Algorithms

Robert Sedgewick | Kevin Wayne

Algorithms

Fourth Edition

Robert Sedgewick | Kevin Wayne

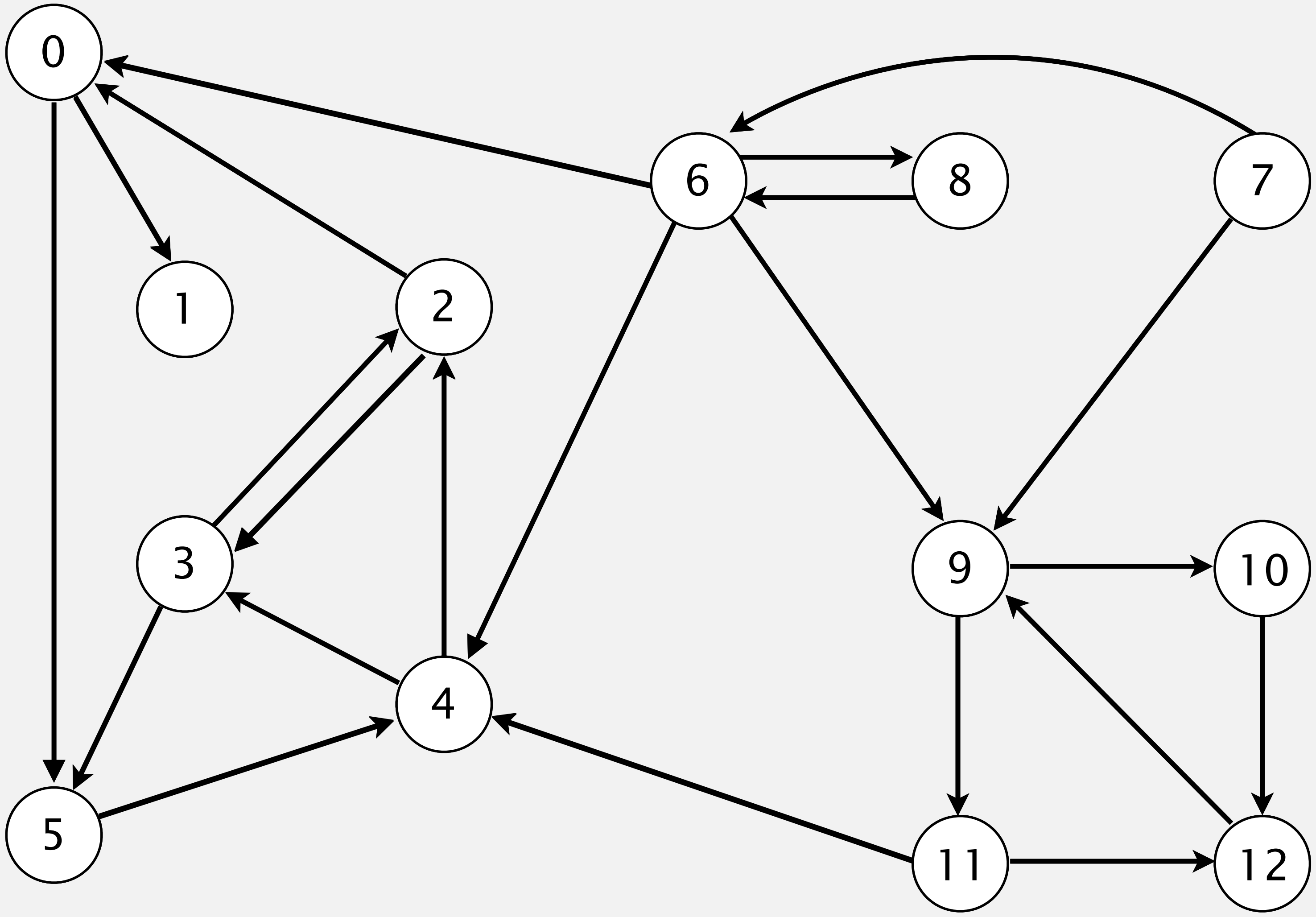
http://algs4.cs.princeton.edu

4.2 Kosaraju-Sharir Demo

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.



digraph G

4→2
2→3
3→2
6→0
0→1
2→0
11→12
12→9
9→10
9→11
7→9
10→12
11→4
4→3
3→5
6→8
8→6
5→4
0→5
6→4
6→9
7→6

Algorithms

Robert Sedgewick | Kevin Wayne

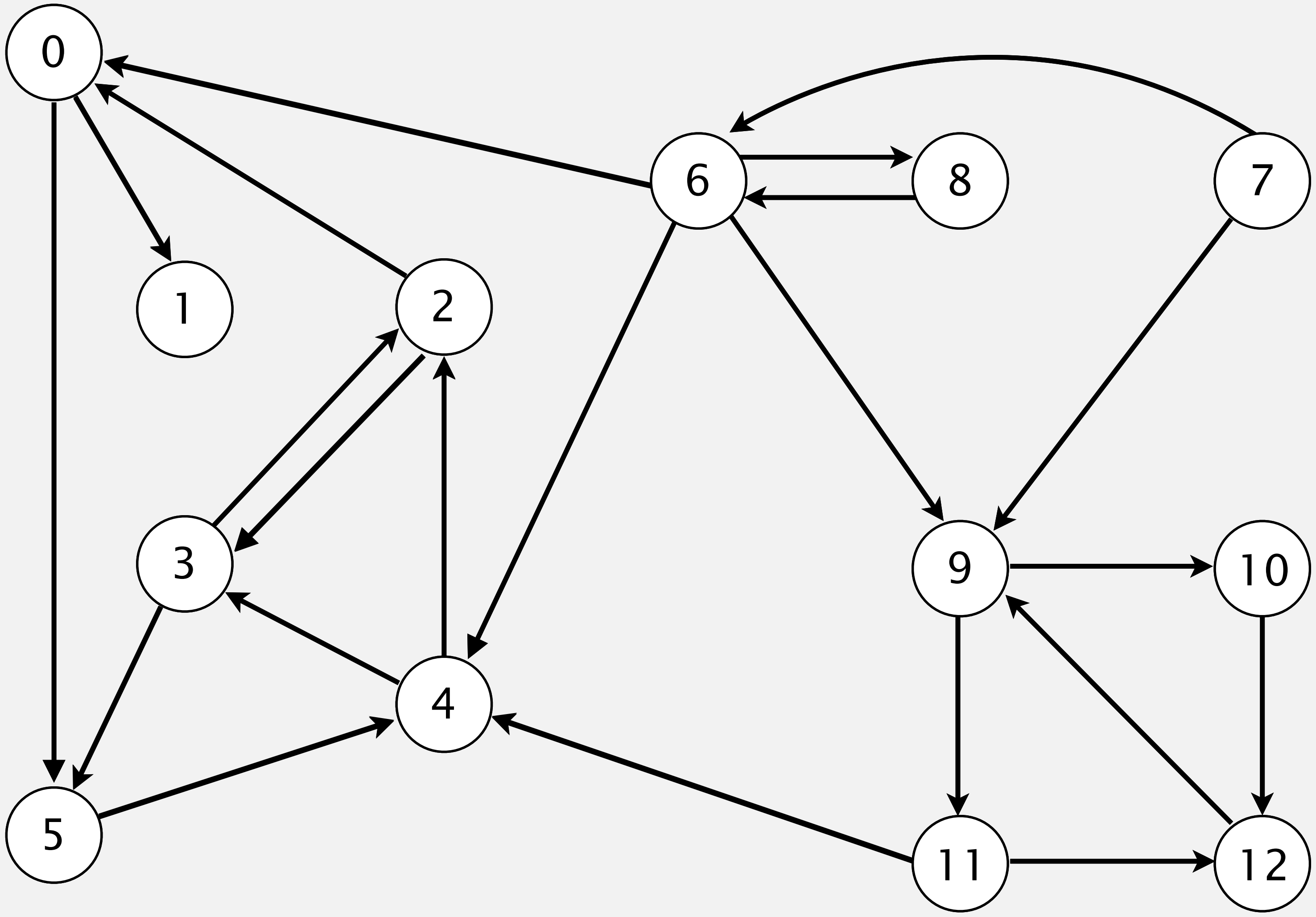
http://algs4.cs.princeton.edu

4.2 Kosaraju-Sharir Demo

- *DFS in reverse graph*
- *DFS in original graph*

Kosaraju-Sharir algorithm demo

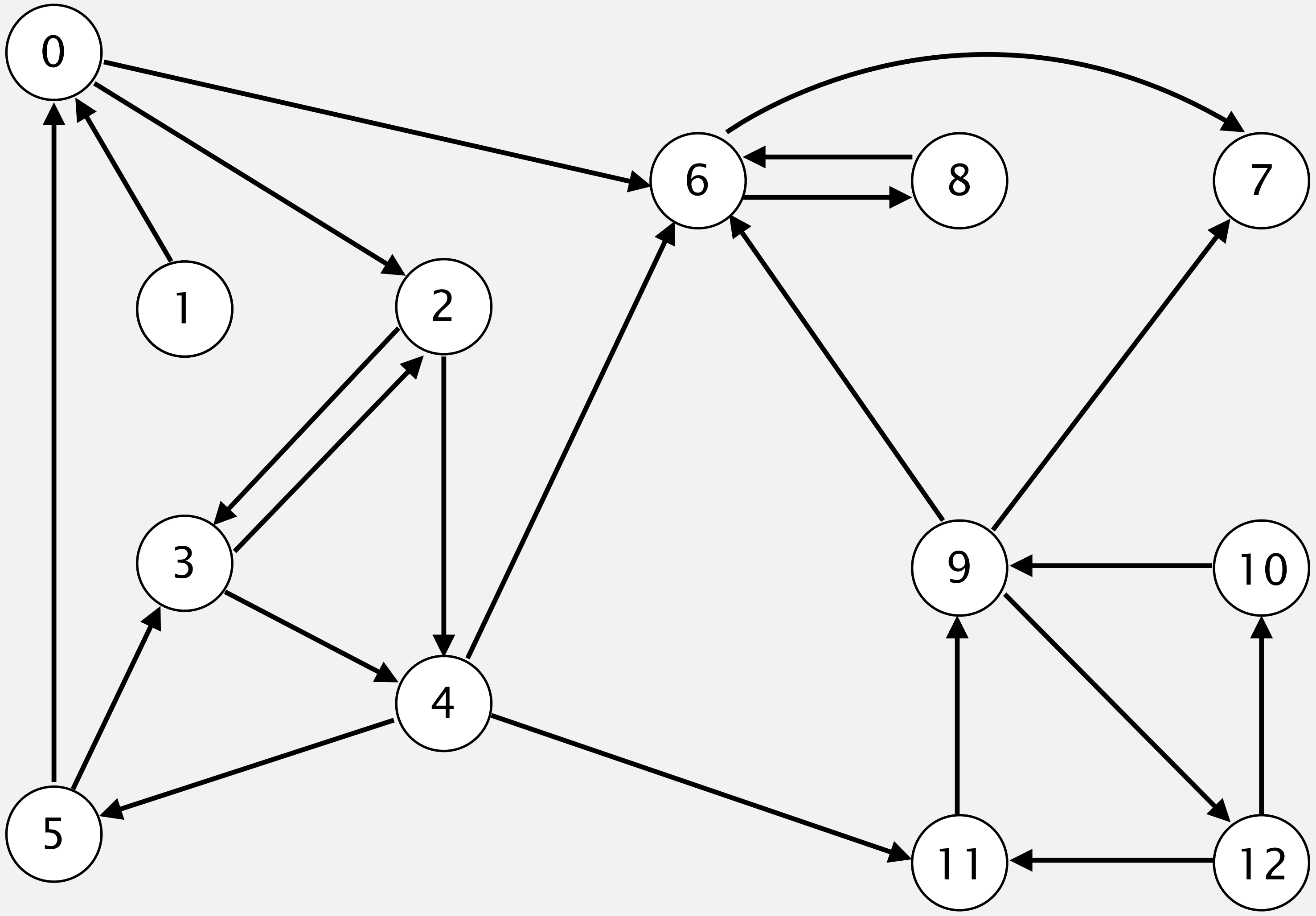
Phase 1. Compute reverse postorder in G^R.



digraph G

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

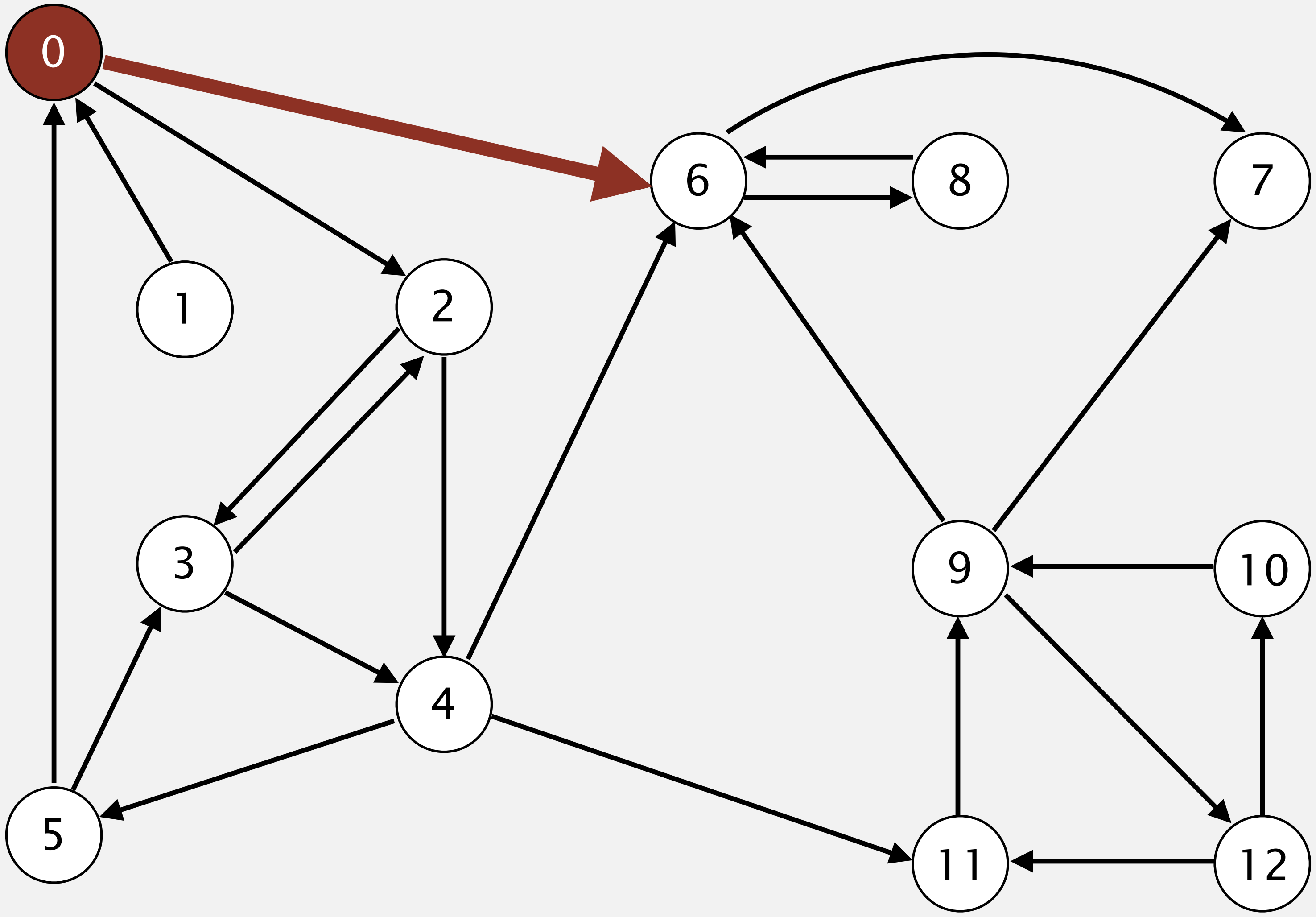


reverse digraph G^R

v	marked[]
0	–
1	–
2	–
3	–
4	–
5	–
6	–
7	–
8	–
9	–
10	–
11	–
12	–

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

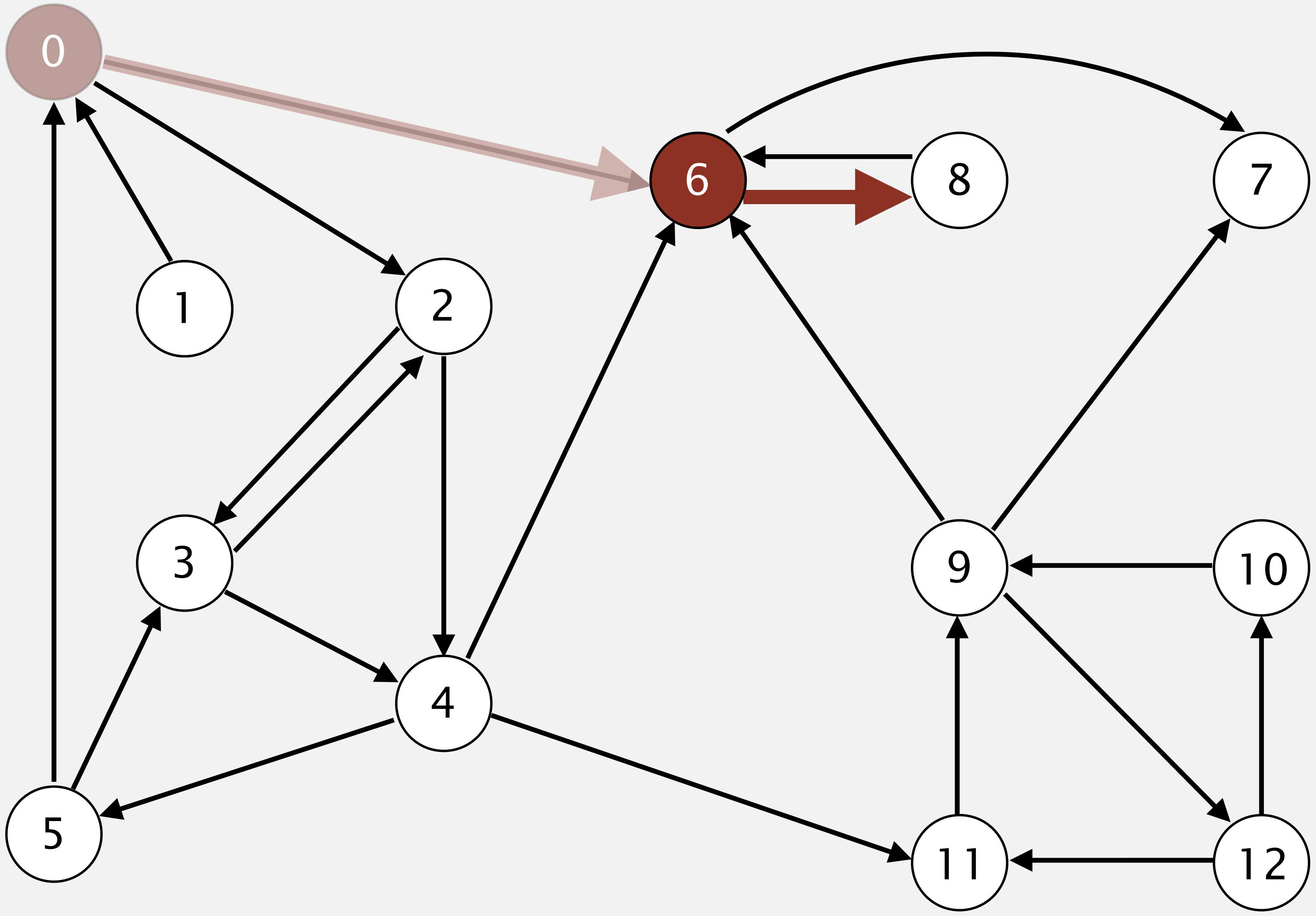


visit 0: check 6 and check 2

v	marked[]
0	T
1	F
2	F
3	F
4	F
5	F
6	F
7	F
8	F
9	F
10	F
11	F
12	F

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

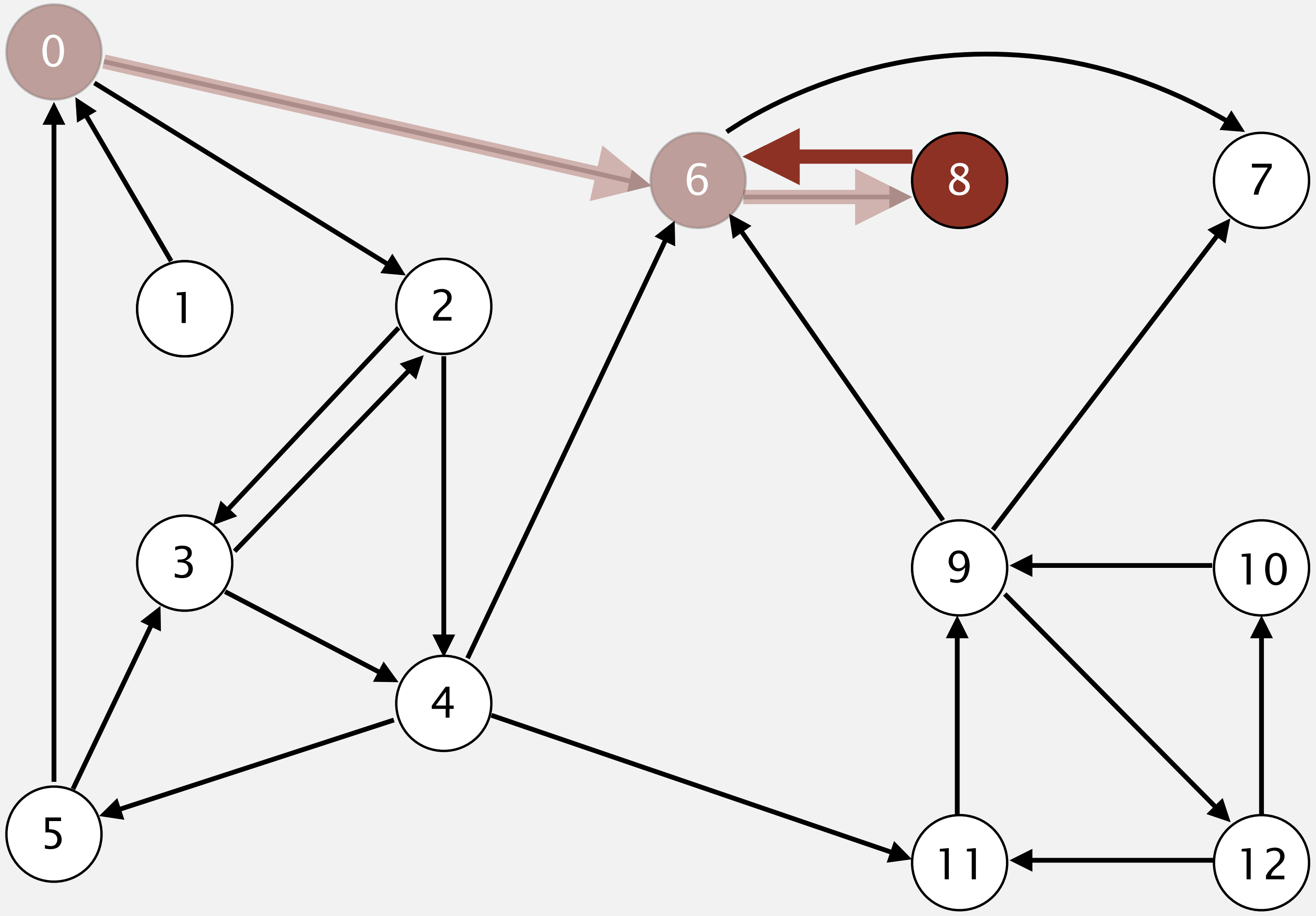


visit 6: check 8 and check 7

v	marked[]
0	T
1	F
2	F
3	F
4	F
5	F
6	T
7	F
8	F
9	F
10	F
11	F
12	F

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

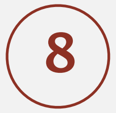


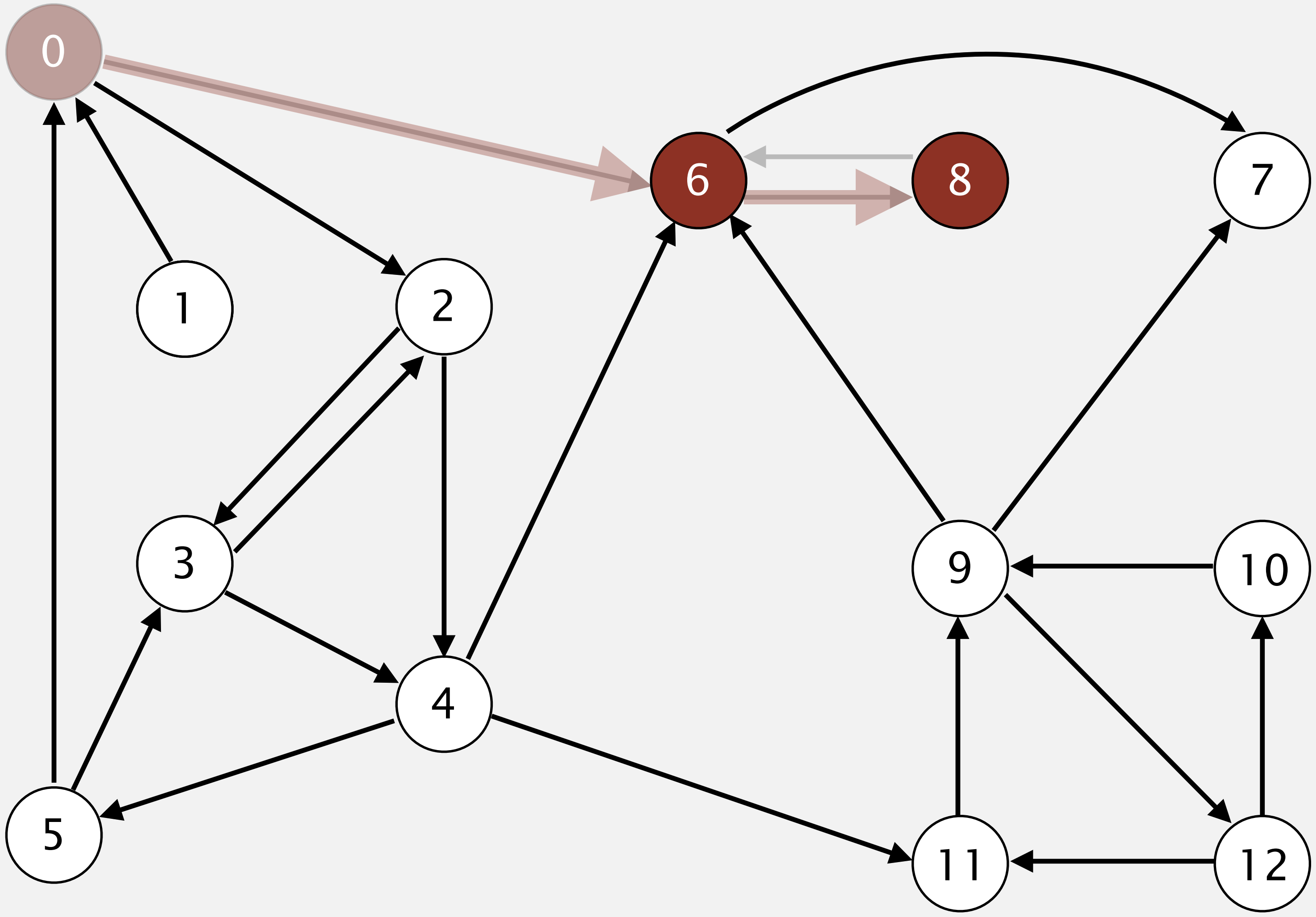
v	marked[]
0	T
1	F
2	F
3	F
4	F
5	F
6	T
7	F
8	T
9	F
10	F
11	F
12	F

visit 8: check 6

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.





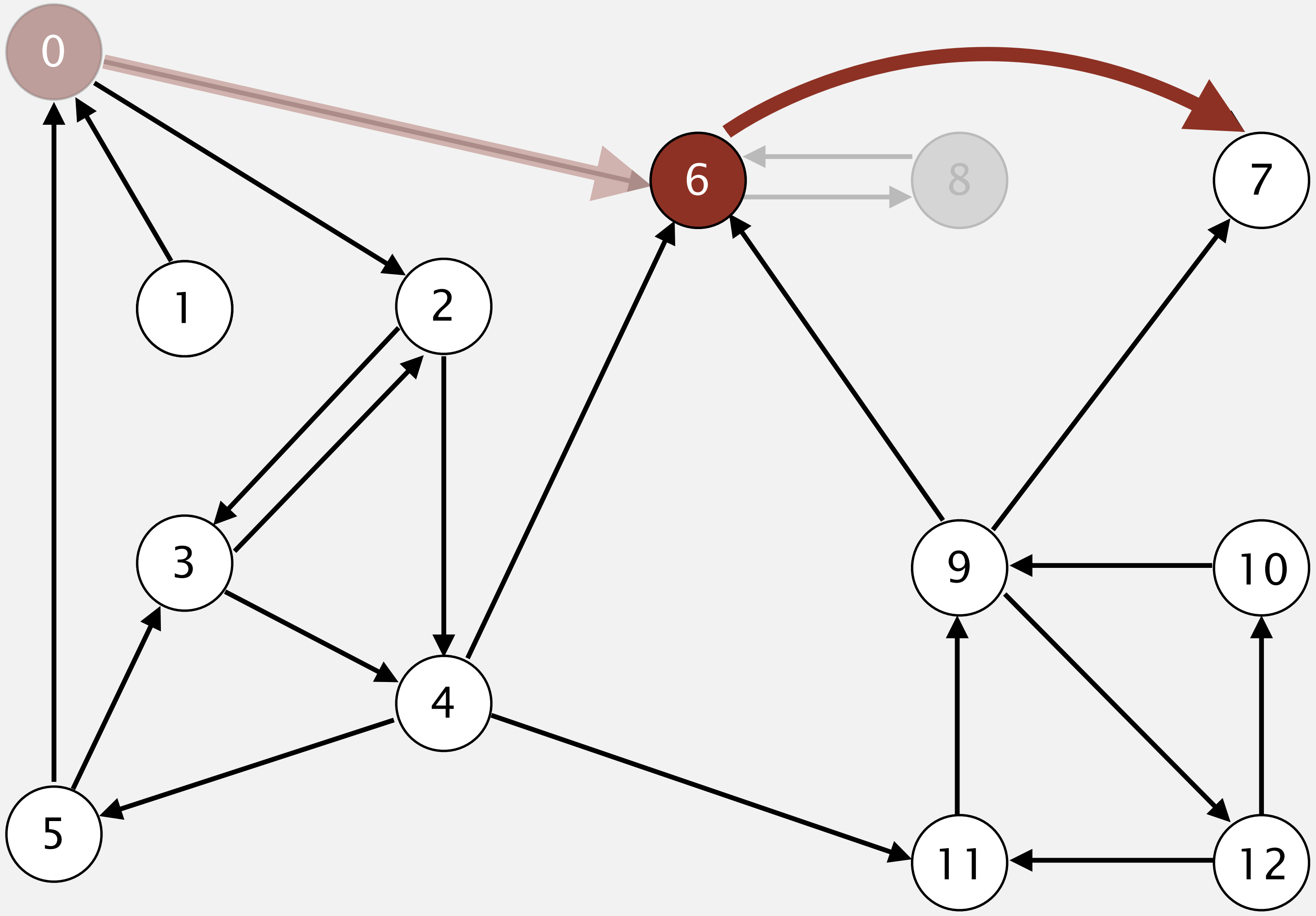
v	marked[]
0	T
1	F
2	F
3	F
4	F
5	F
6	T
7	F
8	T
9	F
10	F
11	F
12	F

8 done

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

8



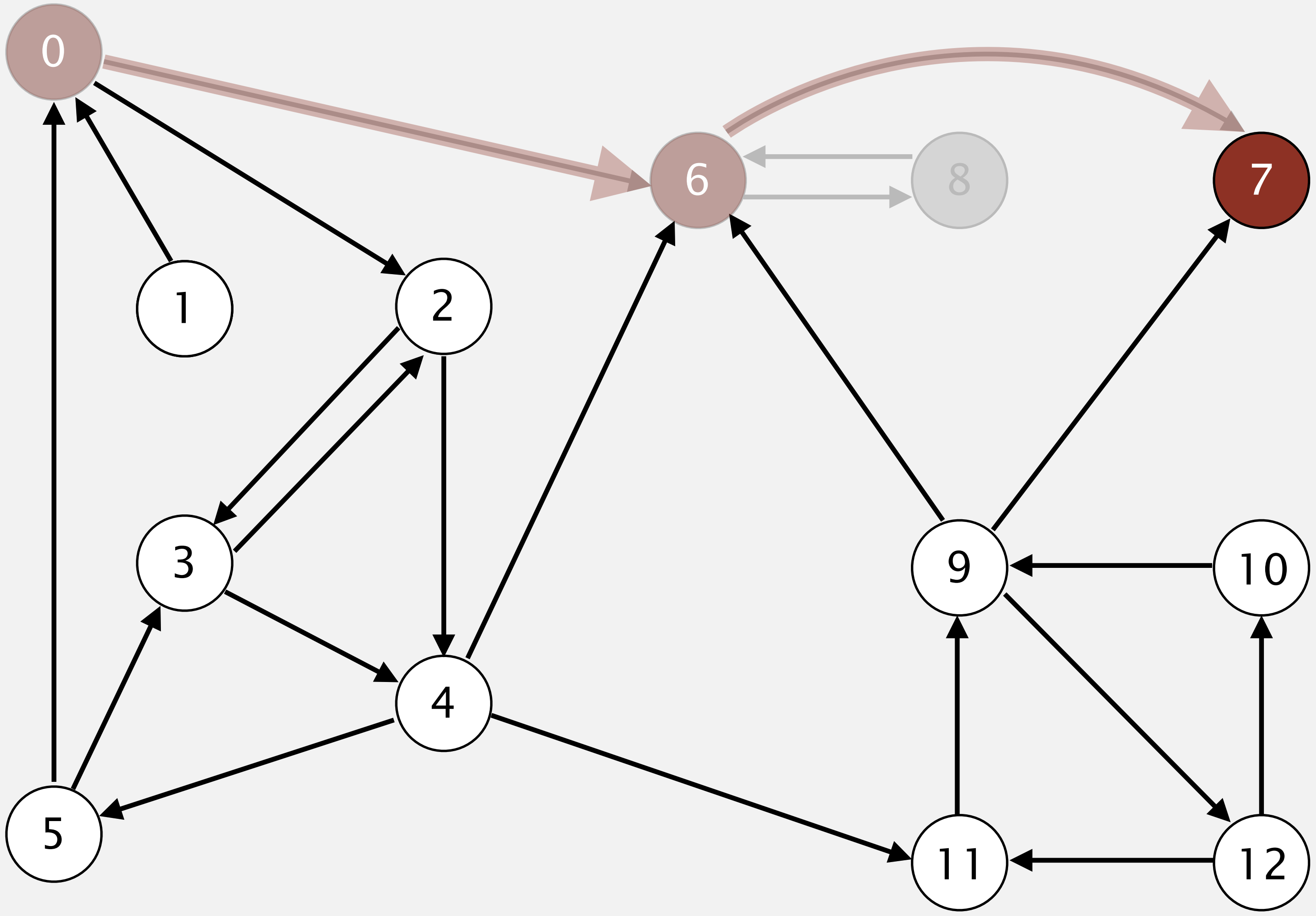
v	marked[]
0	T
1	F
2	F
3	F
4	F
5	F
6	T
7	F
8	T
9	F
10	F
11	F
12	F

visit 6: check 8 and check 7

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

8



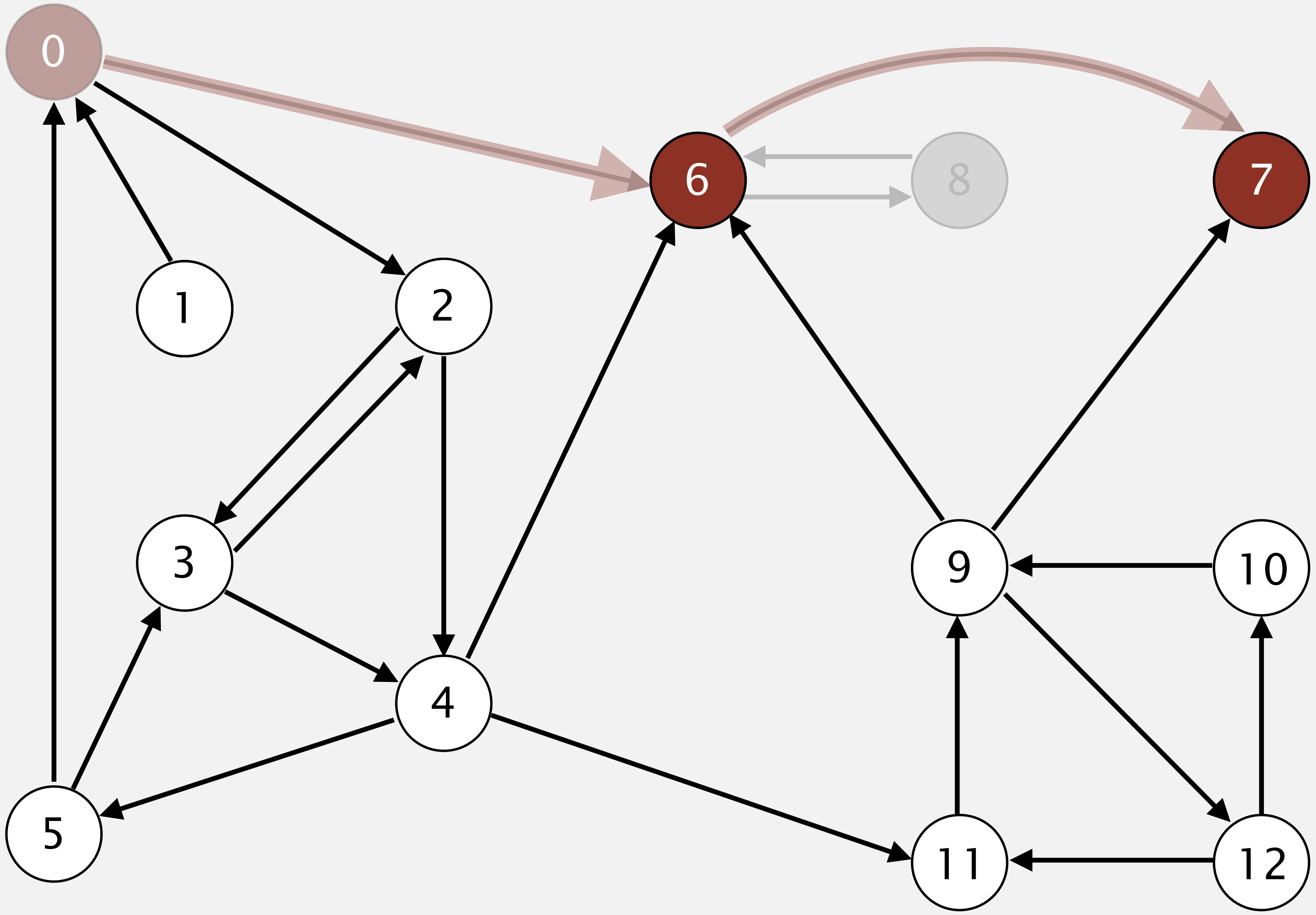
v	marked[]
0	T
1	F
2	F
3	F
4	F
5	F
6	T
7	T
8	T
9	F
10	F
11	F
12	F

visit 7

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

7 8



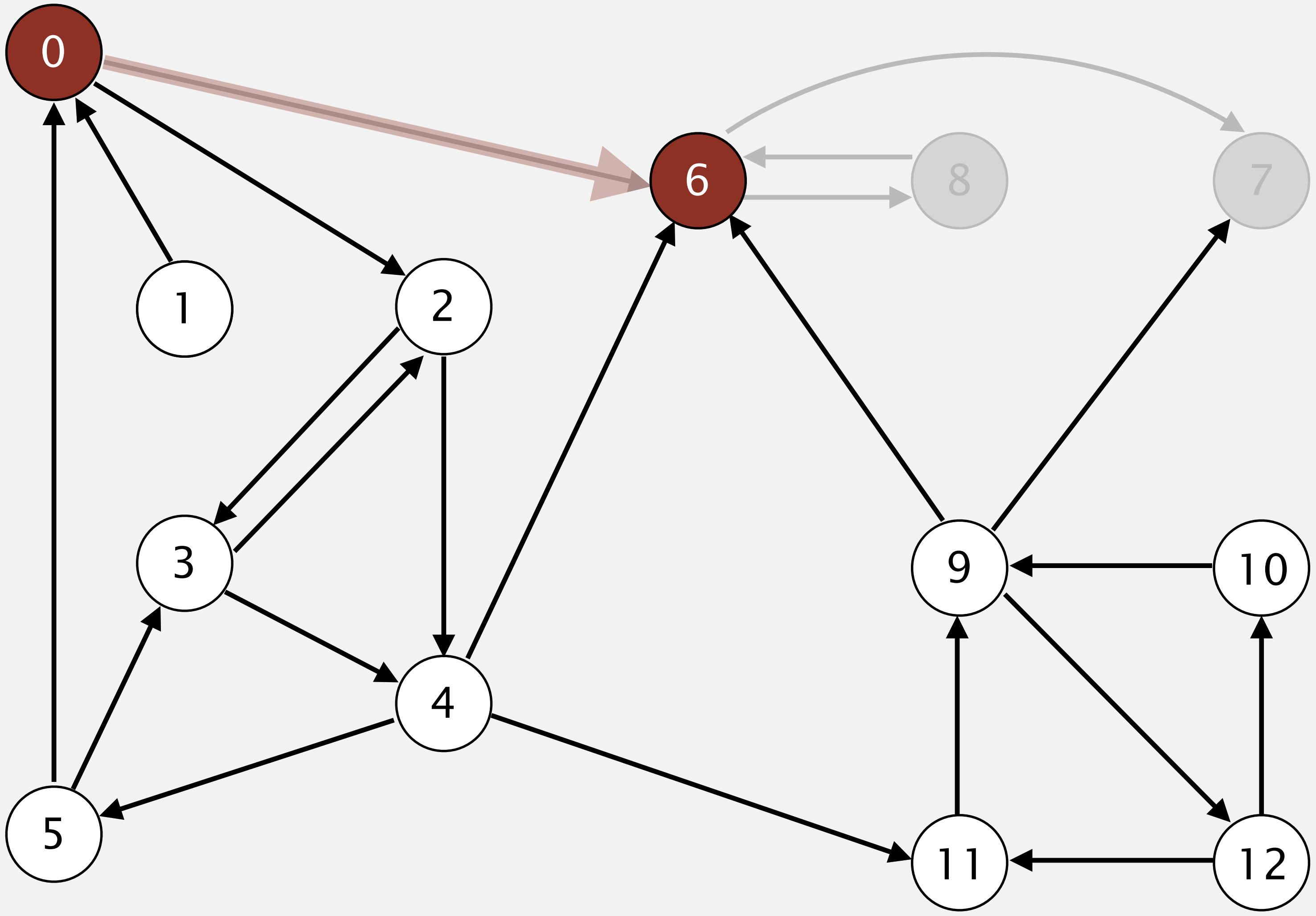
v	marked[]
0	T
1	F
2	F
3	F
4	F
5	F
6	T
7	T
8	T
9	F
10	F
11	F
12	F

7 done

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

6 7 8



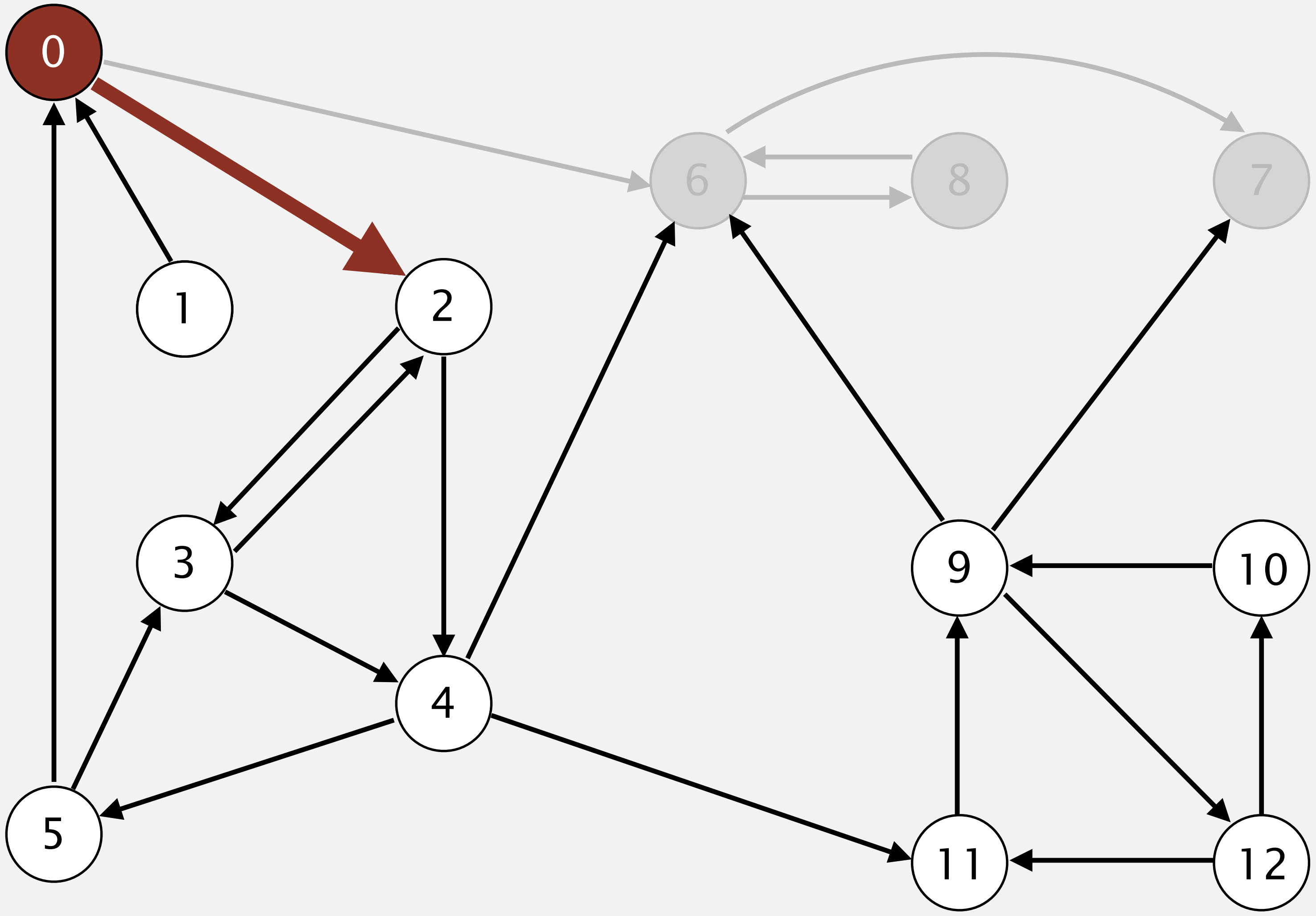
v	marked[]
0	T
1	F
2	F
3	F
4	F
5	F
6	T
7	T
8	T
9	F
10	F
11	F
12	F

6 done

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

6 7 8



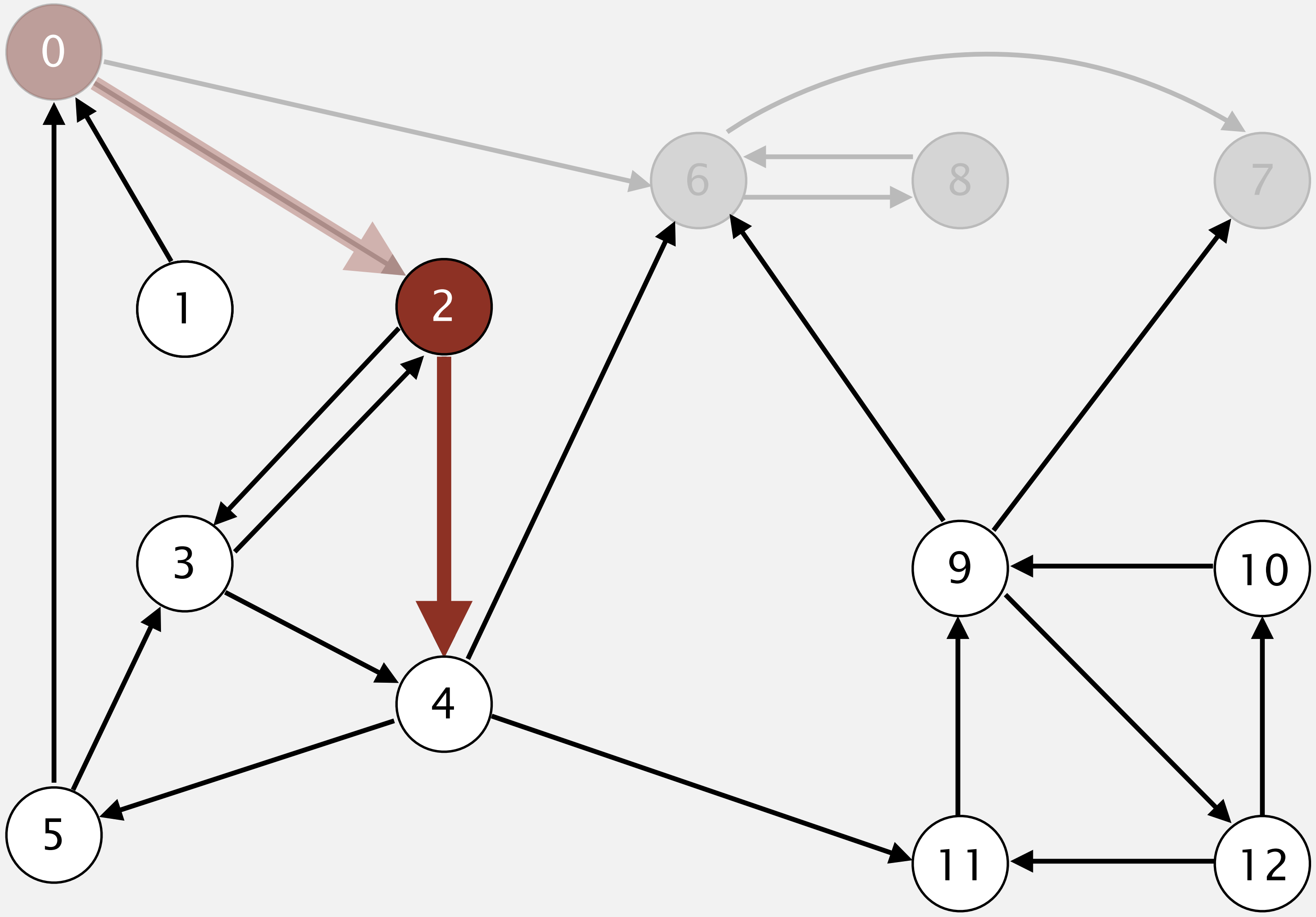
v	marked[]
0	T
1	F
2	F
3	F
4	F
5	F
6	T
7	T
8	T
9	F
10	F
11	F
12	F

visit 0: check 6 and check 2

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

6 7 8



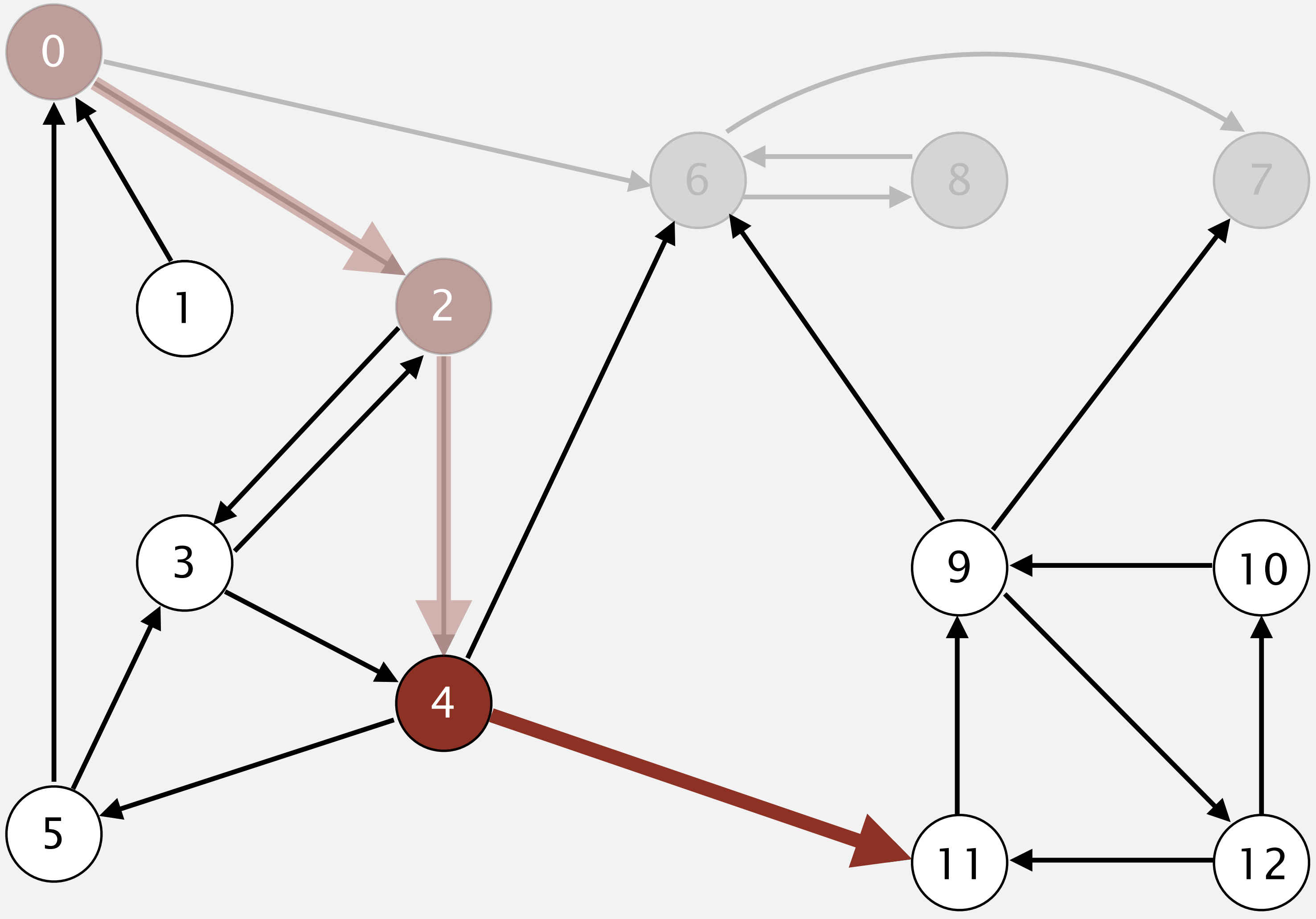
visit 2: check 4 and check 3

v	marked[]
0	T
1	F
2	T
3	F
4	F
5	F
6	T
7	T
8	T
9	F
10	F
11	F
12	F

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

6 7 8



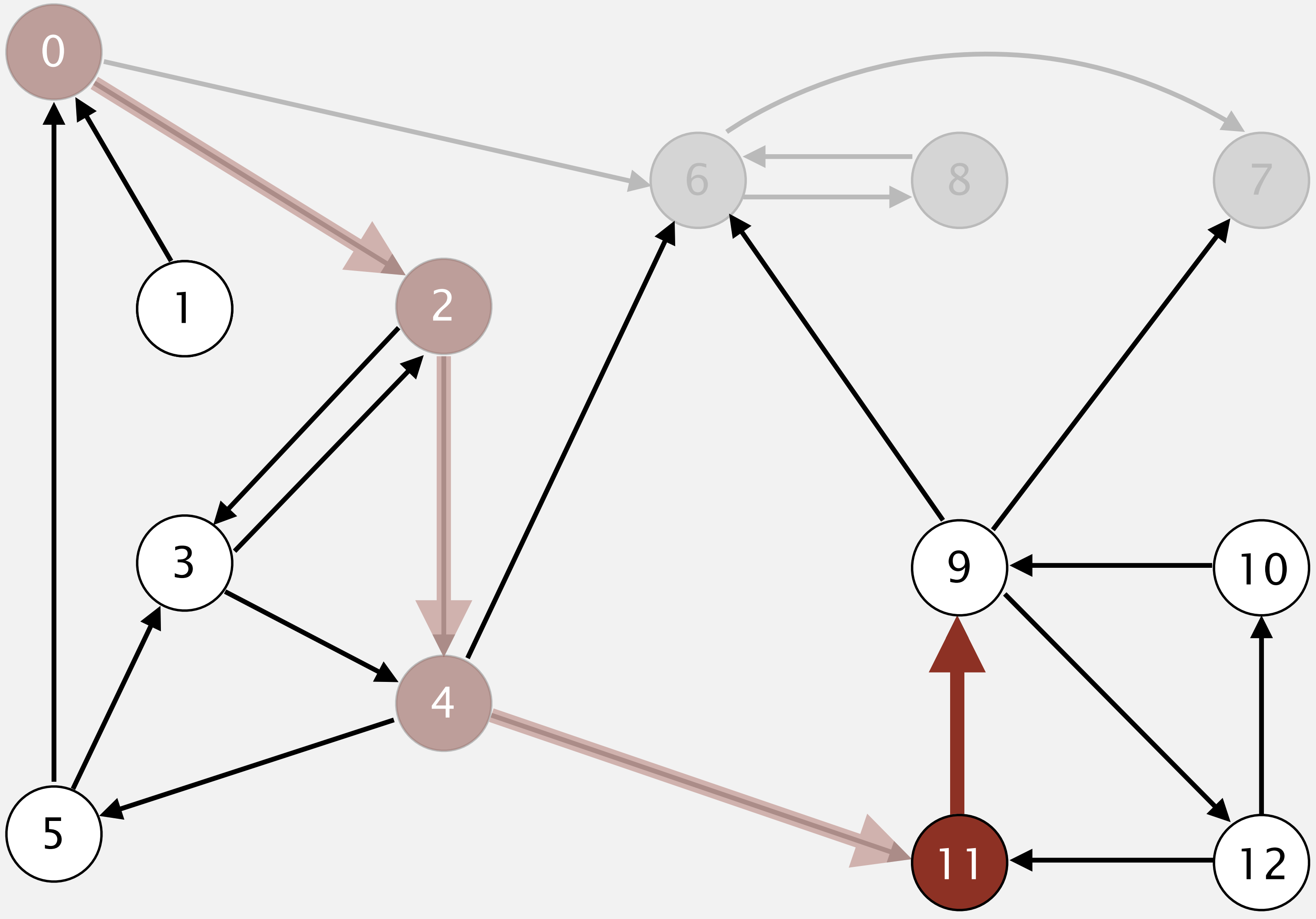
v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	F
10	F
11	F
12	F

visit 4: check 11, check 6, and check 5

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

6 7 8



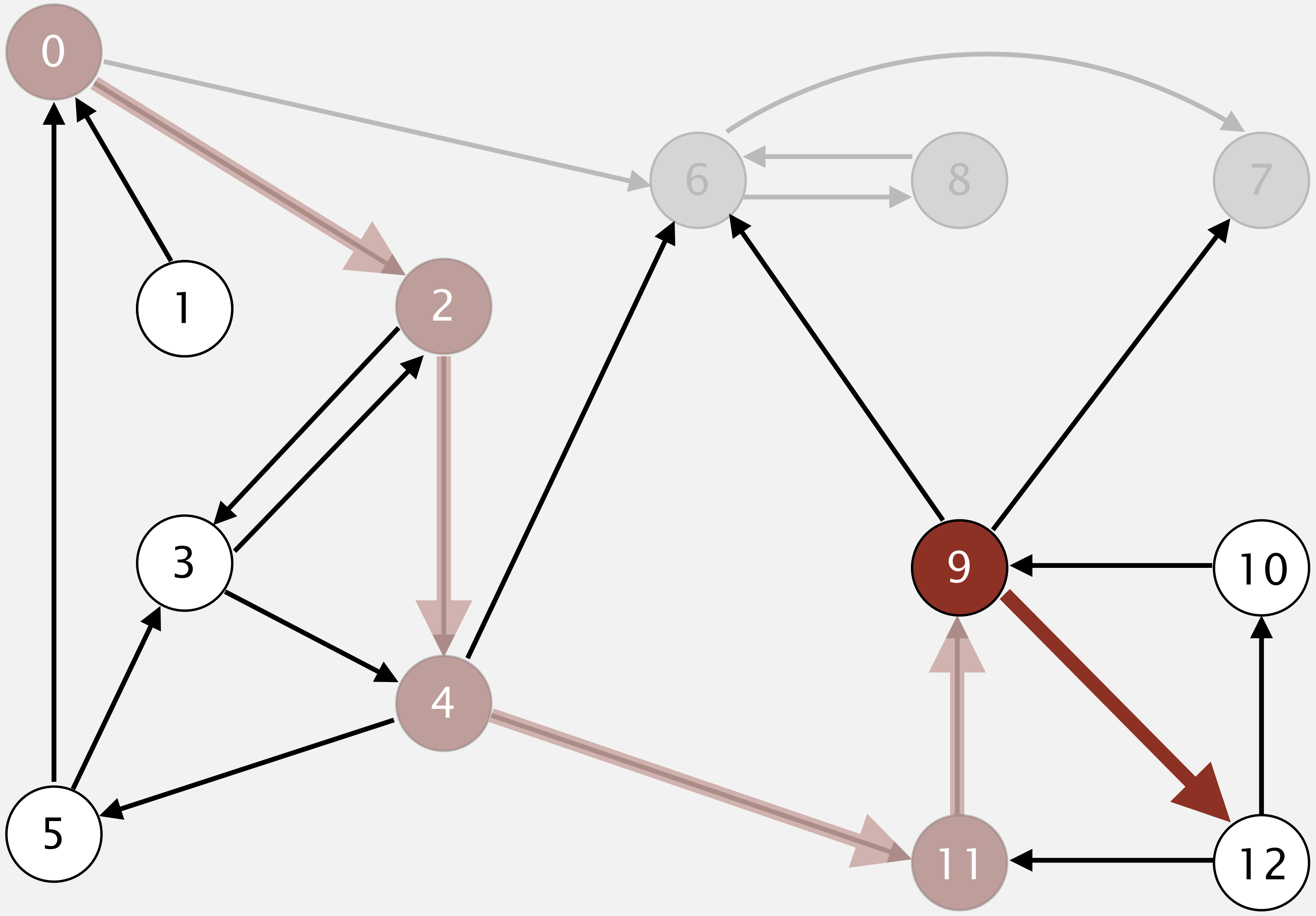
visit 11: check 9

v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	F
10	F
11	T
12	F

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

6 7 8



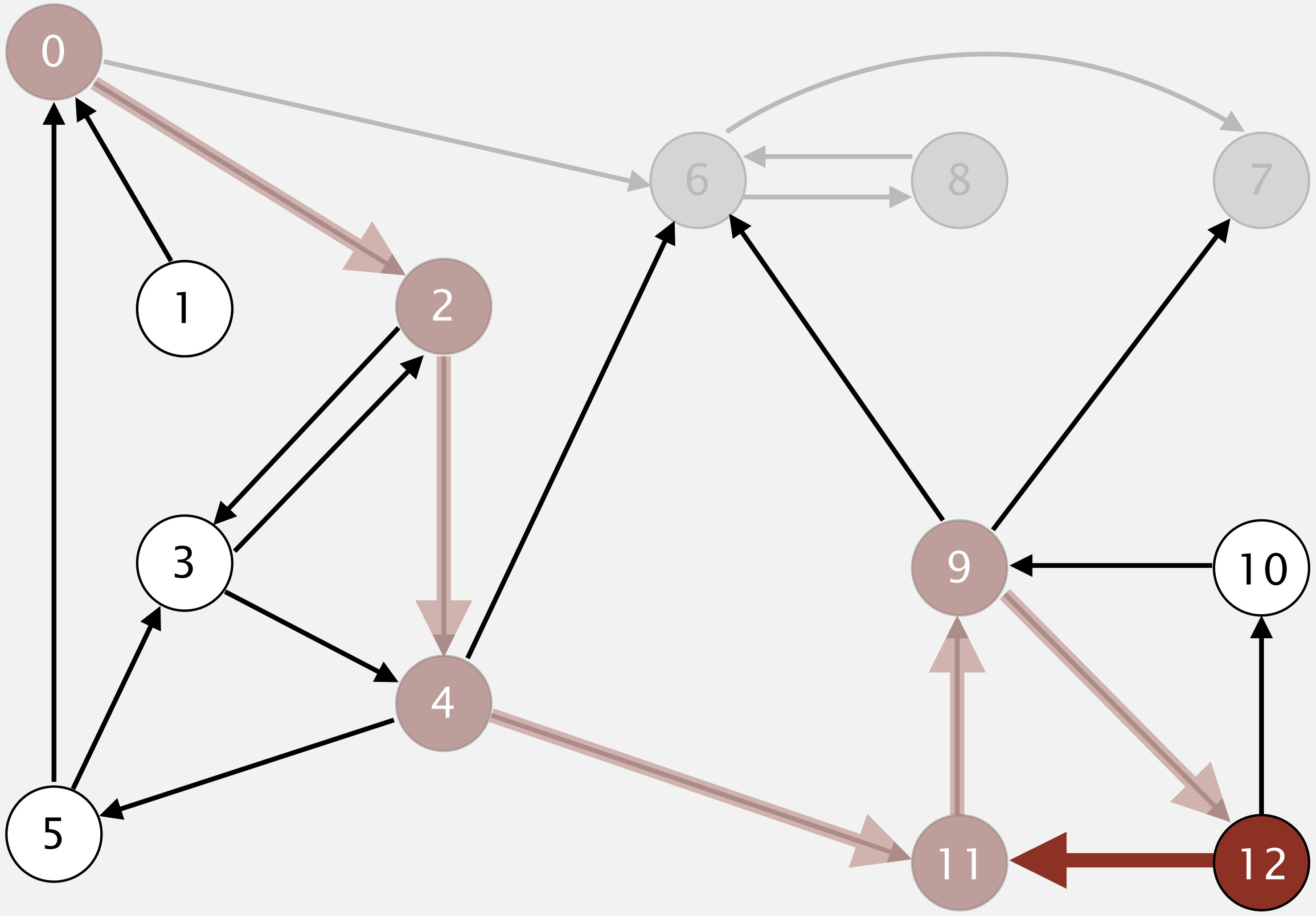
v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	F
11	T
12	F

visit 9: check 12, check 7, and check 6

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

6 7 8



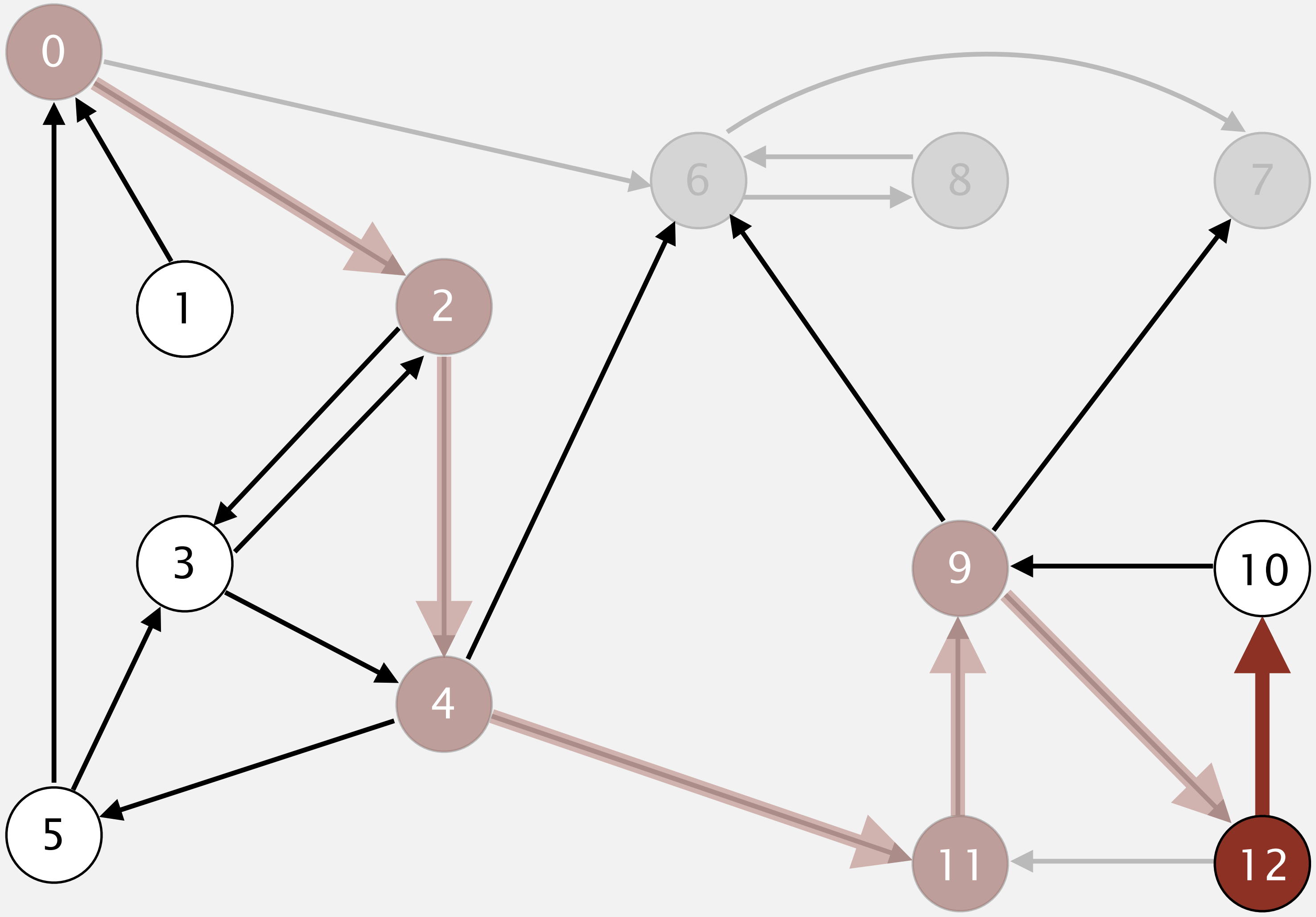
visit 12: check 11 and check 10

v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	F
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

6 7 8



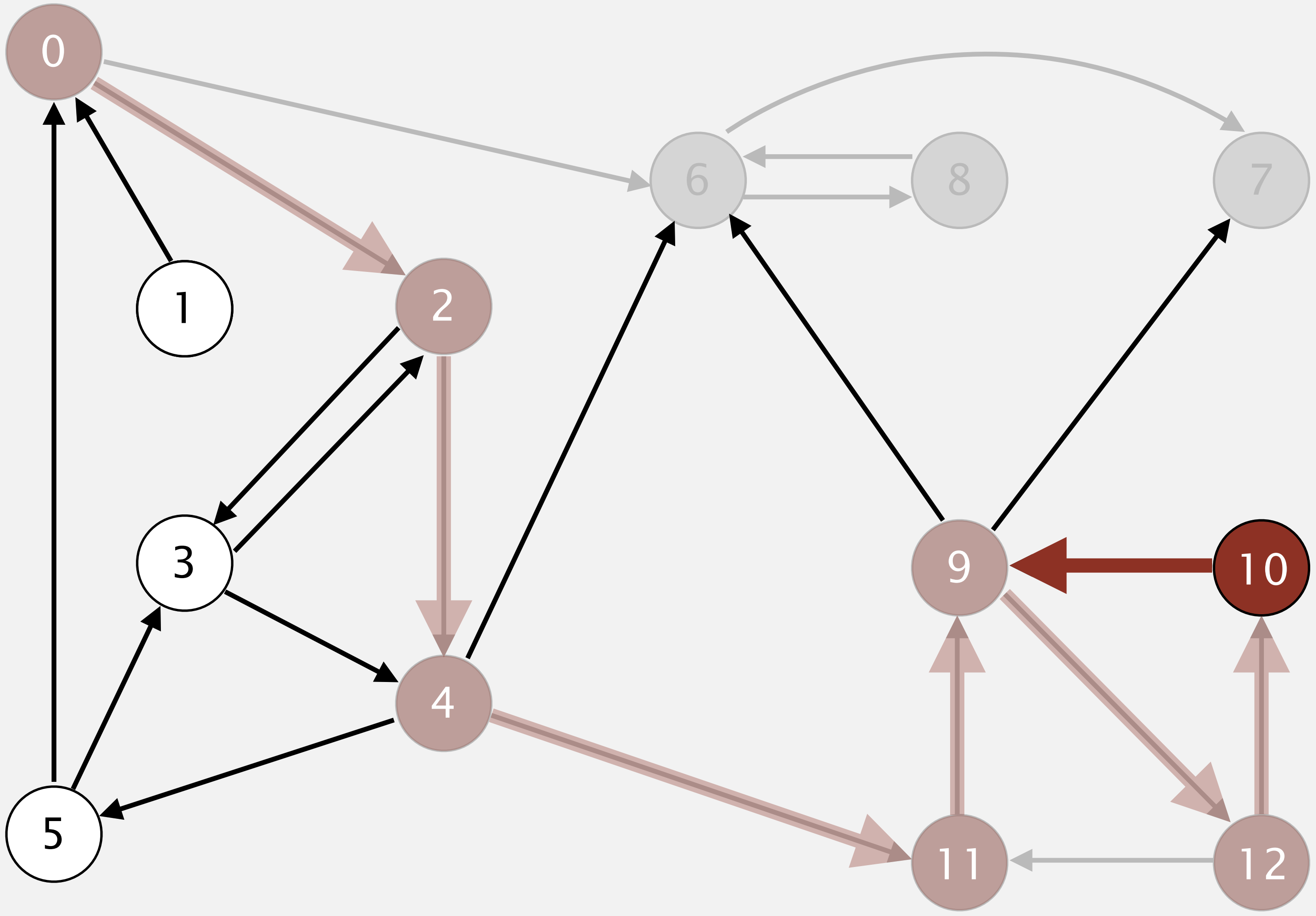
v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	F
11	T
12	T

visit 12: check 11 and check 10

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

6 7 8



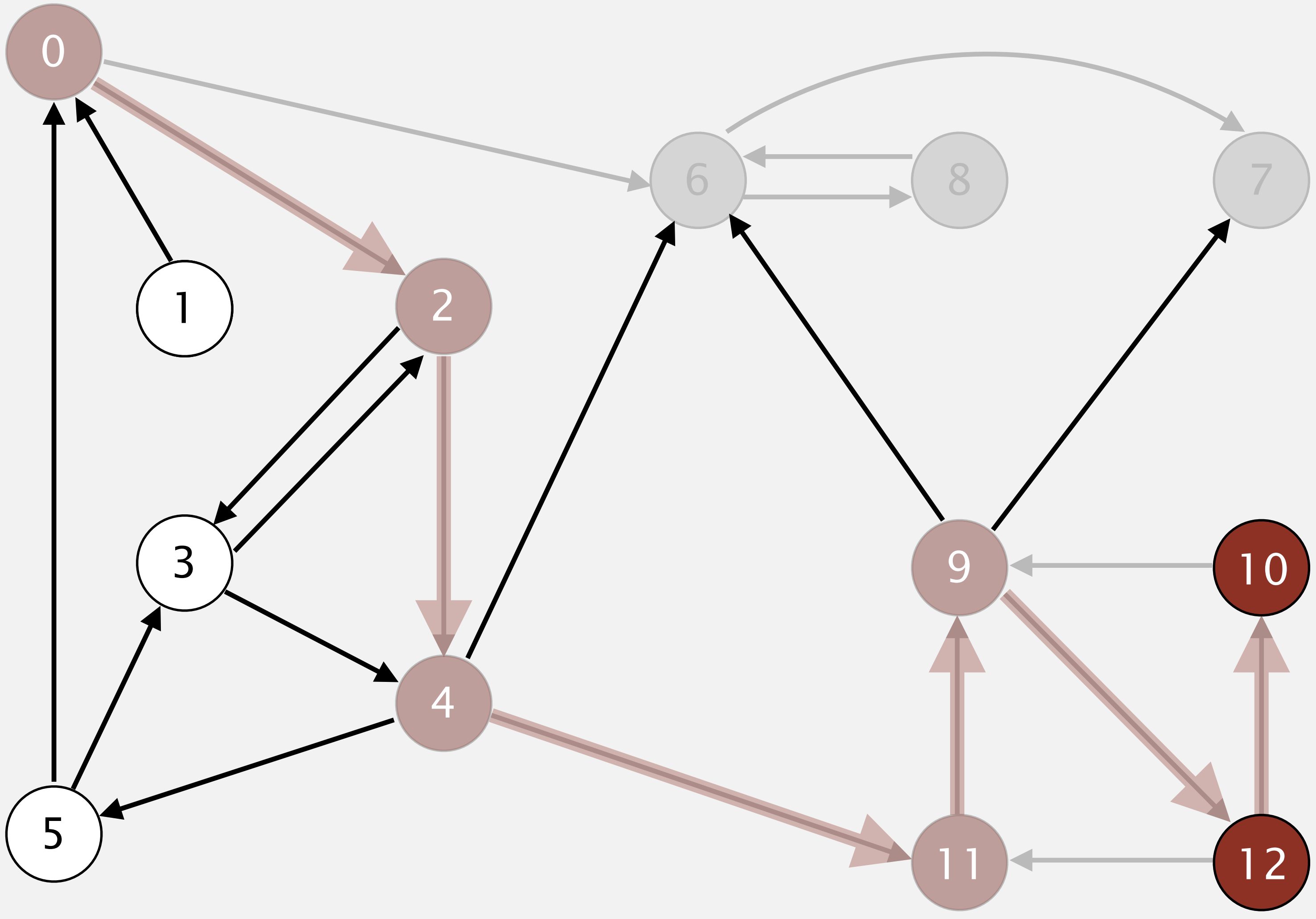
visit 10: check 9

v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

10 6 7 8



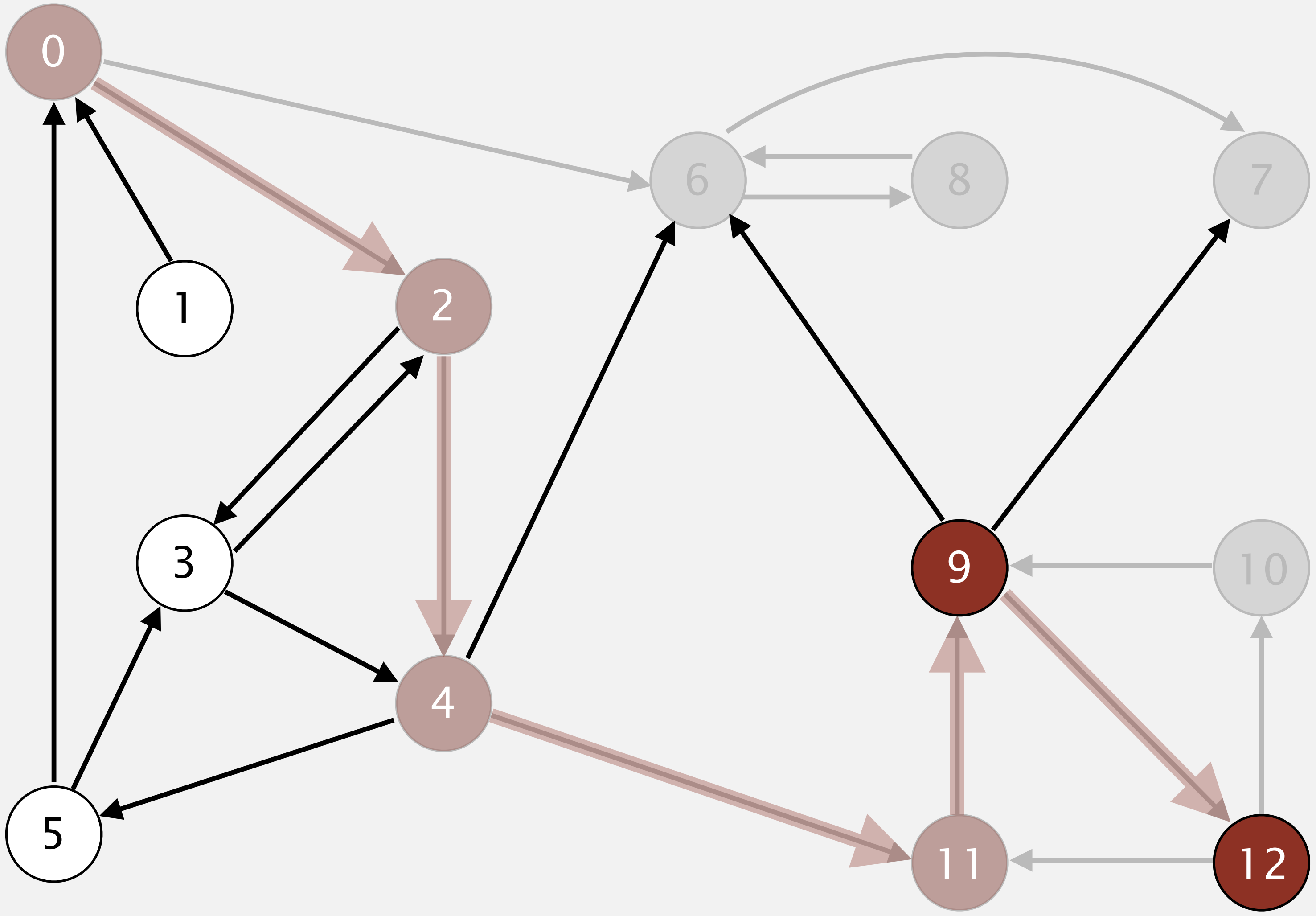
10 done

v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

12 10 6 7 8



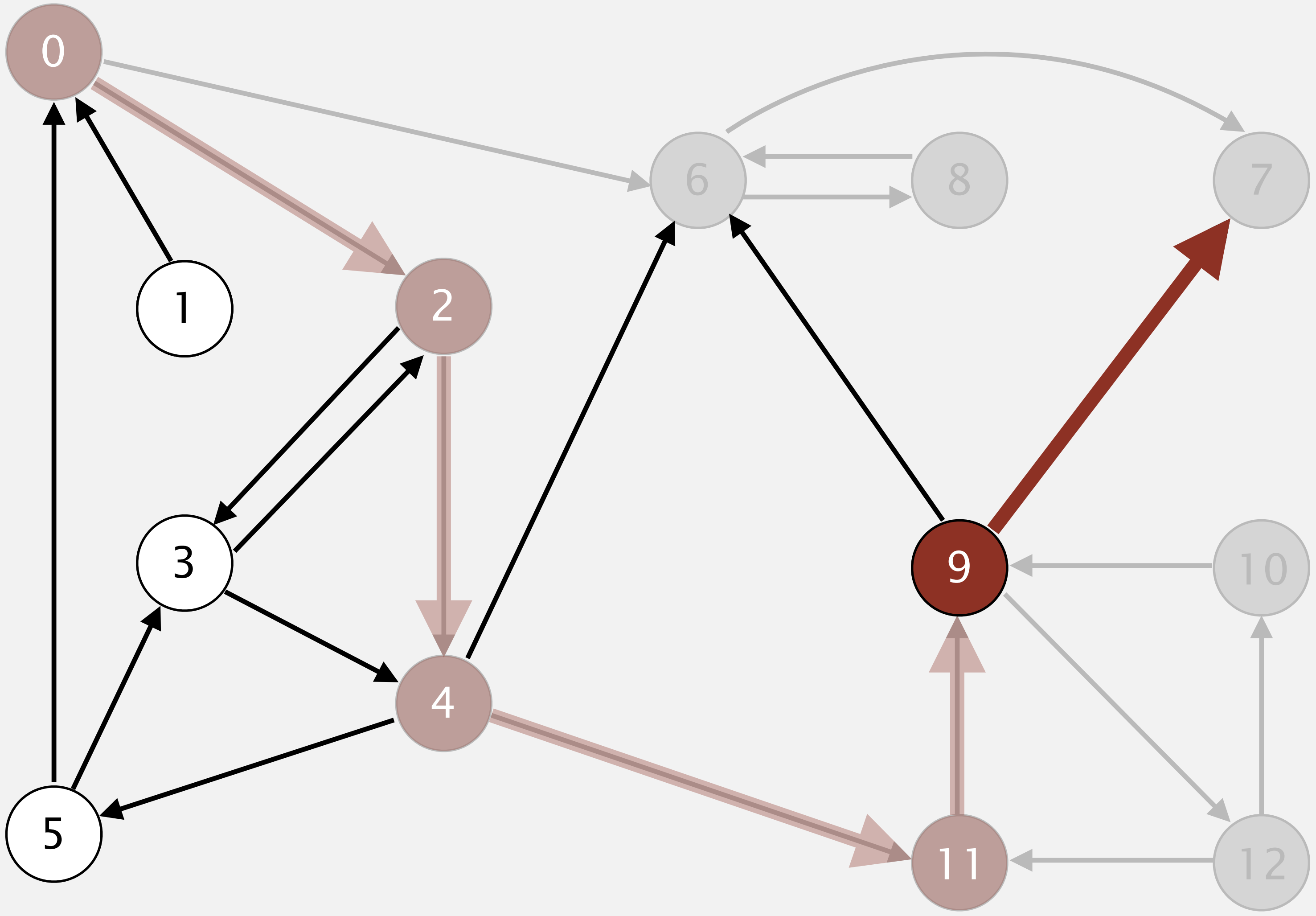
12 done

v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

12 10 6 7 8



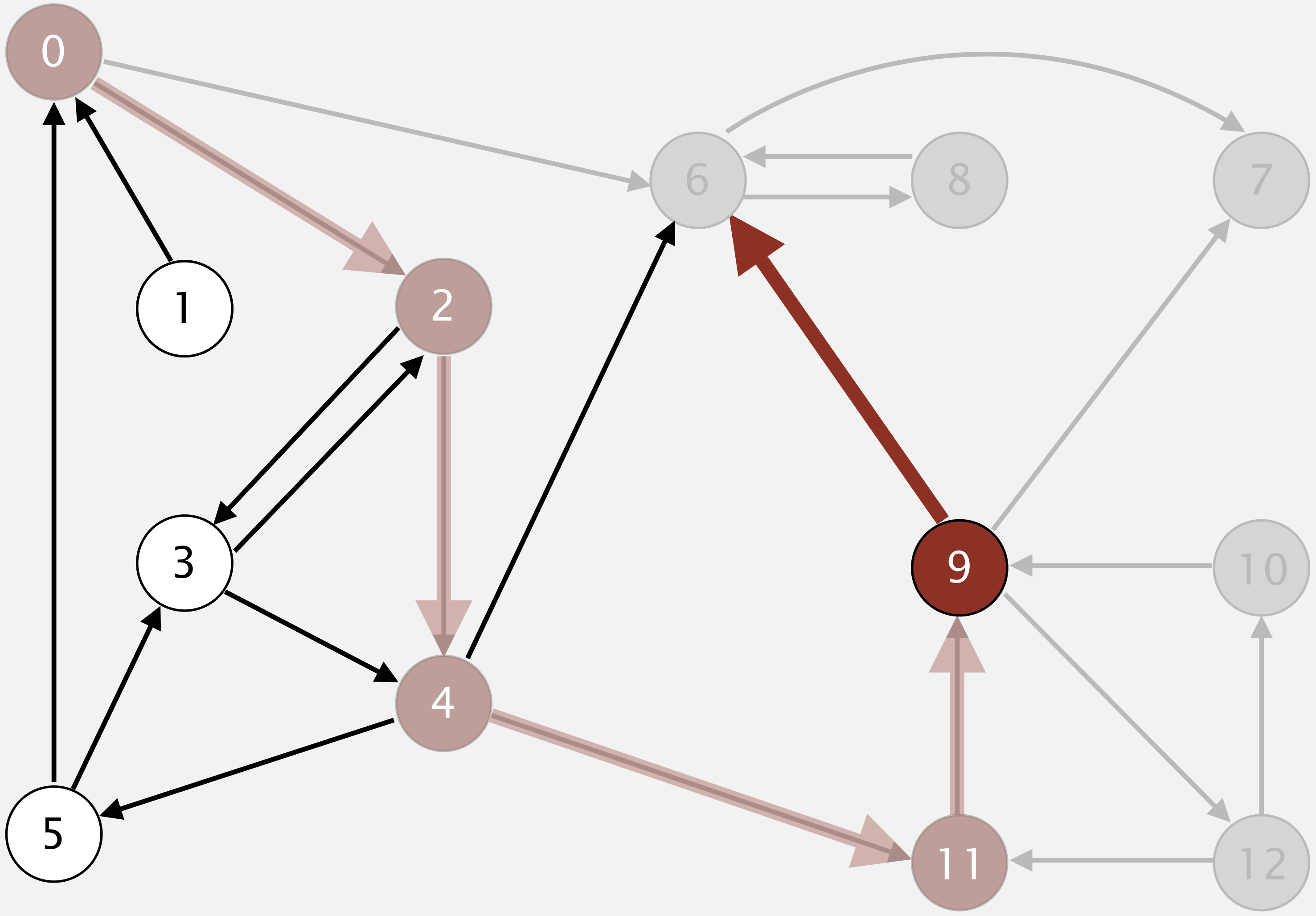
visit 9: check 12, check 7 and check 6

v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

12 10 6 7 8



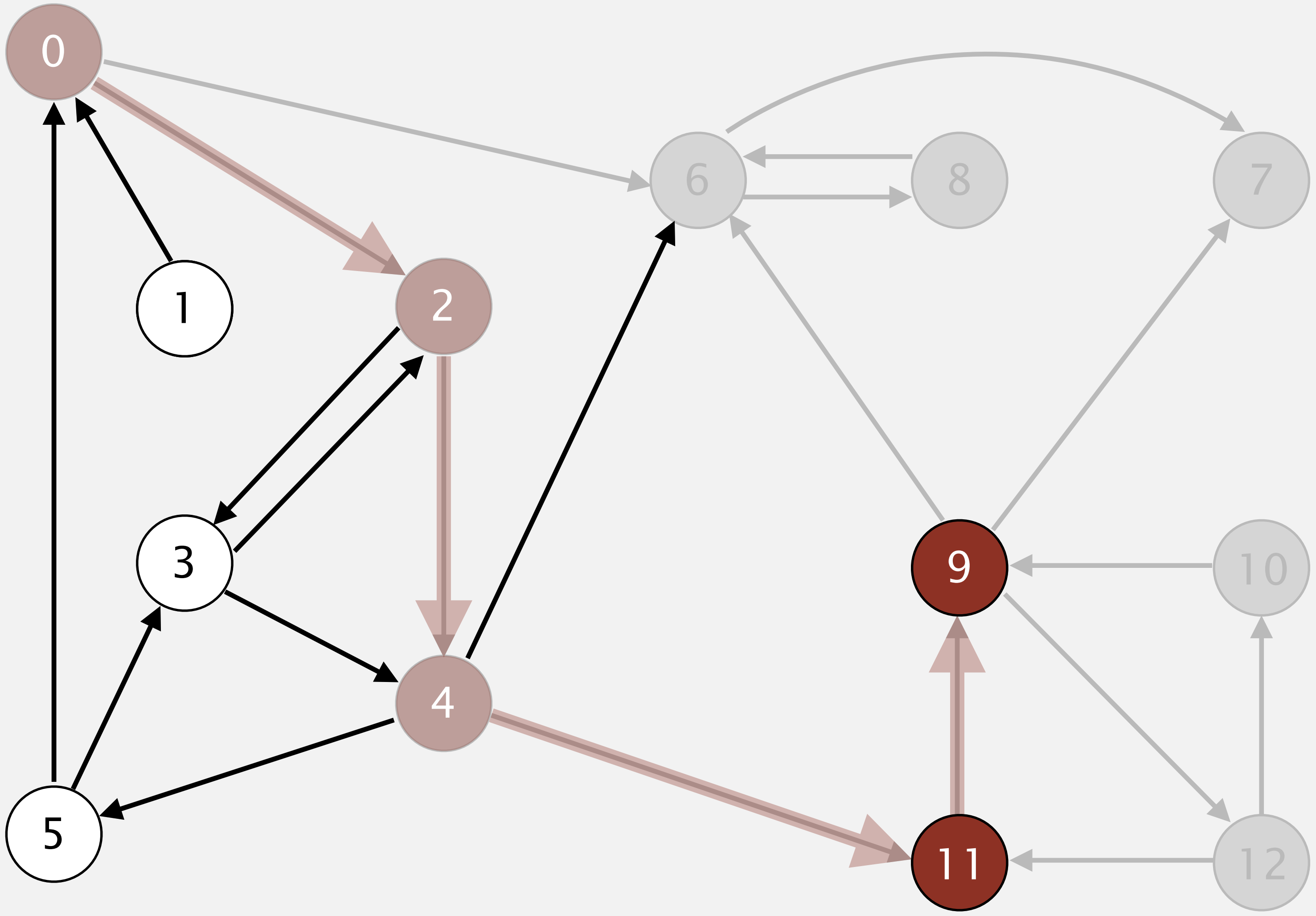
v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	T
11	T
12	T

visit 9: check 12, check 7, and check 6

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

9 12 10 6 7 8



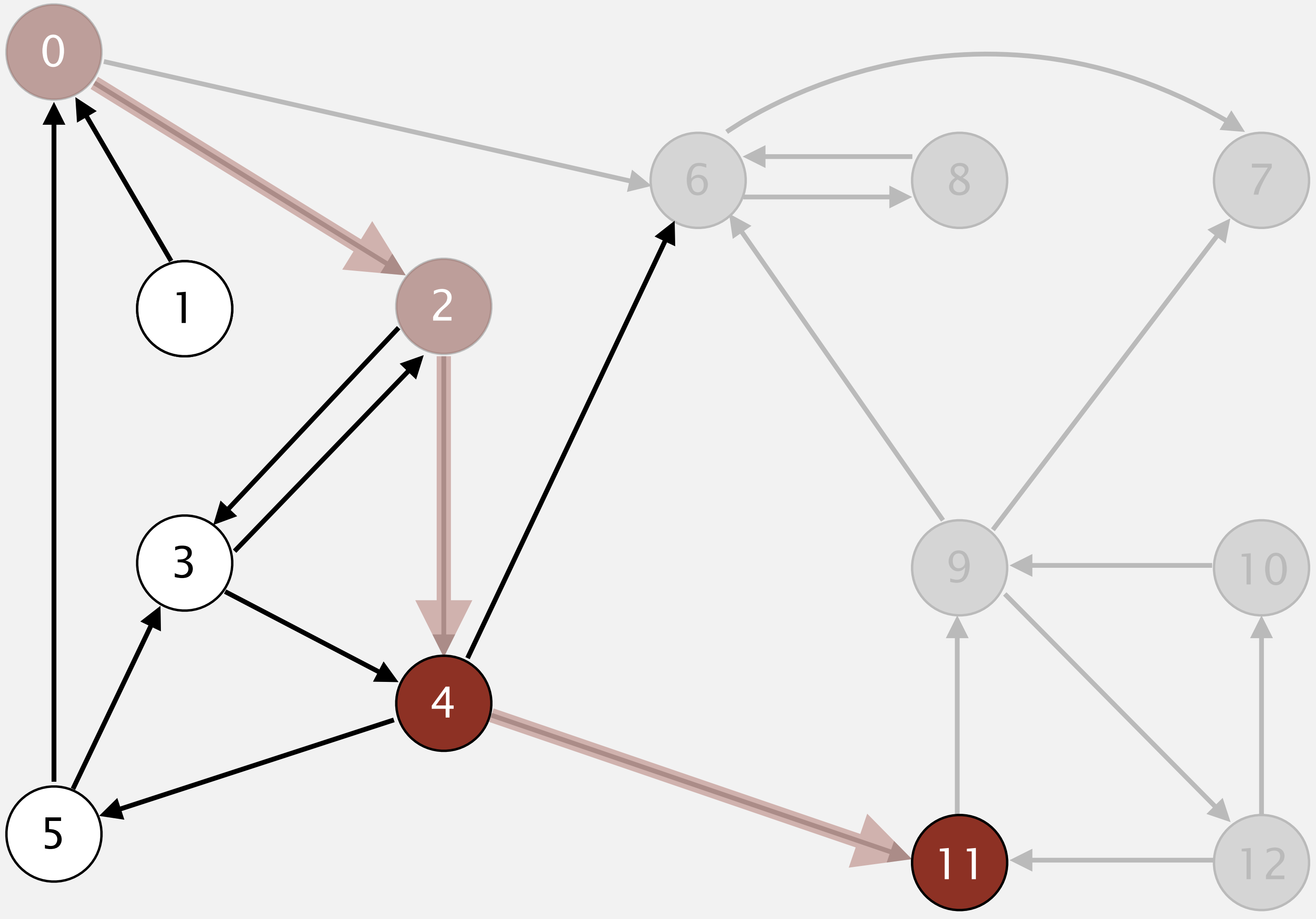
9 done

v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

11 9 12 10 6 7 8



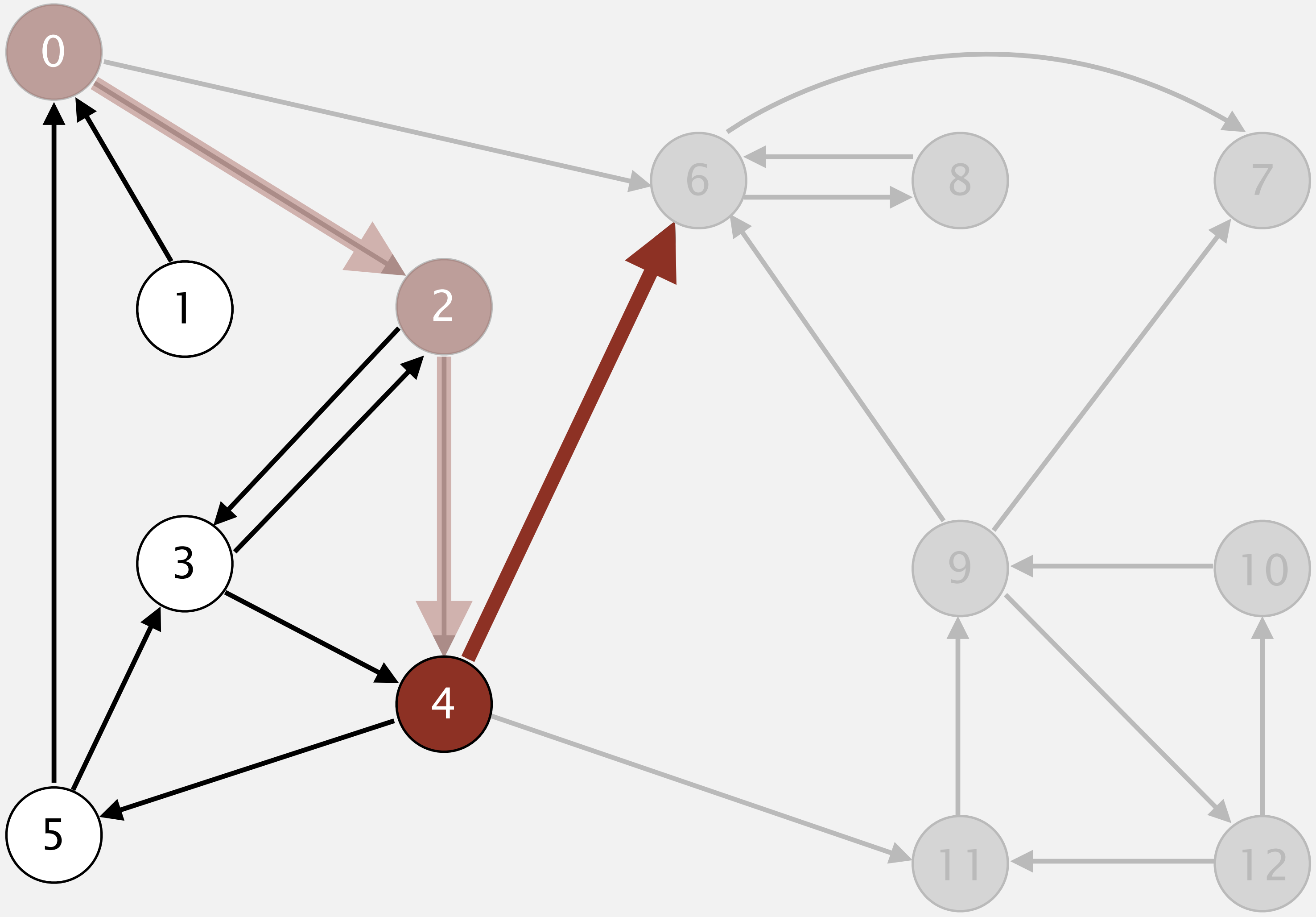
11 done

v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

11 9 12 10 6 7 8



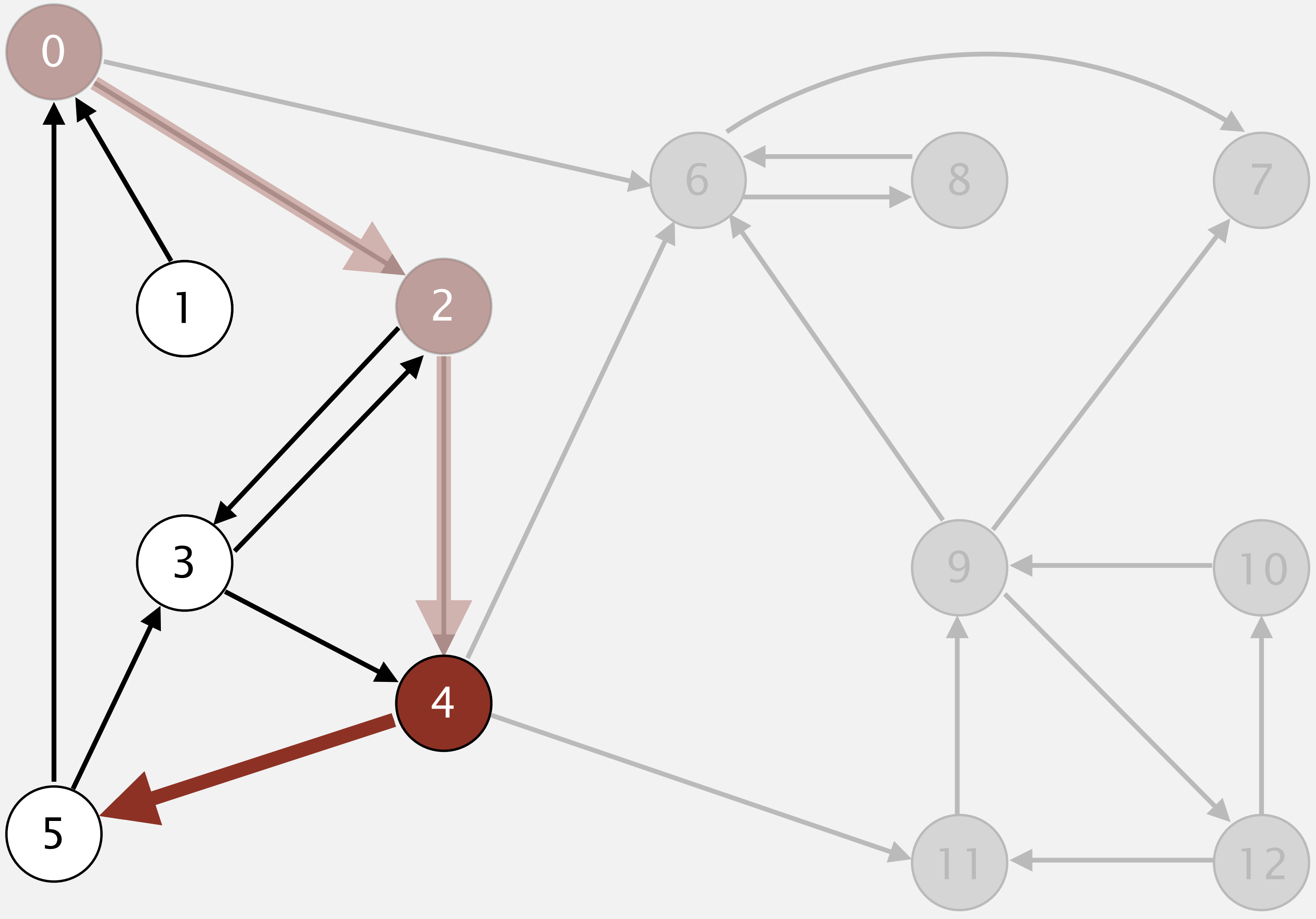
visit 4: check 11, check 6, and check 5

v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

11 9 12 10 6 7 8



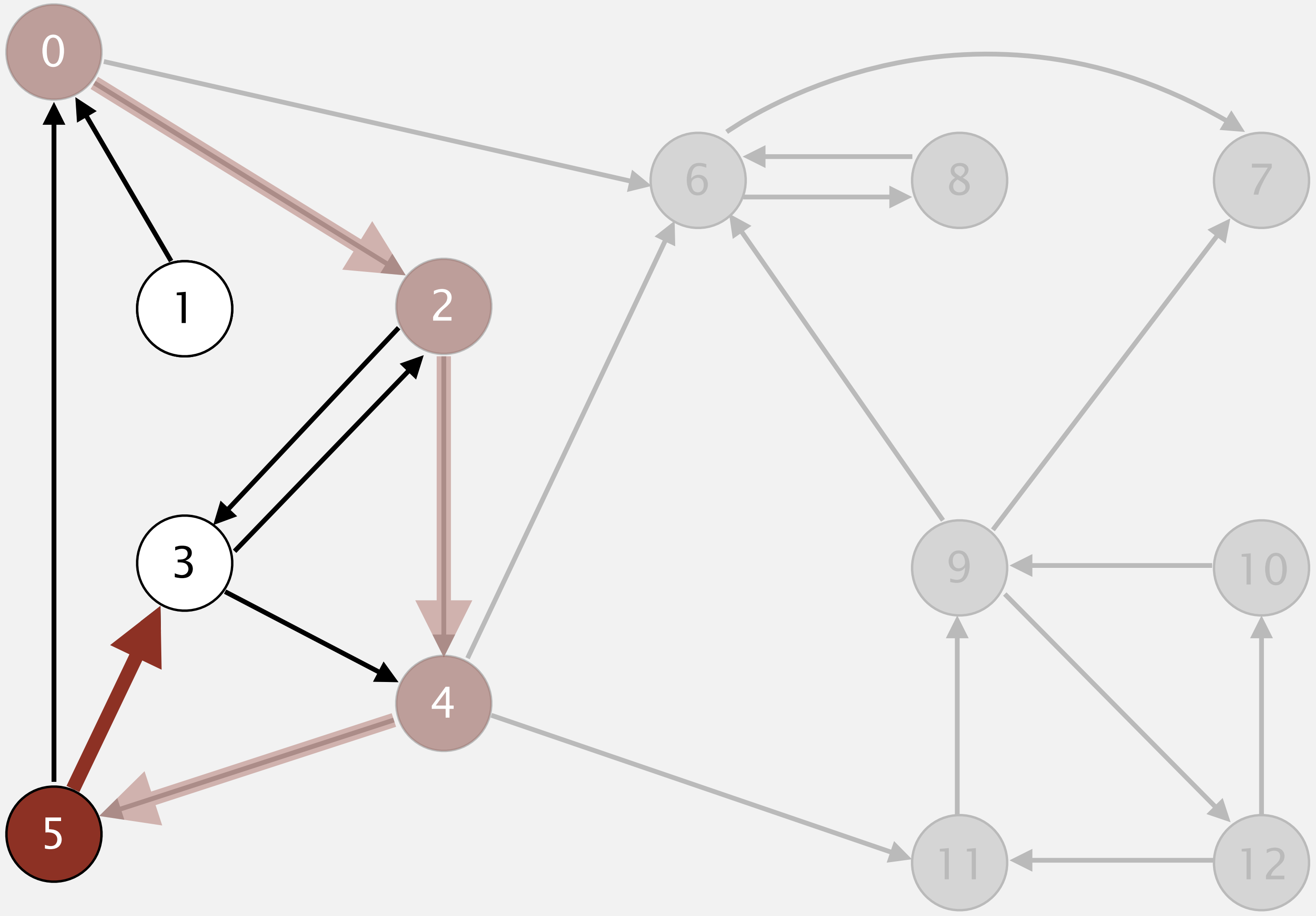
visit 4: check 11, check 6, and check 5

v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

11 9 12 10 6 7 8



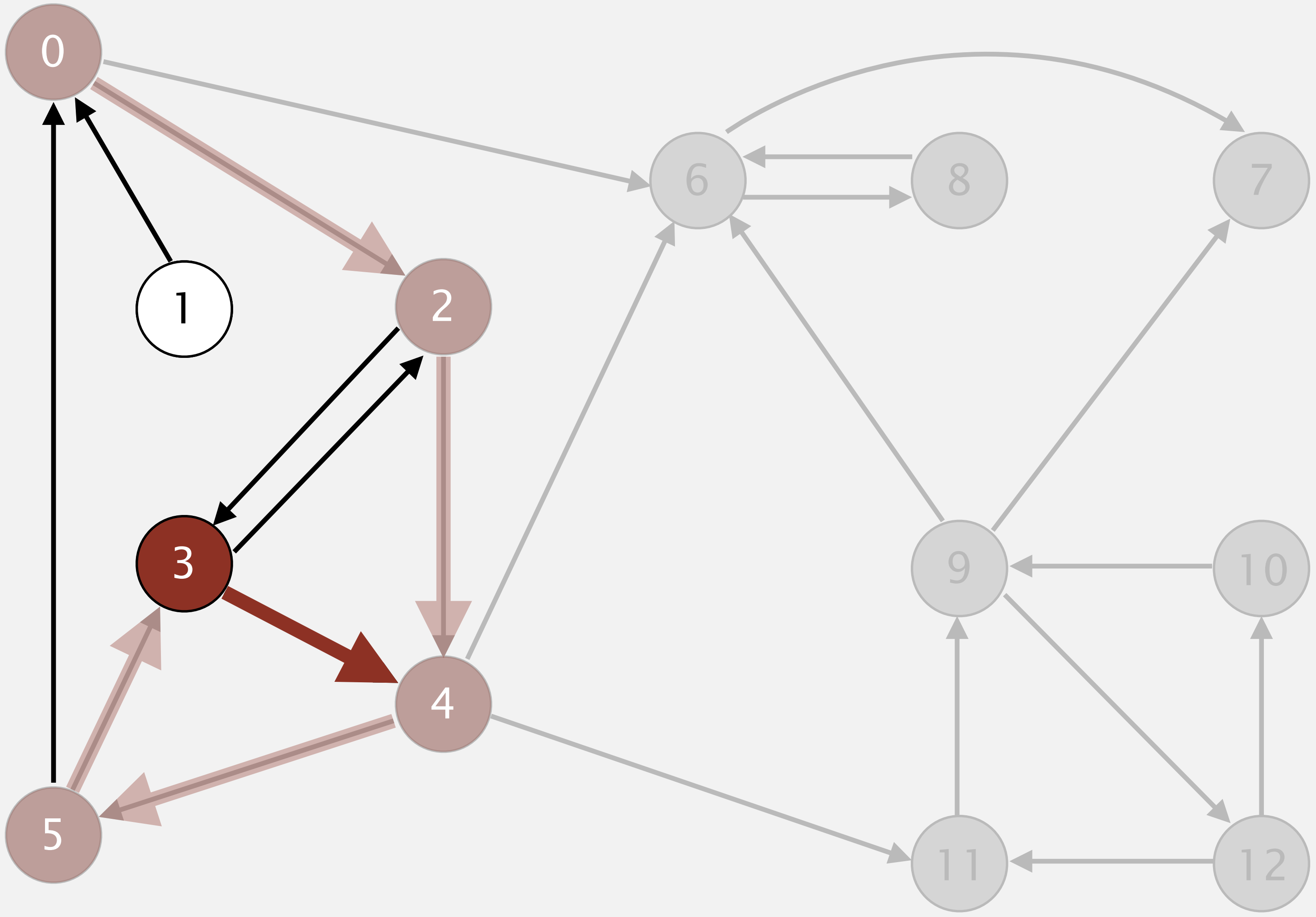
v	marked[]
0	T
1	F
2	T
3	F
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

visit 5: check 3 and check 0

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

11 9 12 10 6 7 8



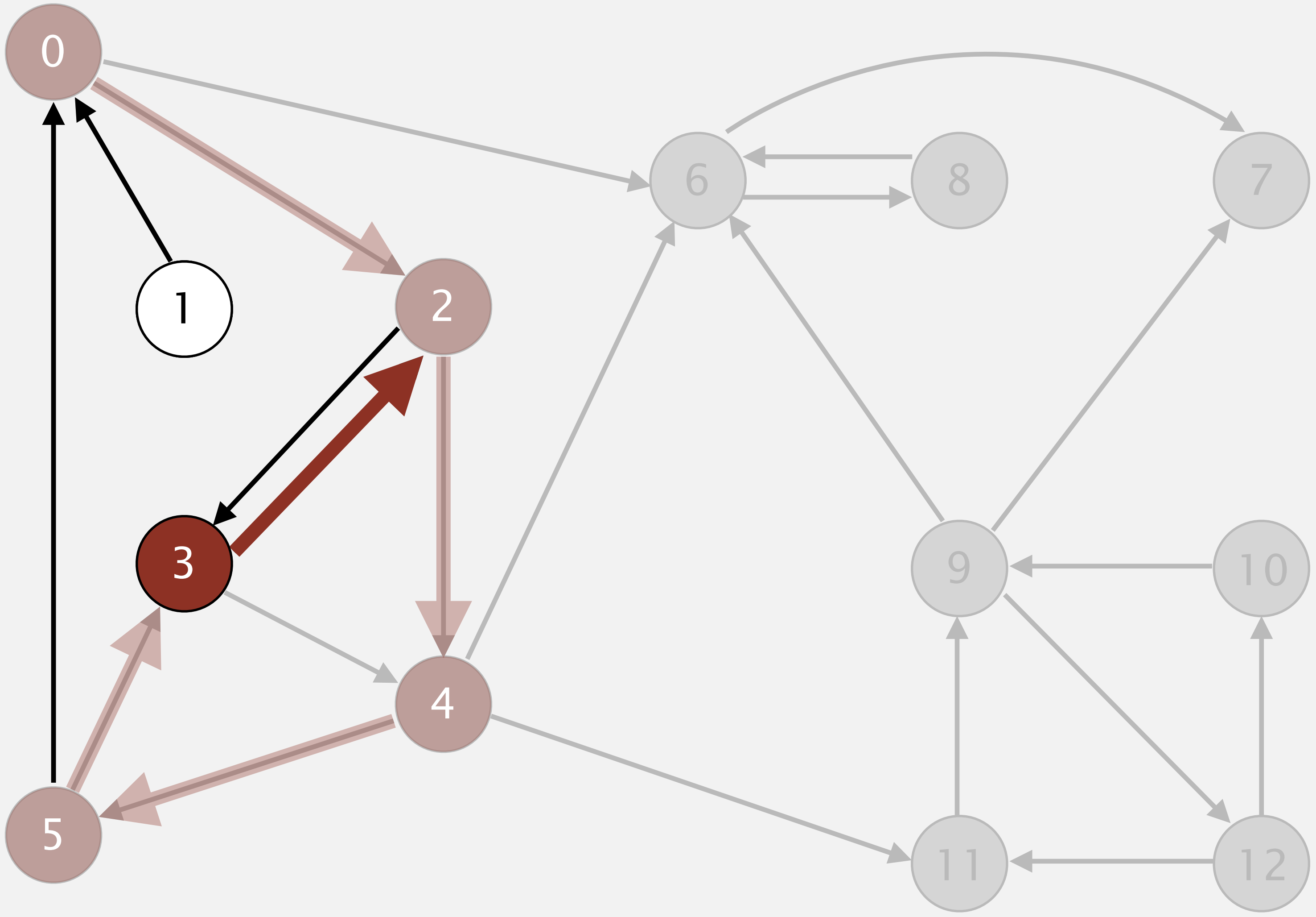
visit 3: check 4 and check 2

v	marked[]
0	T
1	F
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

11 9 12 10 6 7 8



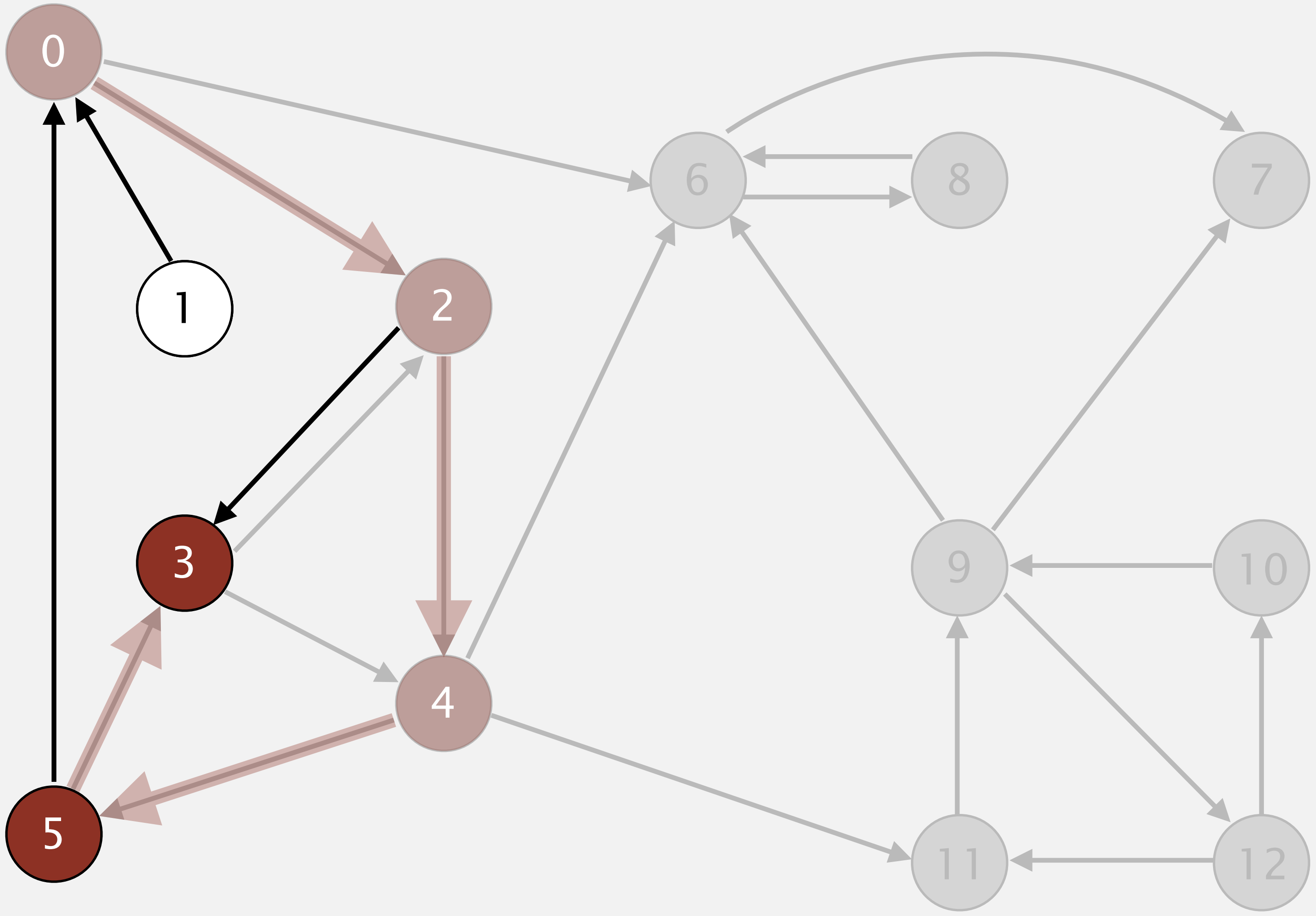
visit 3: check 4 and check 2

v	marked[]
0	T
1	F
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

3 11 9 12 10 6 7 8



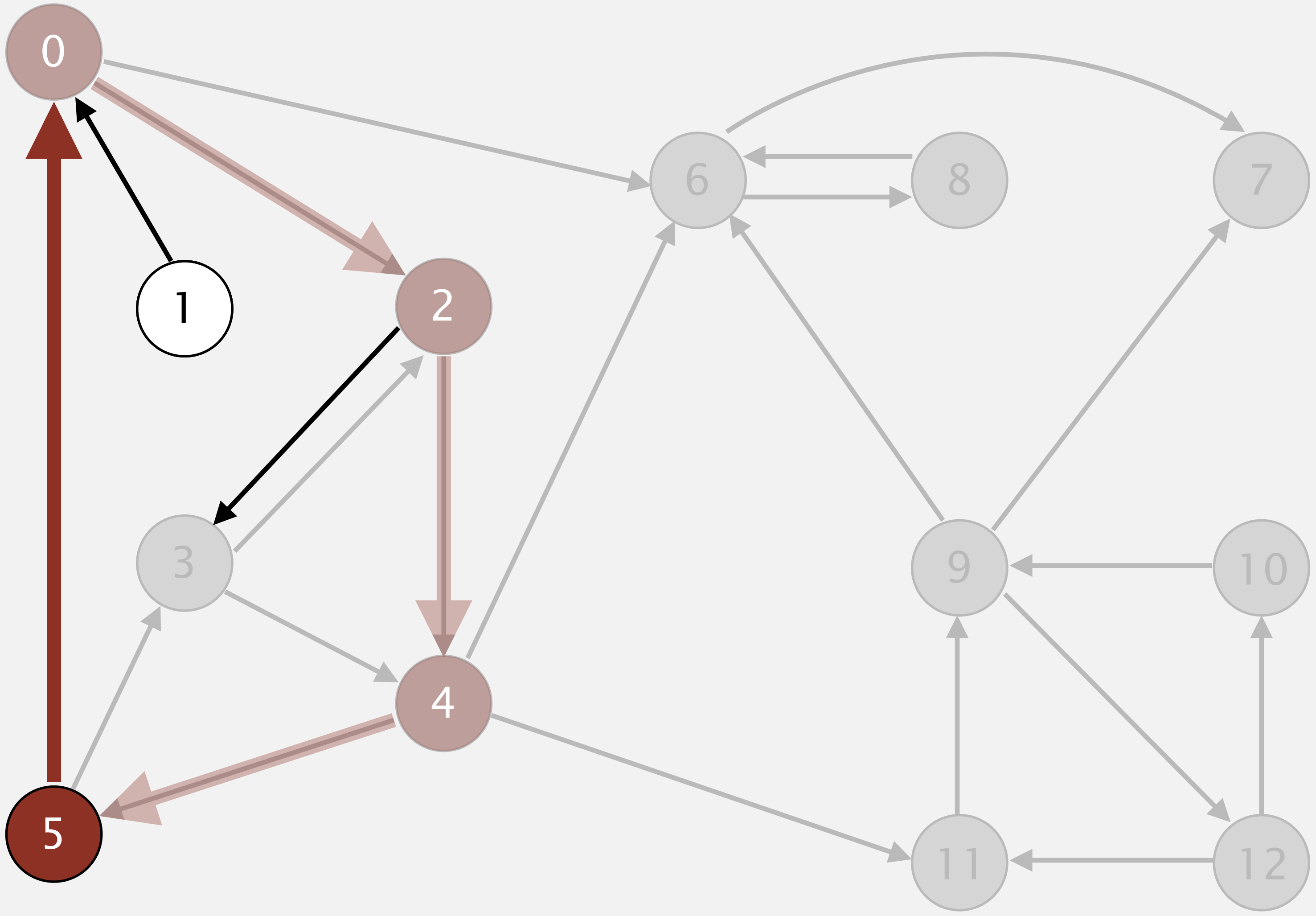
3 done

v	marked[]
0	T
1	F
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

3 11 9 12 10 6 7 8



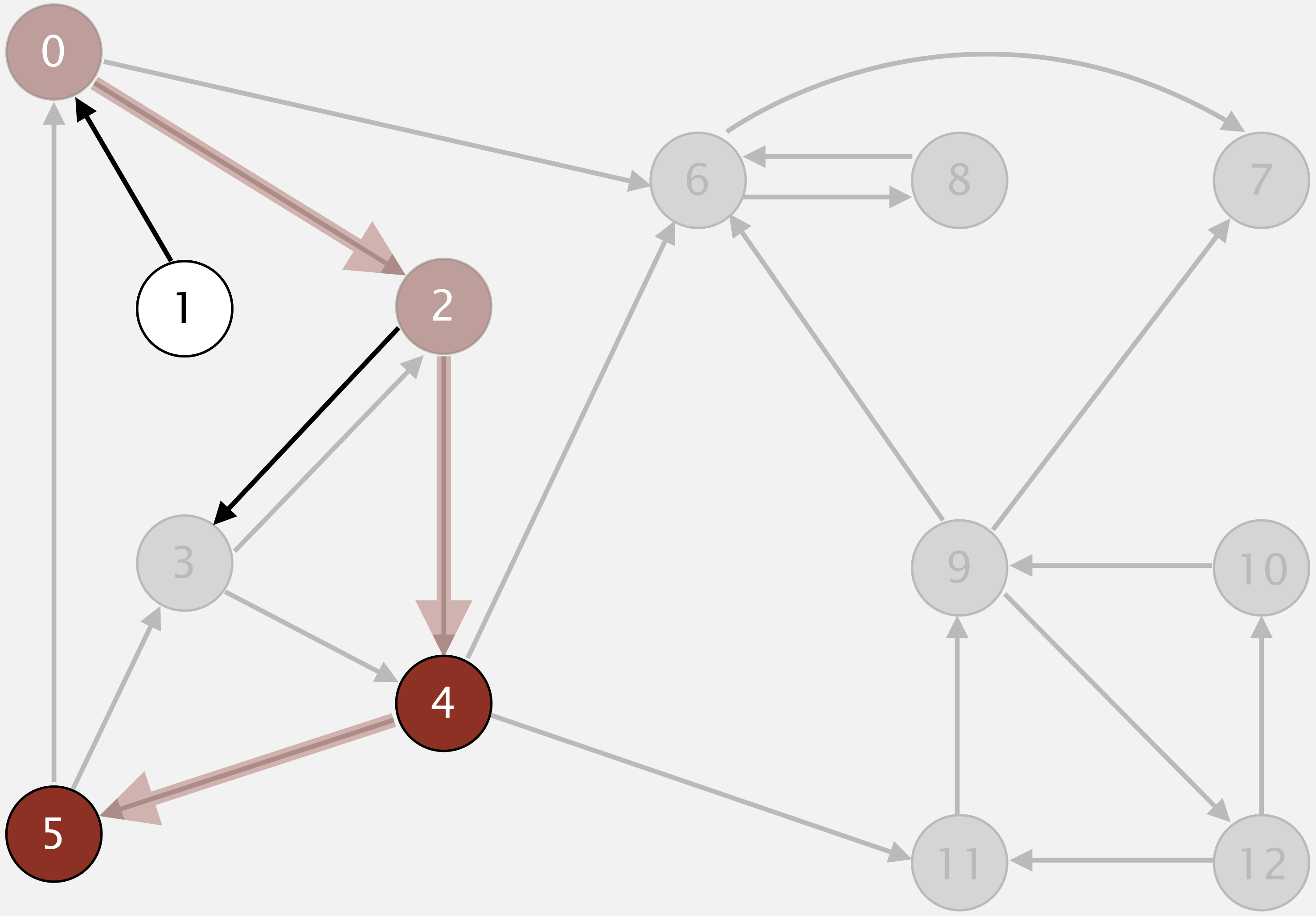
v	marked[]
0	T
1	F
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

visit 5: check 3 and check 0

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

5 3 11 9 12 10 6 7 8



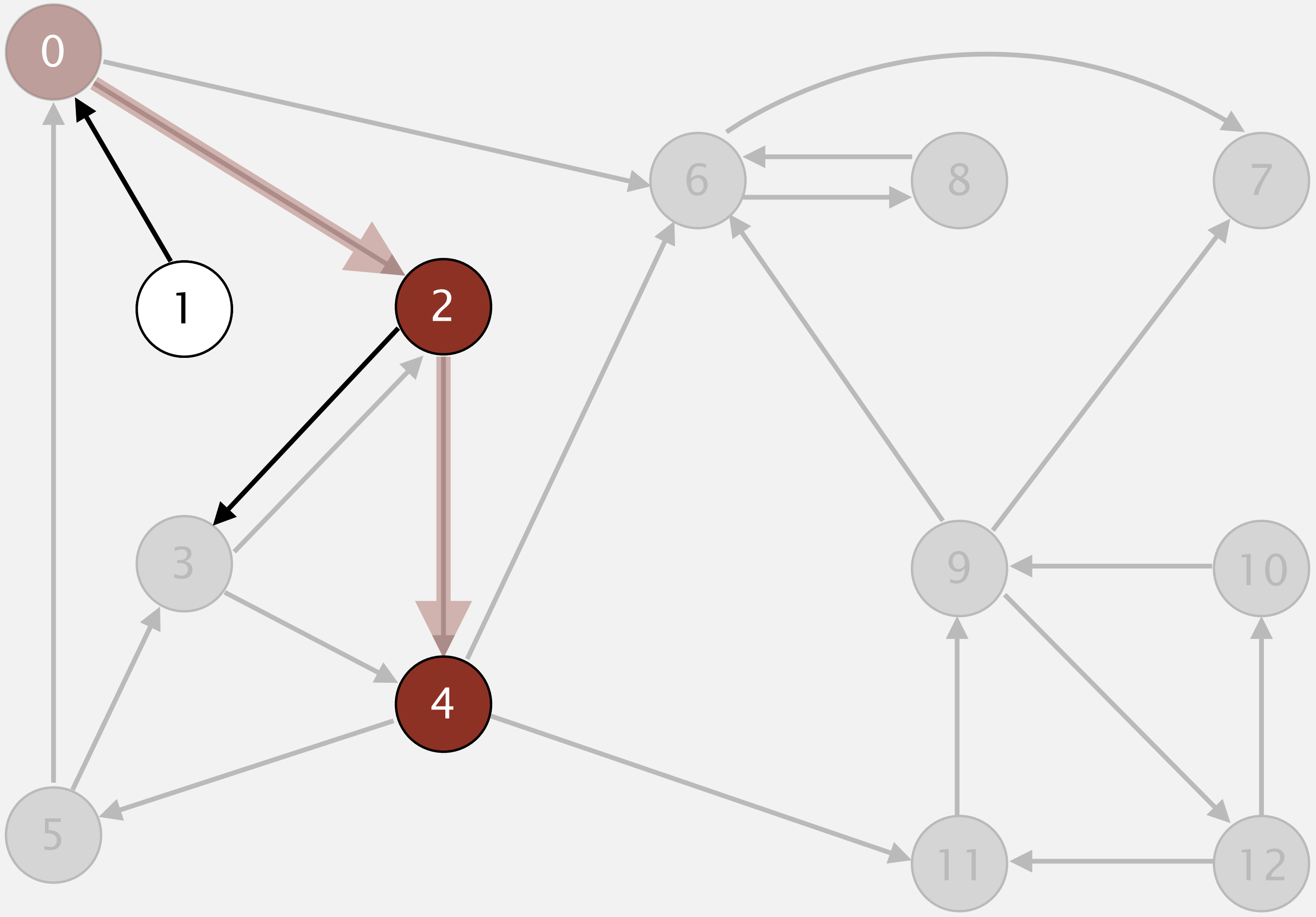
5 done

v	marked[]
0	T
1	F
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

4 5 3 11 9 12 10 6 7 8



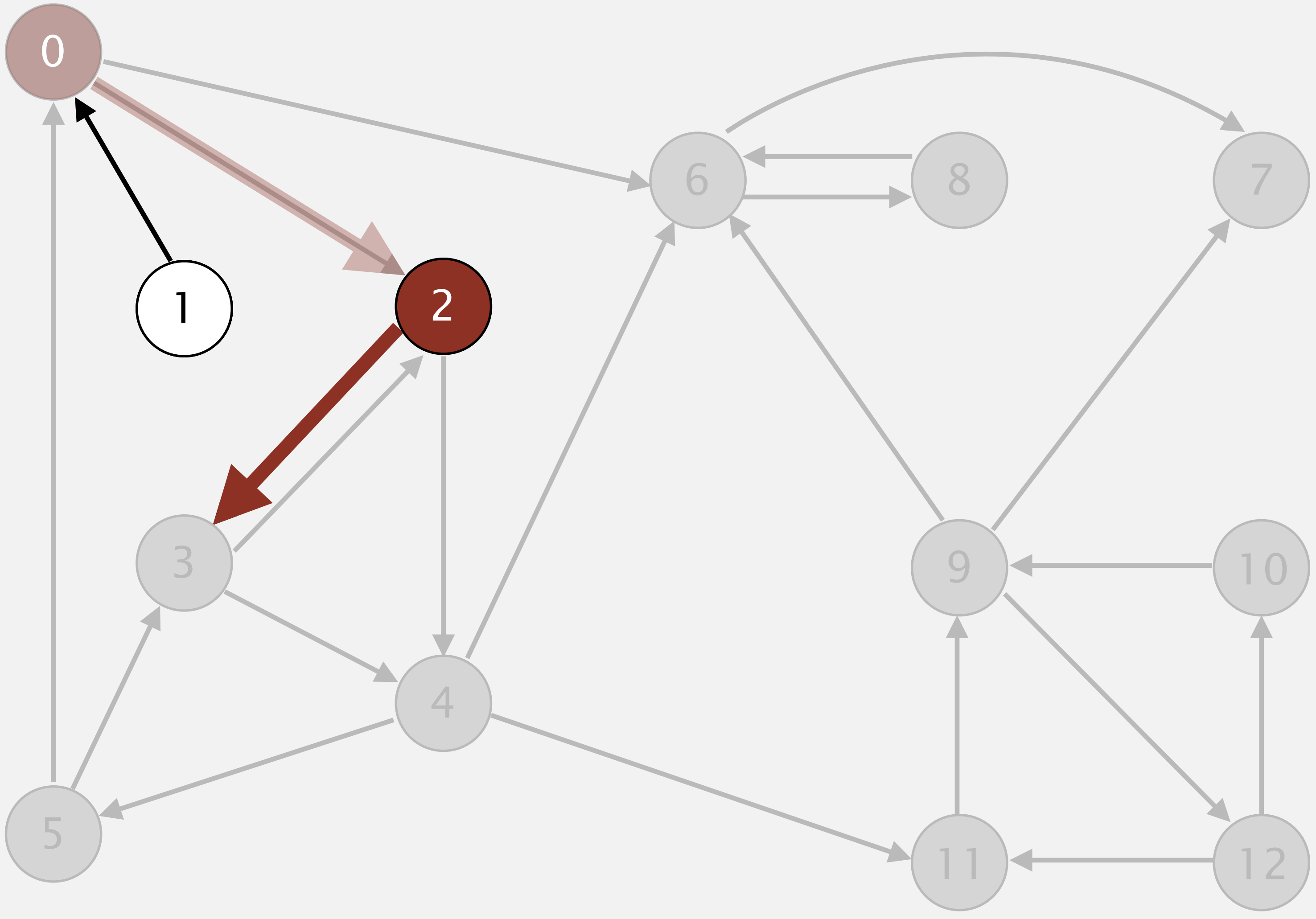
v	marked[]
0	T
1	F
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

4 done

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

4 5 3 11 9 12 10 6 7 8



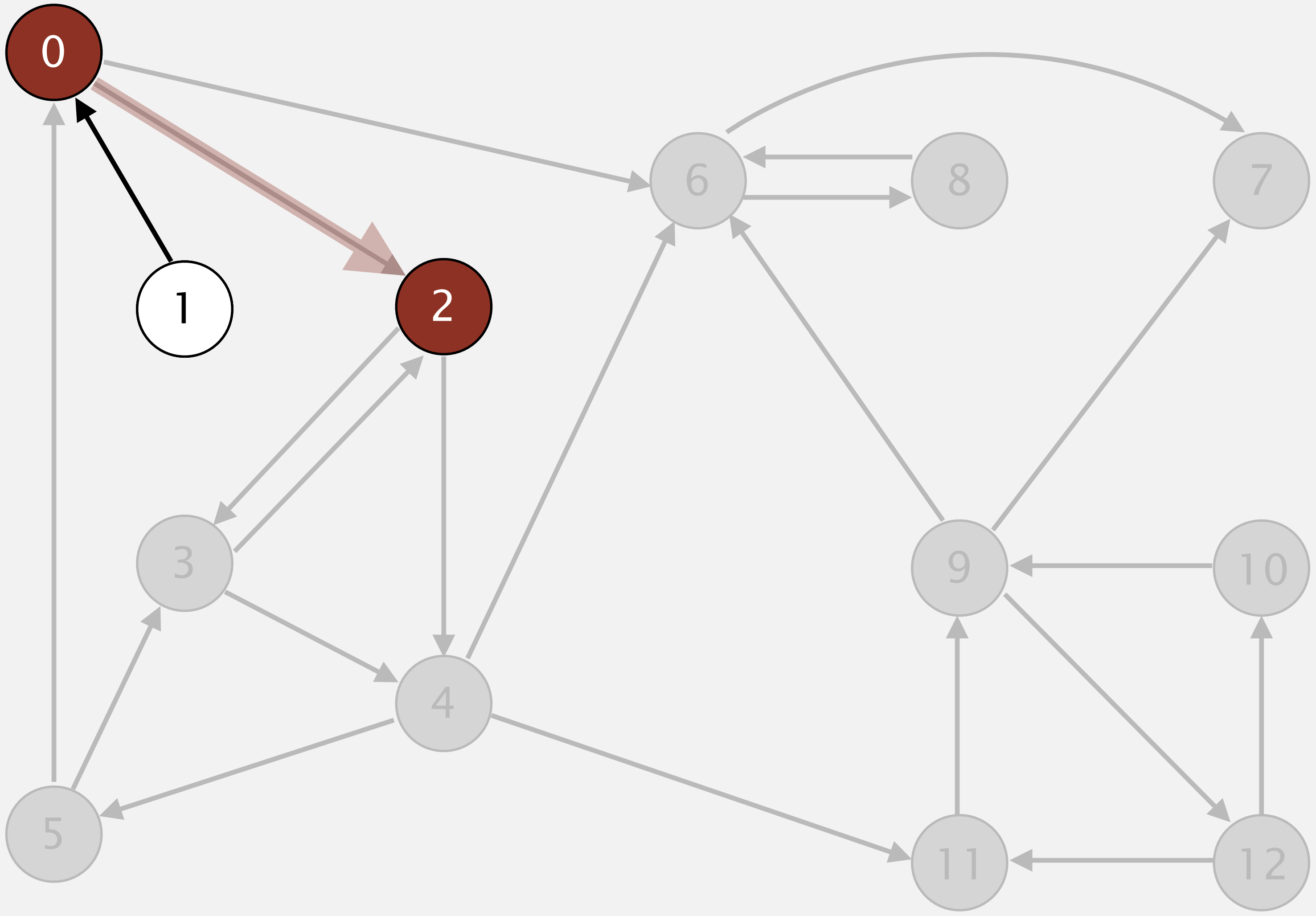
visit 2: check 4 and check 3

v	marked[]
0	T
1	F
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

2 4 5 3 11 9 12 10 6 7 8



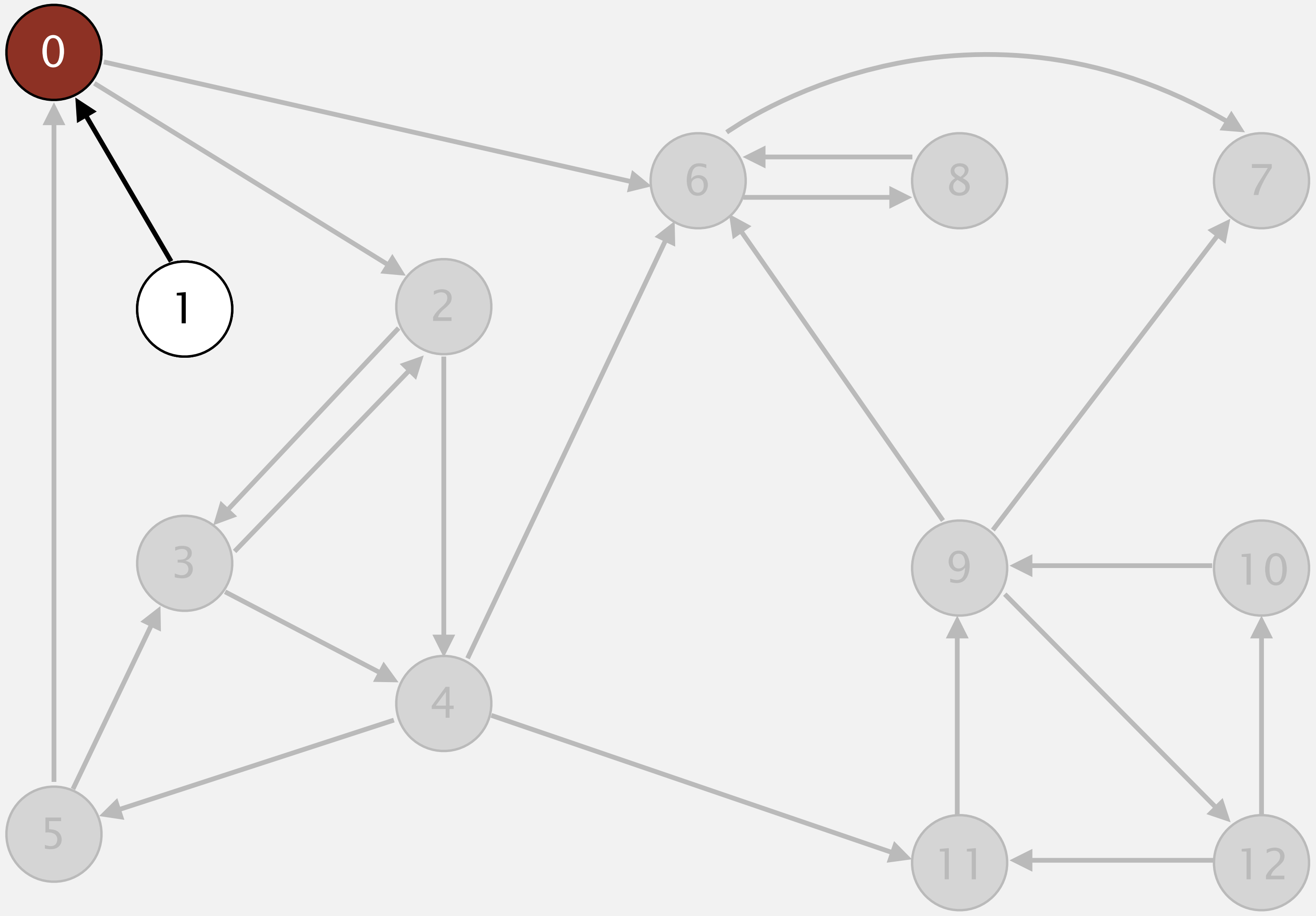
2 done

v	marked[]
0	T
1	F
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

0 2 4 5 3 11 9 12 10 6 7 8



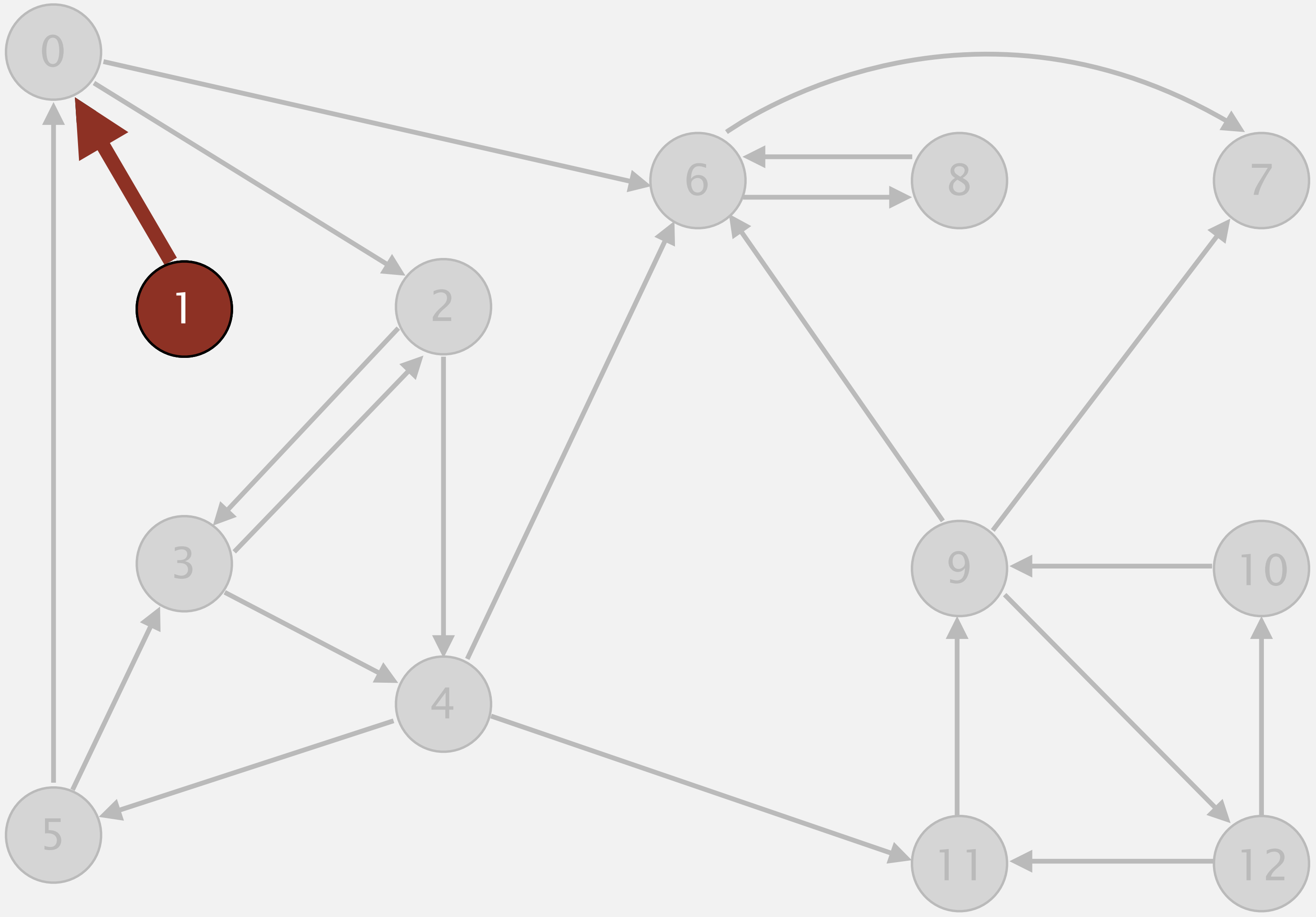
0 done

v	marked[]
0	T
1	F
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

0 2 4 5 3 11 9 12 10 6 7 8



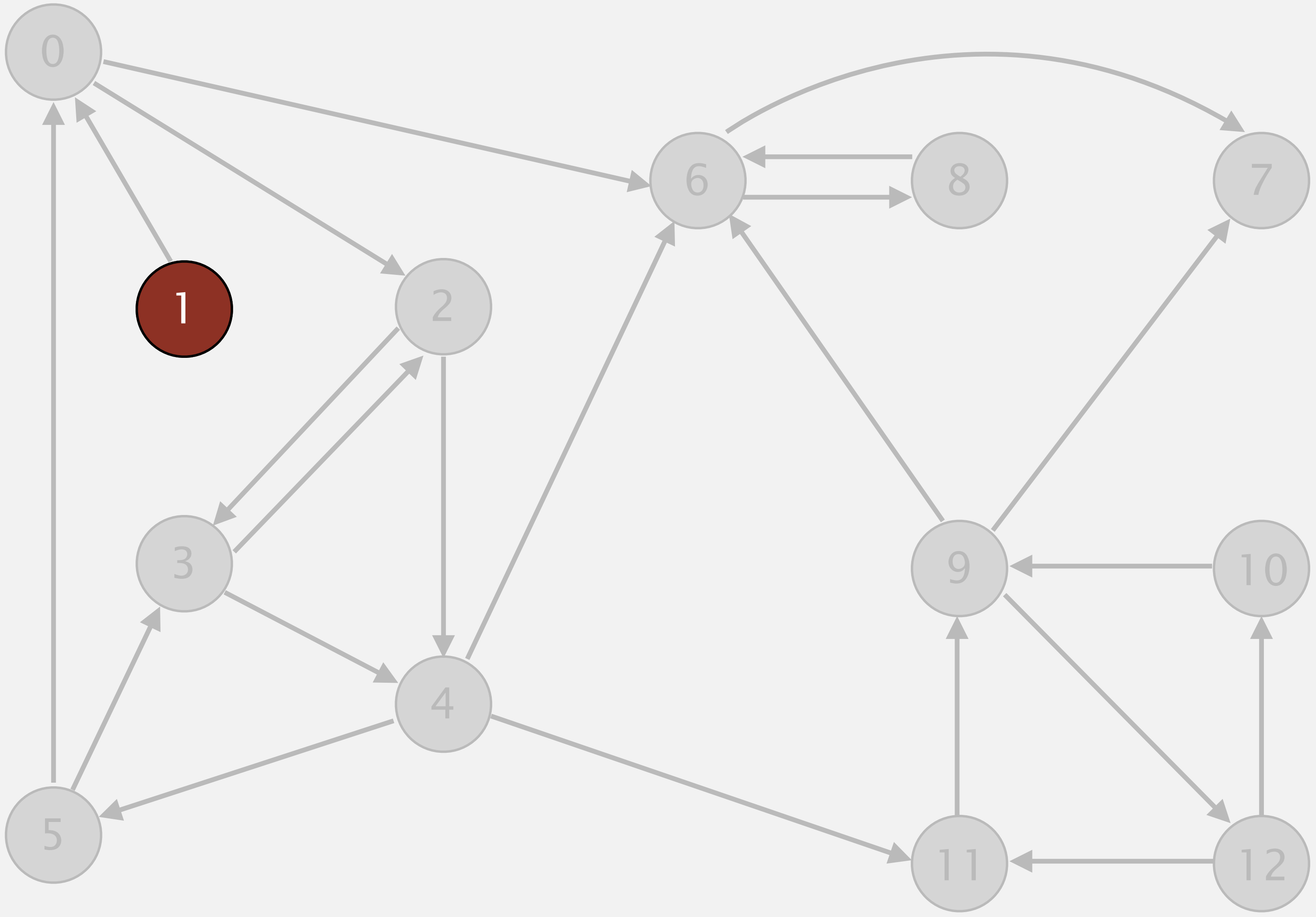
visit 1: check 0

v	marked[]
0	T
1	T
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



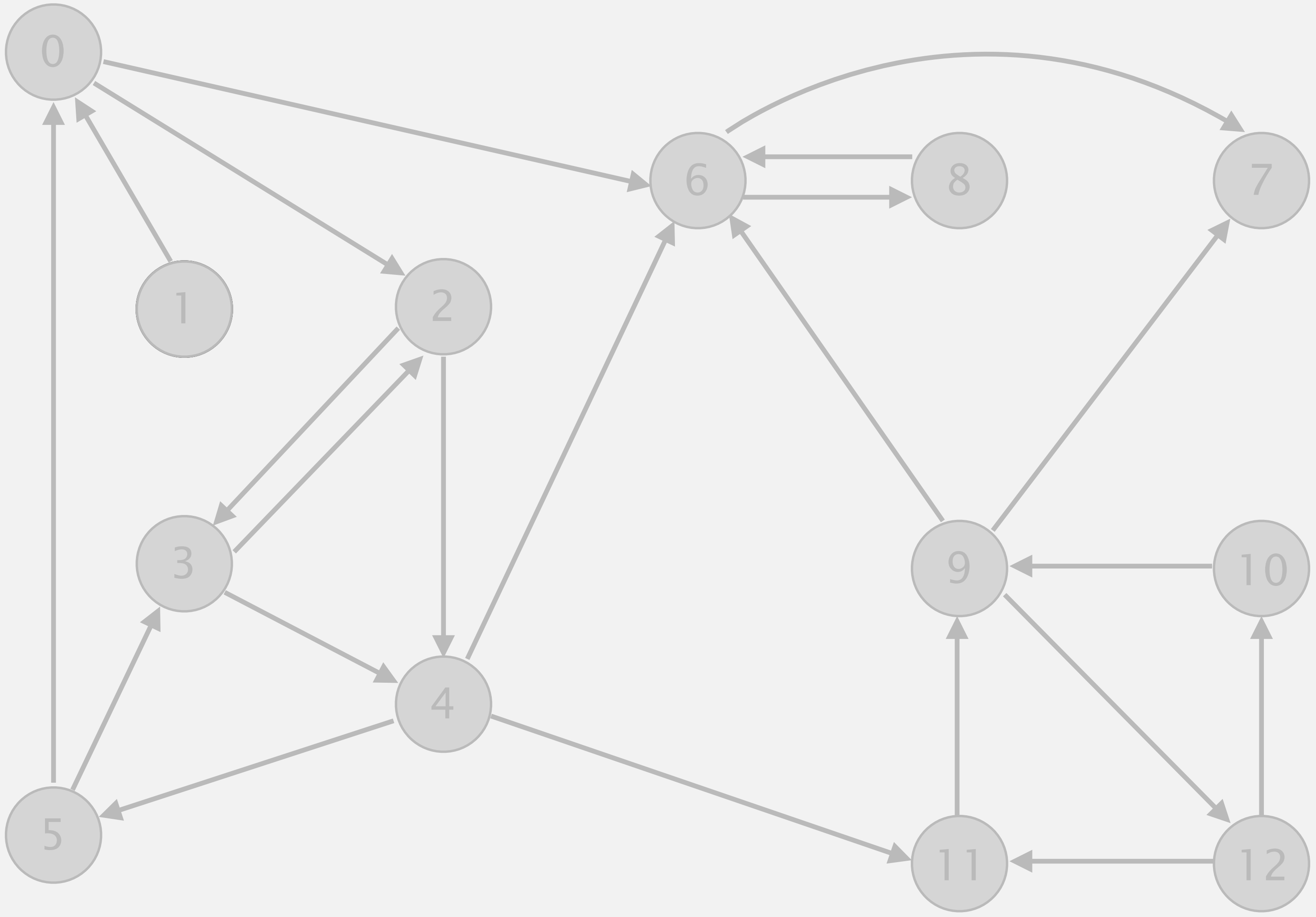
1 done

v	marked[]
0	T
1	T
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



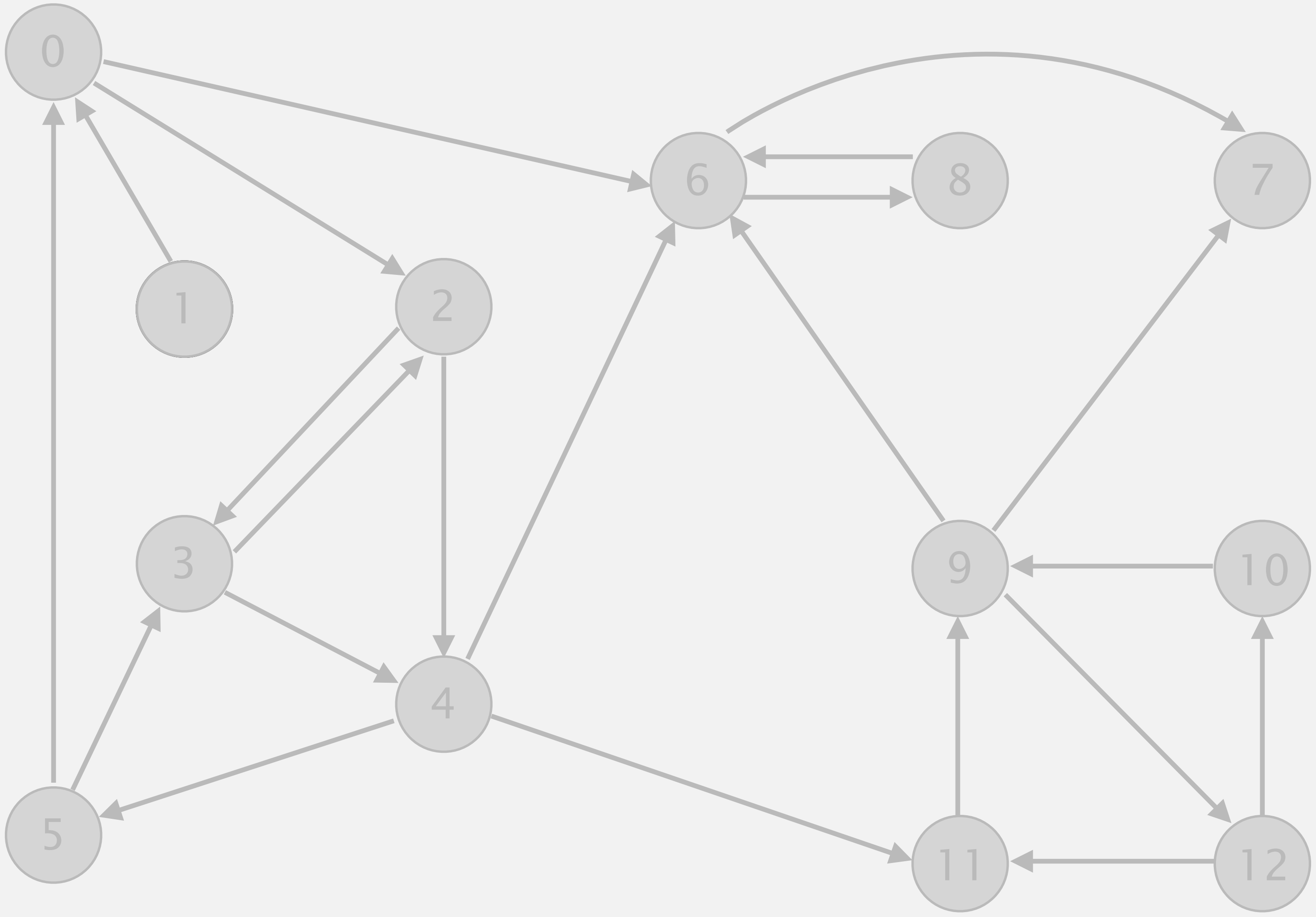
check 2 3 4 5 6 7 8 9 10 11 12

v	marked[]
0	T
1	T
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



reverse digraph G^R

Algorithms

Robert Sedgewick | Kevin Wayne

http://algs4.cs.princeton.edu

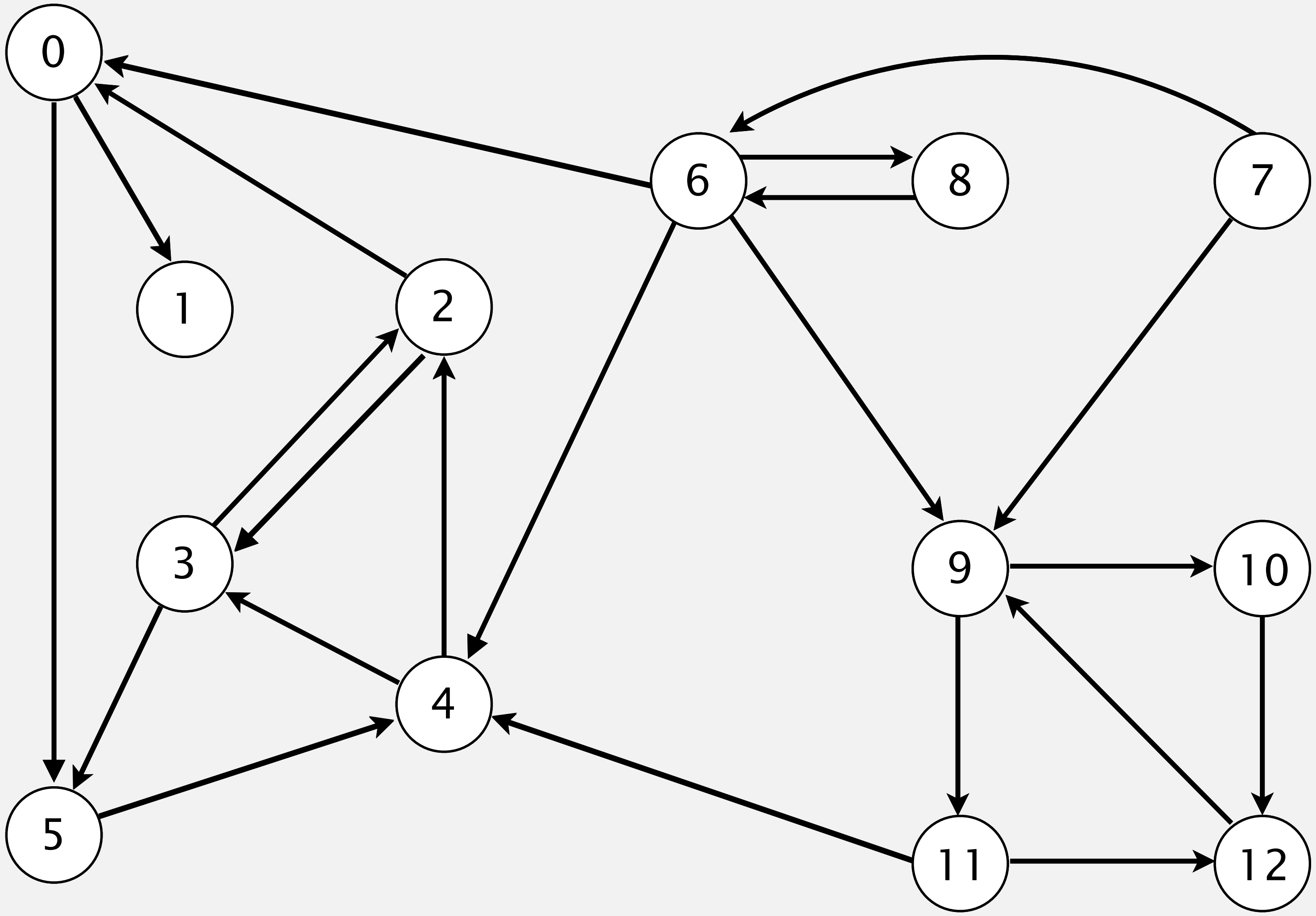
4.2 Kosaraju-Sharir Demo

- *DFS in reverse graph*
- *DFS in original graph*

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



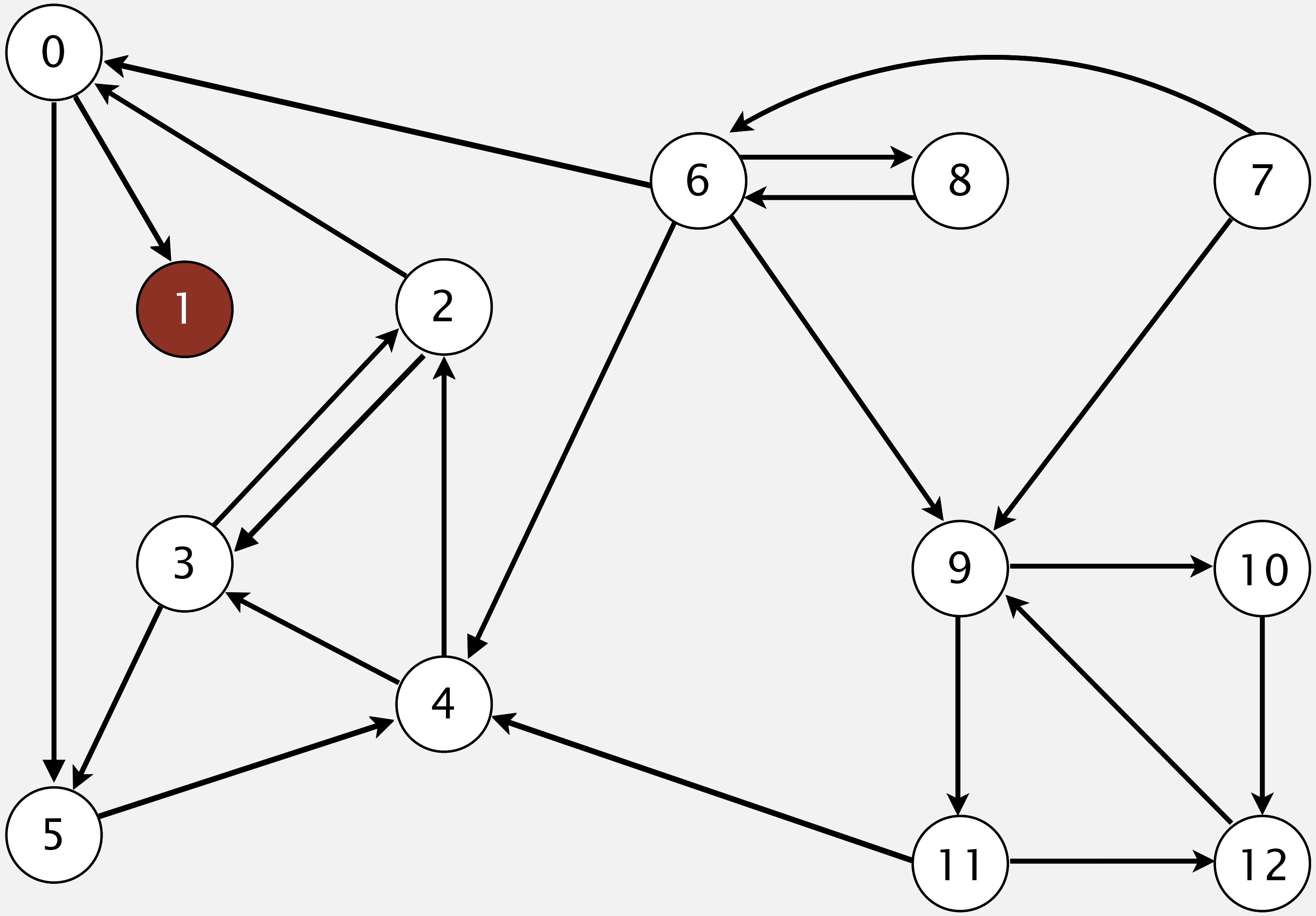
original digraph G

v	id[]
0	–
1	–
2	–
3	–
4	–
5	–
6	–
7	–
8	–
9	–
10	–
11	–
12	–

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



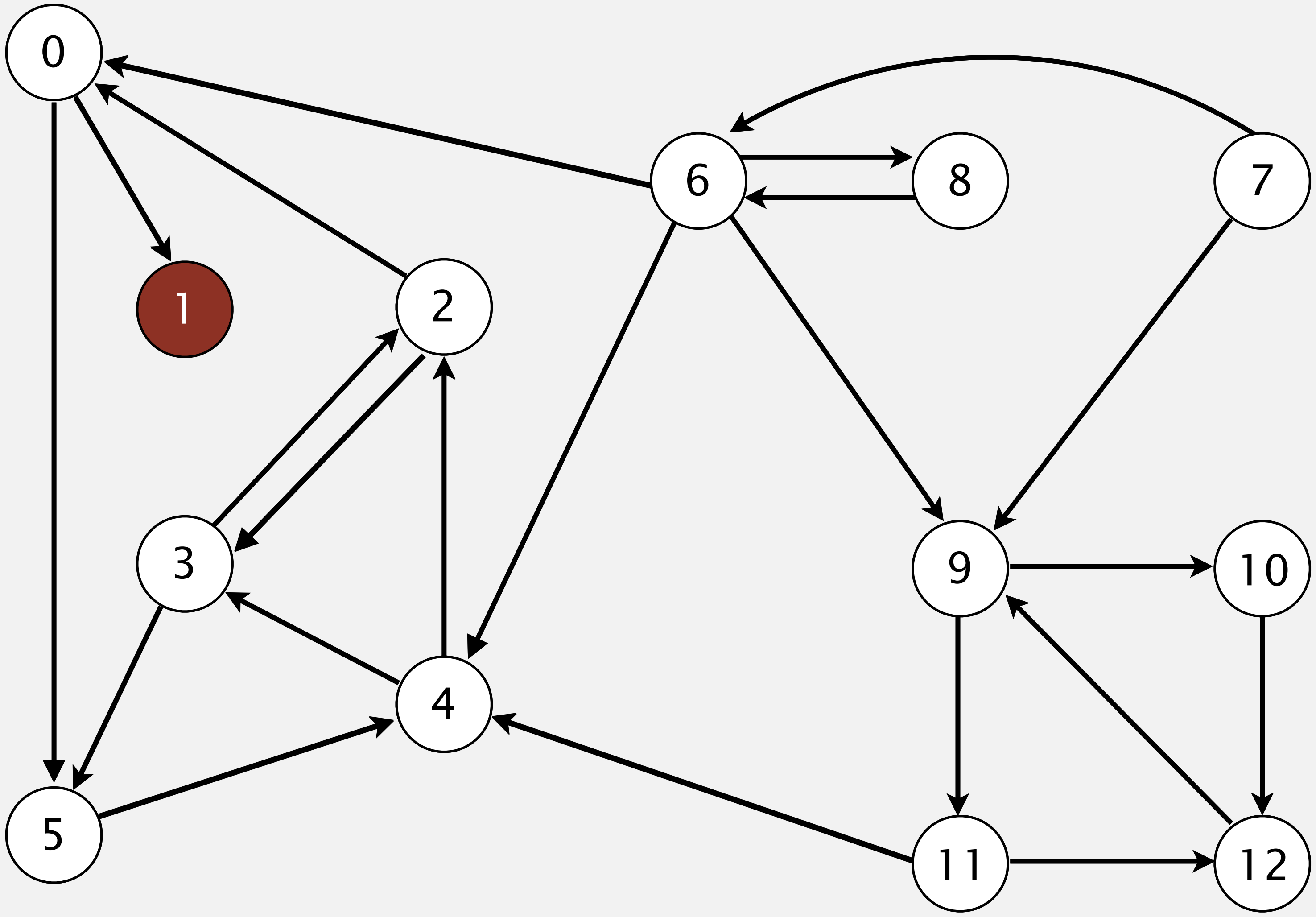
v	id[]
0	–
1	0
2	–
3	–
4	–
5	–
6	–
7	–
8	–
9	–
10	–
11	–
12	–

visit 1

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



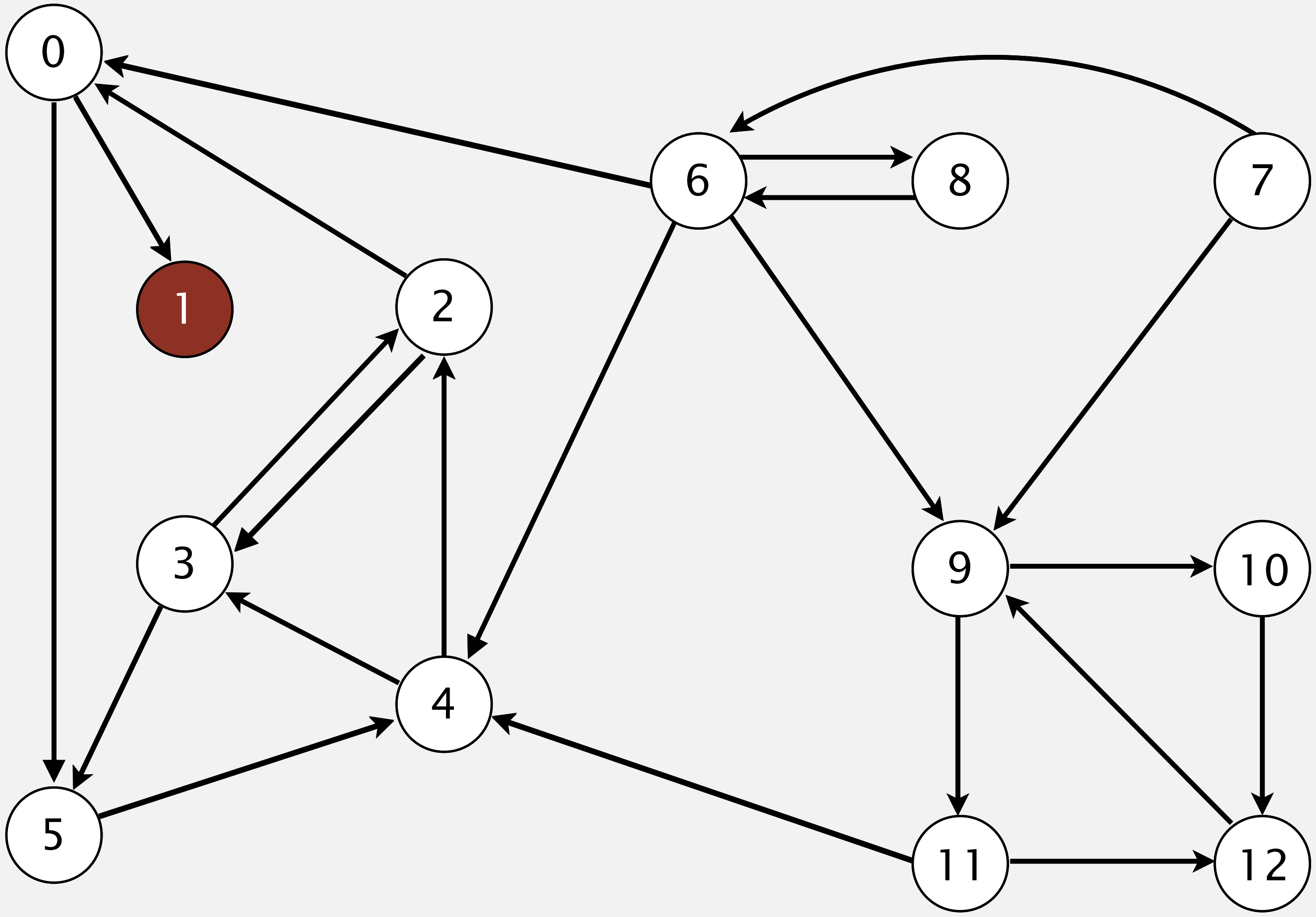
1 done

v	id[]
0	–
1	0
2	–
3	–
4	–
5	–
6	–
7	–
8	–
9	–
10	–
11	–
12	–

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



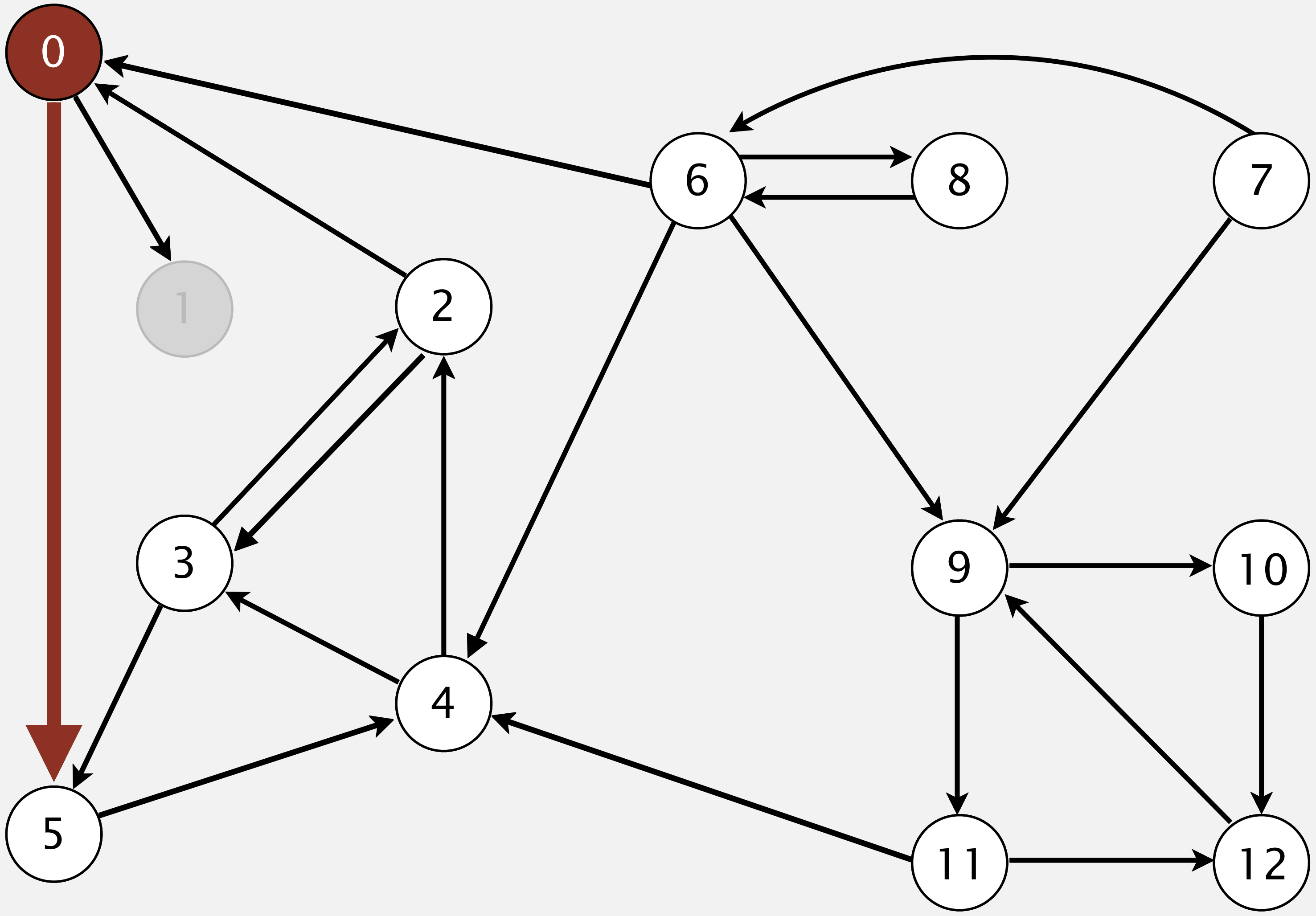
strong component: 1

v	id[]
0	–
1	0
2	–
3	–
4	–
5	–
6	–
7	–
8	–
9	–
10	–
11	–
12	–

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



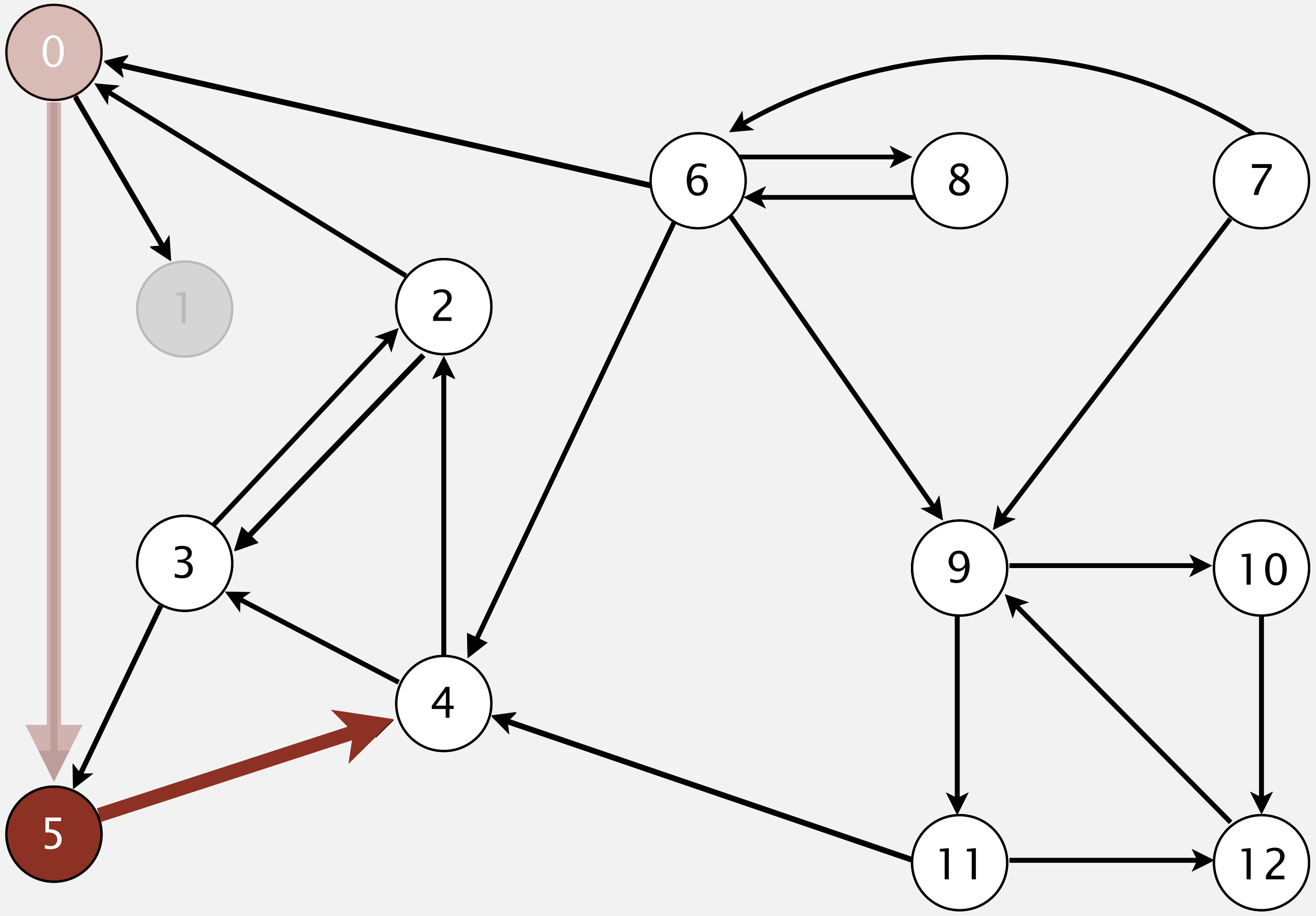
v	id[]
0	1
1	0
2	–
3	–
4	–
5	–
6	–
7	–
8	–
9	–
10	–
11	–
12	–

visit 0: check 5 and check 1

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



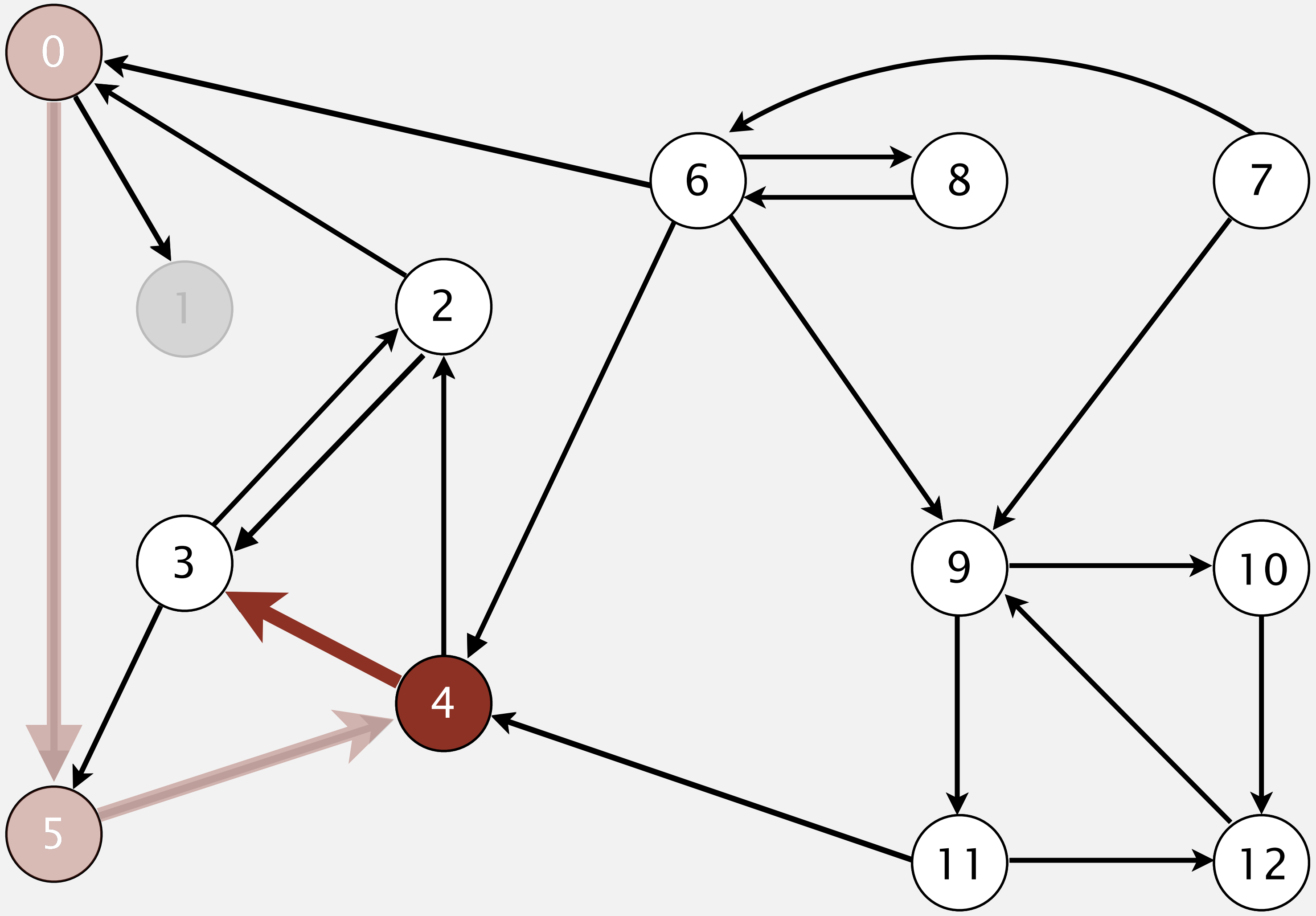
visit 5: check 4

v	id[]
0	1
1	0
2	–
3	–
4	–
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



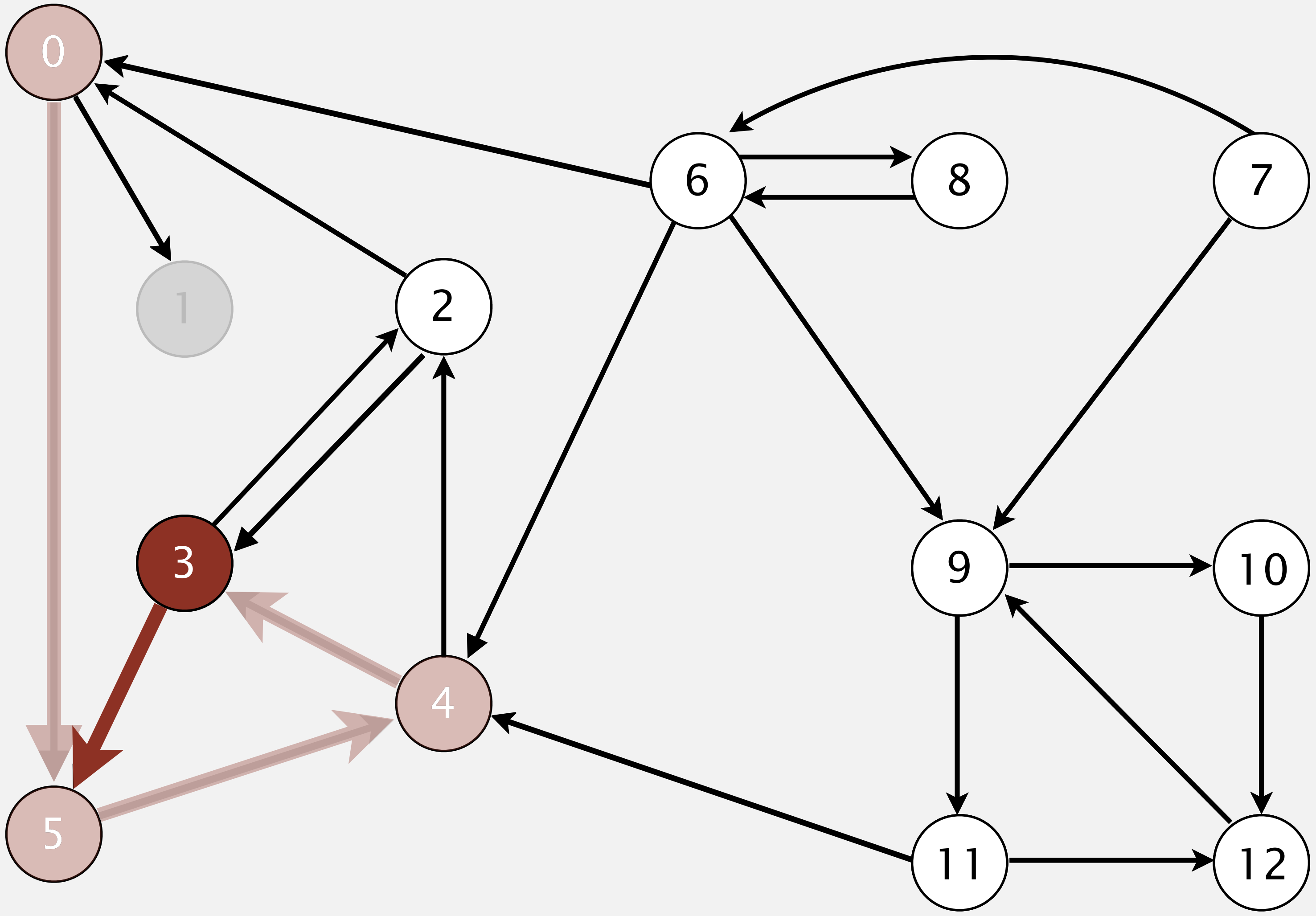
visit 4: check 3 and check 2

v	id[]
0	1
1	0
2	–
3	–
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



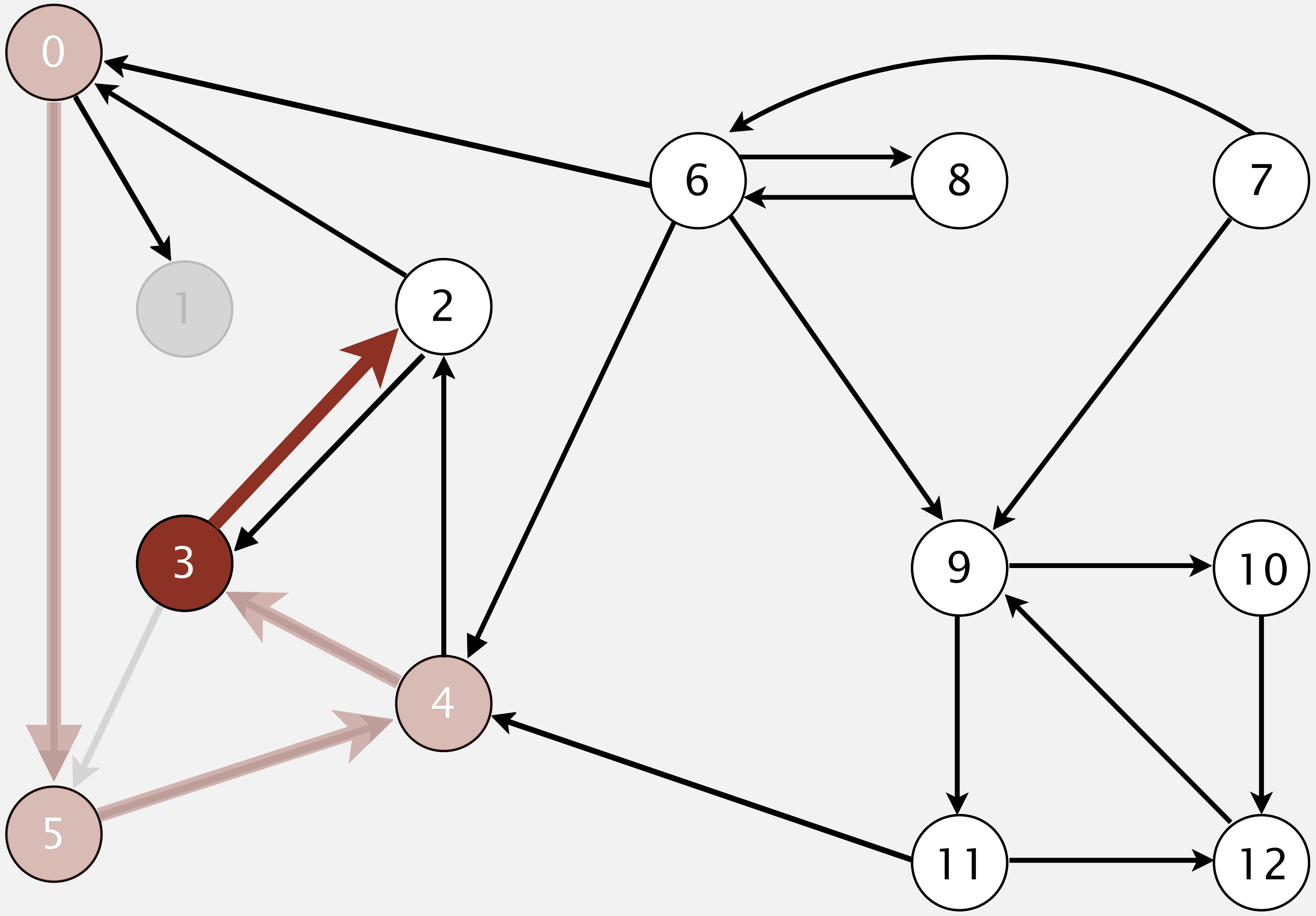
v	id[]
0	1
1	0
2	–
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

visit 3: check 5 and check 2

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



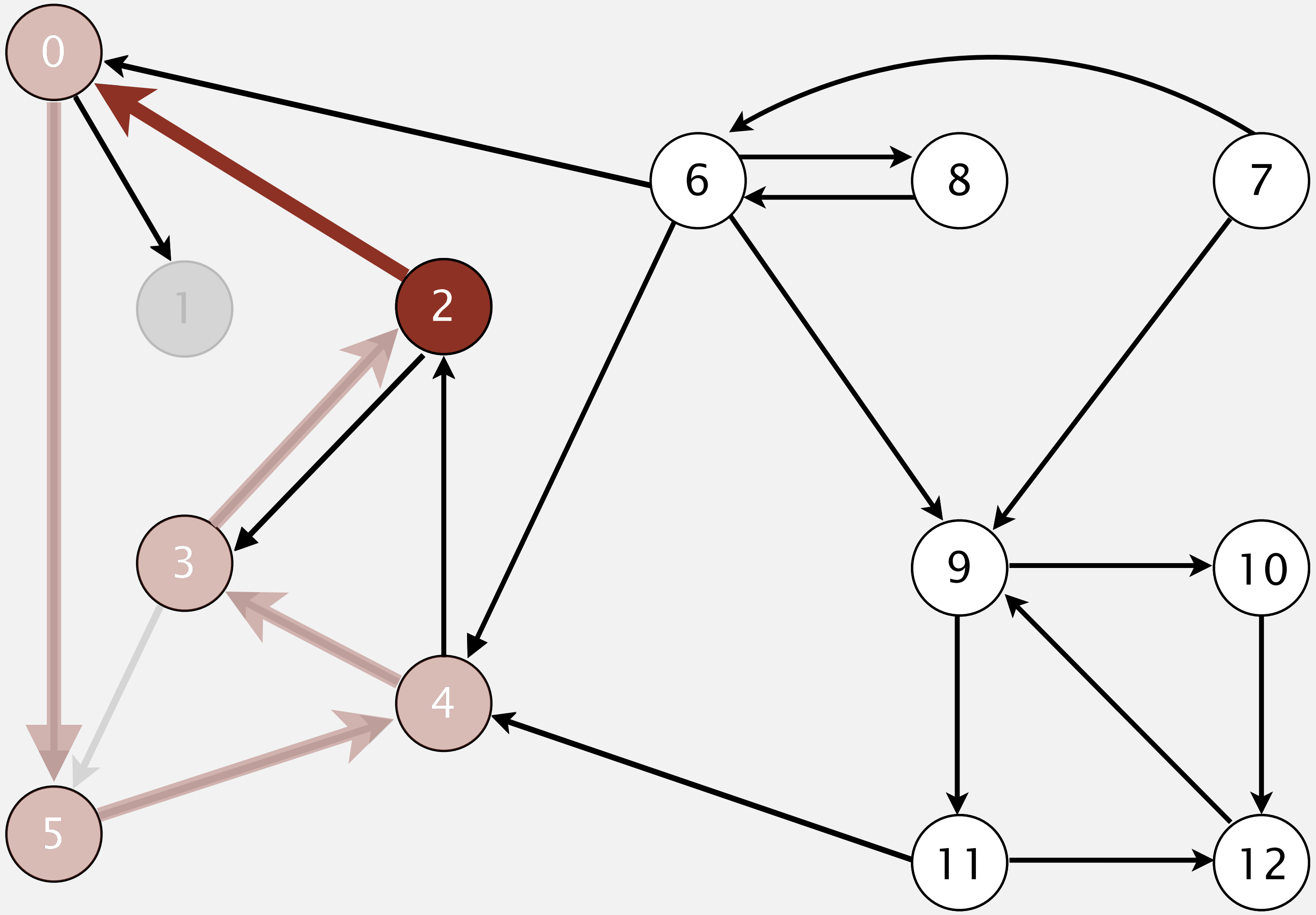
visit 3: check 5 and check 2

v	id[]
0	1
1	0
2	–
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



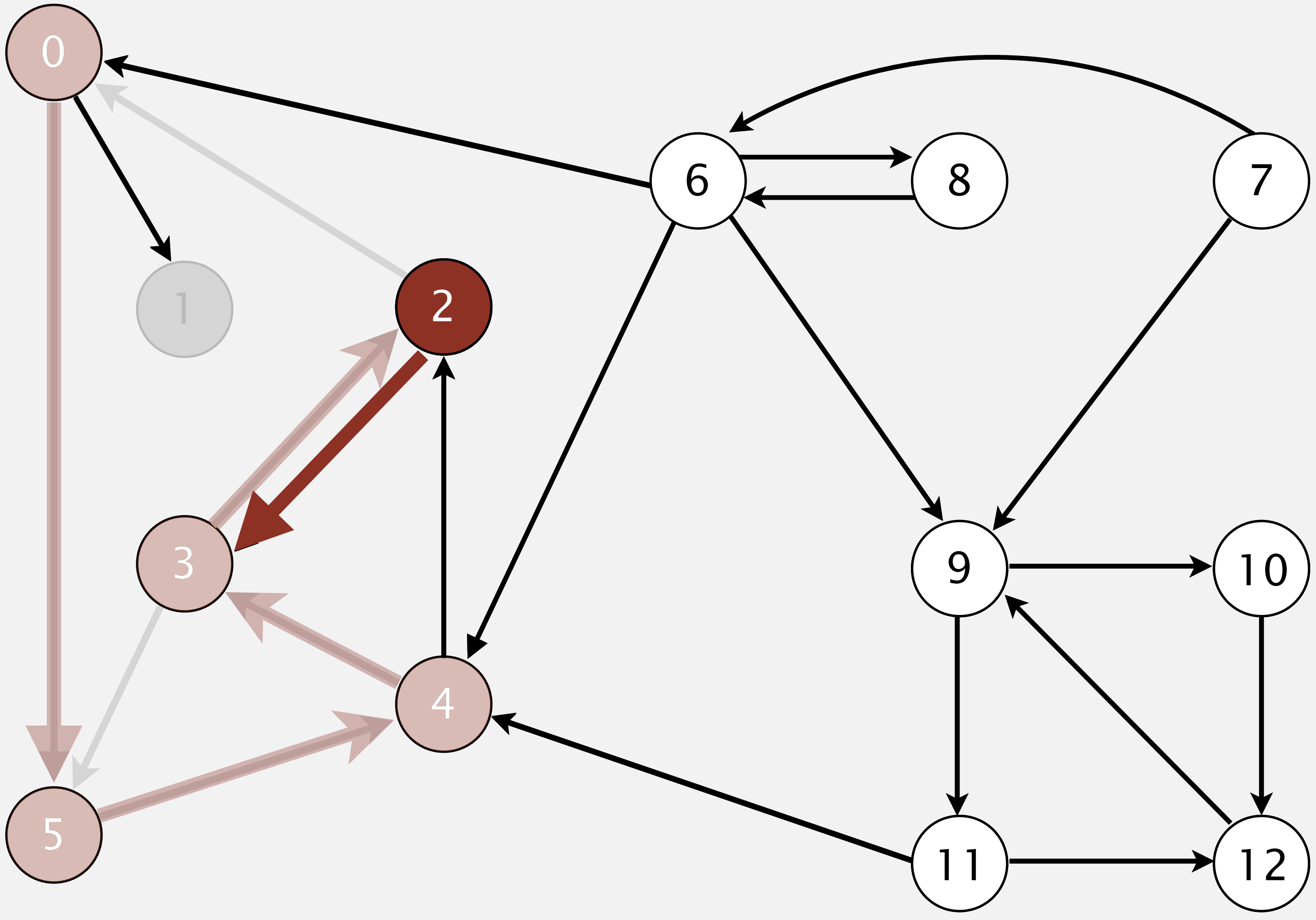
visit 2: check 0 and check 3

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



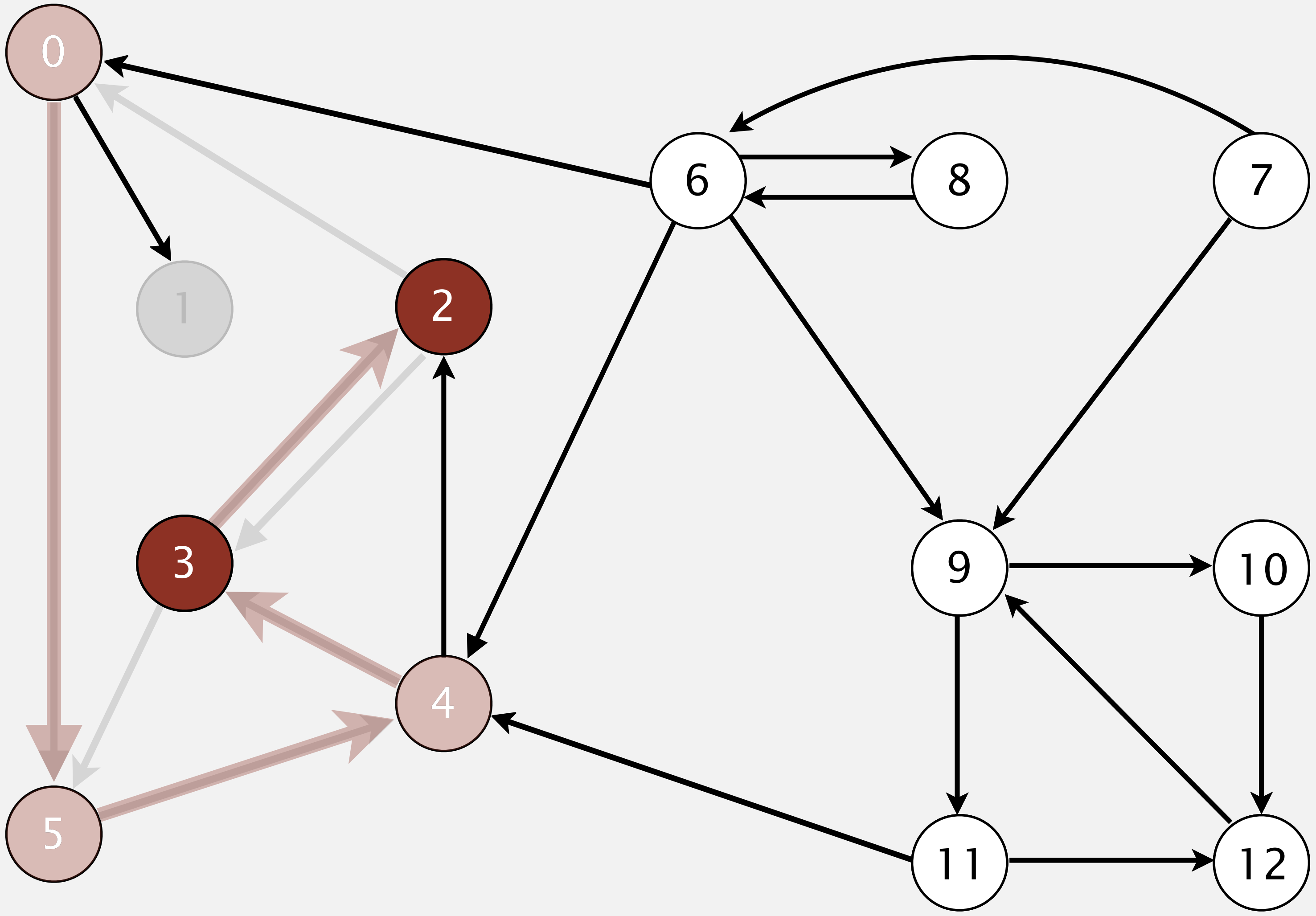
visit 2: check 0 and check 3

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



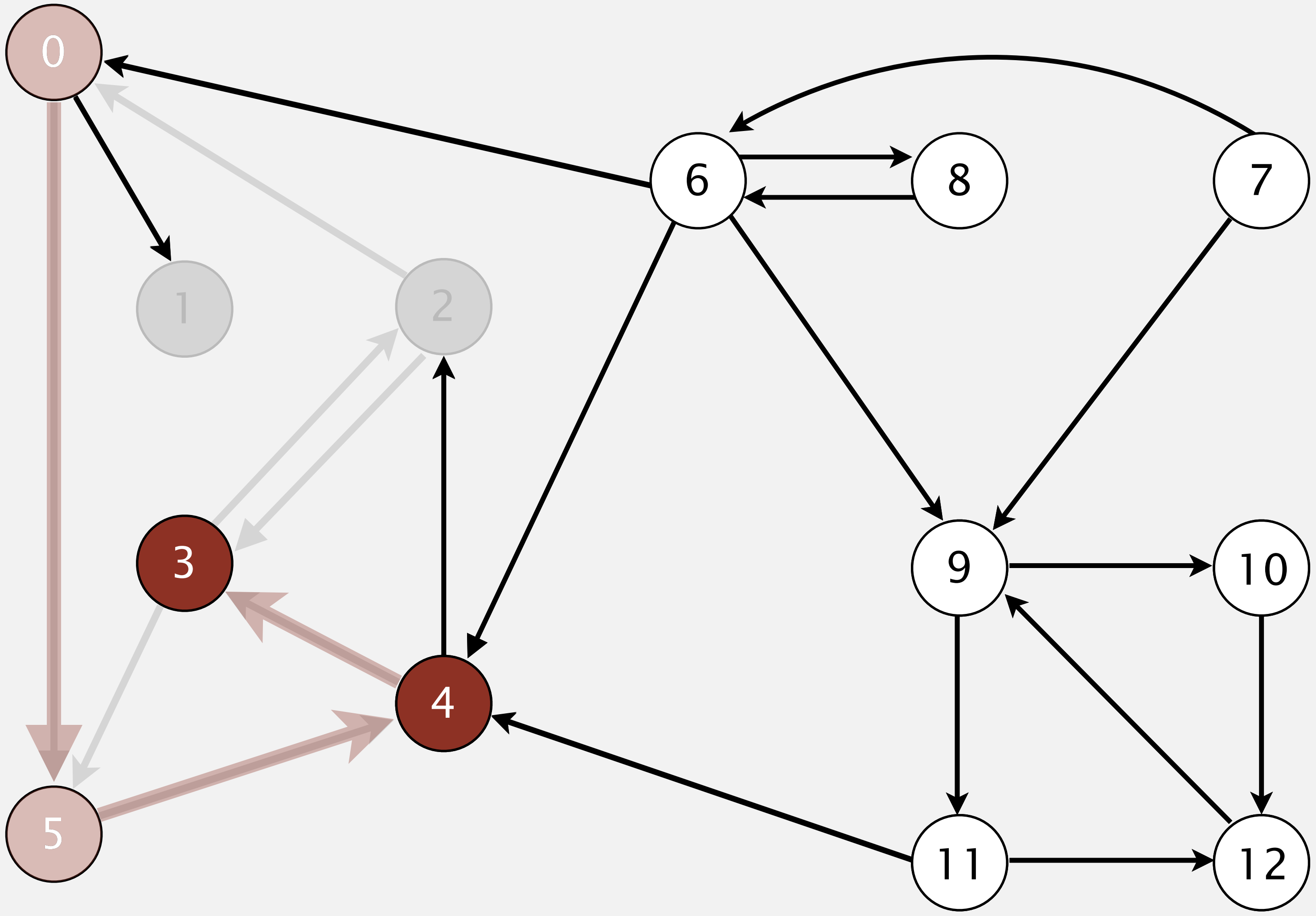
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

2 done

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



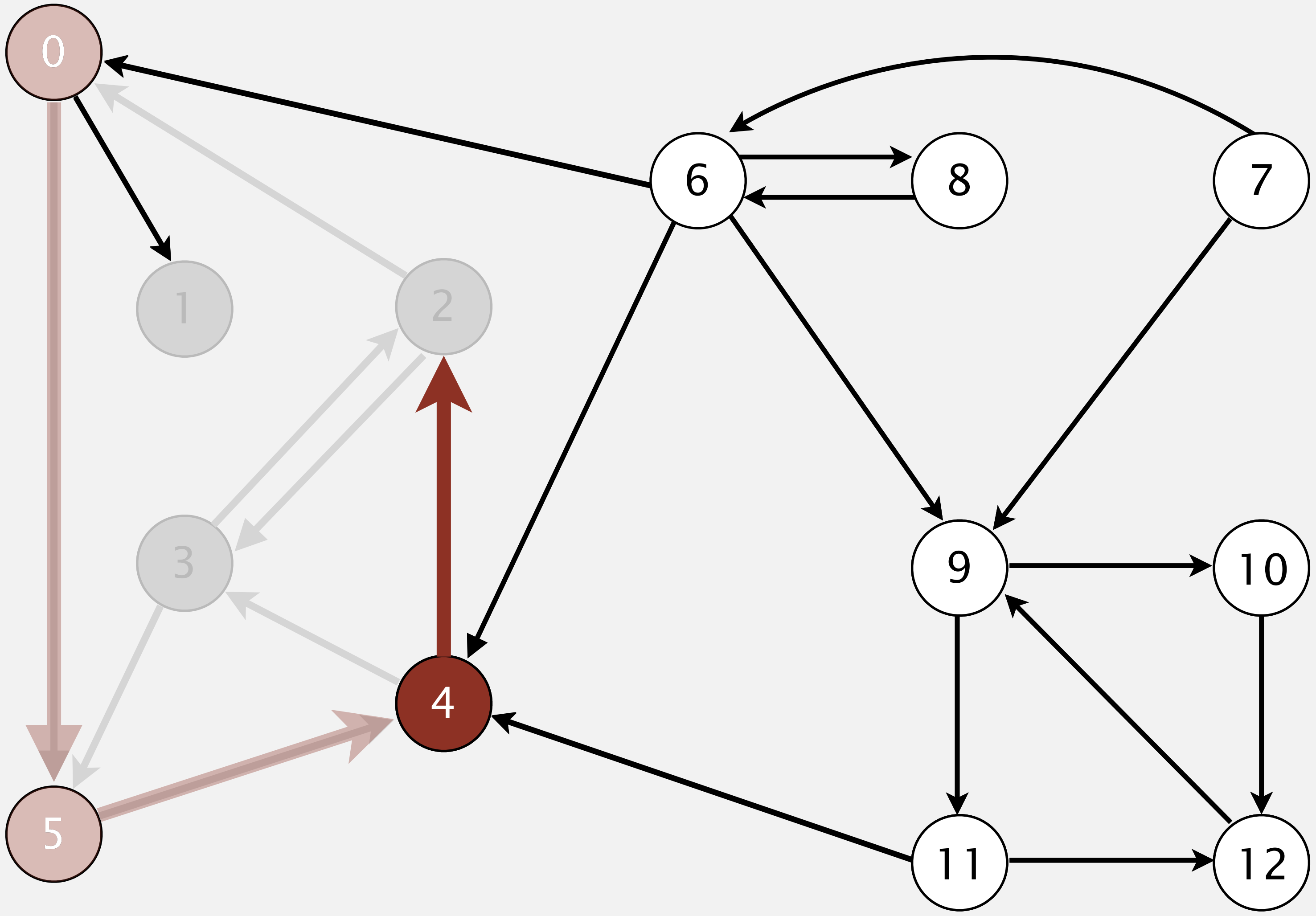
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

3 done

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



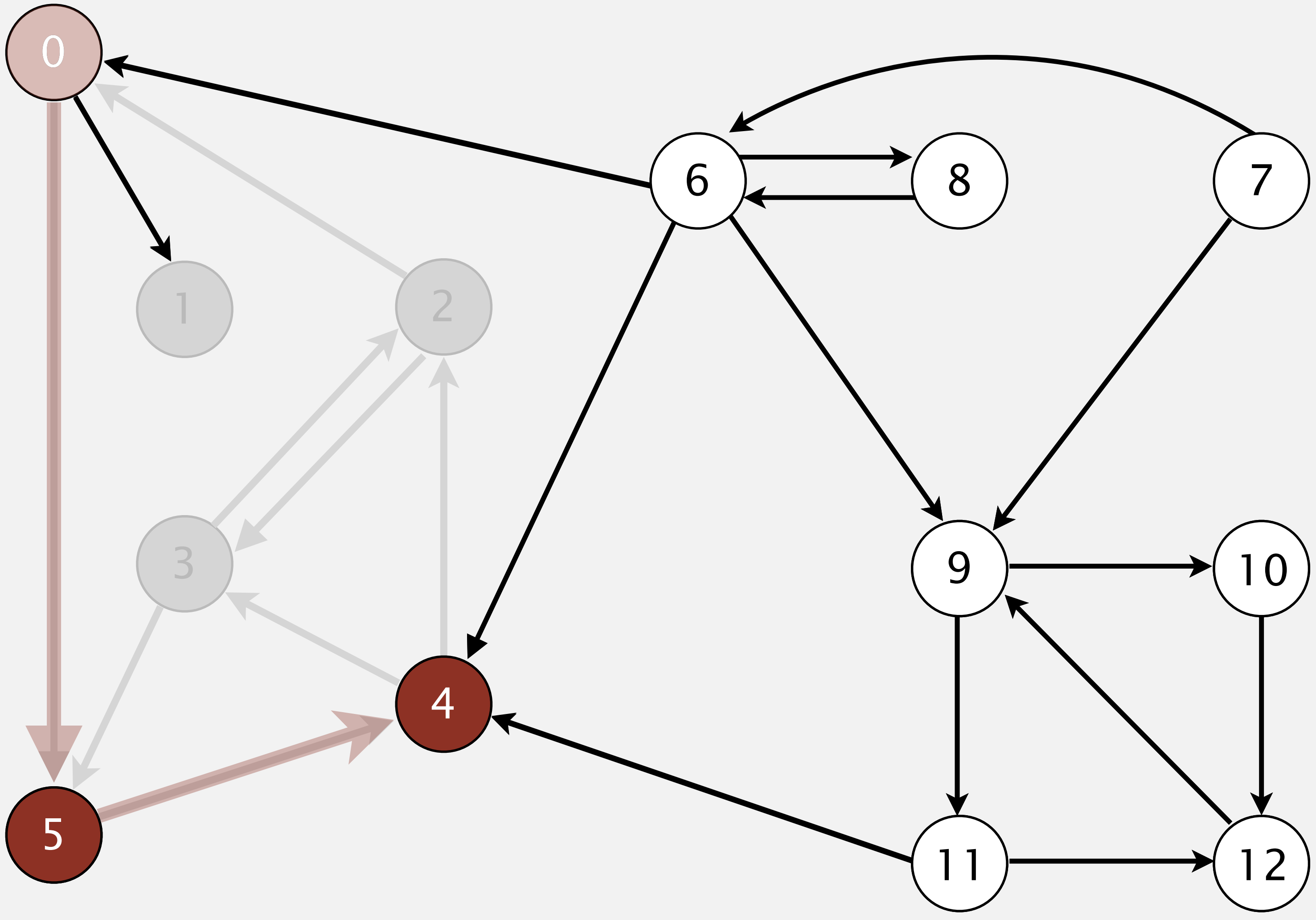
visit 4: check 3 and check 2

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



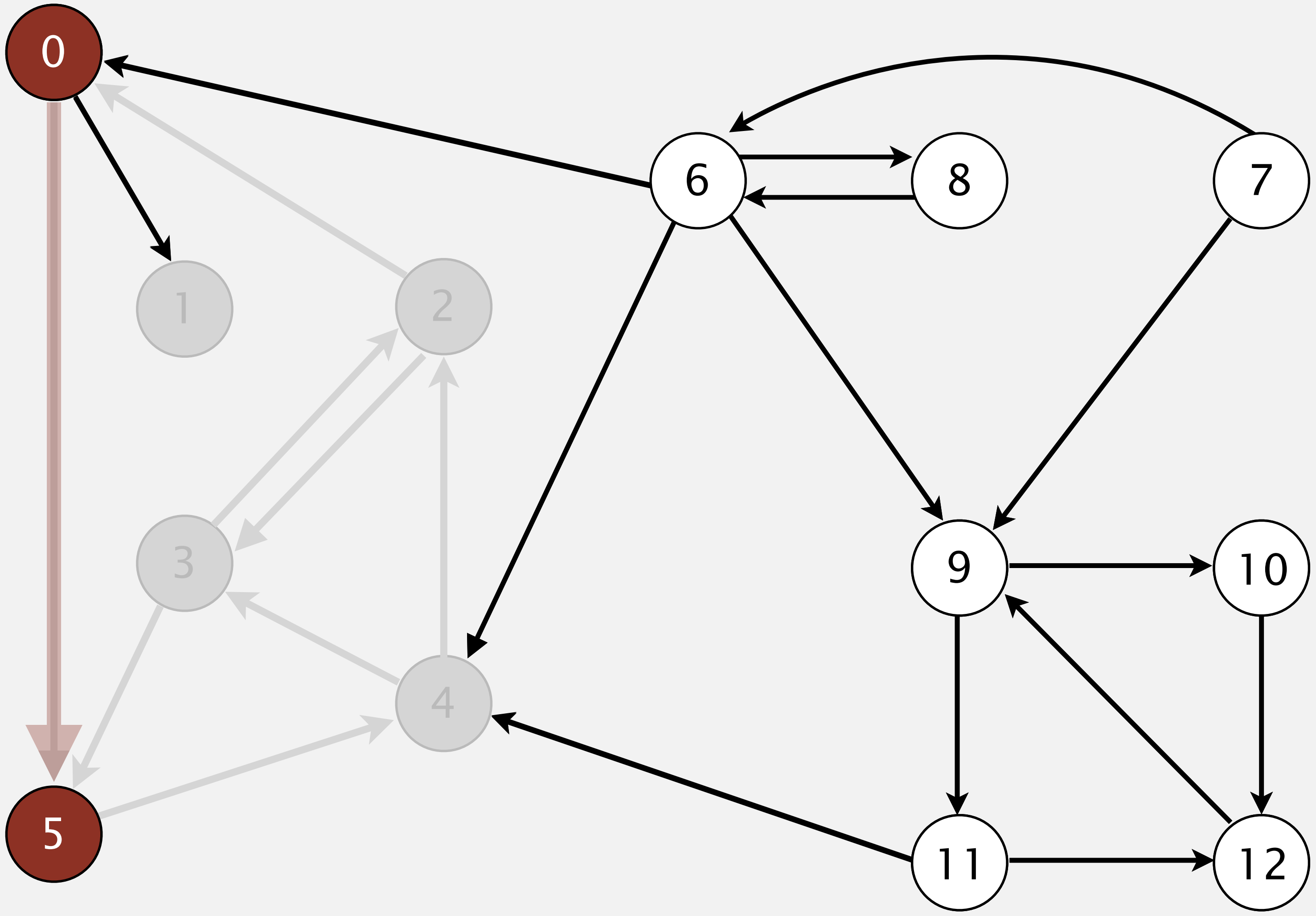
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

4 done

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



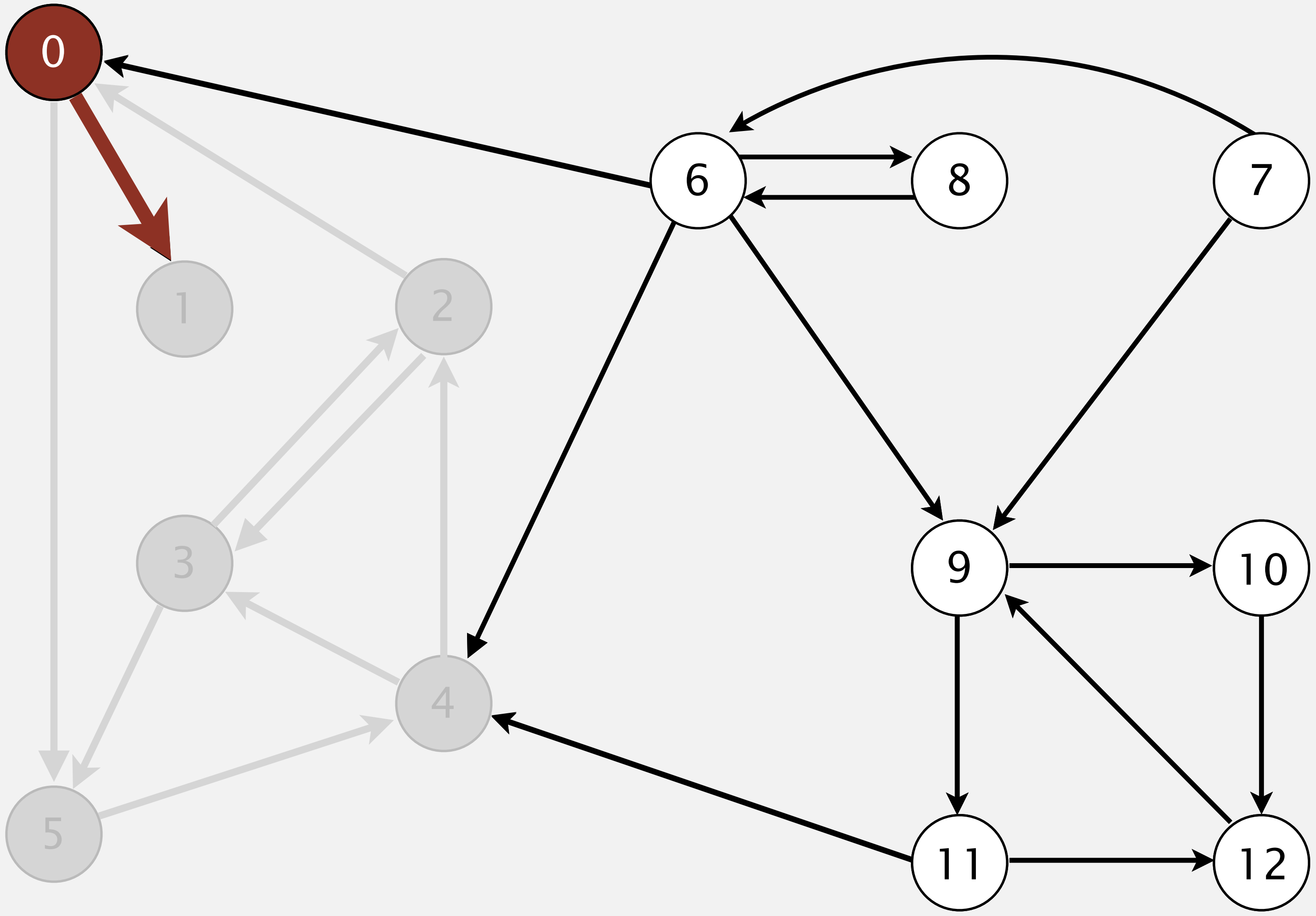
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

5 done

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



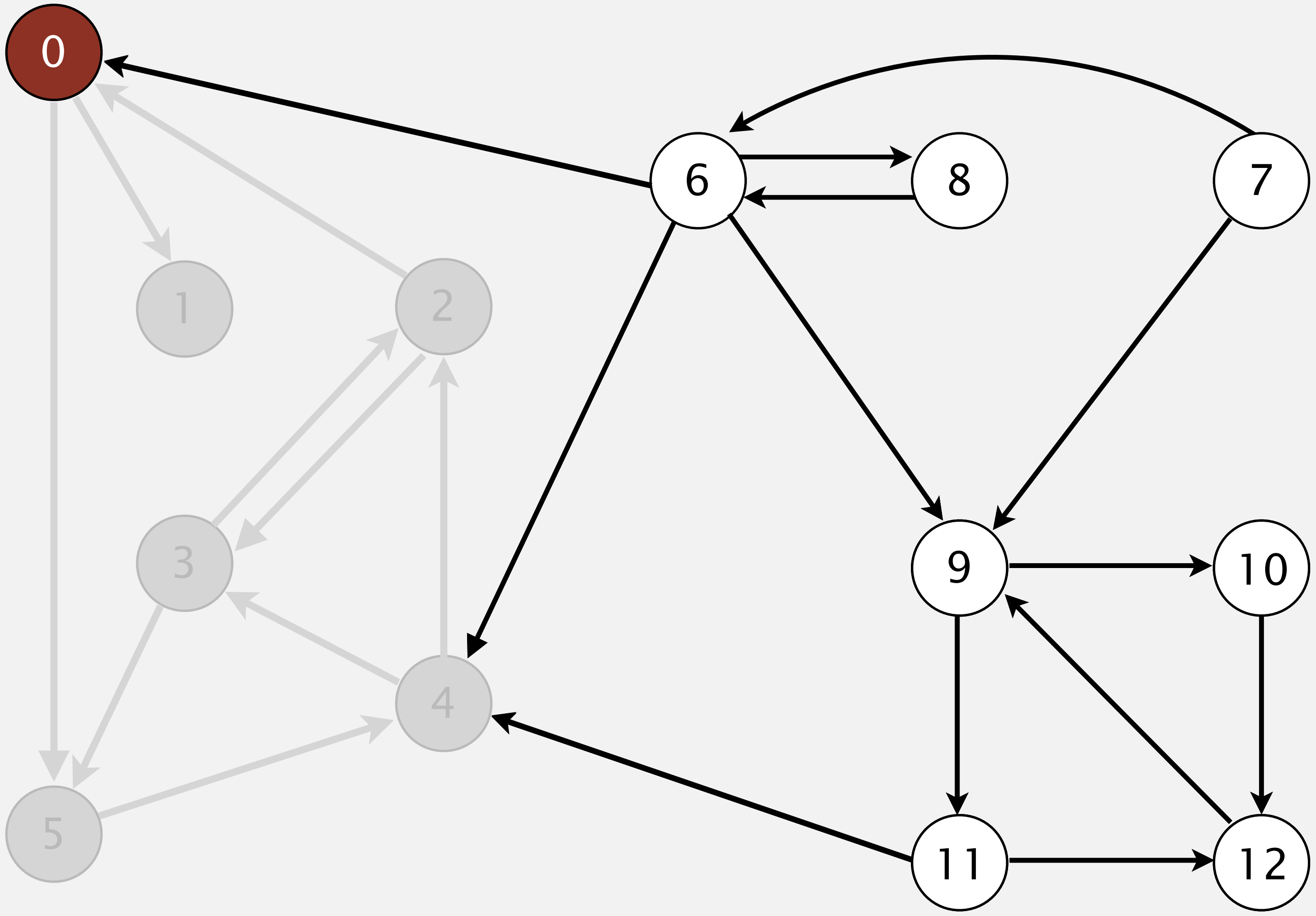
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

visit 0: check 5 and check 1

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



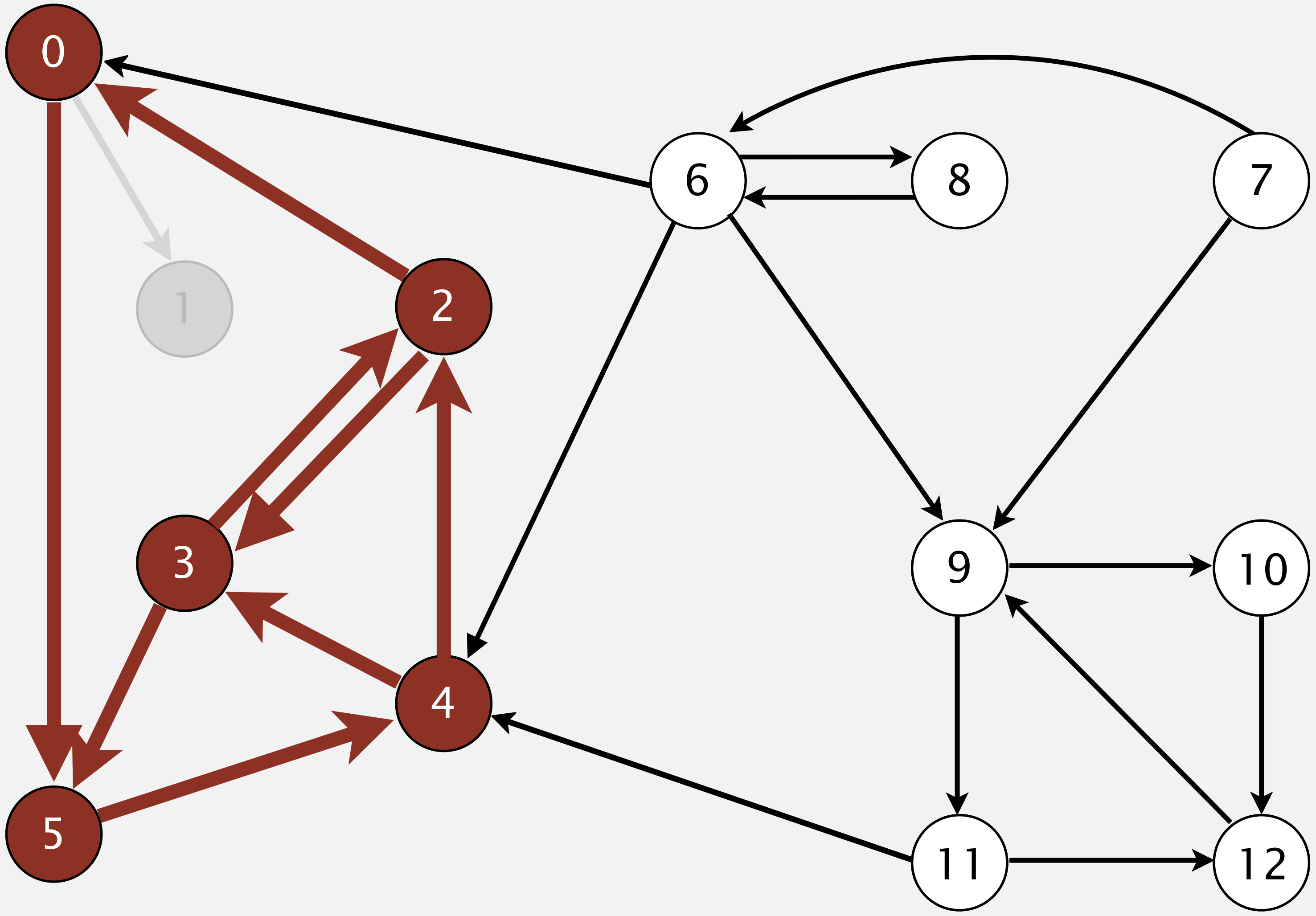
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

0 done

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



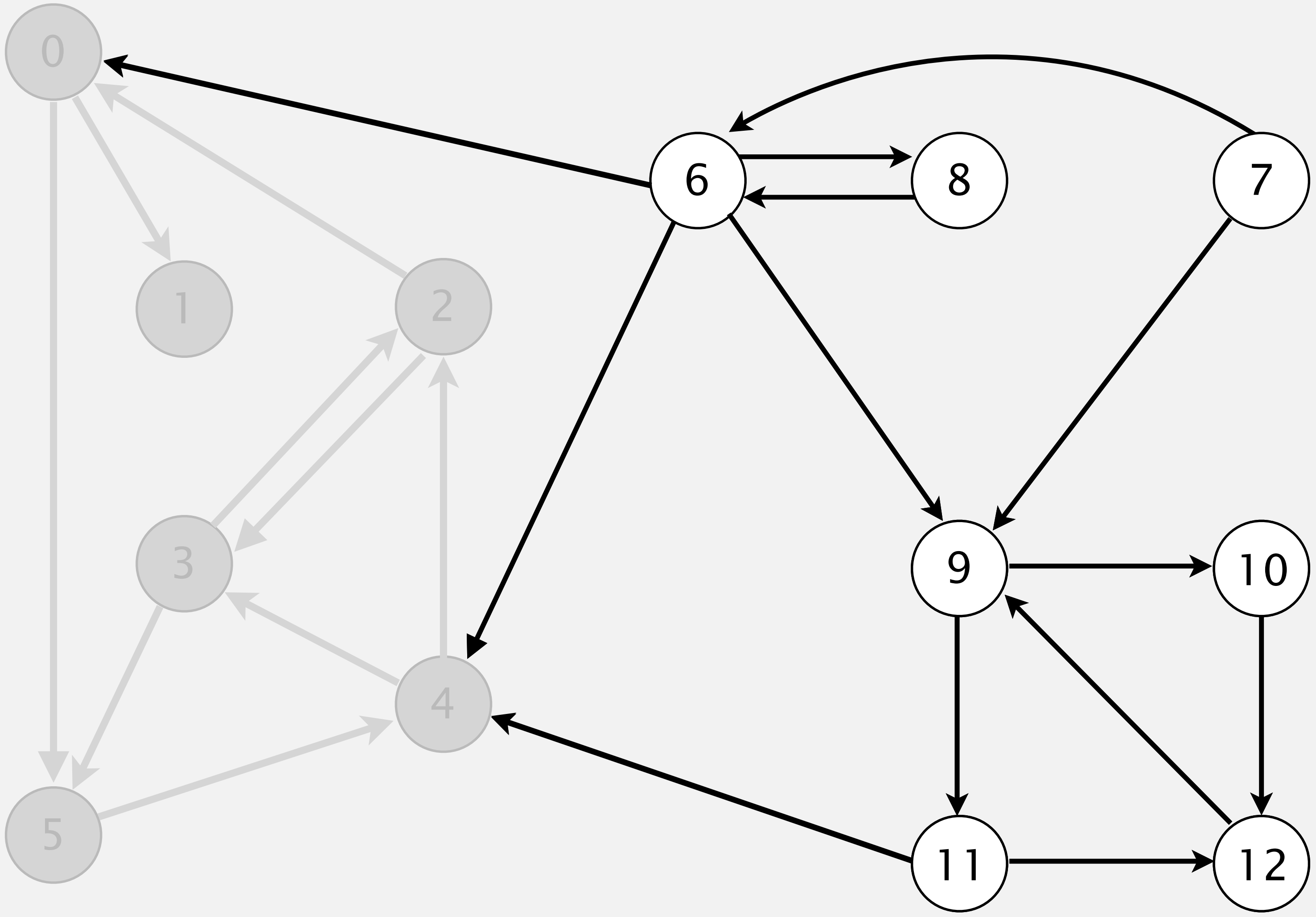
strong component: 0 2 3 4 5

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



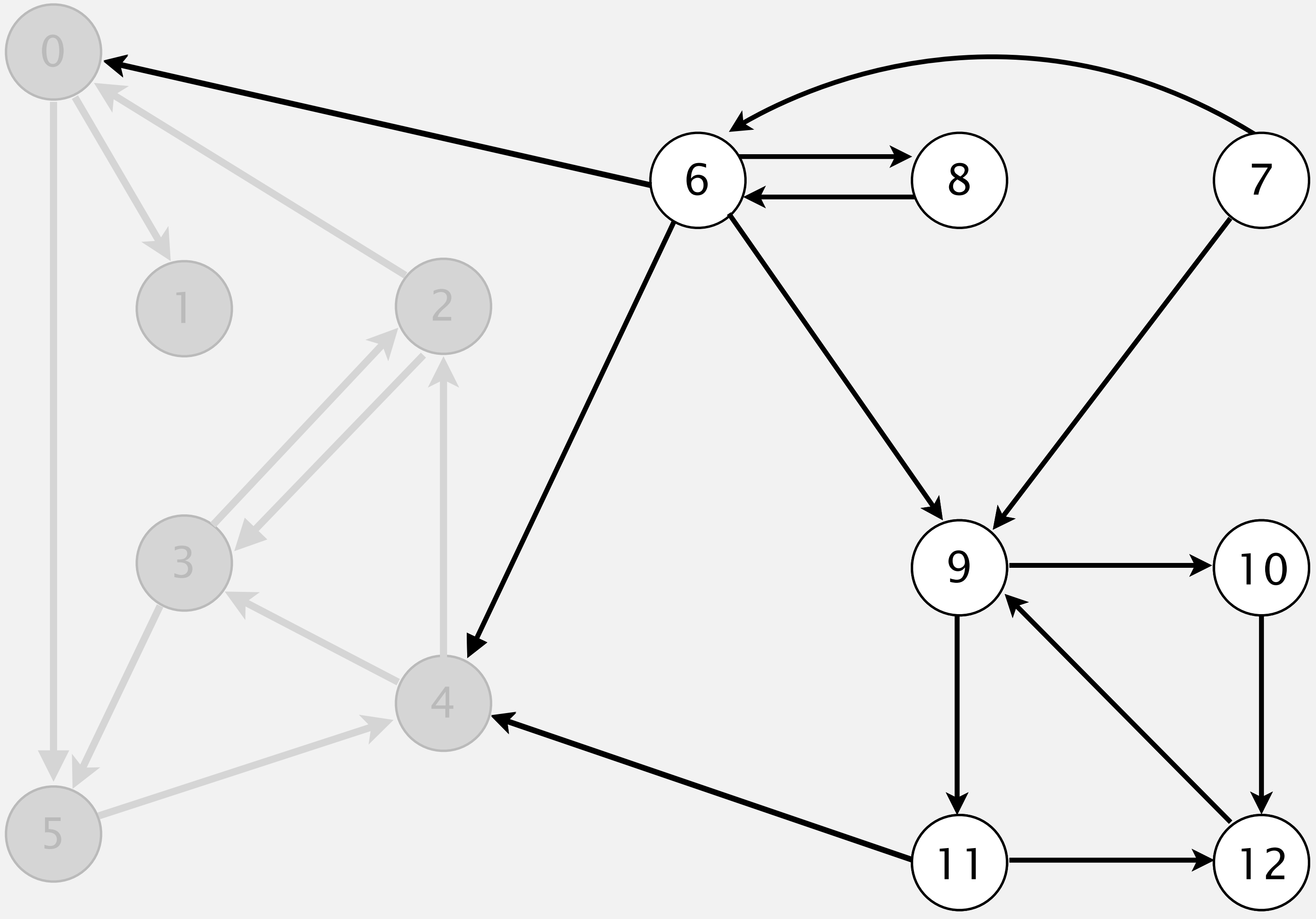
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

check 2

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 **4** **5 3 11 9 12 10 6 7 8**



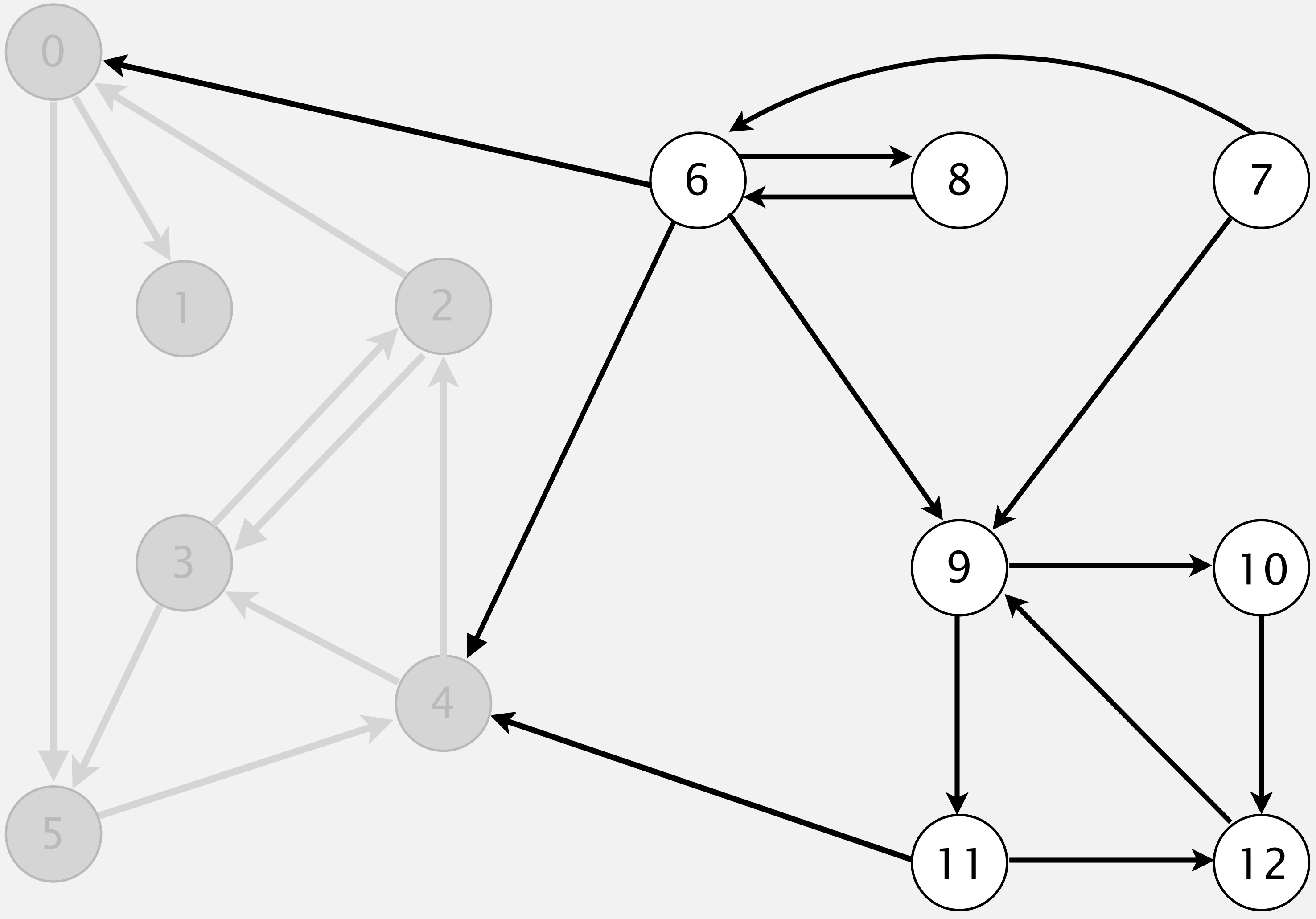
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

check 4

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



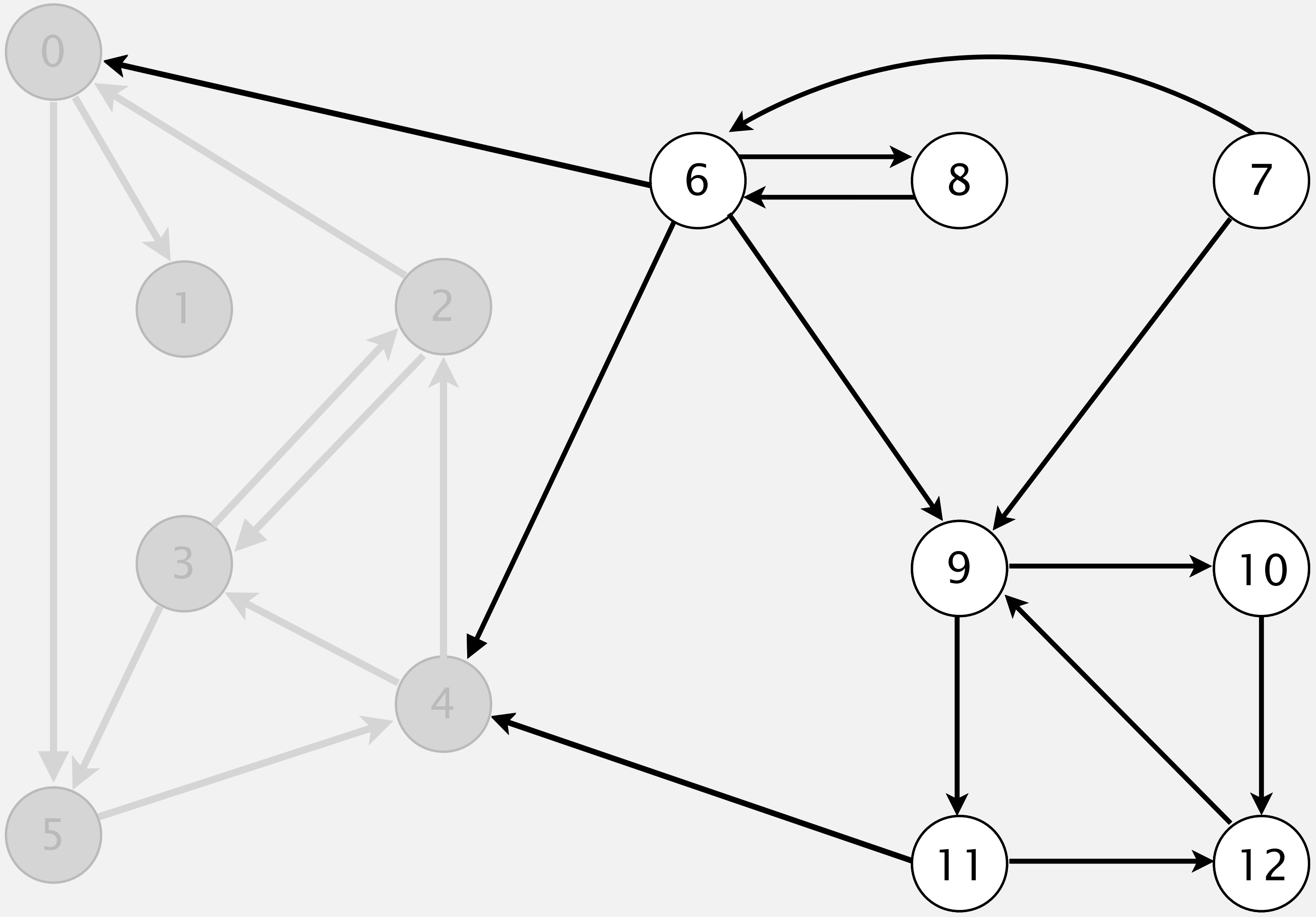
check 5

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



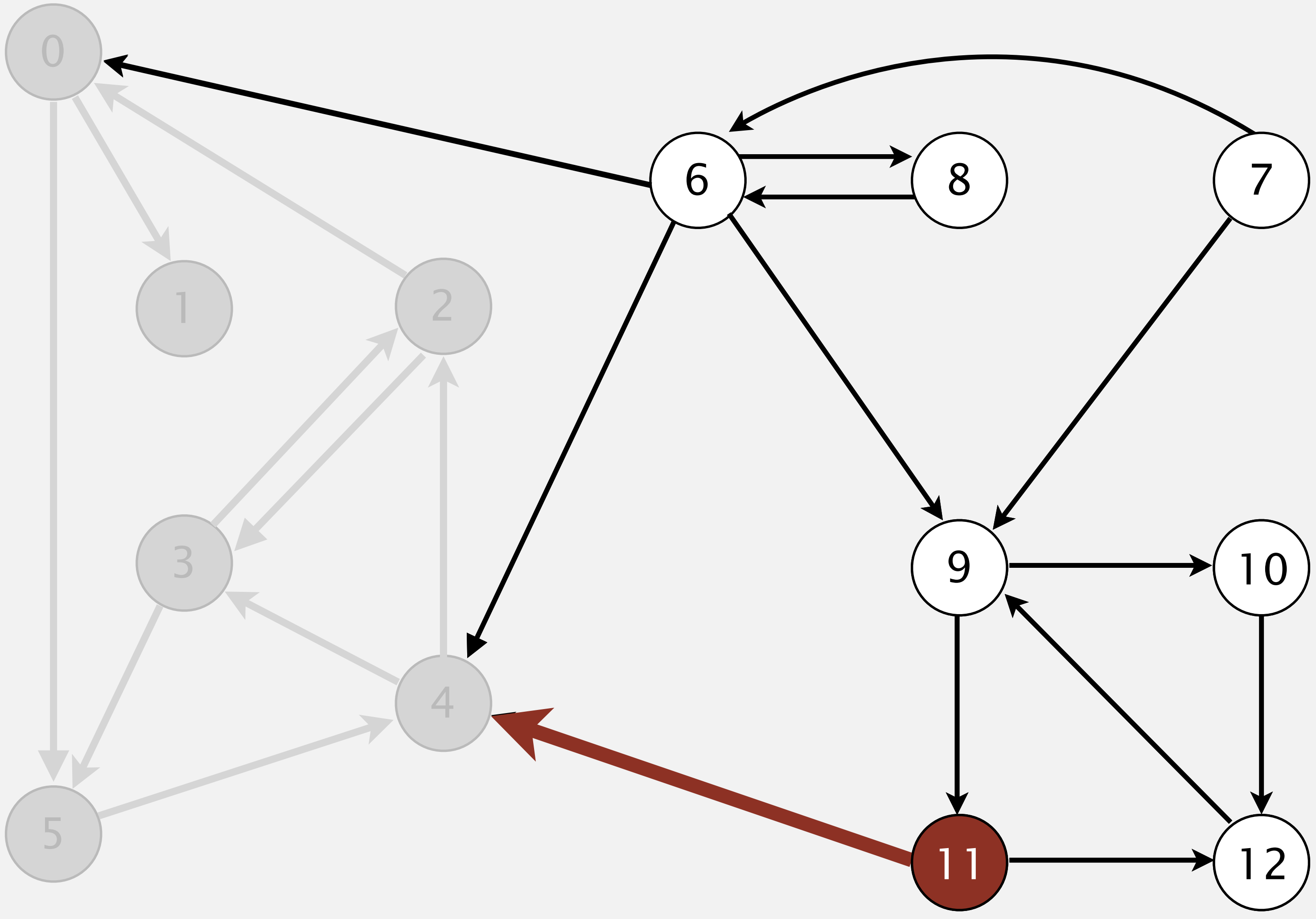
check 3

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



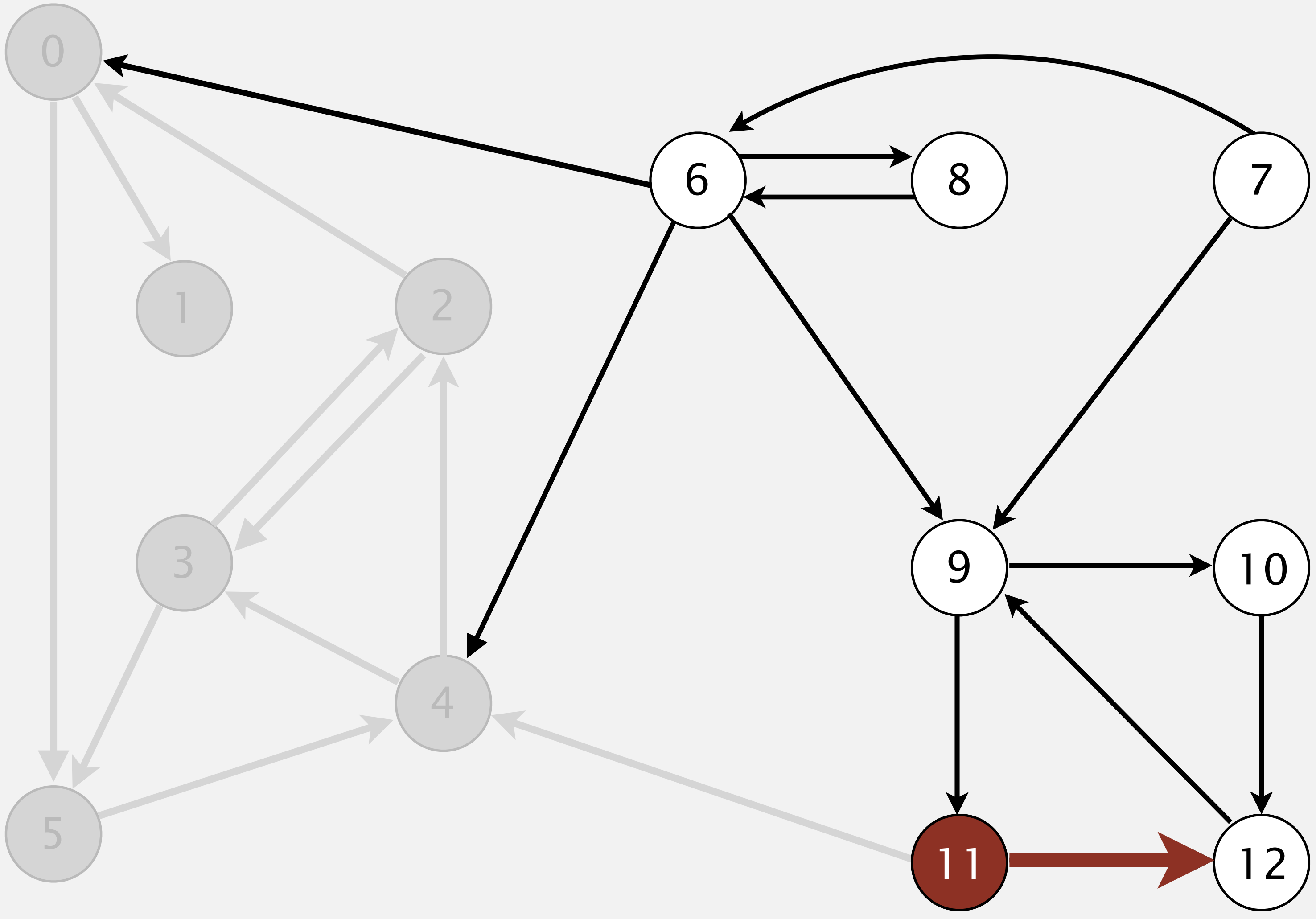
visit 11: check 4 and check 12

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	2
12	–

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



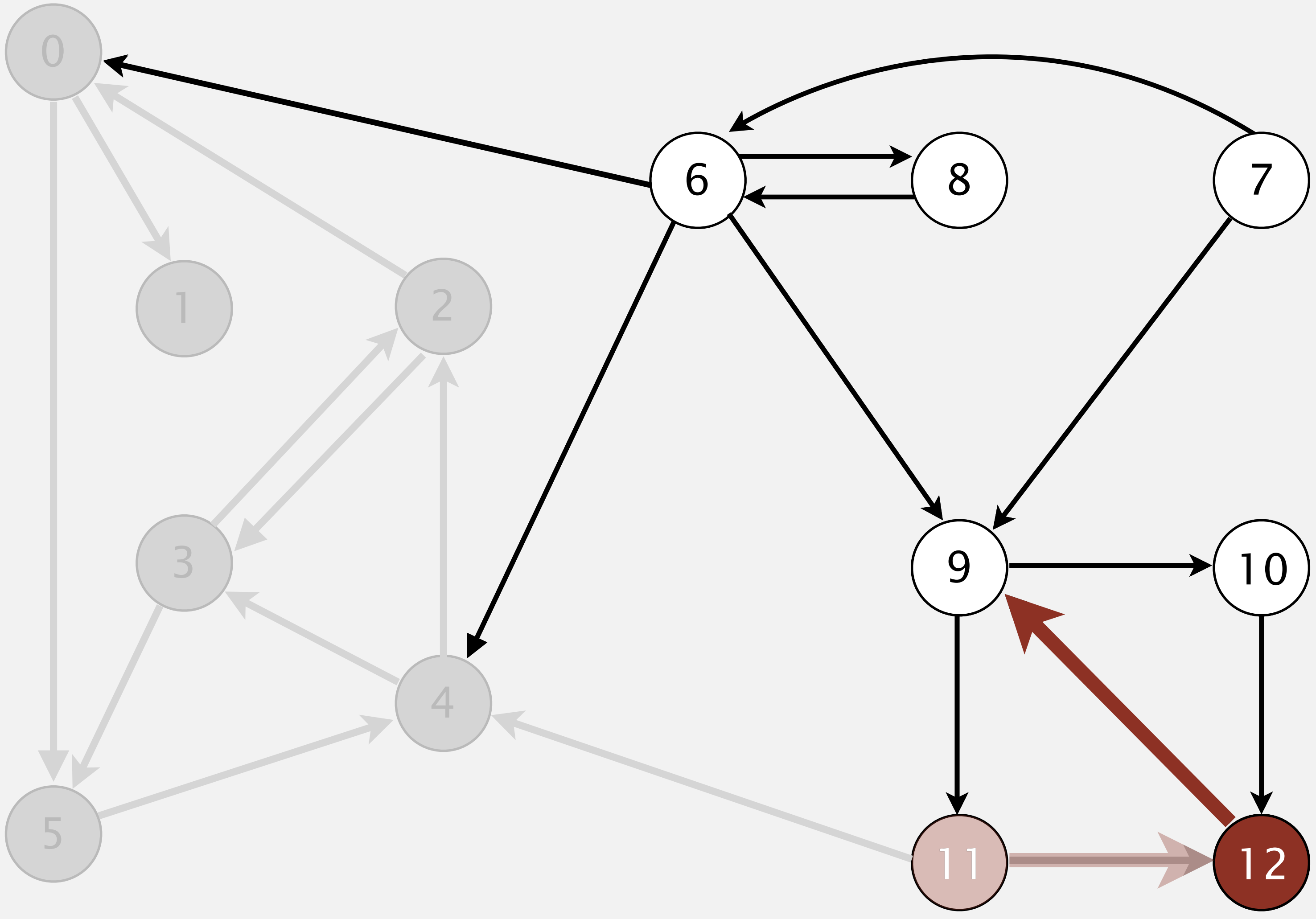
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	2
12	–

visit 11: check 4 and check 12

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



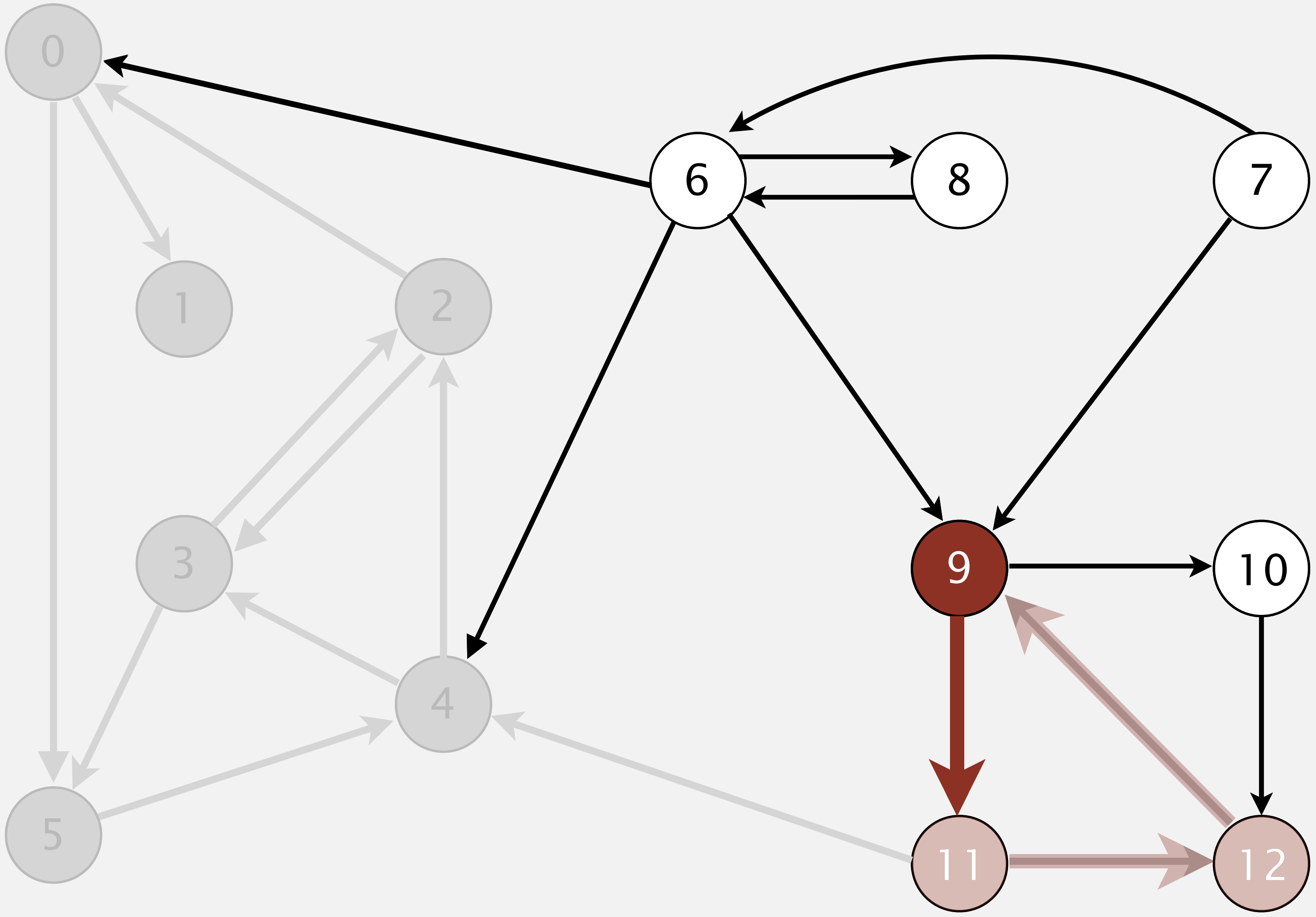
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	2
12	2

visit 12: check 9

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



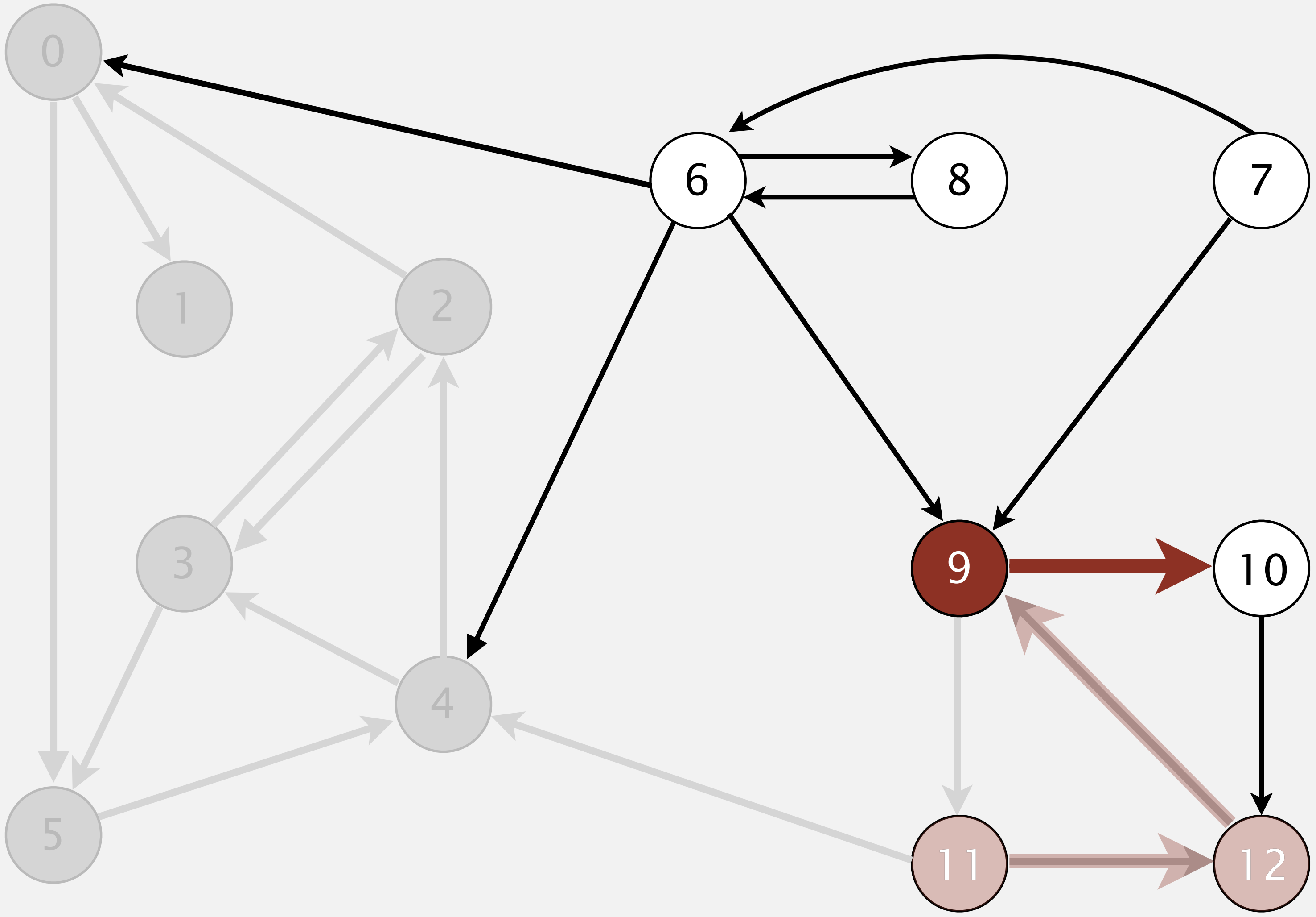
visit 9: check 11 and check 10

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	–
11	2
12	2

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



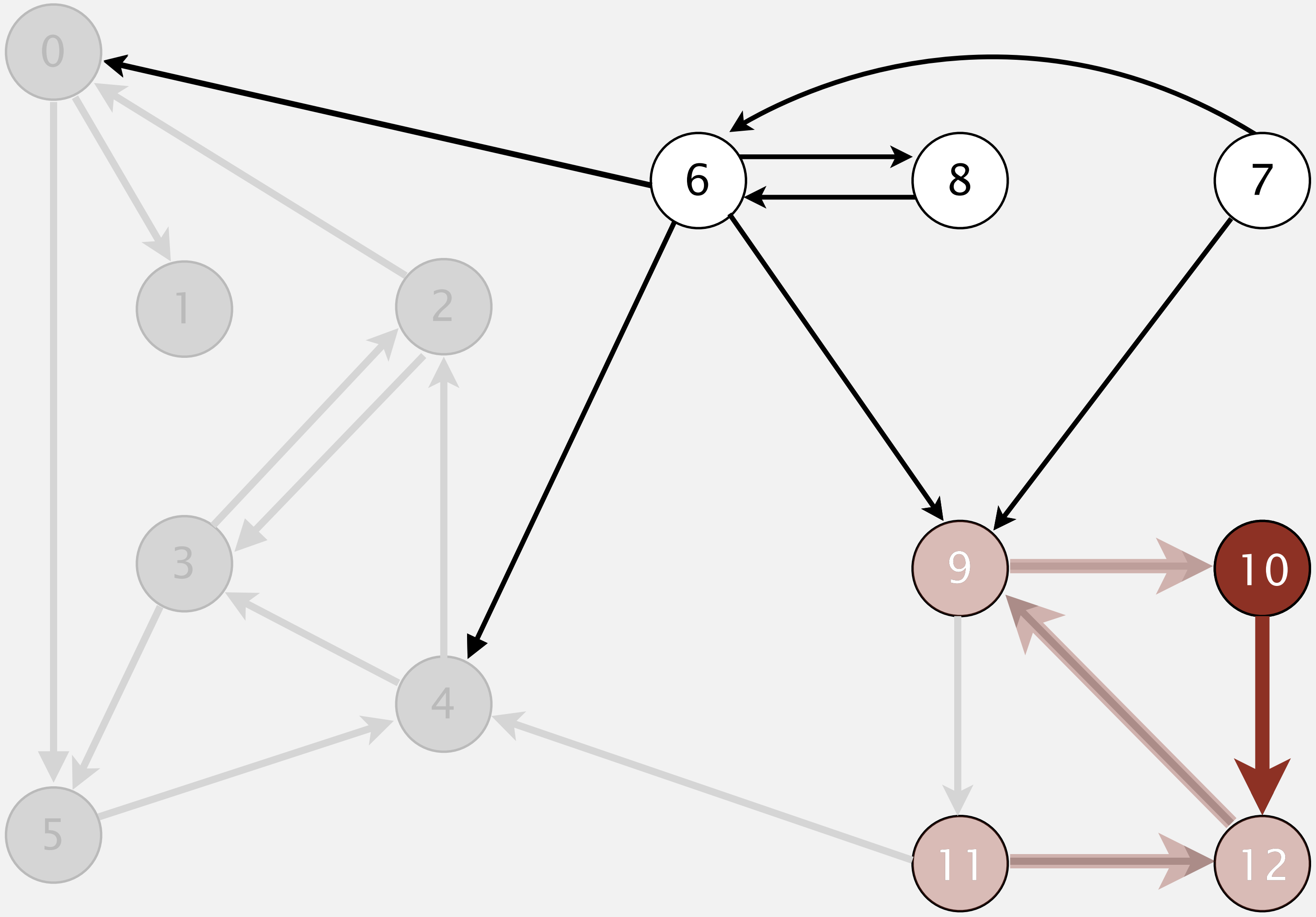
visit 9: check 11 and check 10

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	–
11	2
12	2

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



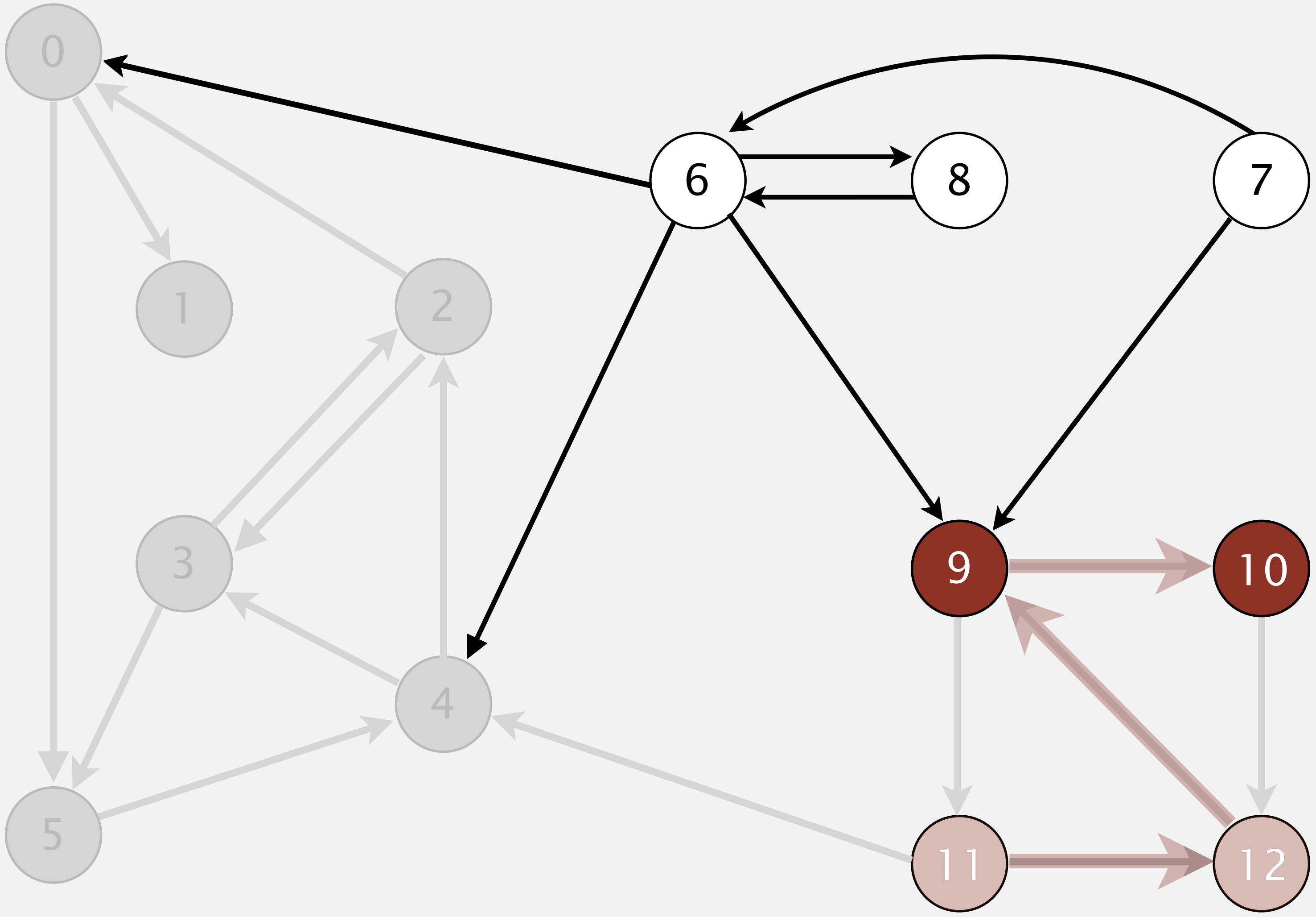
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	2
11	2
12	2

visit 10: check 12

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



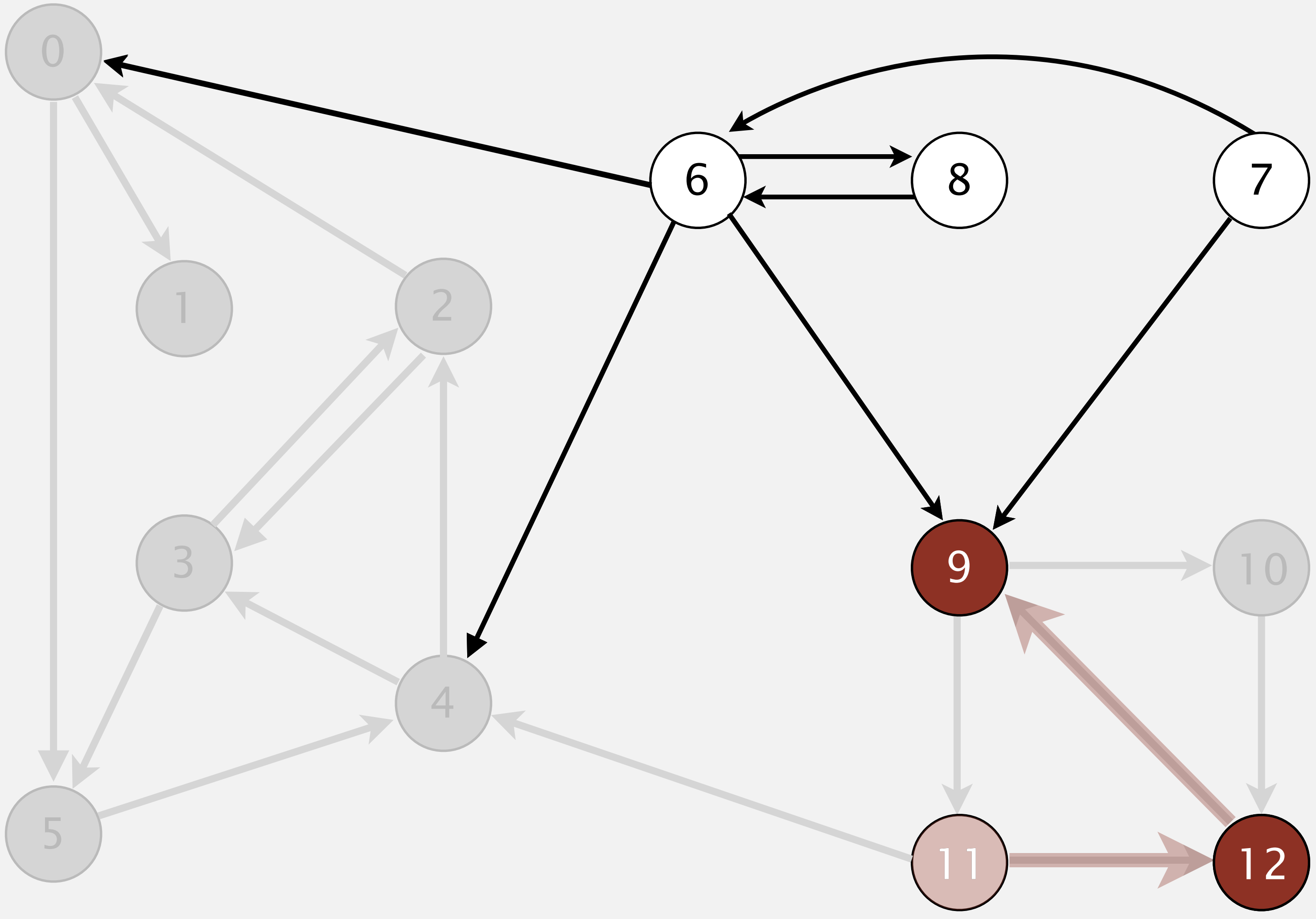
10 done

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	2
11	2
12	2

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



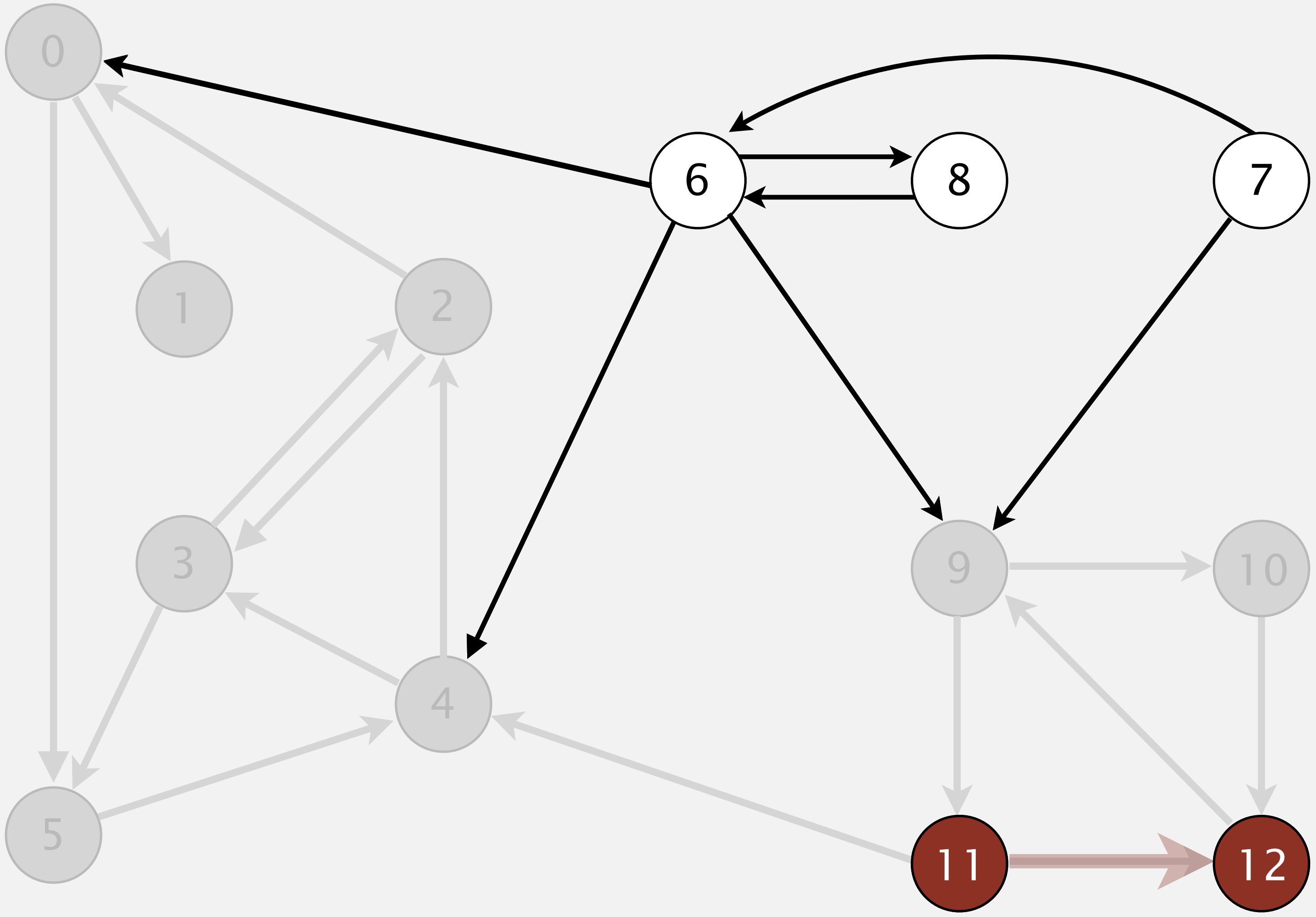
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	2
11	2
12	2

9 done

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



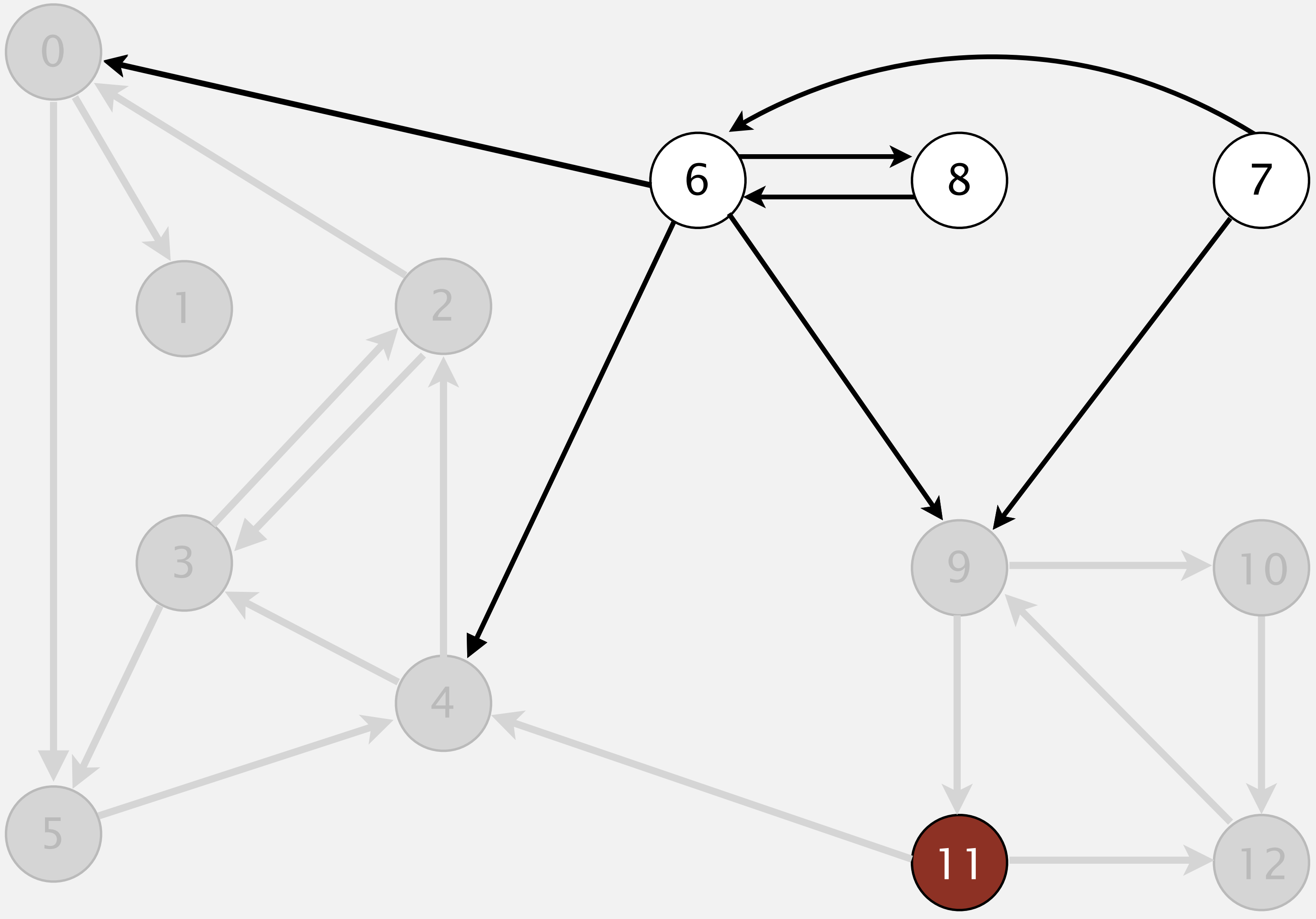
12 done

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	2
11	2
12	2

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



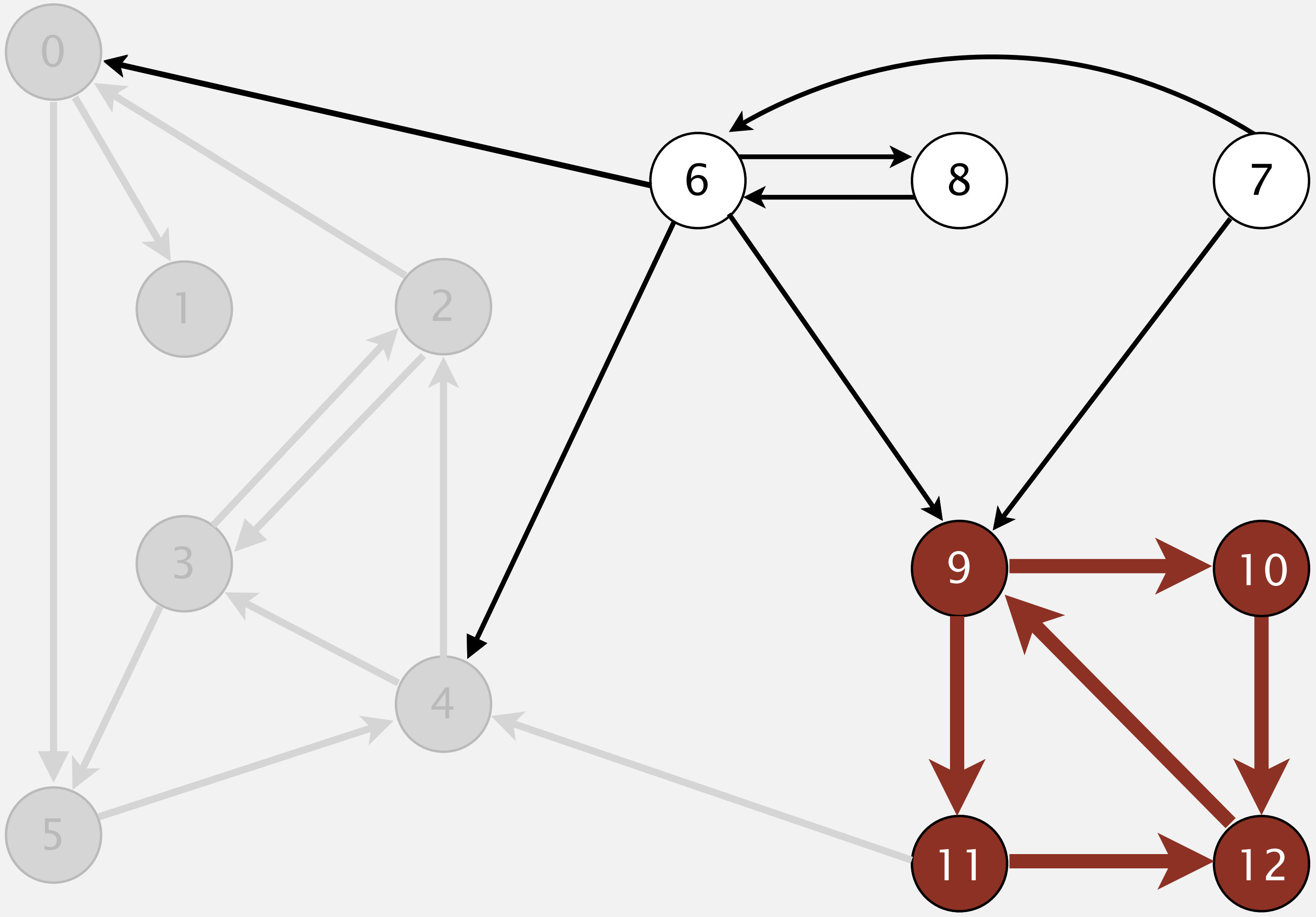
11 done

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	2
11	2
12	2

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



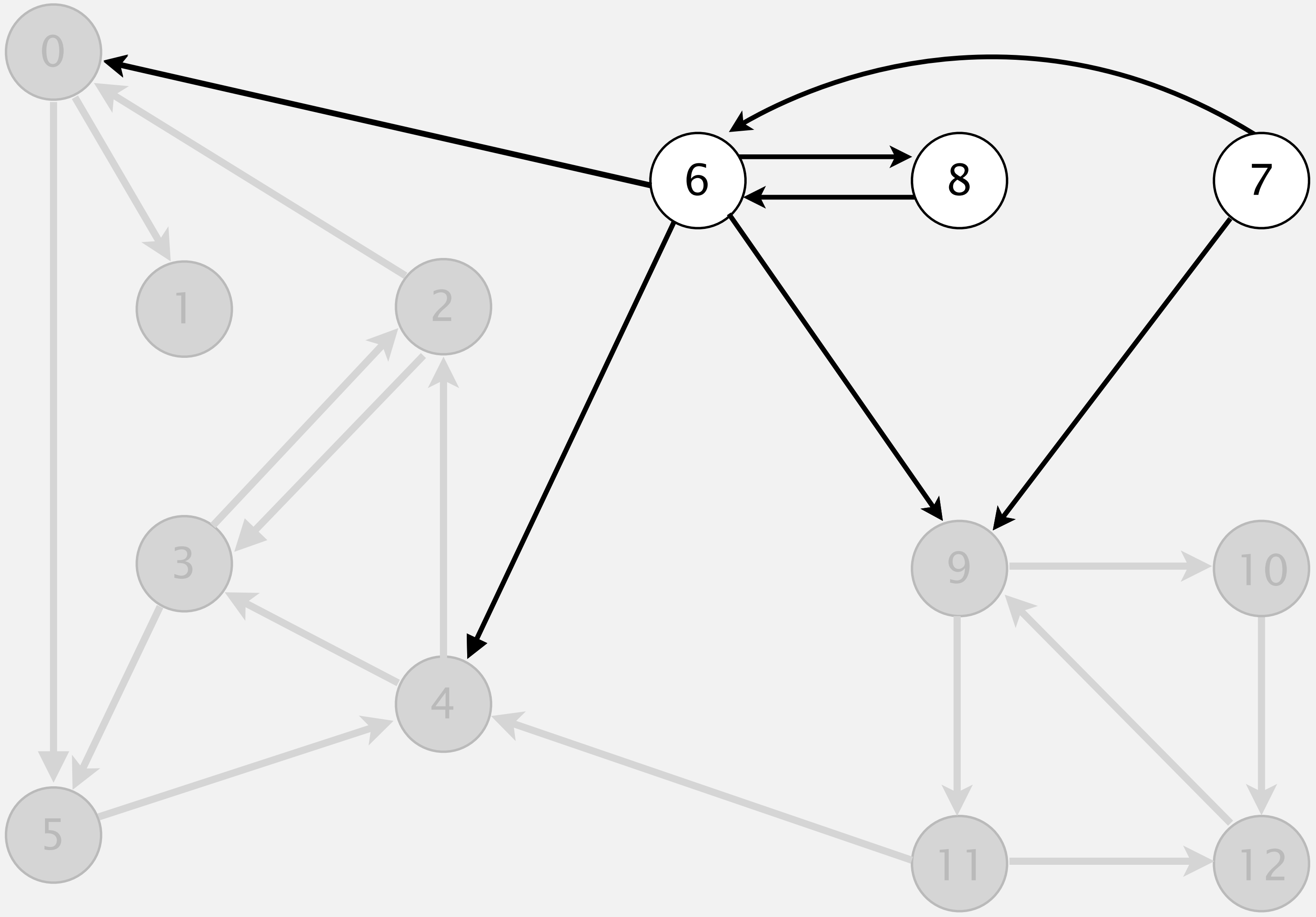
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	2
11	2
12	2

strong component: 9 10 11 12

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 **9** **12 10 6 7 8**



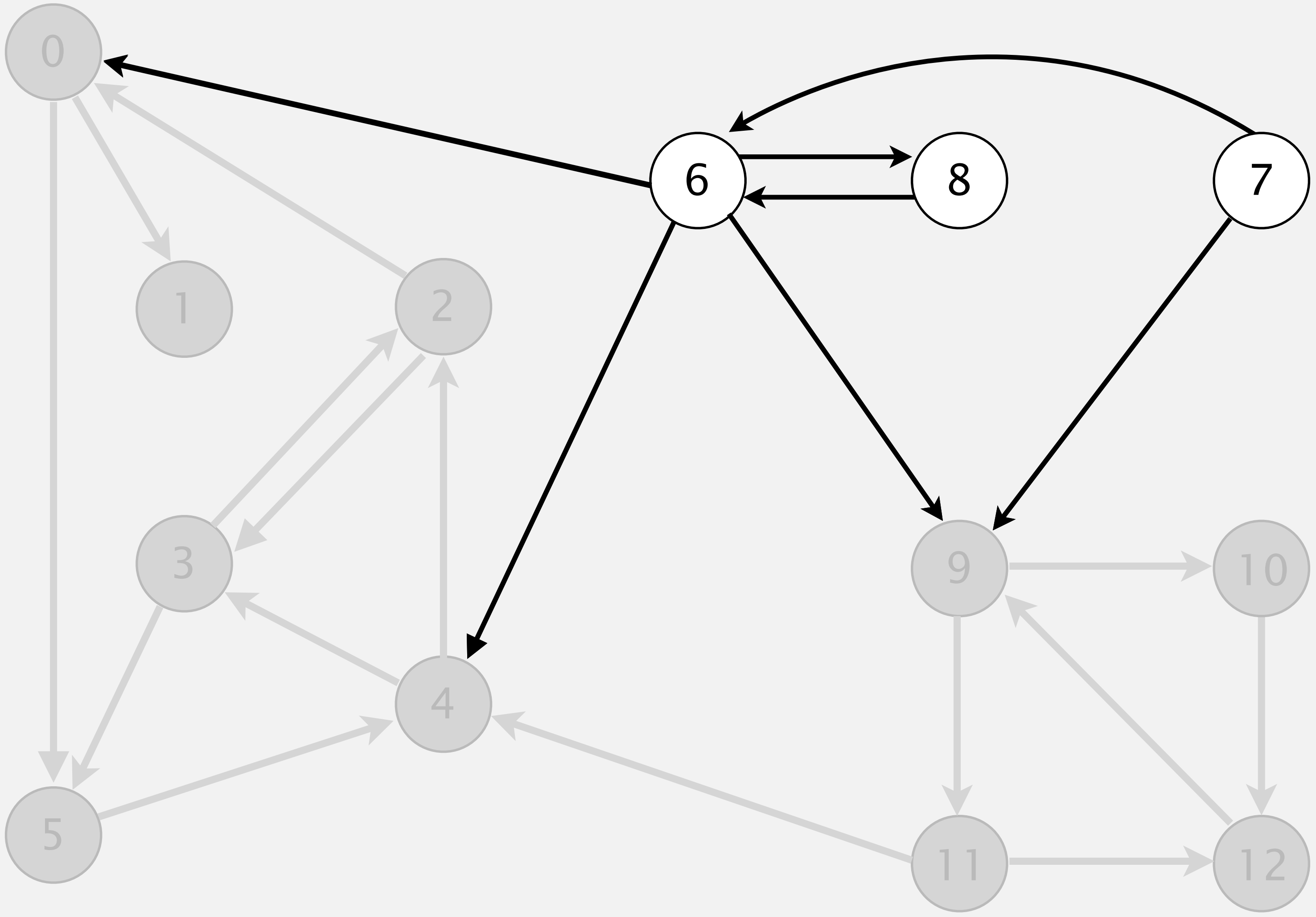
check 9

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	2
11	2
12	2

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



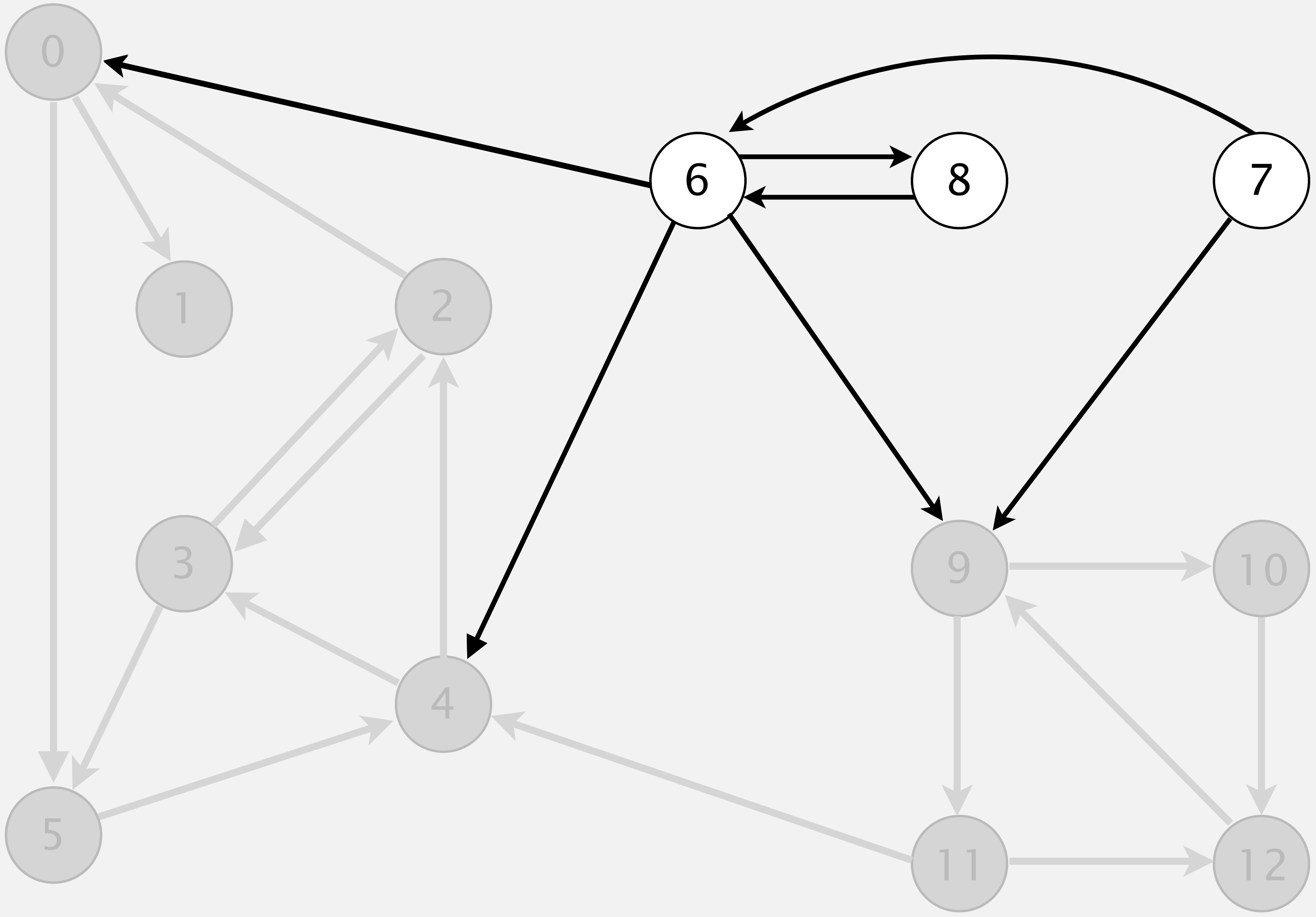
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	2
11	2
12	2

check 12

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 **6 7 8**



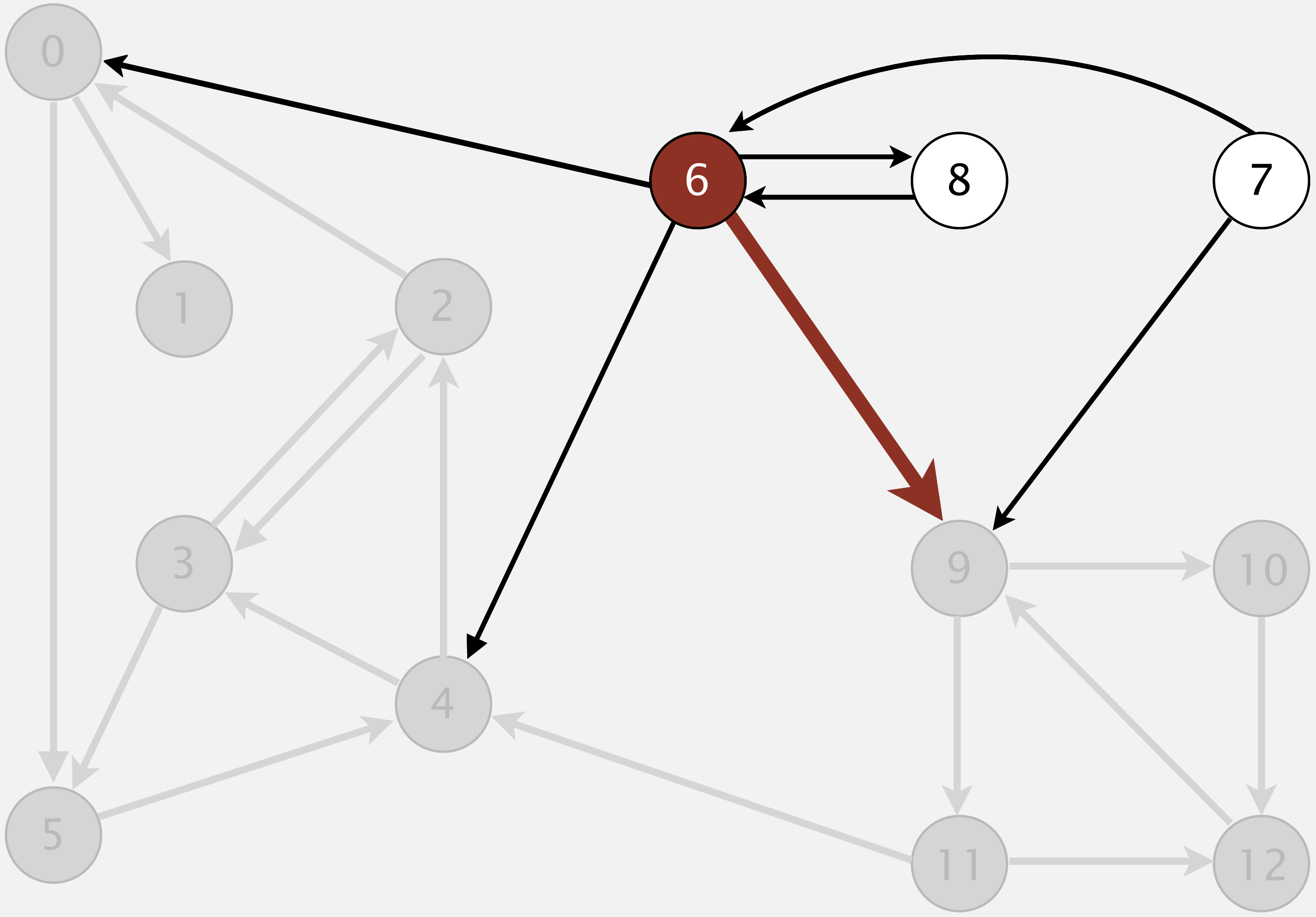
check 10

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	2
11	2
12	2

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



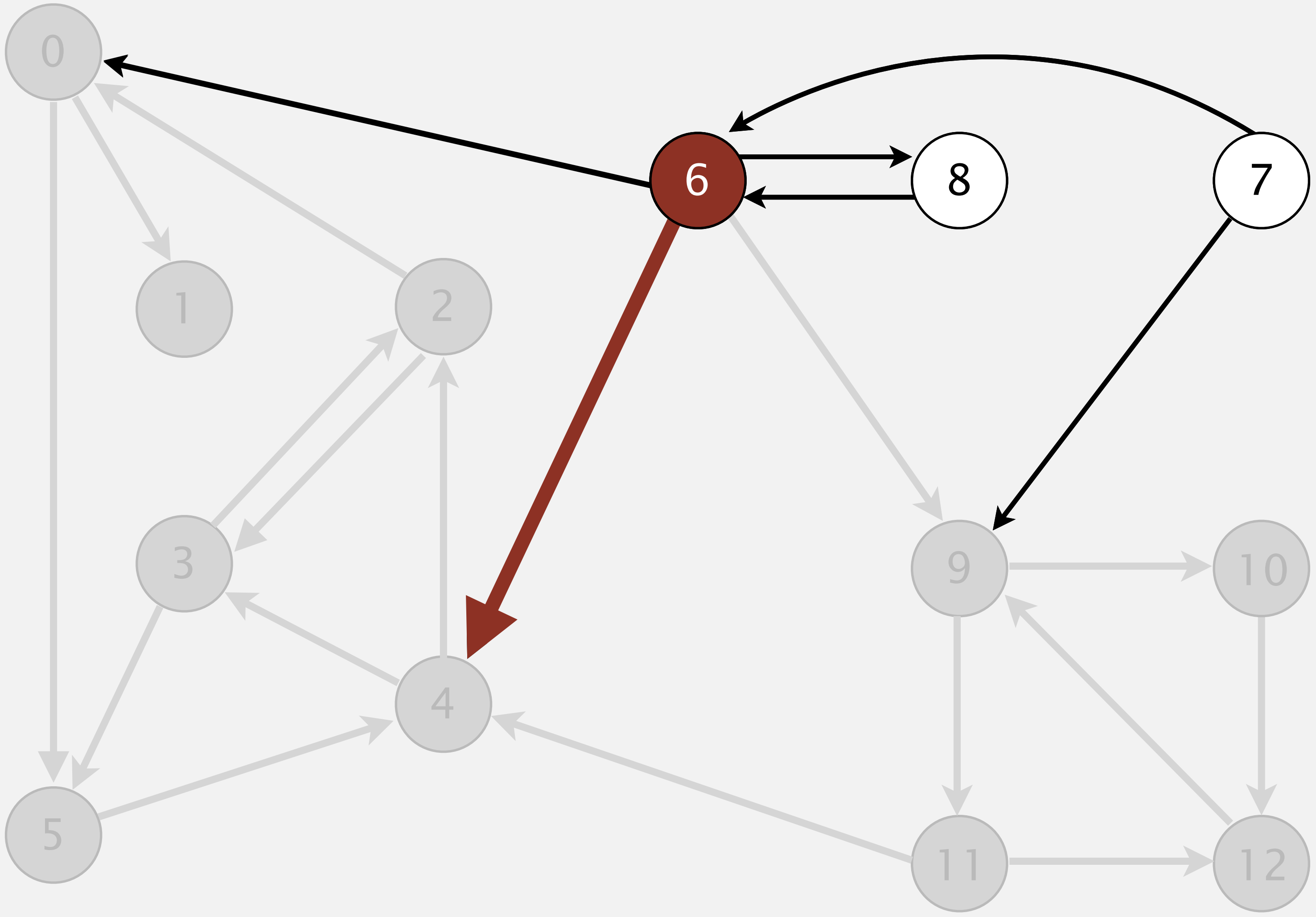
visit 6: check 9, check 4, check 8, and check 0

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	–
8	–
9	2
10	2
11	2
12	2

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



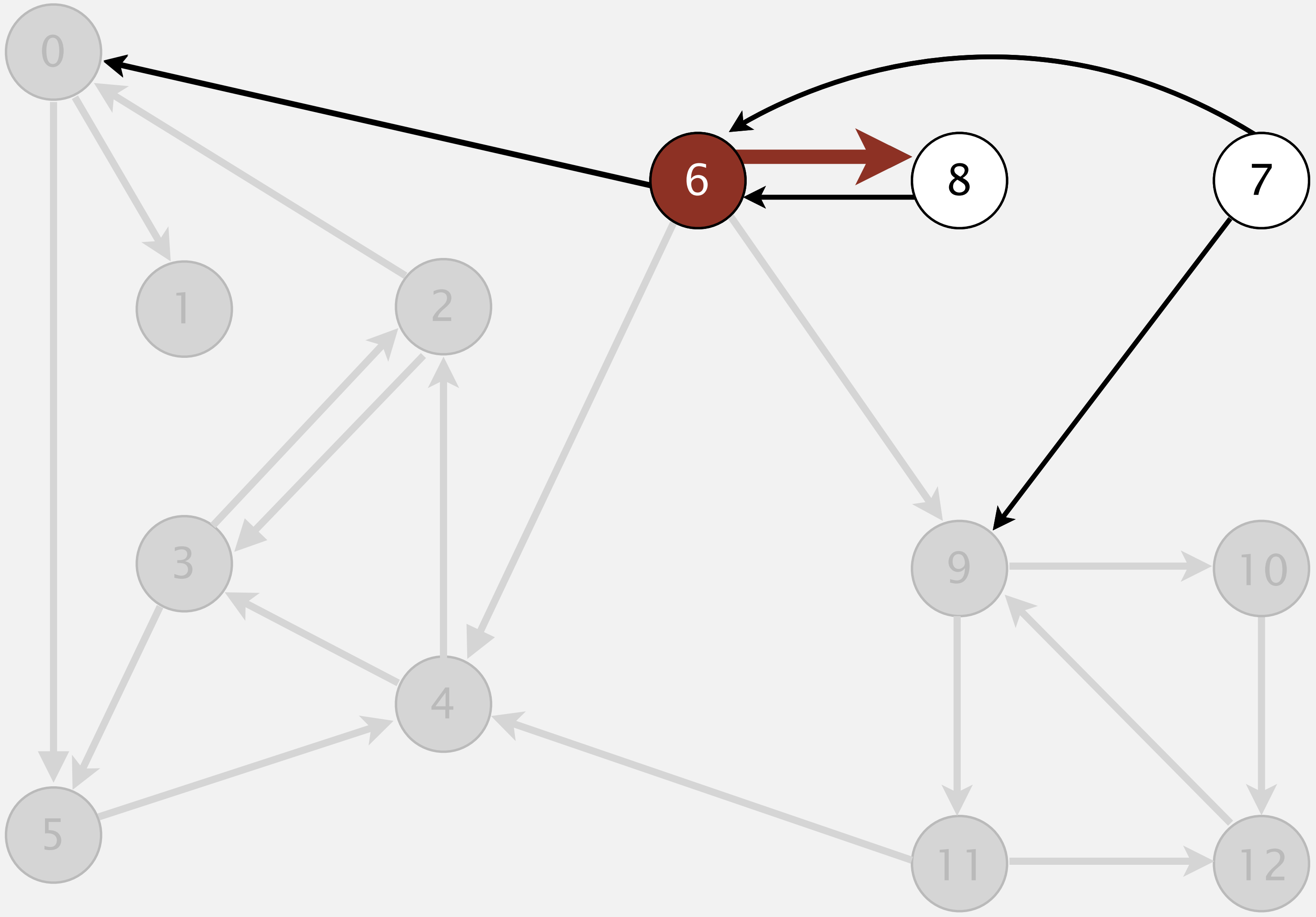
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	–
8	–
9	2
10	2
11	2
12	2

visit 6: check 9, check 4, check 8, and check 0

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



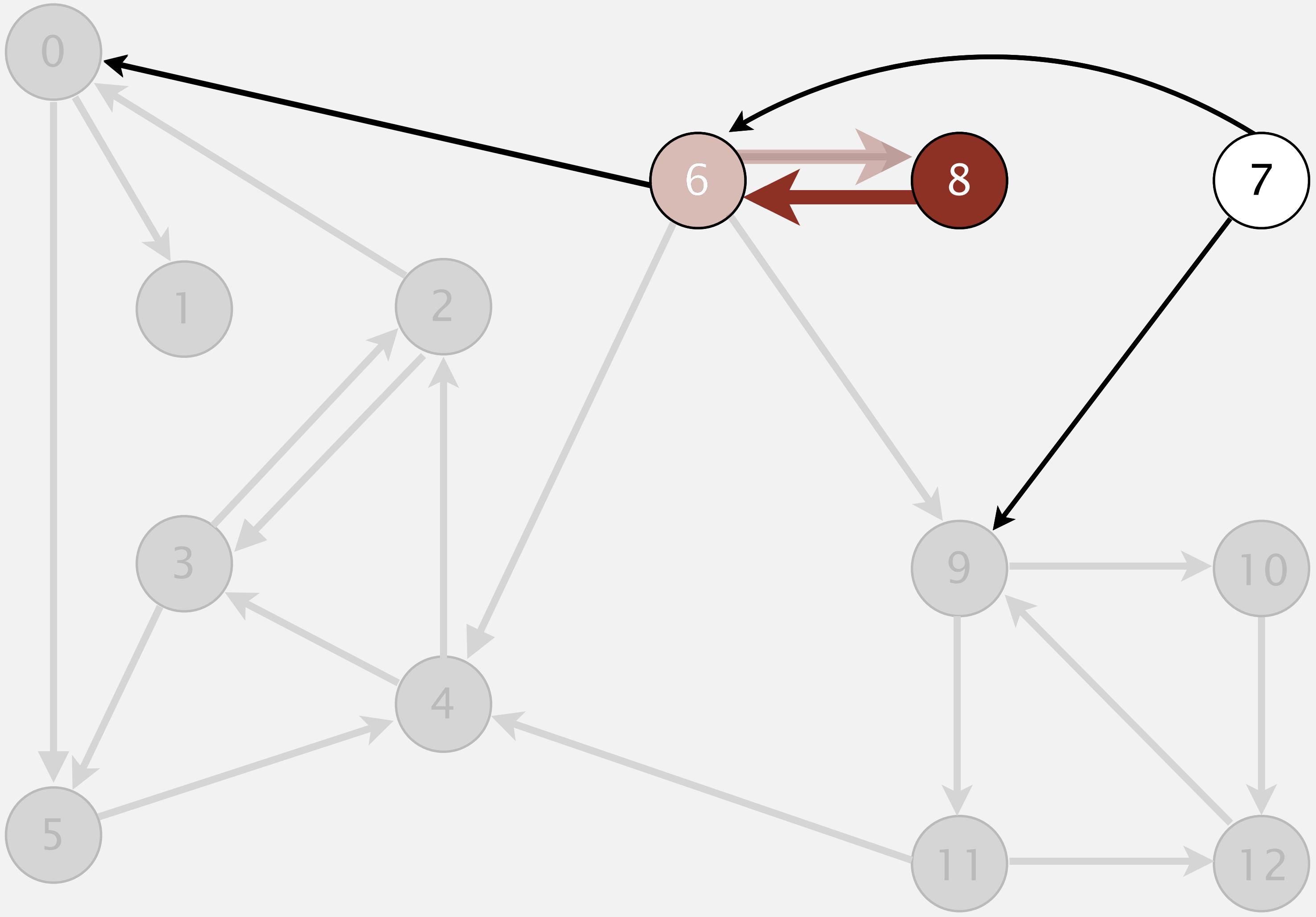
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	–
8	–
9	2
10	2
11	2
12	2

visit 6: check 9, check 4, check 8, and check 0

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



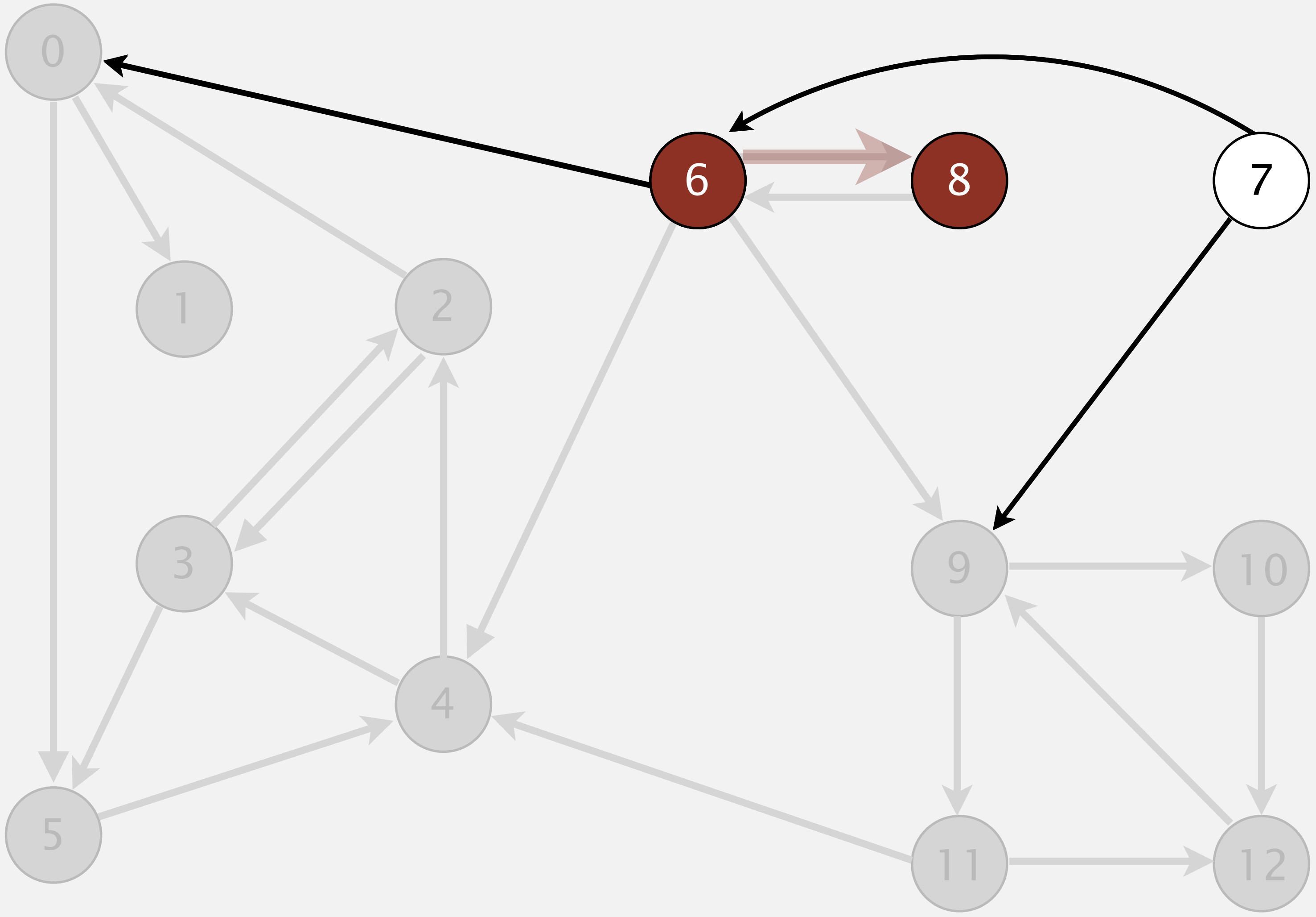
visit 8: check 6

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	–
8	3
9	2
10	2
11	2
12	2

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



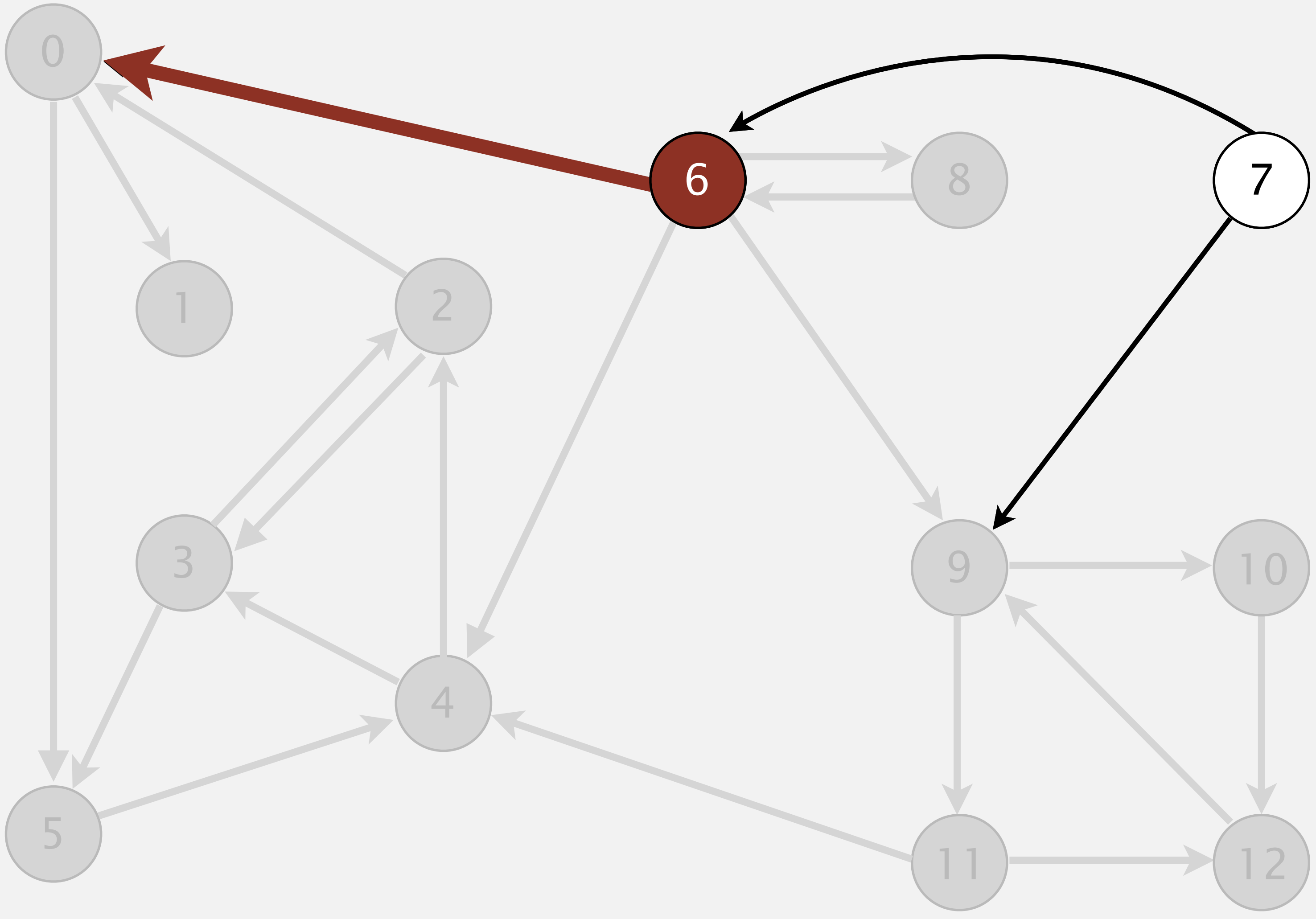
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	–
8	3
9	2
10	2
11	2
12	2

8 done

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



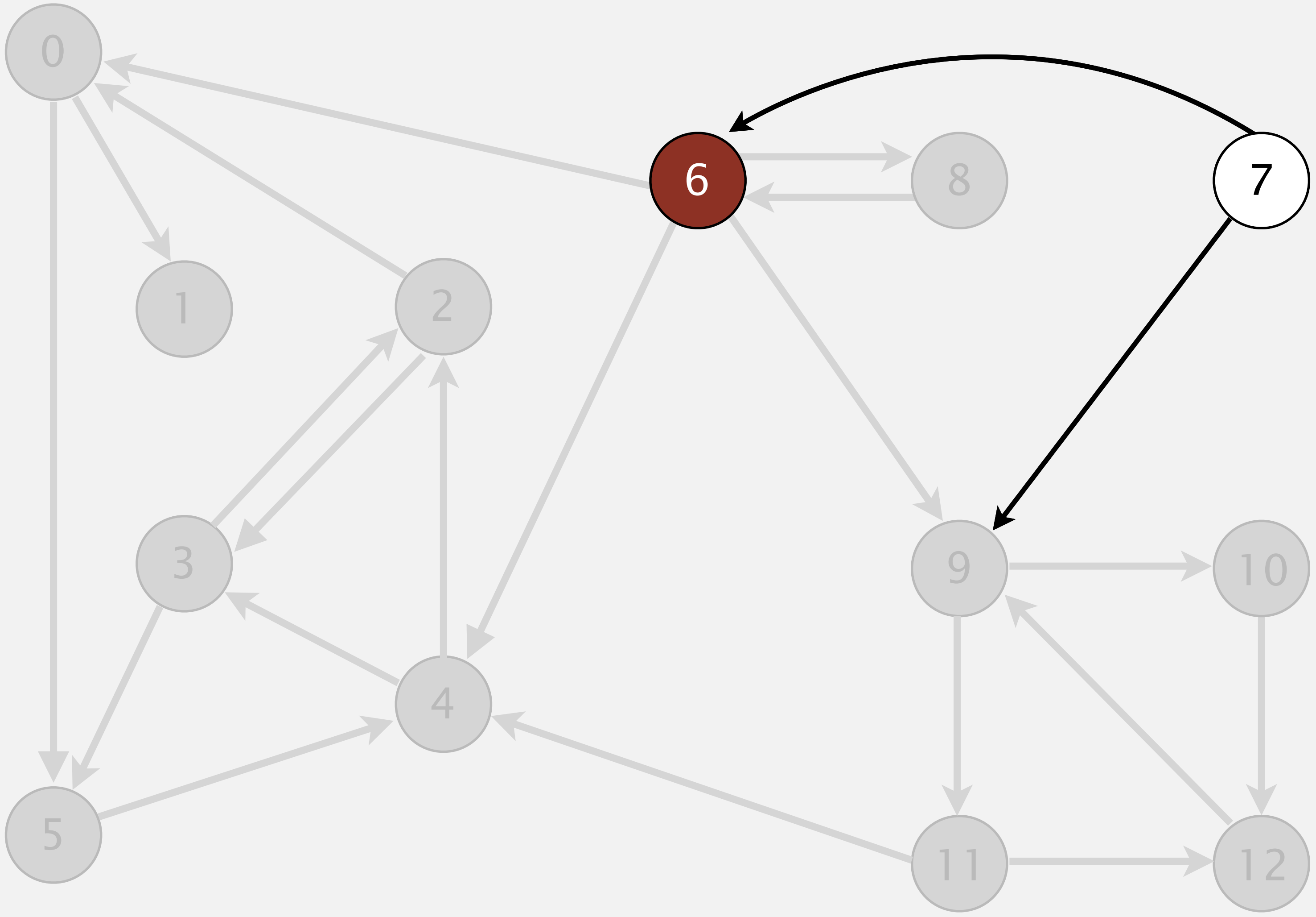
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	–
8	3
9	2
10	2
11	2
12	2

visit 6: check 9, check 4, check 8, and check 0

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



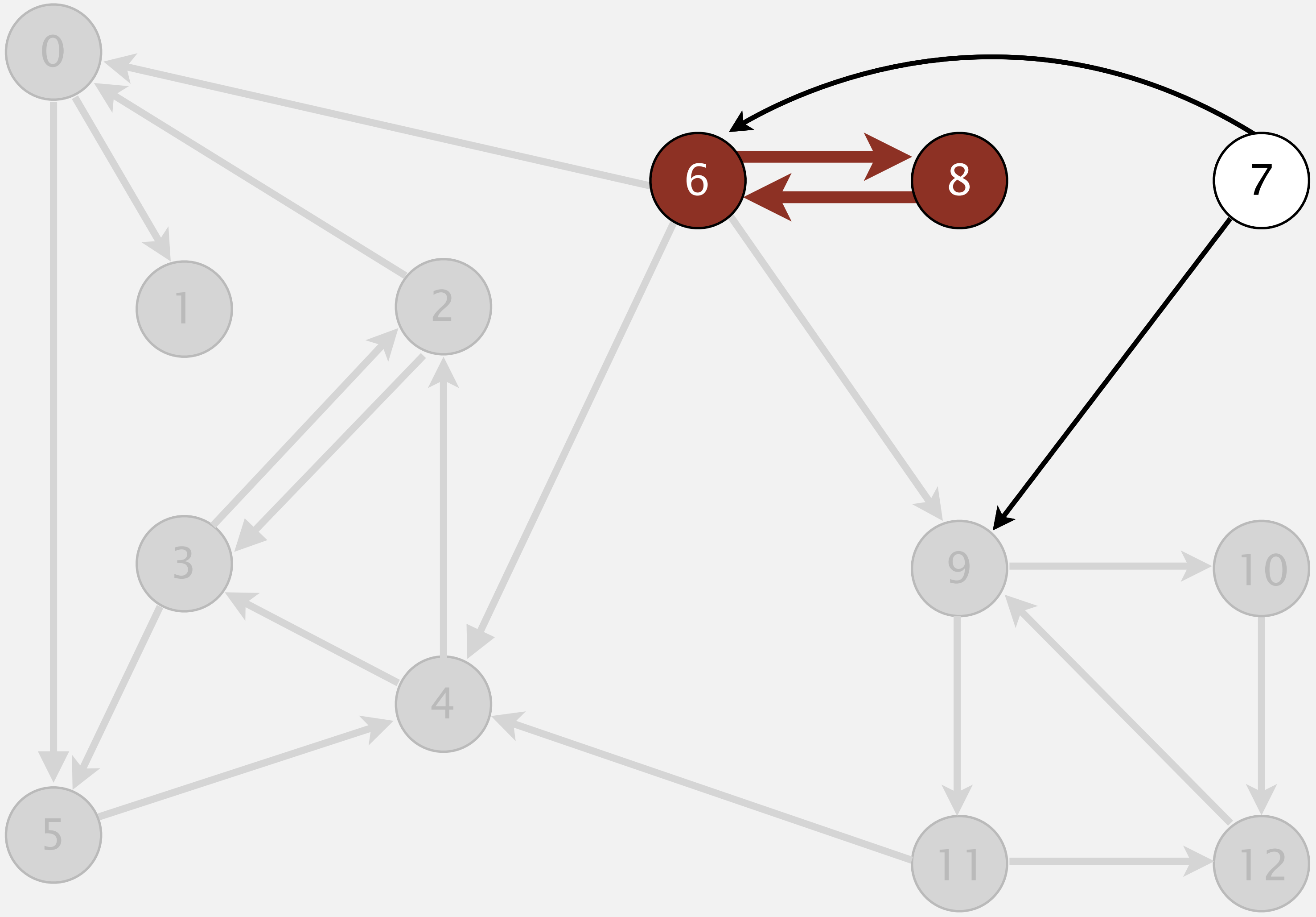
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	–
8	3
9	2
10	2
11	2
12	2

6 done

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



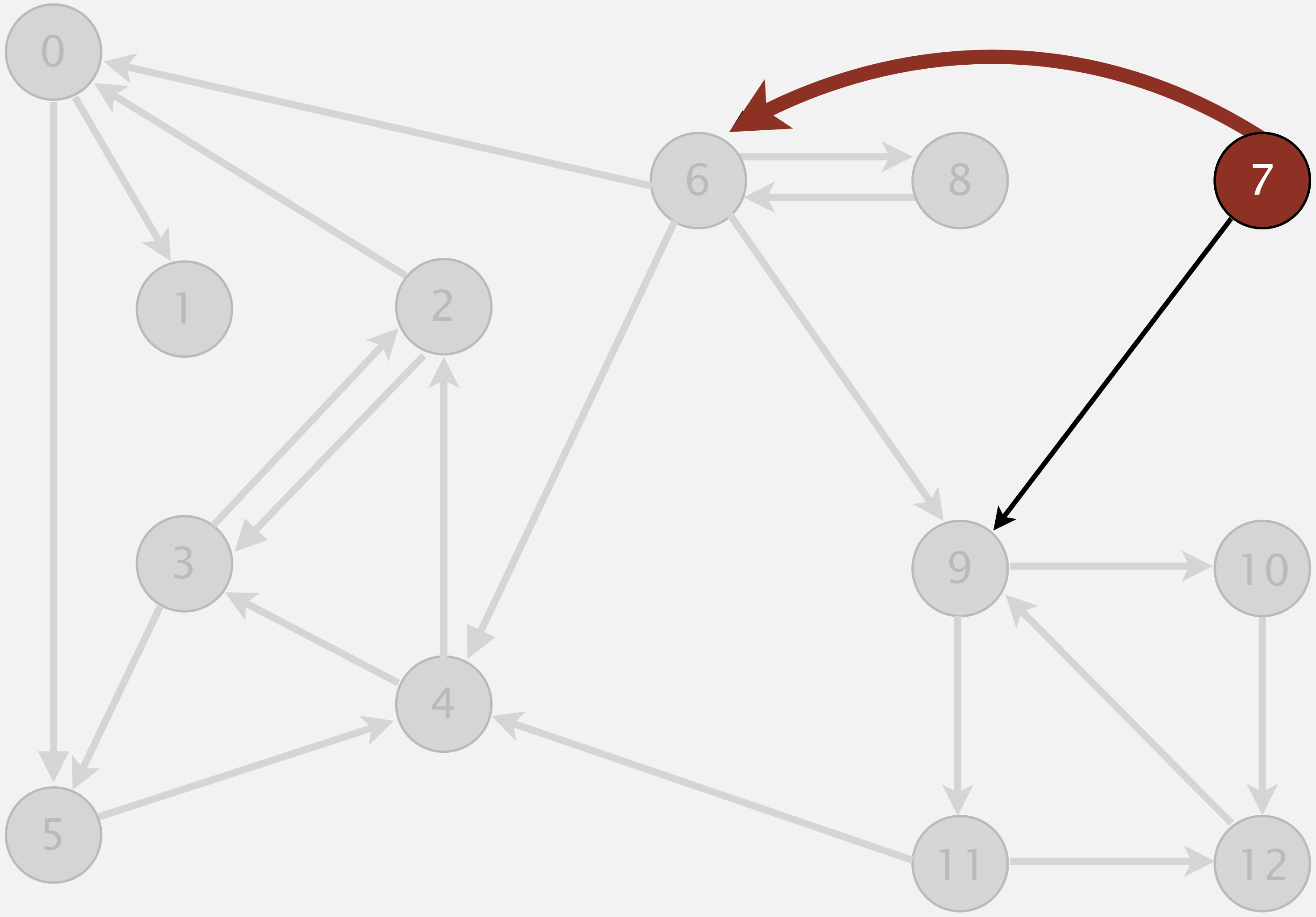
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	–
8	3
9	2
10	2
11	2
12	2

strong component: 6 8

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



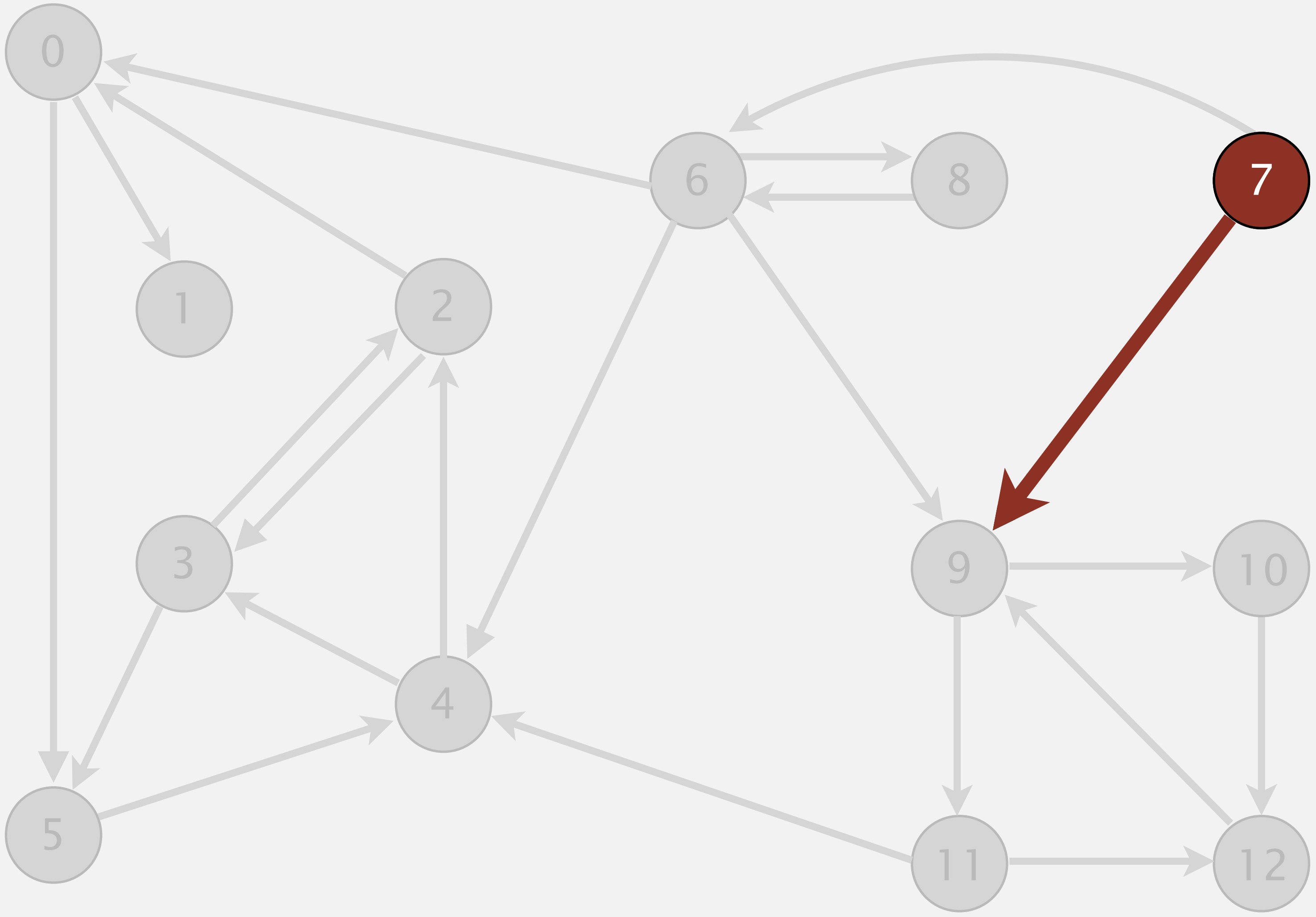
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	4
8	3
9	2
10	2
11	2
12	2

visit 7: check 6 and check 9

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



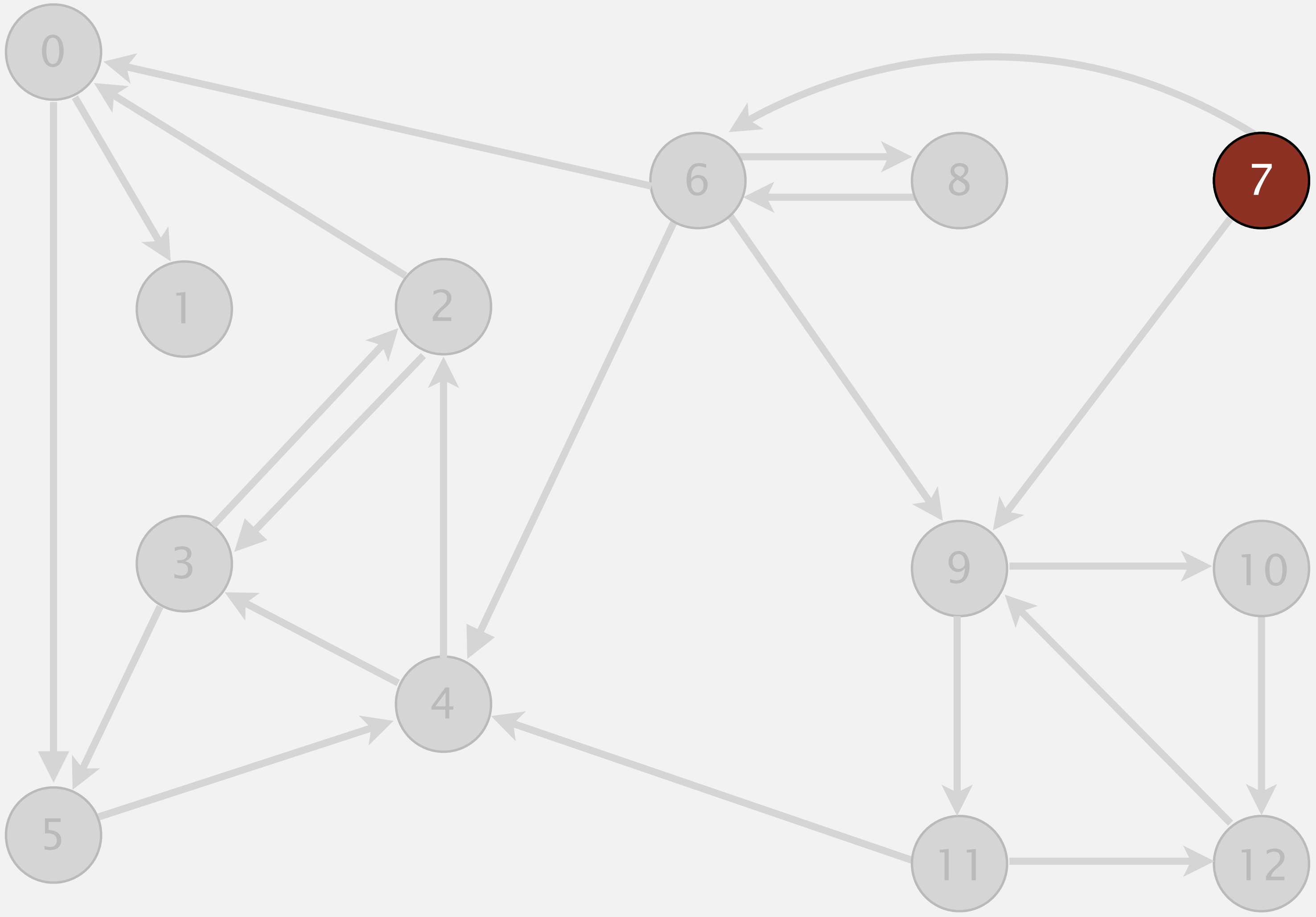
visit 7: check 6 and check 9

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	4
8	3
9	2
10	2
11	2
12	2

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



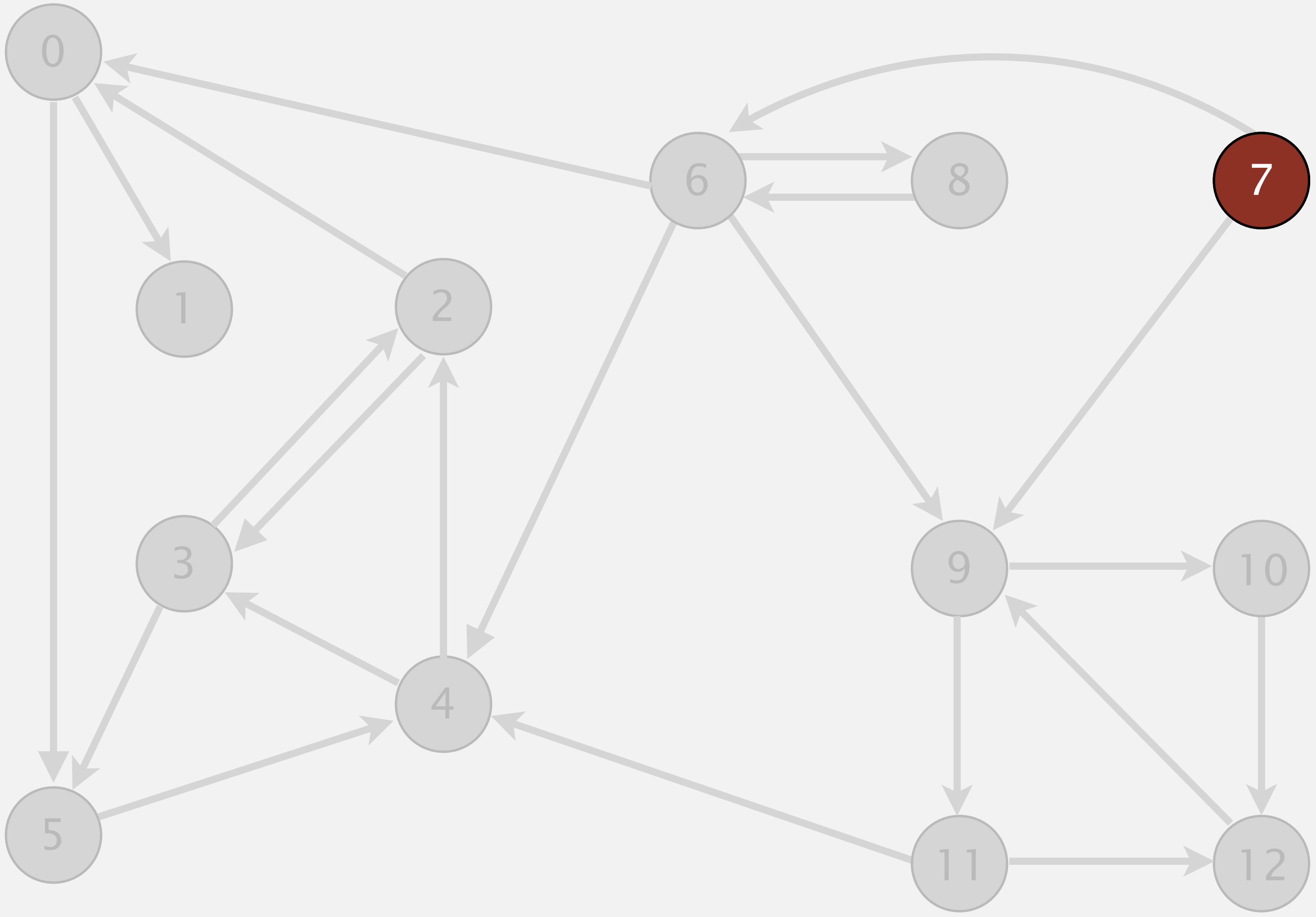
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	4
8	3
9	2
10	2
11	2
12	2

7 done

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 **8**



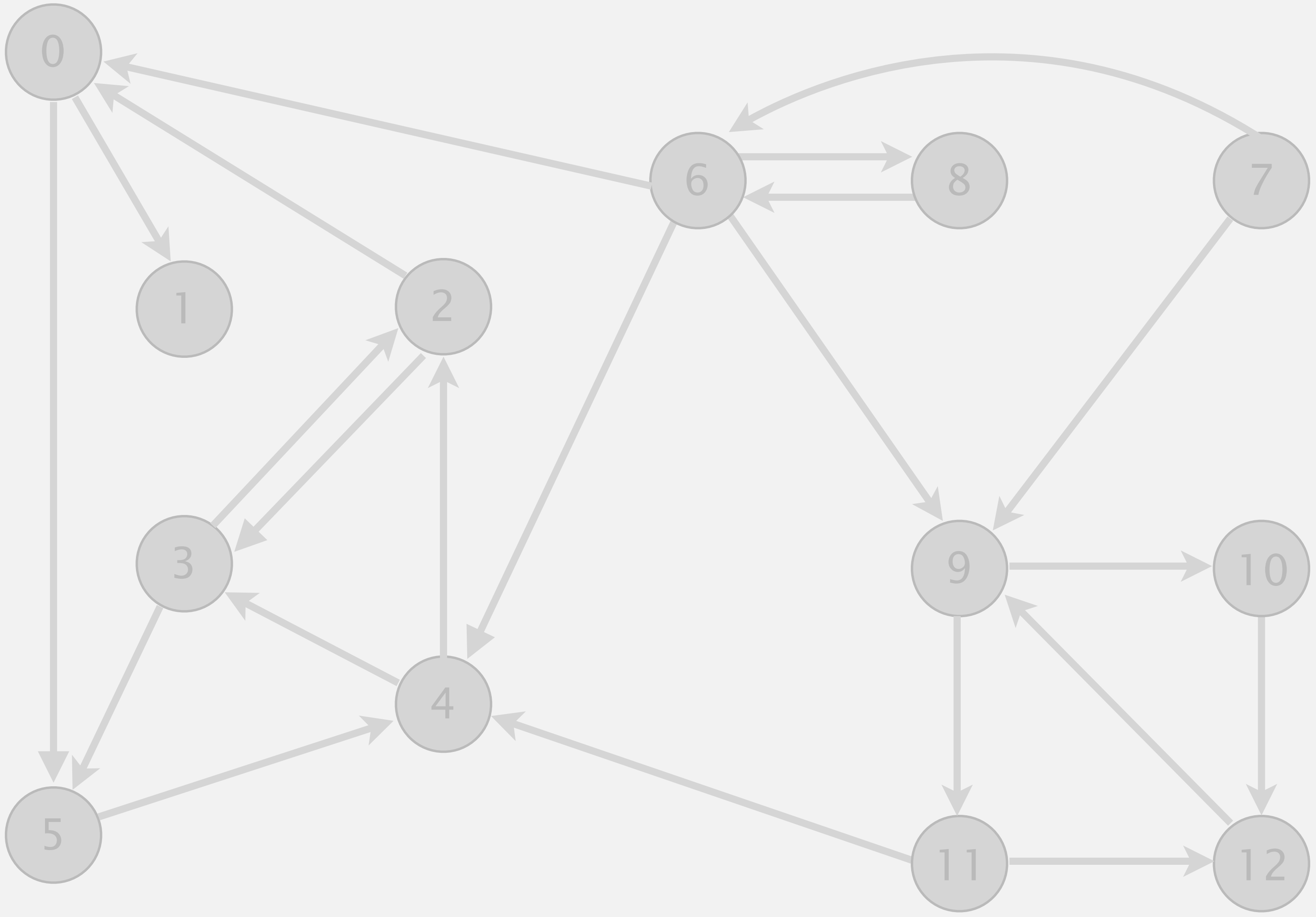
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	4
8	3
9	2
10	2
11	2
12	2

strong component: 7

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



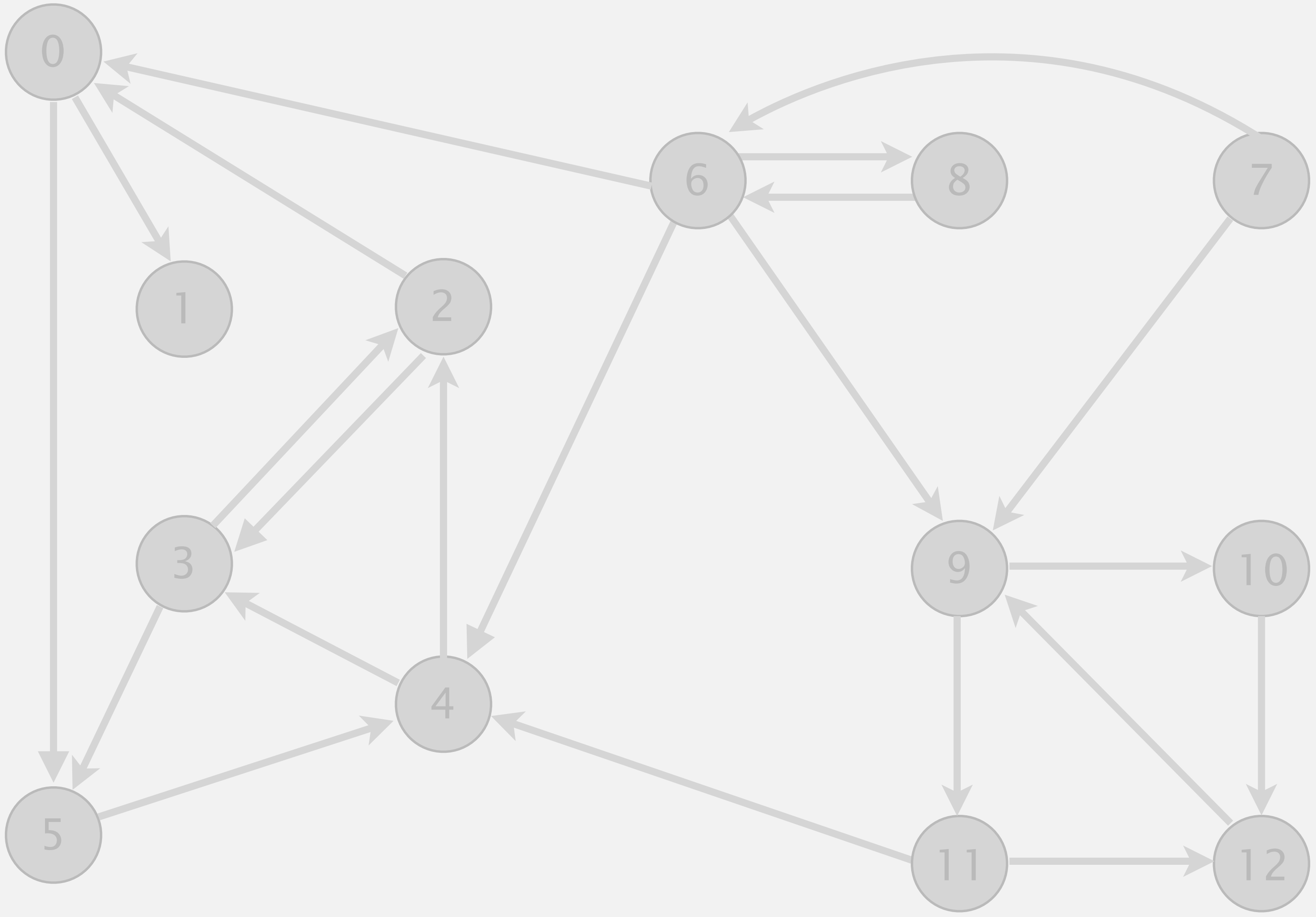
check 8

v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	4
8	3
9	2
10	2
11	2
12	2

Kosaraju-Sharir algorithm demo

Phase 2. Run DFS in G, visiting unmarked vertices in reverse postorder of G^R.

1 0 2 4 5 3 11 9 12 10 6 7 8



v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	4
8	3
9	2
10	2
11	2
12	2

done