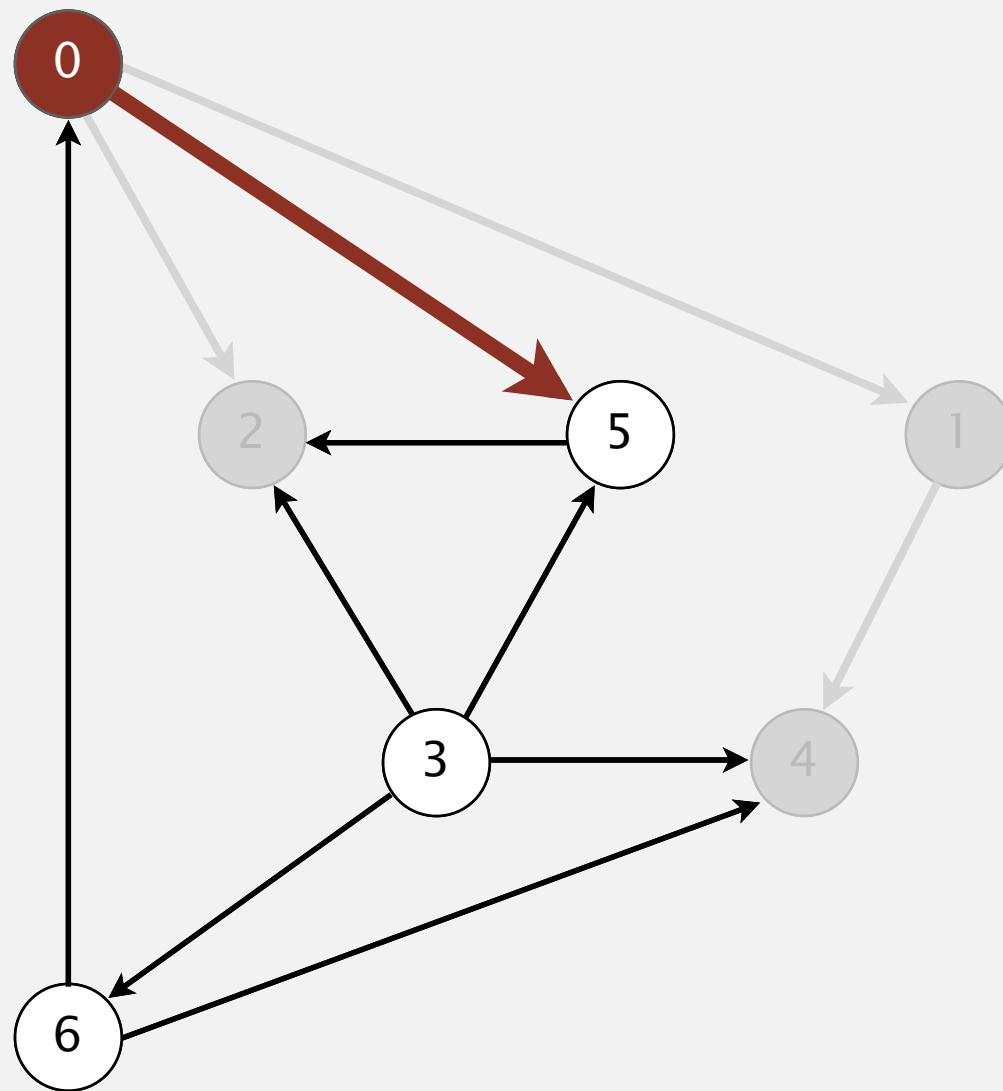
4 1 2

v	marked[]
0	T
1	T
2	T
3	F
4	T
5	F
6	F

2 done

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

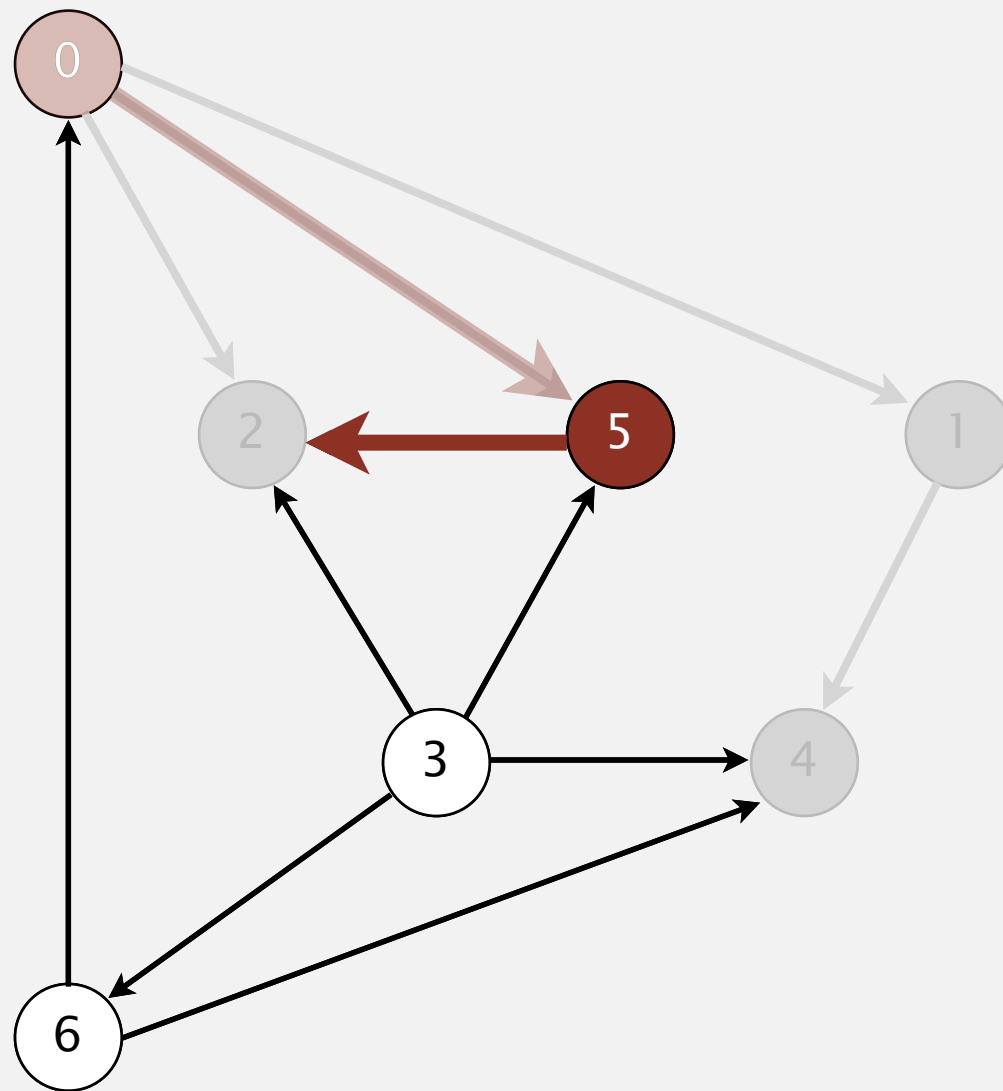
4 1 2

v	marked[]
0	T
1	T
2	T
3	F
4	T
5	F
6	F

visit 0: check 1, check 2, and **check 5**

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

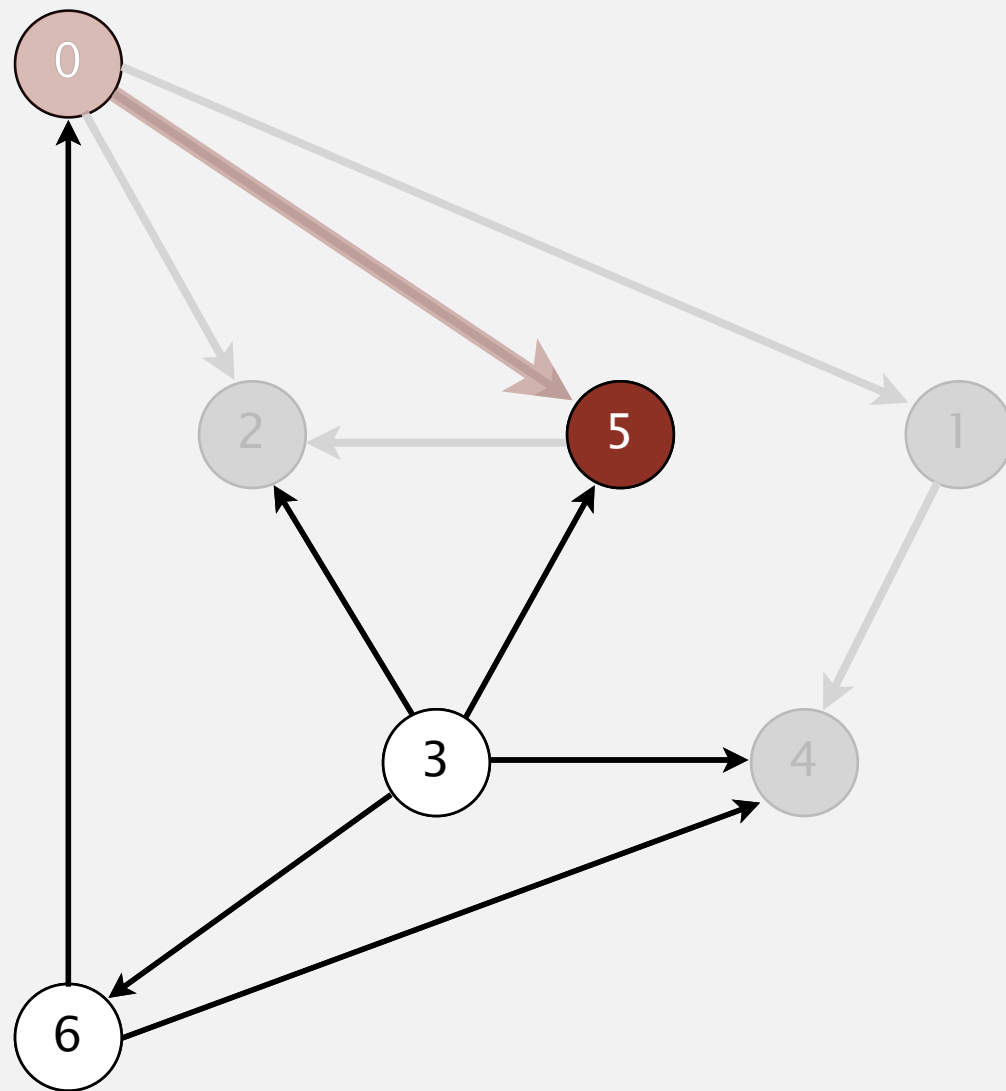
4 1 2

v	marked[]
0	T
1	T
2	T
3	F
4	T
5	T
6	F

visit 5: check 2

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

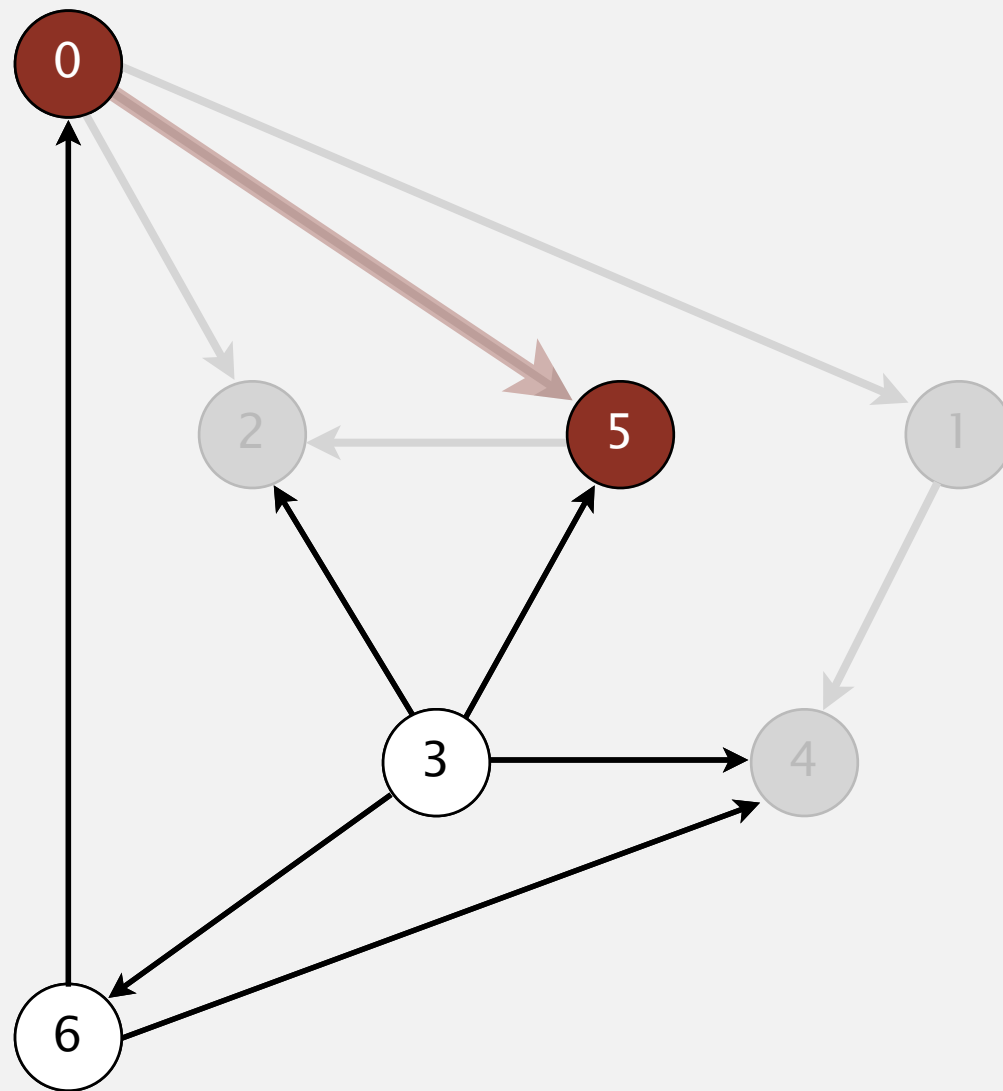
4 1 2

v	marked[]
0	T
1	T
2	T
3	F
4	T
5	T
6	F

visit 5

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

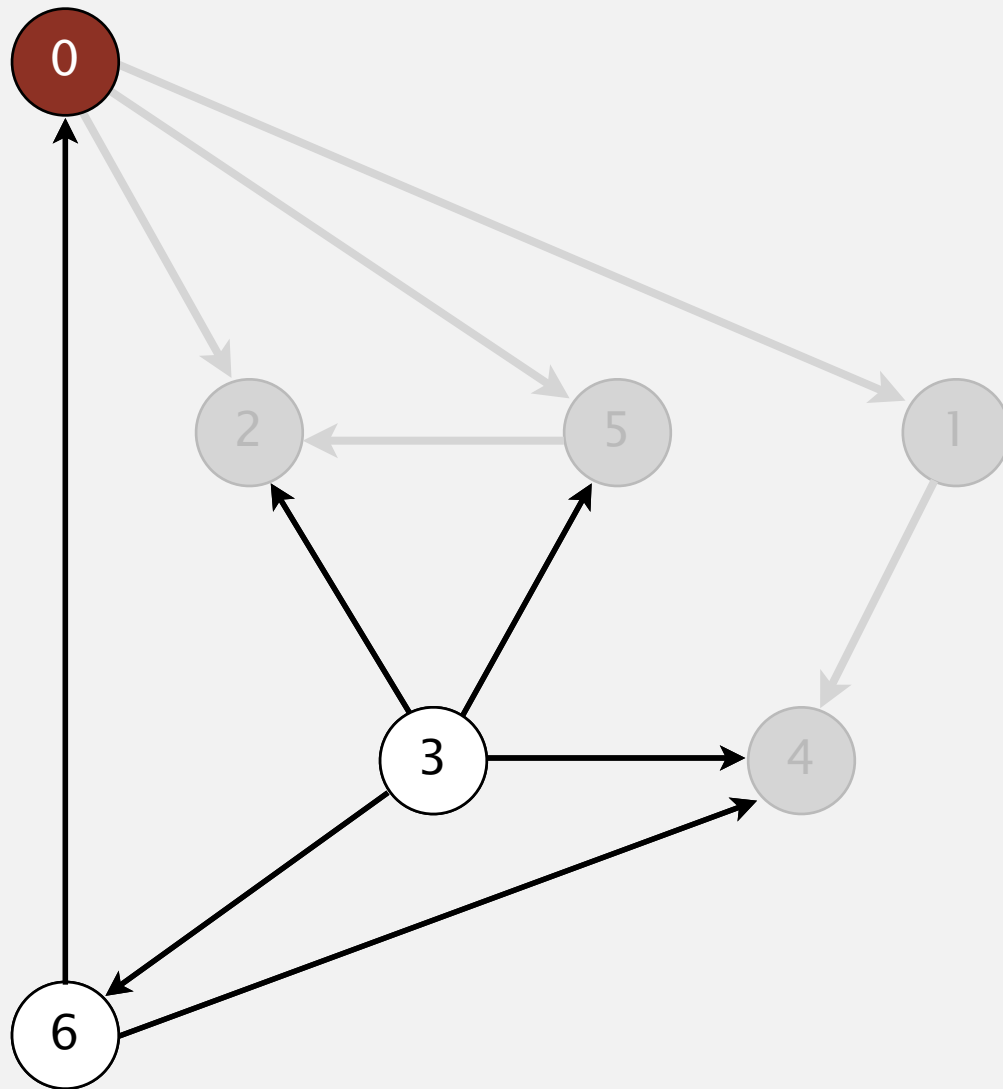
4 1 2 5

v	marked[]
0	T
1	T
2	T
3	F
4	T
5	T
6	F

5 done

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

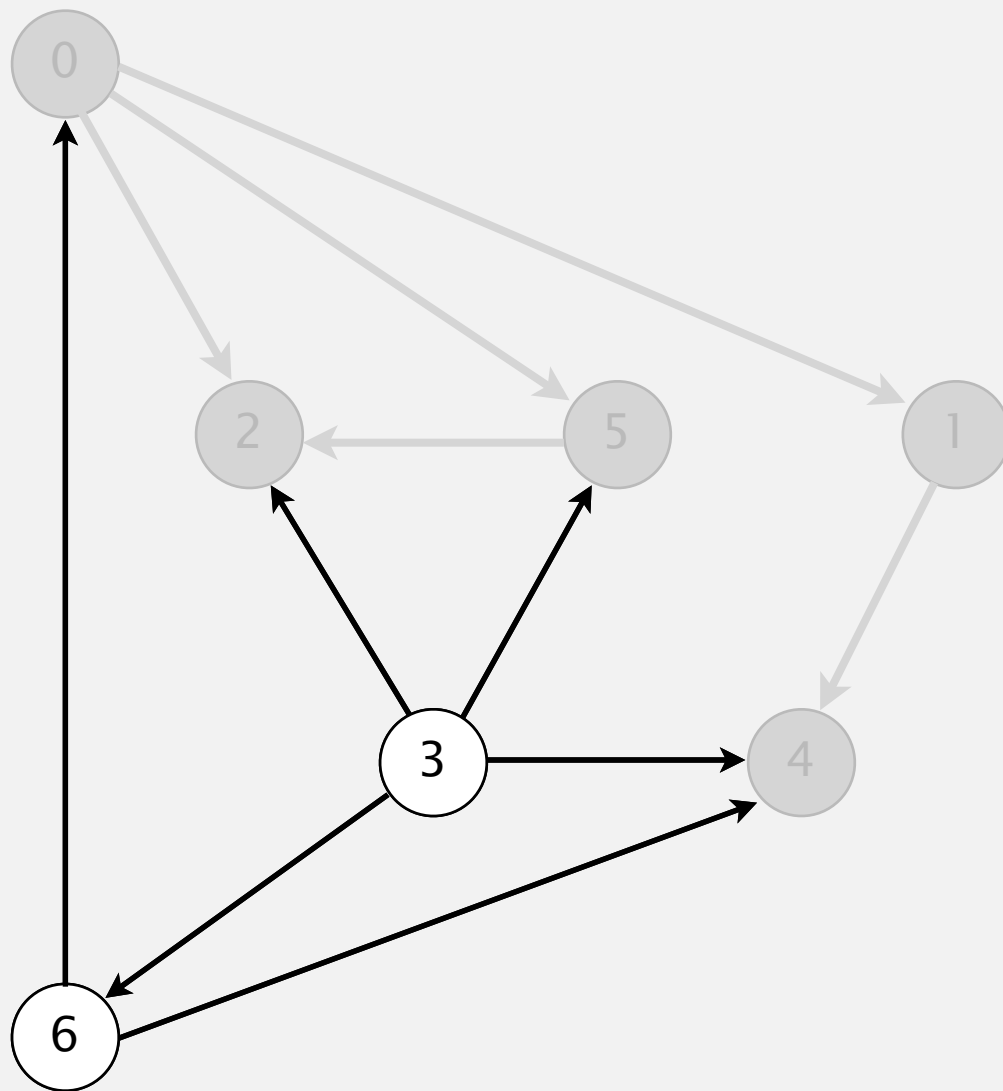
4 1 2 5 0

v	marked[]
0	T
1	T
2	T
3	F
4	T
5	T
6	F

0 done

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

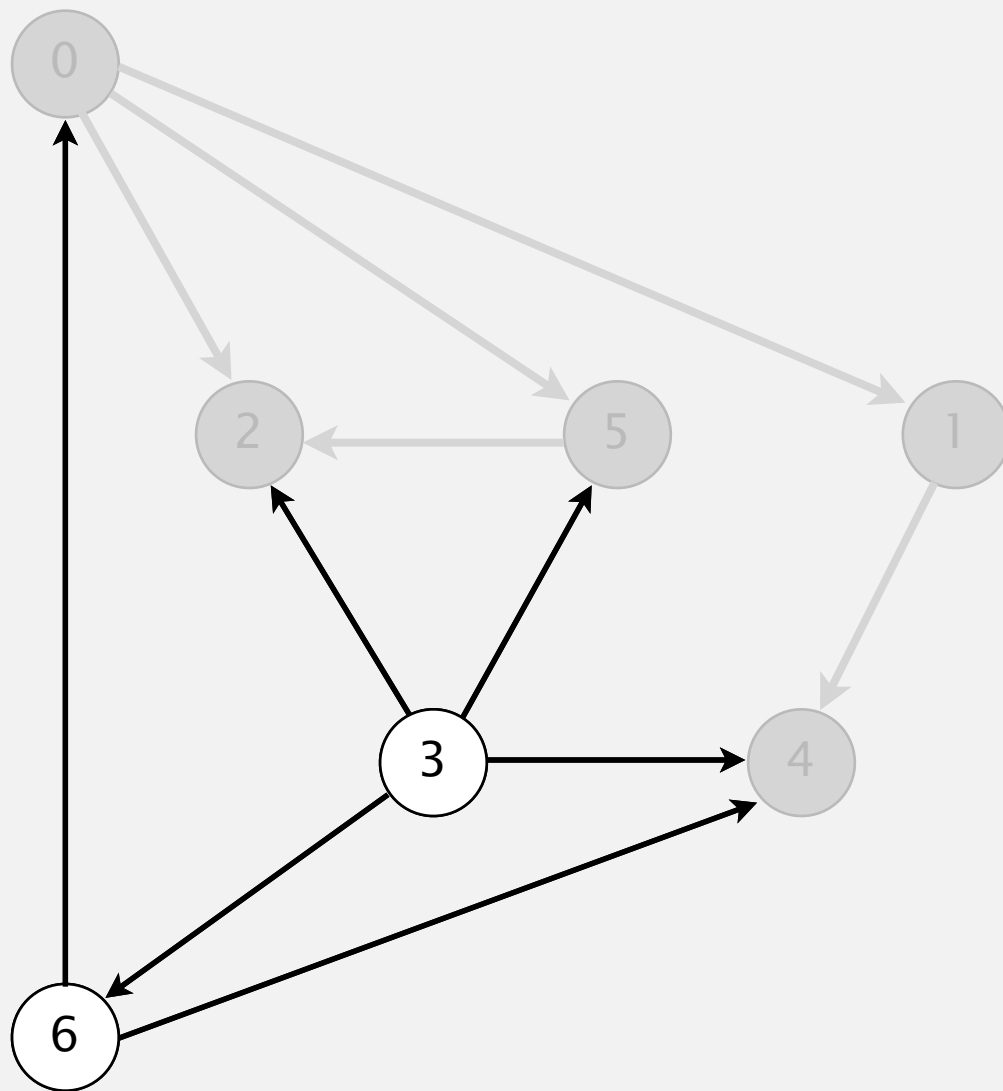
4 1 2 5 0

v	marked[]
0	T
1	T
2	T
3	F
4	T
5	T
6	F

check 1

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

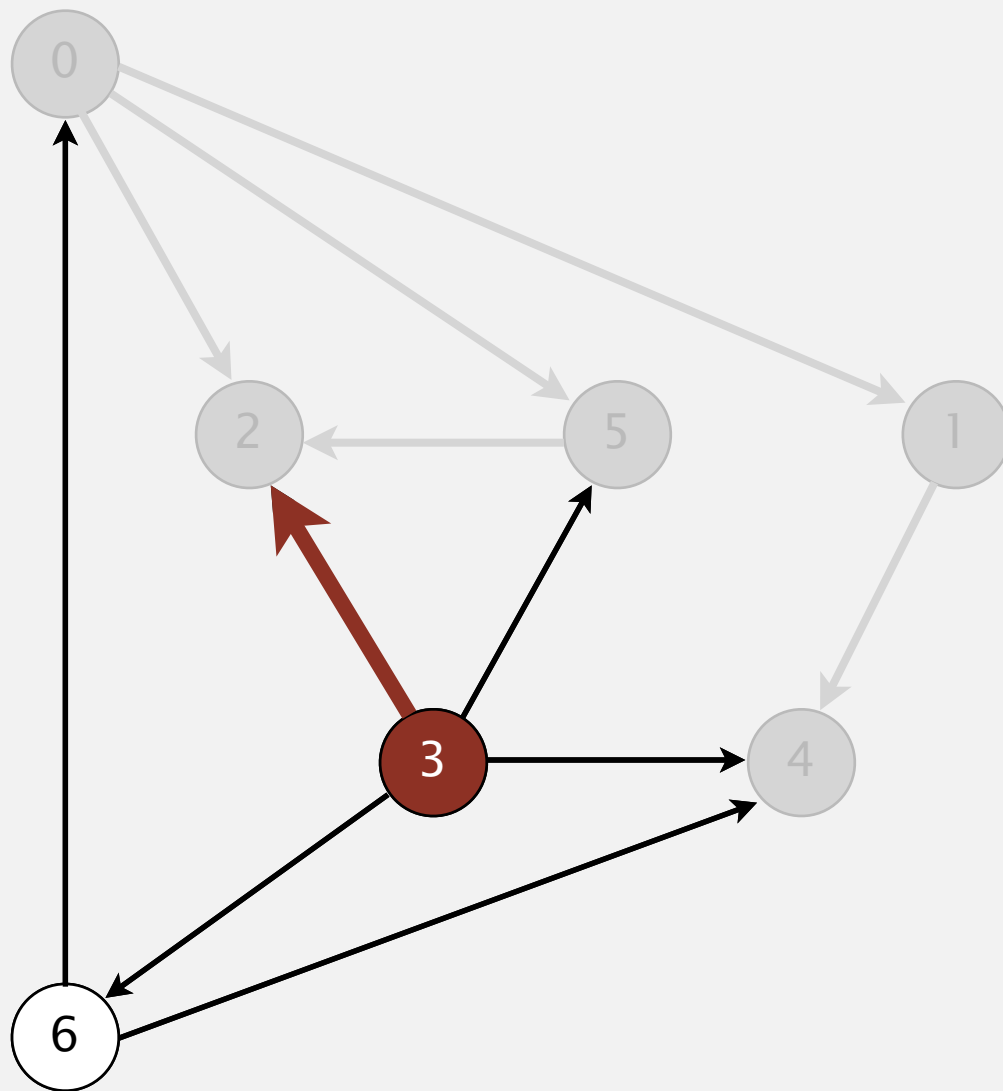
4 1 2 5 0

v	marked[]
0	T
1	T
2	T
3	F
4	T
5	T
6	F

check 2

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

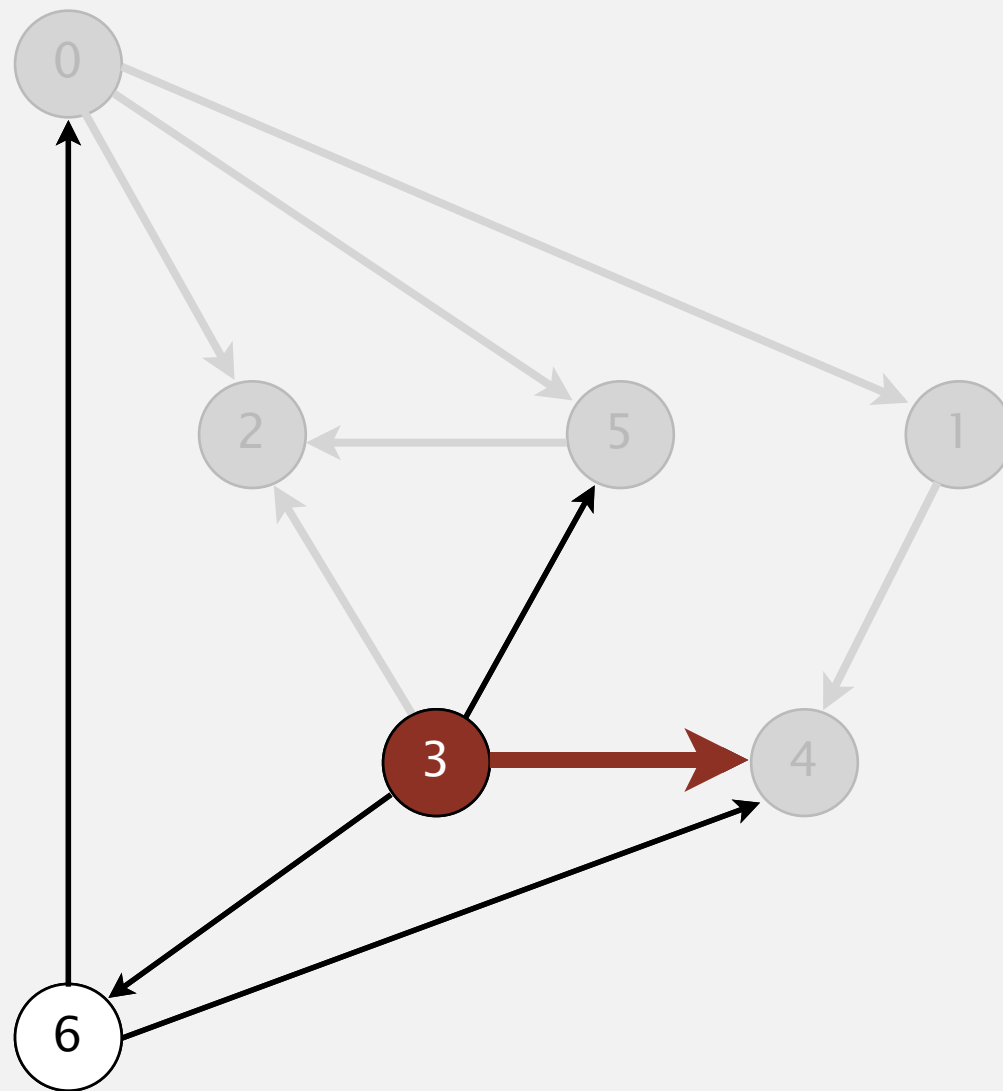
4 1 2 5 0

v	marked[]
0	T
1	T
2	T
3	T
4	T
5	T
6	F

visit 3: check 2, check 4, check 5, and check 6

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

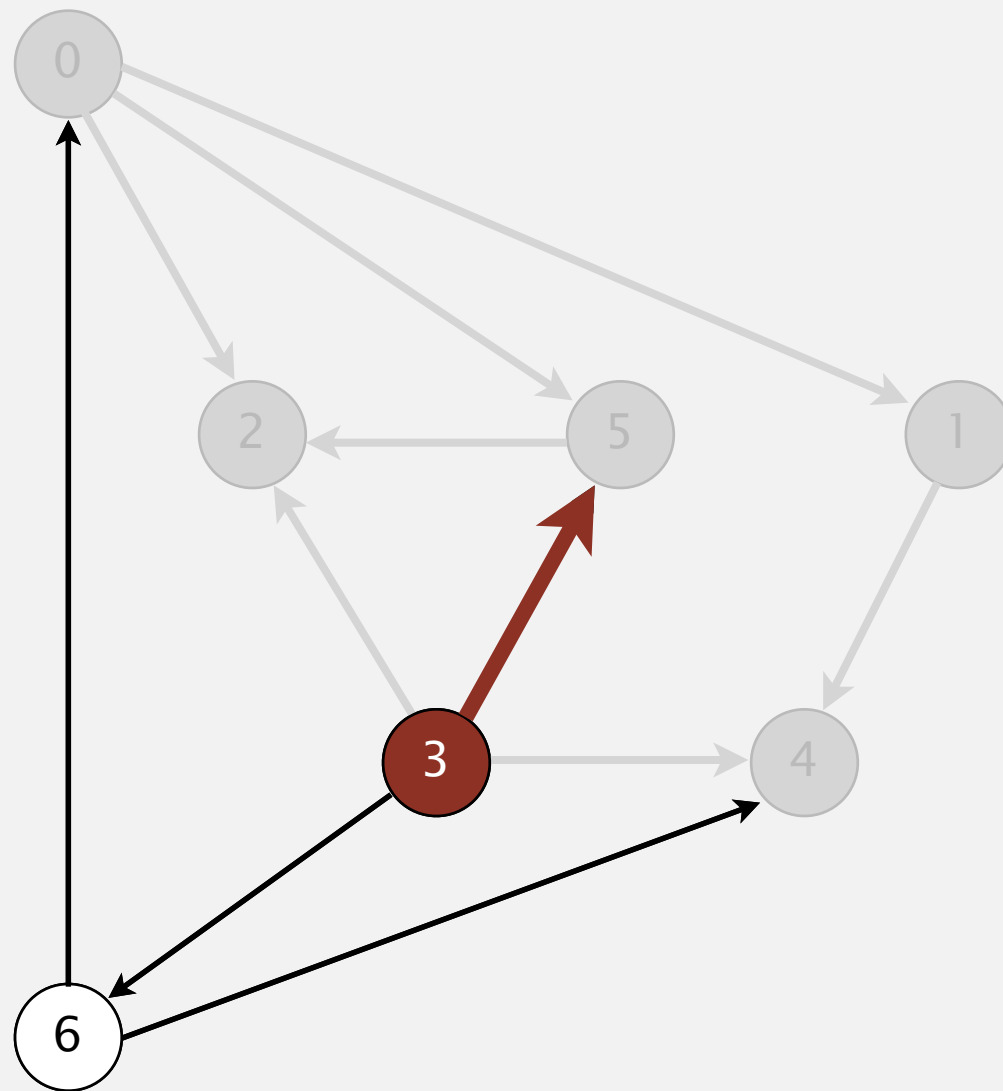
4 1 2 5 0

v	marked[]
0	T
1	T
2	T
3	T
4	T
5	T
6	F

visit 3: check 2, **check 4**, check 5, and check 6

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

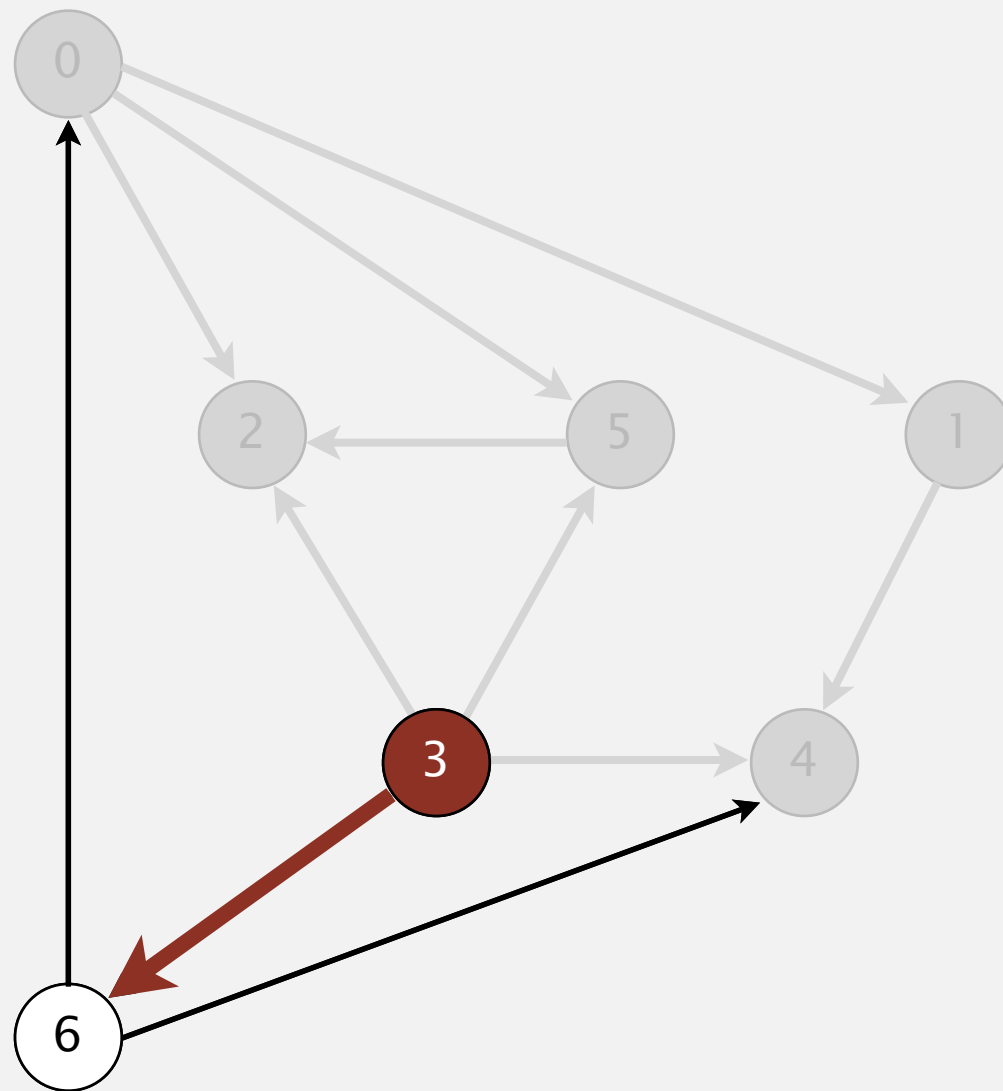
4 1 2 5 0

v	marked[]
0	T
1	T
2	T
3	T
4	T
5	T
6	F

visit 3: check 2, check 4, **check 5**, and check 6

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

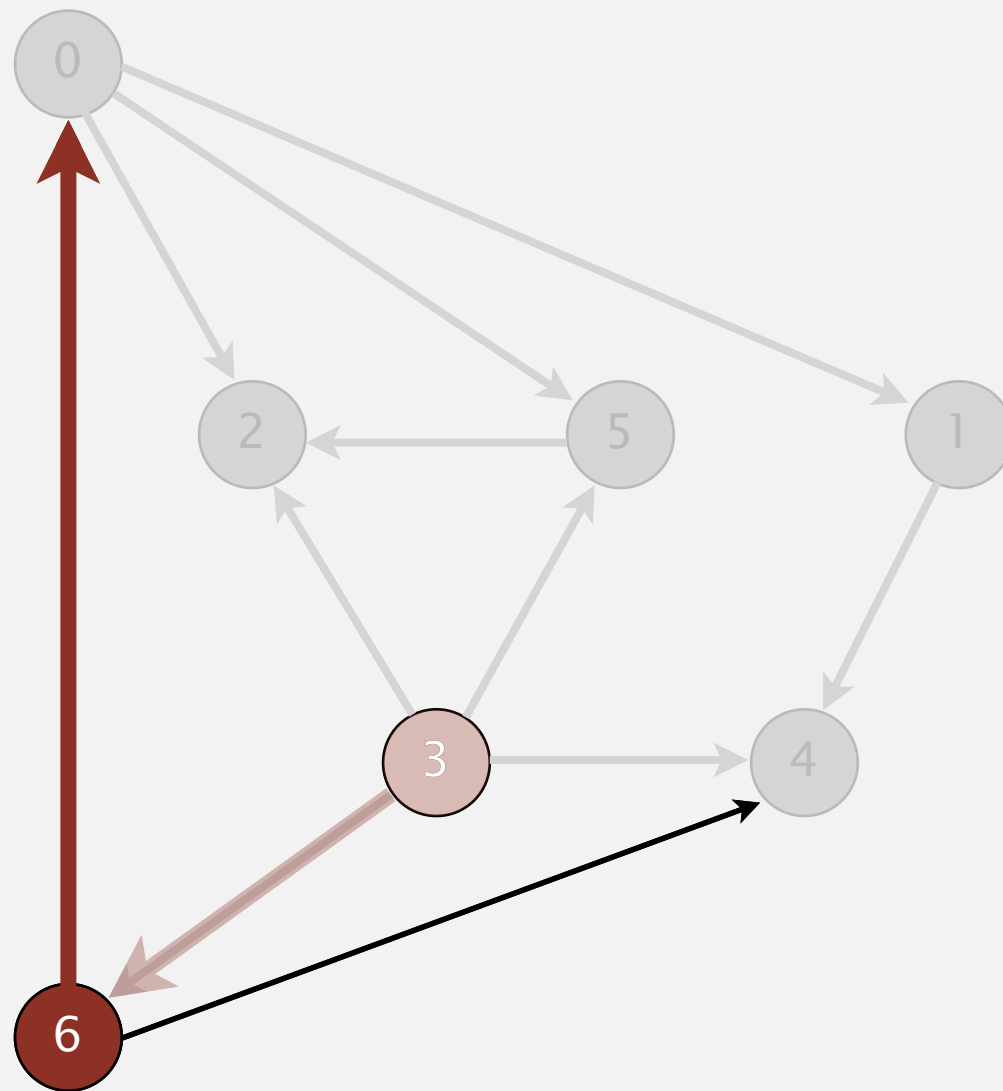
4 1 2 5 0

v	marked[]
0	T
1	T
2	T
3	T
4	T
5	T
6	F

visit 3: check 2, check 4, check 5, and **check 6**

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

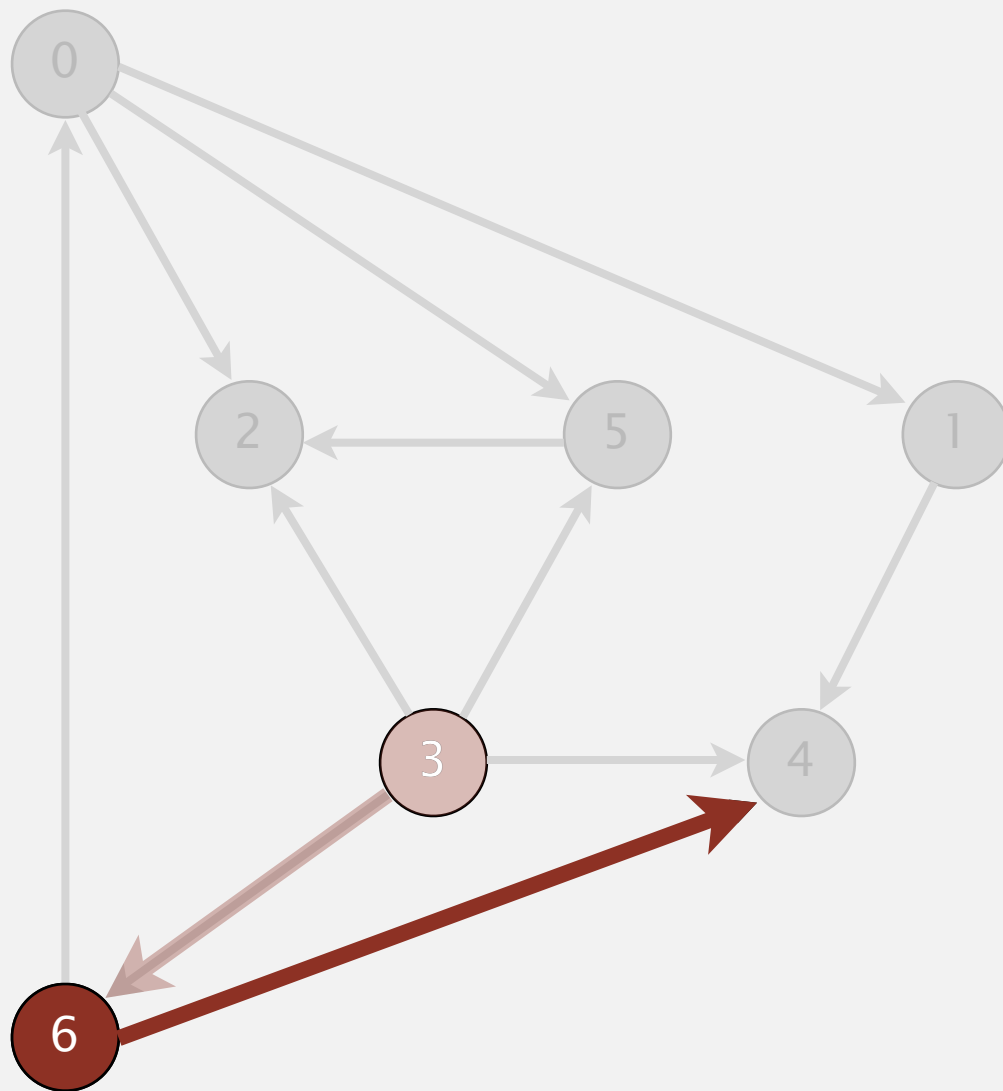
4 1 2 5 0

v	marked[]
0	T
1	T
2	T
3	T
4	T
5	T
6	T

visit 6: check 0 and check 4

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

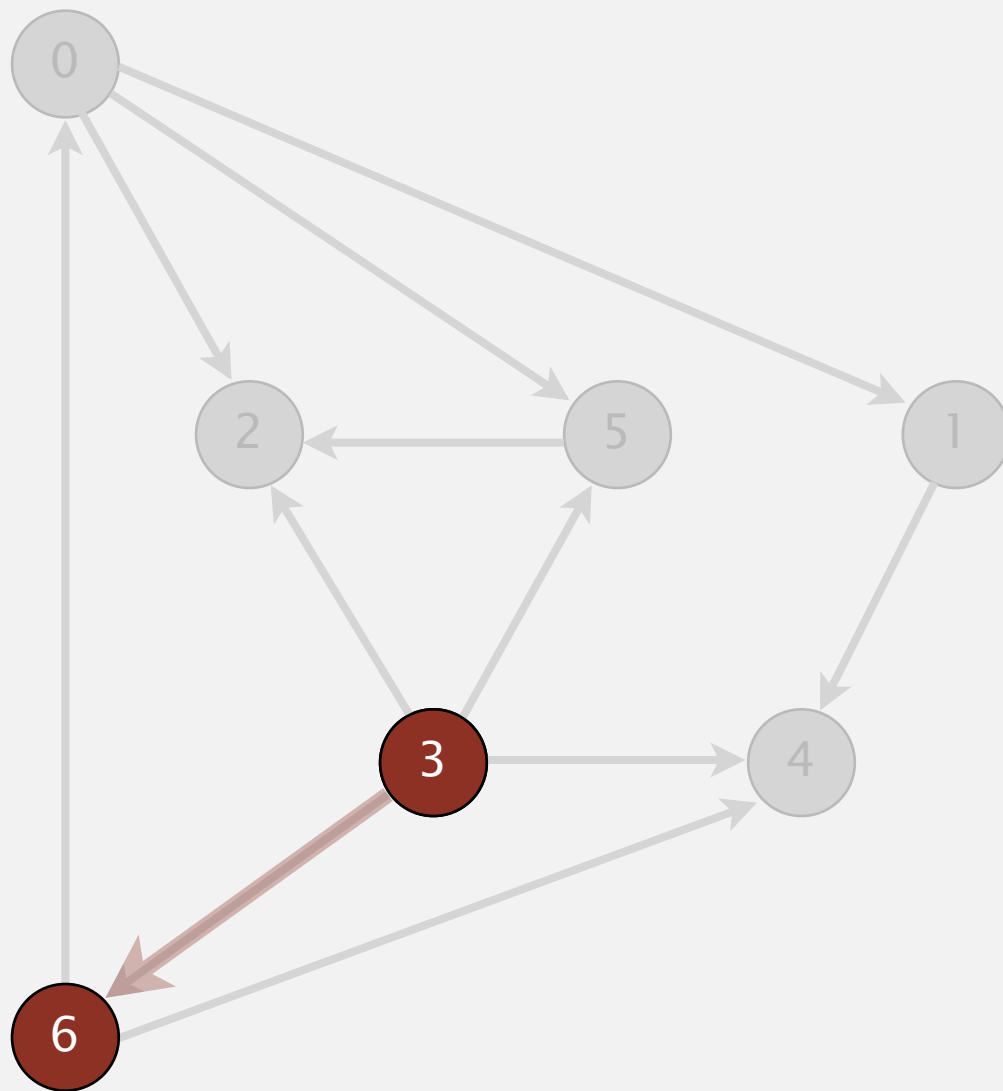
4 1 2 5 0

v	marked[]
0	T
1	T
2	T
3	T
4	T
5	T
6	T

visit 6: check 0 and **check 4**

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

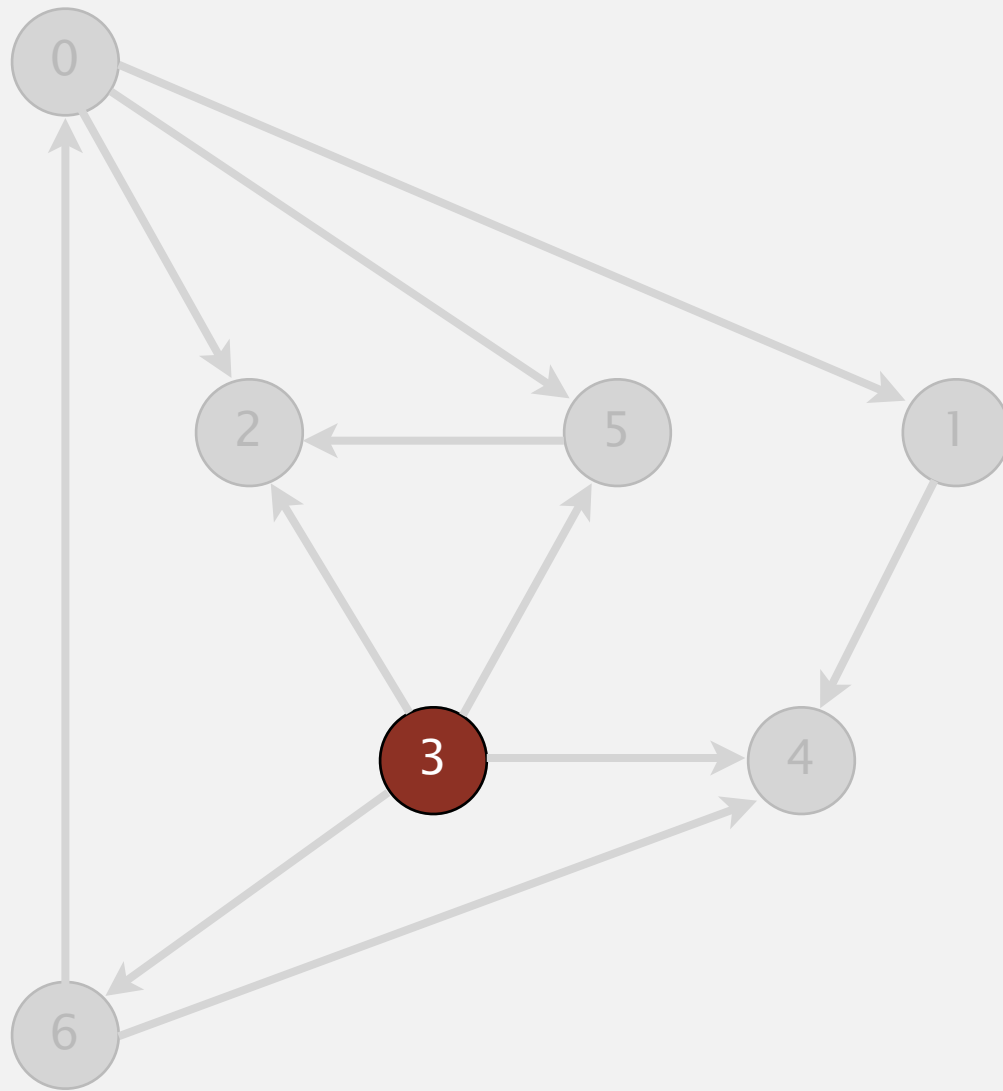
4 1 2 5 0 6

v	marked[]
0	T
1	T
2	T
3	T
4	T
5	T
6	T

6 done

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

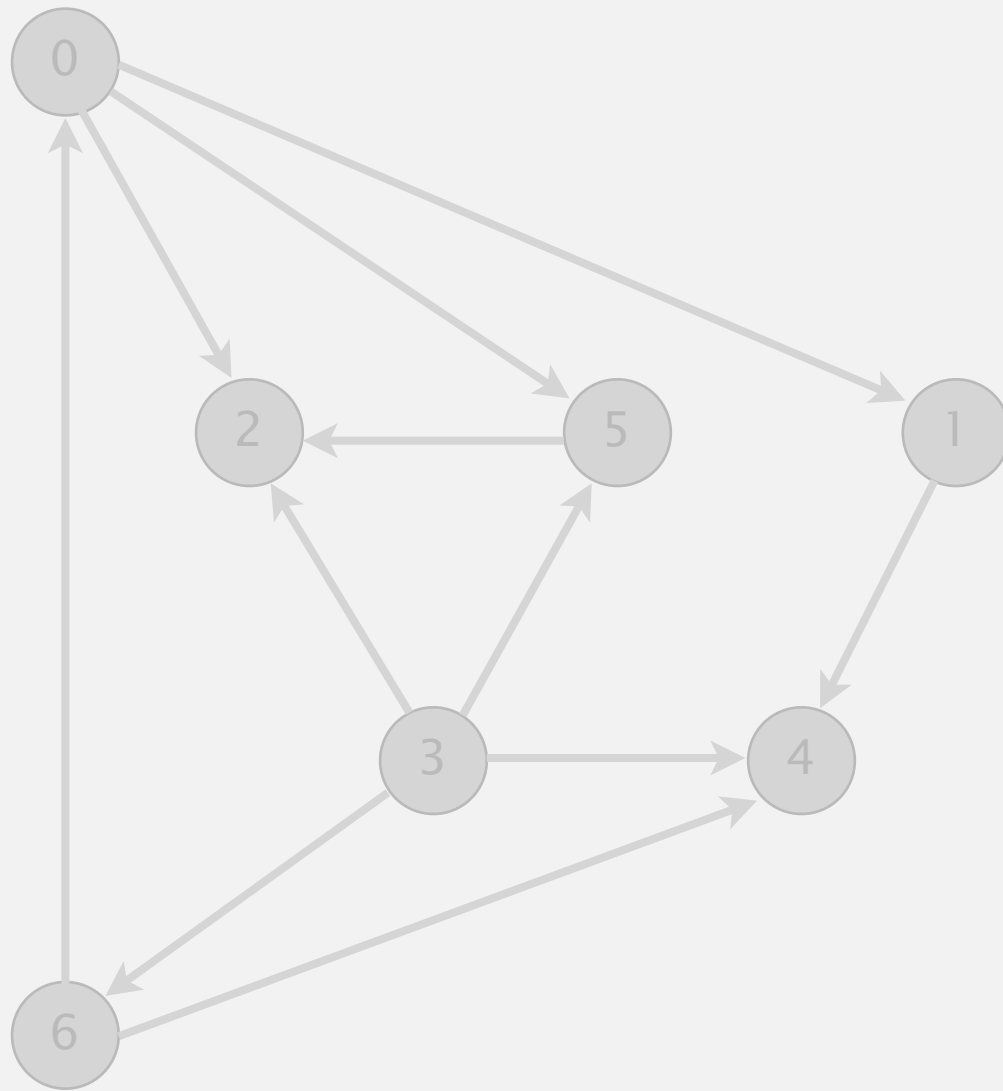
4 1 2 5 0 6 3

v	marked[]
0	T
1	T
2	T
3	T
4	T
5	T
6	T

3 done

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

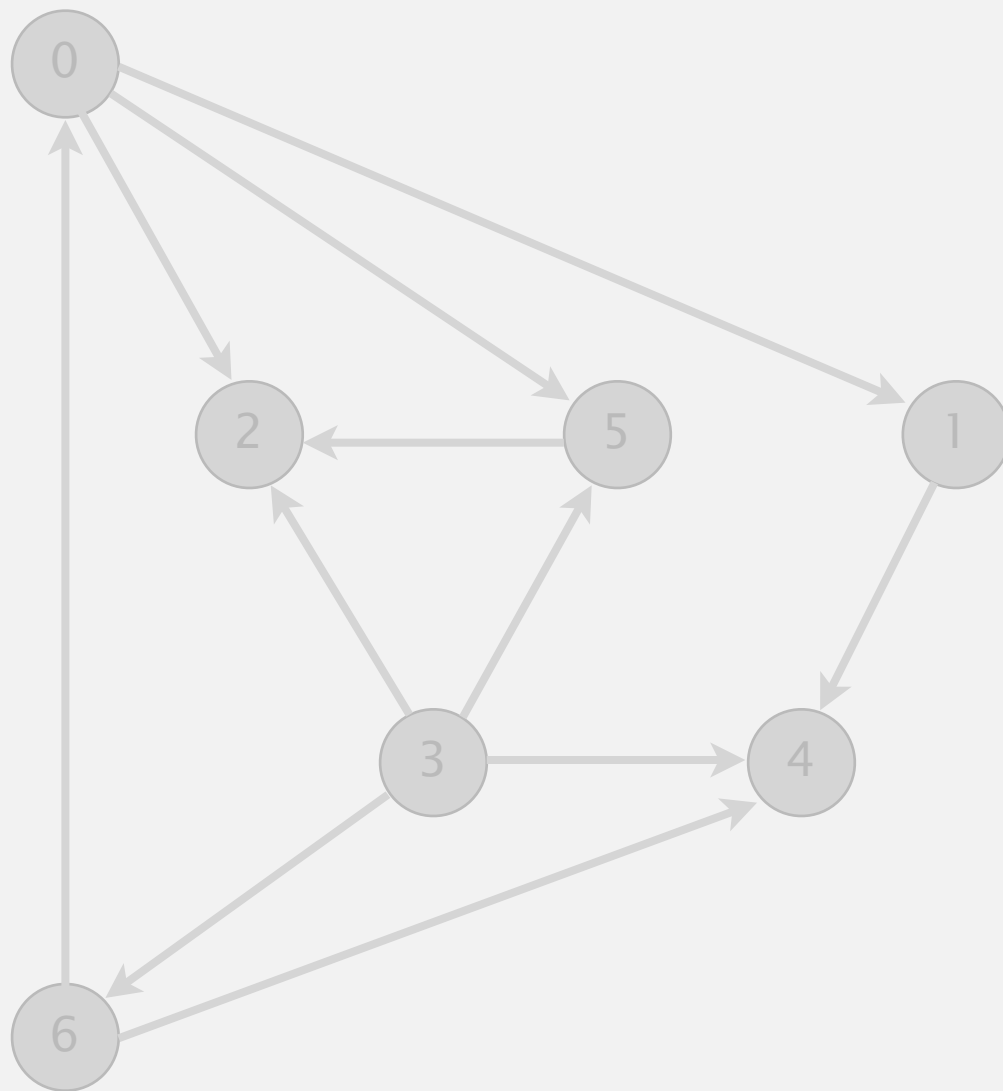
4 1 2 5 0 6 3

v	marked[]
0	T
1	T
2	T
3	T
4	T
5	T
6	T

check 4

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

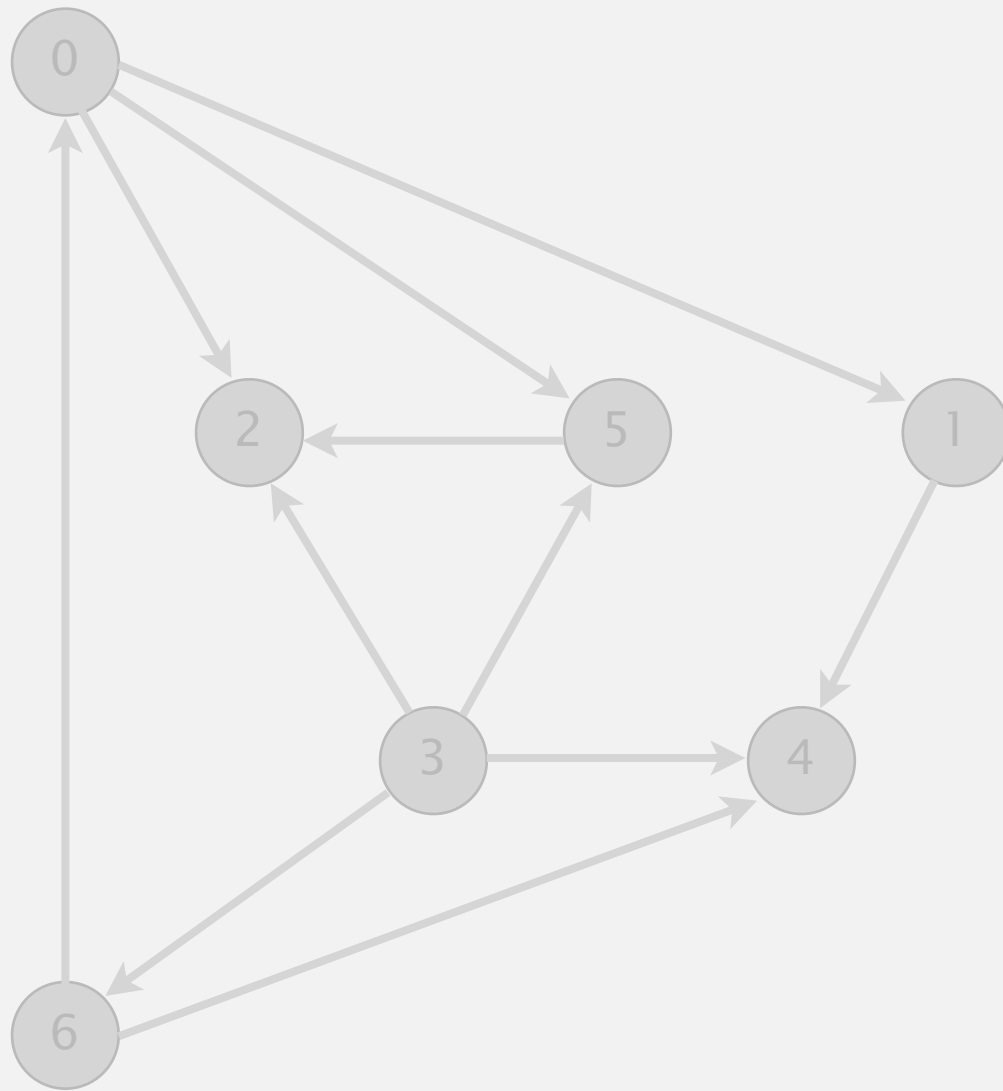
4 1 2 5 0 6 3

v	marked[]
0	T
1	T
2	T
3	T
4	T
5	T
6	T

check 5

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

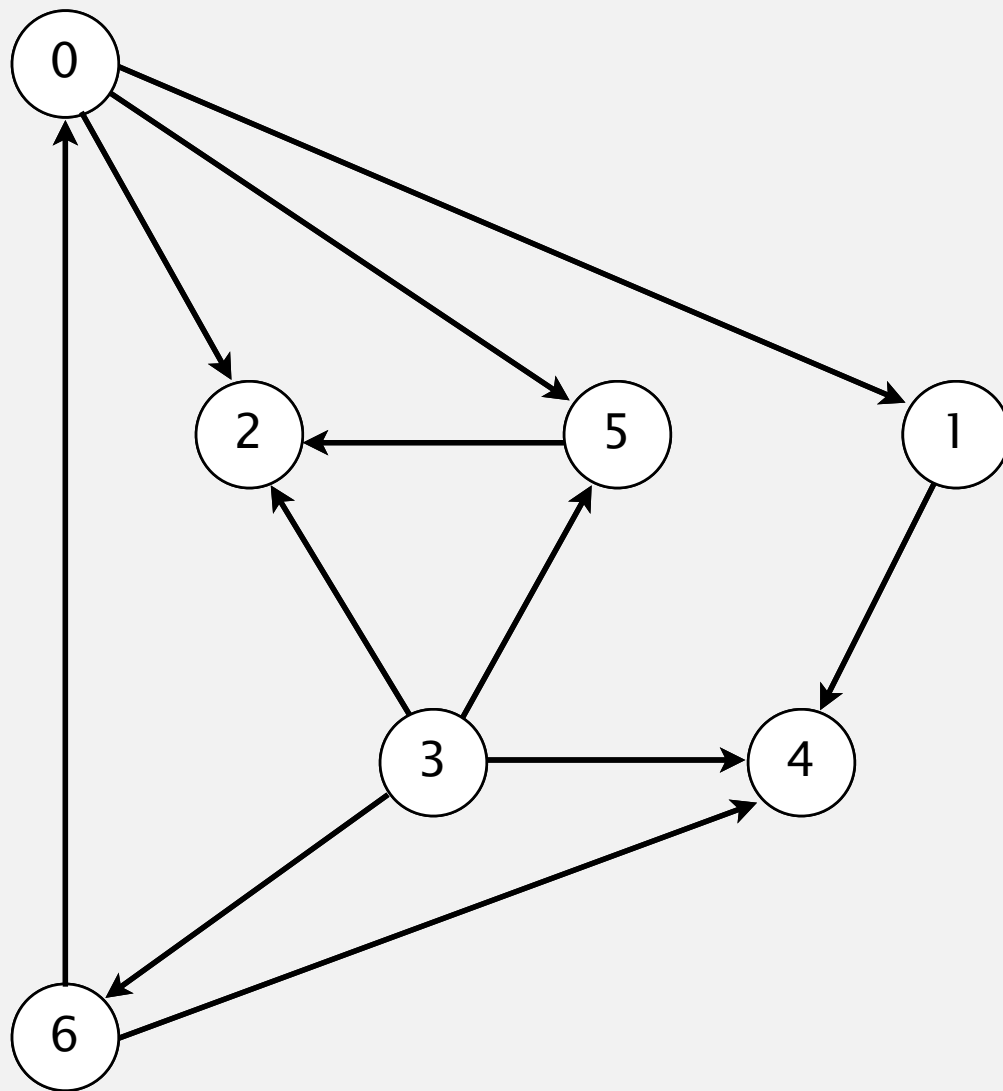
4 1 2 5 0 6 3

v	marked[]
0	T
1	T
2	T
3	T
4	T
5	T
6	T

check 6

Topological sort demo

- Run depth-first search.
- Return vertices in reverse postorder.



postorder

4 1 2 5 0 6 3

topological order

3 6 0 5 2 1 4

done