



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

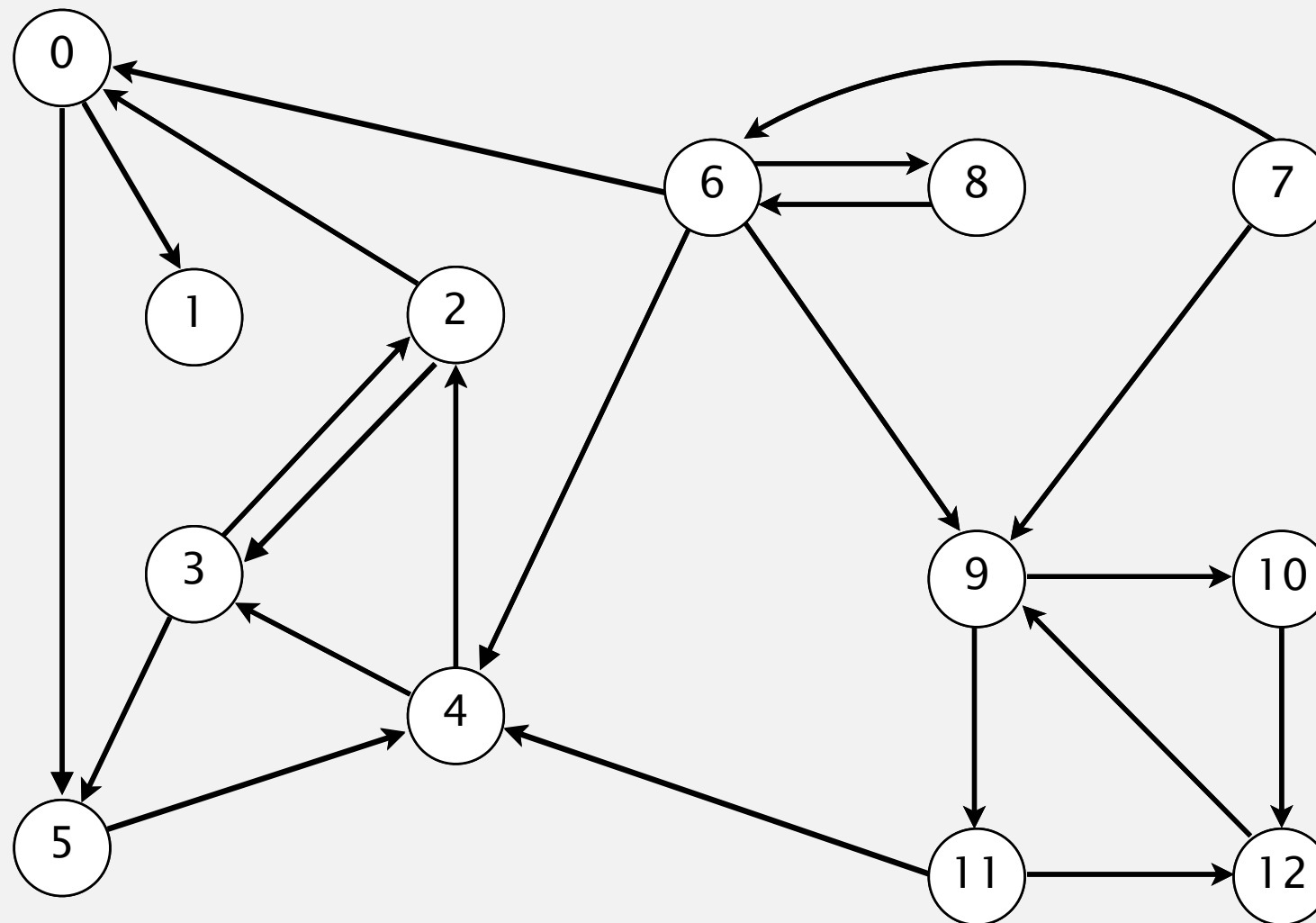
## 4.2 KOSARAJU-SHARIR DEMO

---

# 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



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

## 4.2 KOSARAJU-SHARIR DEMO

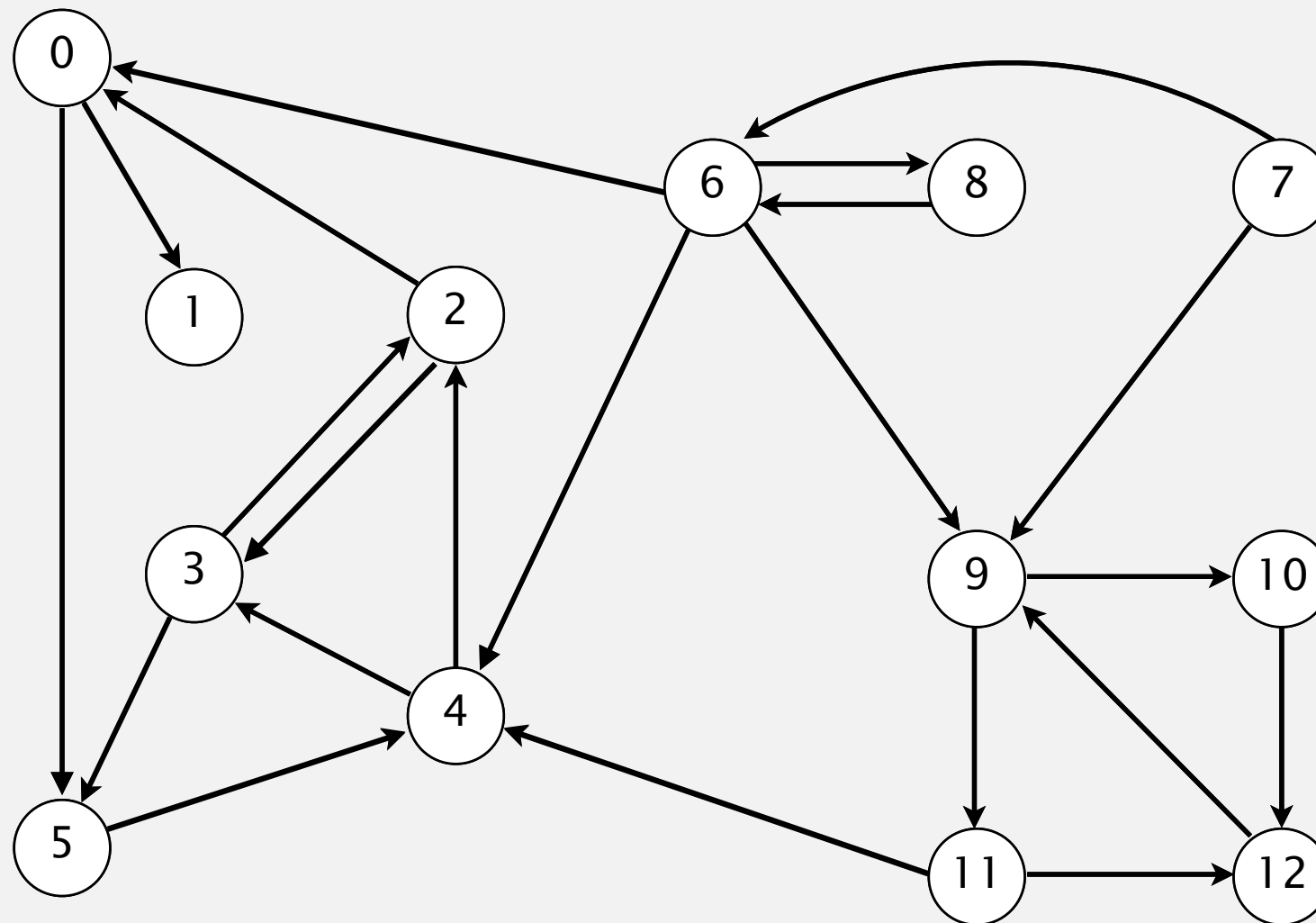
---

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

# Kosaraju-Sharir algorithm demo

---

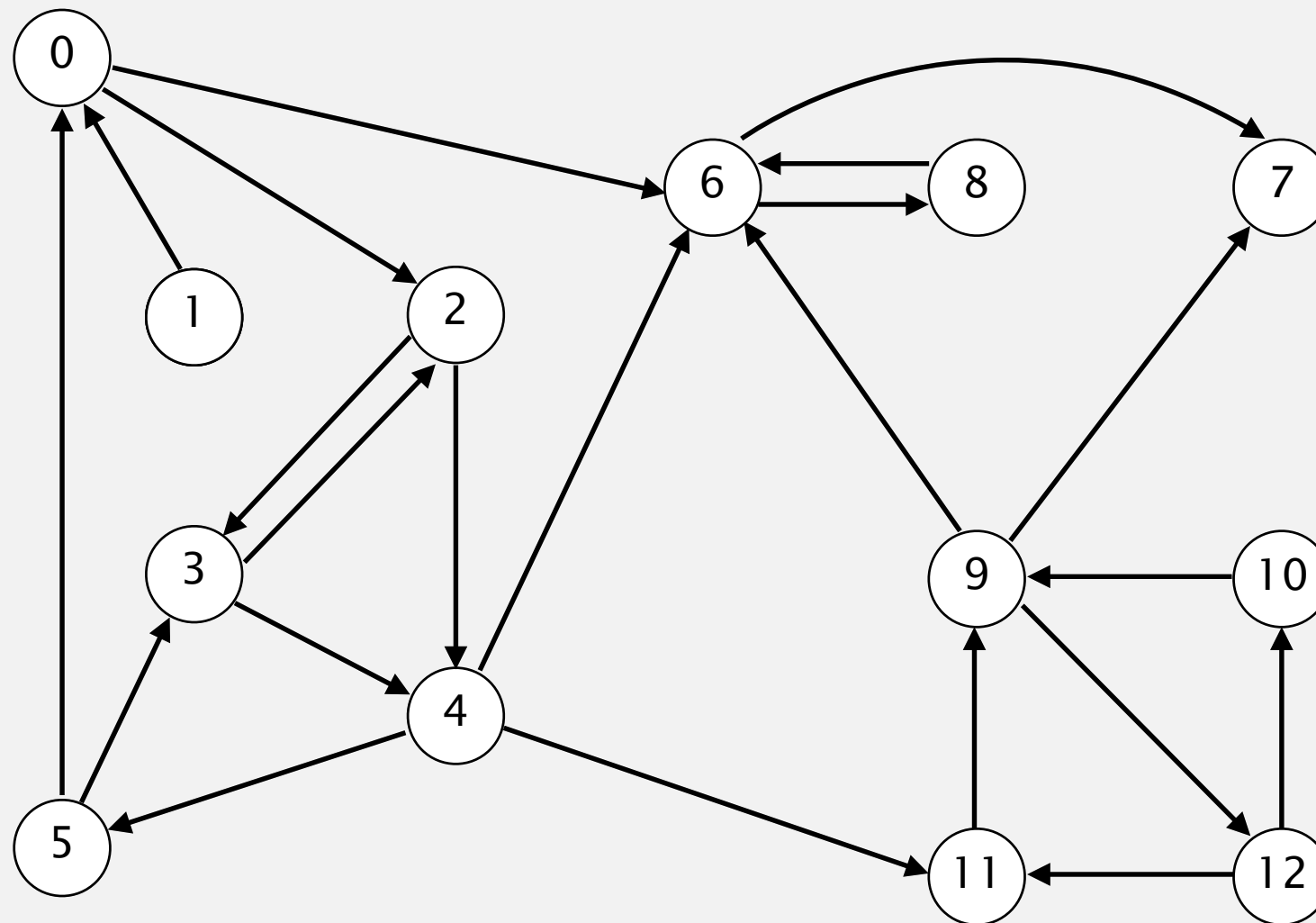
Phase 1. Compute reverse postorder in  $G^R$ .



**digraph G**

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

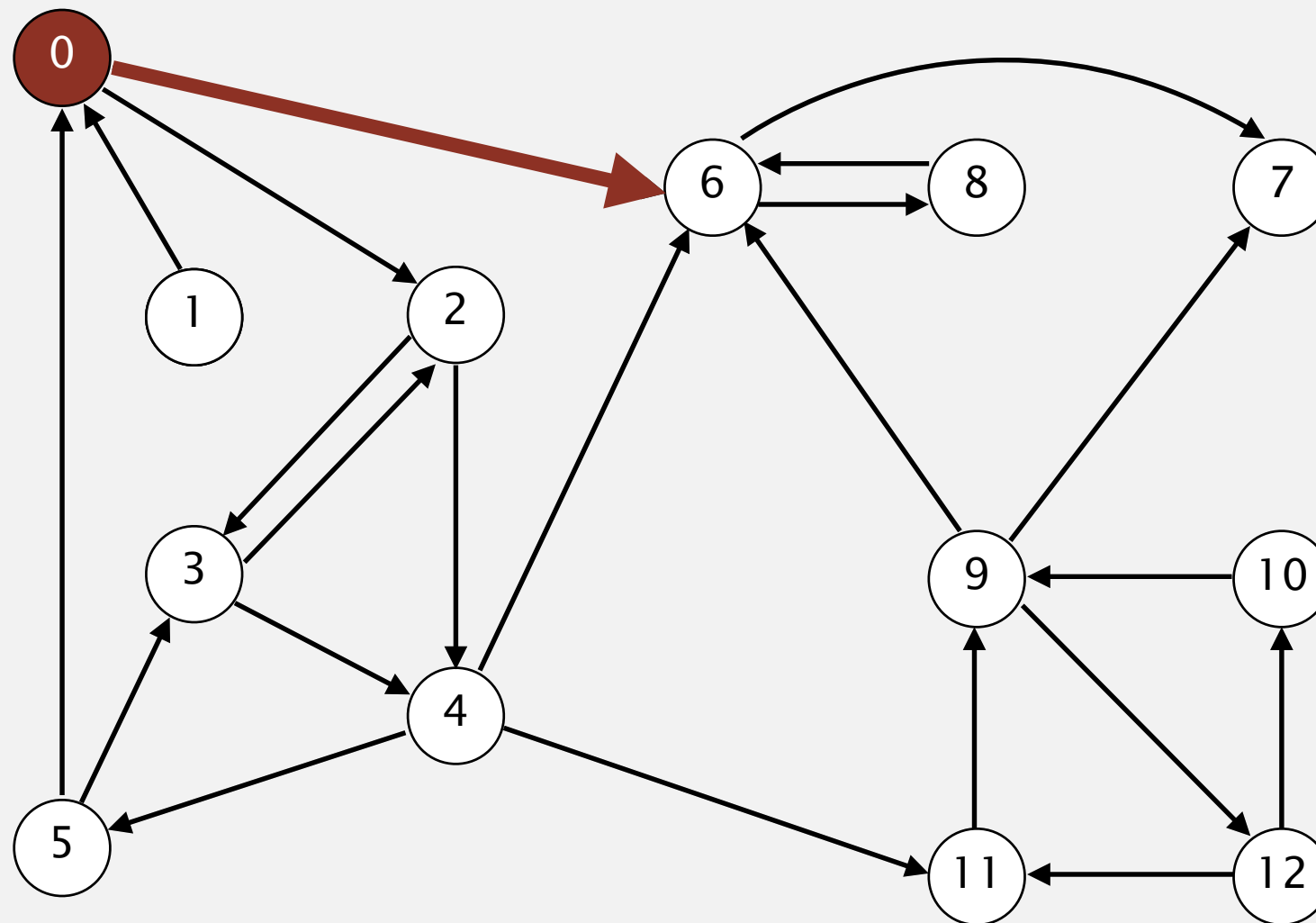


v	marked[]
0	—
1	—
2	—
3	—
4	—
5	—
6	—
7	—
8	—
9	—
10	—
11	—
12	—

reverse digraph  $G^R$

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

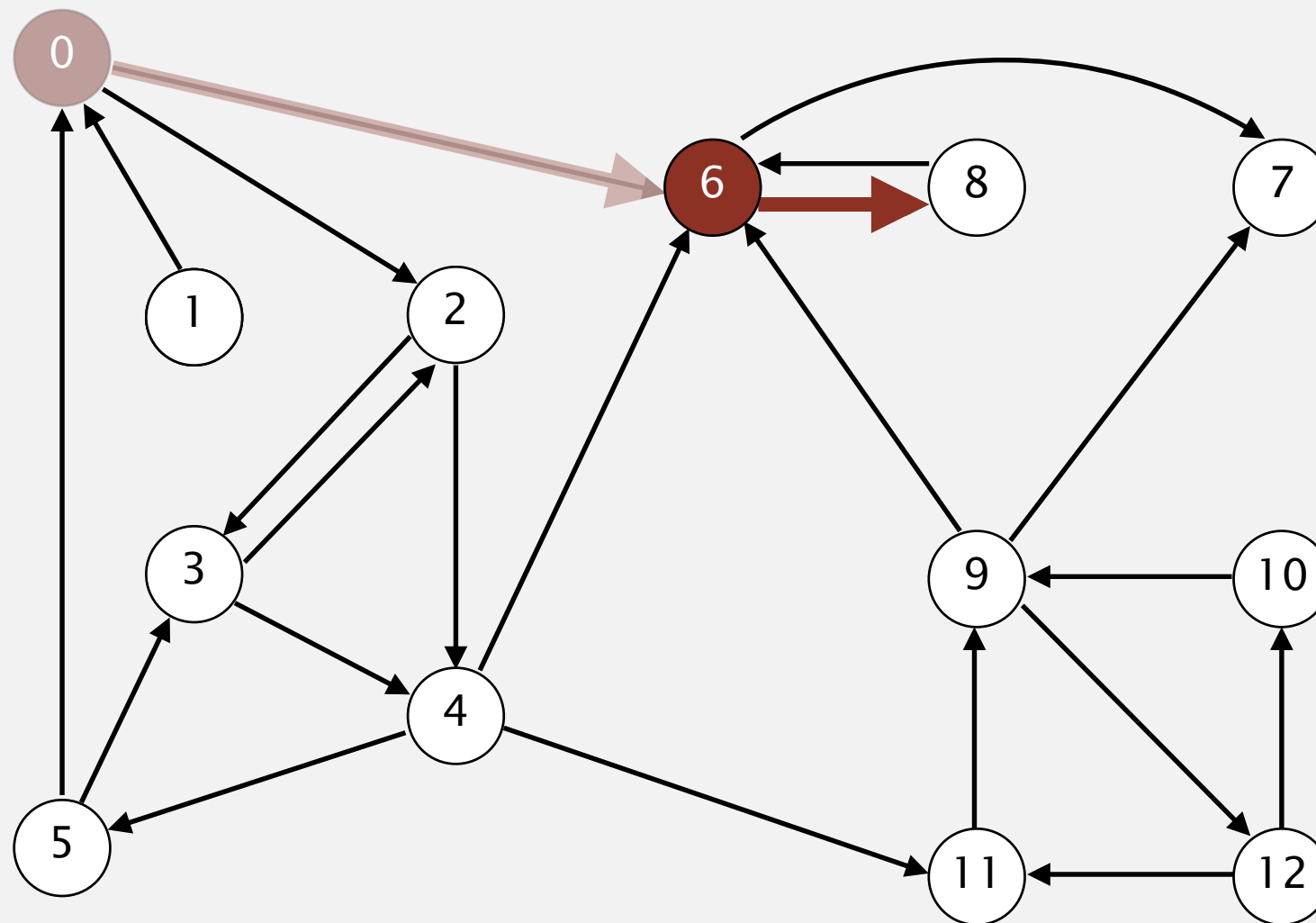


v	marked[]
0	T
1	F
2	F
3	F
4	F
5	F
6	F
7	F
8	F
9	F
10	F
11	F
12	F

visit 0: check 6 and check 2

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

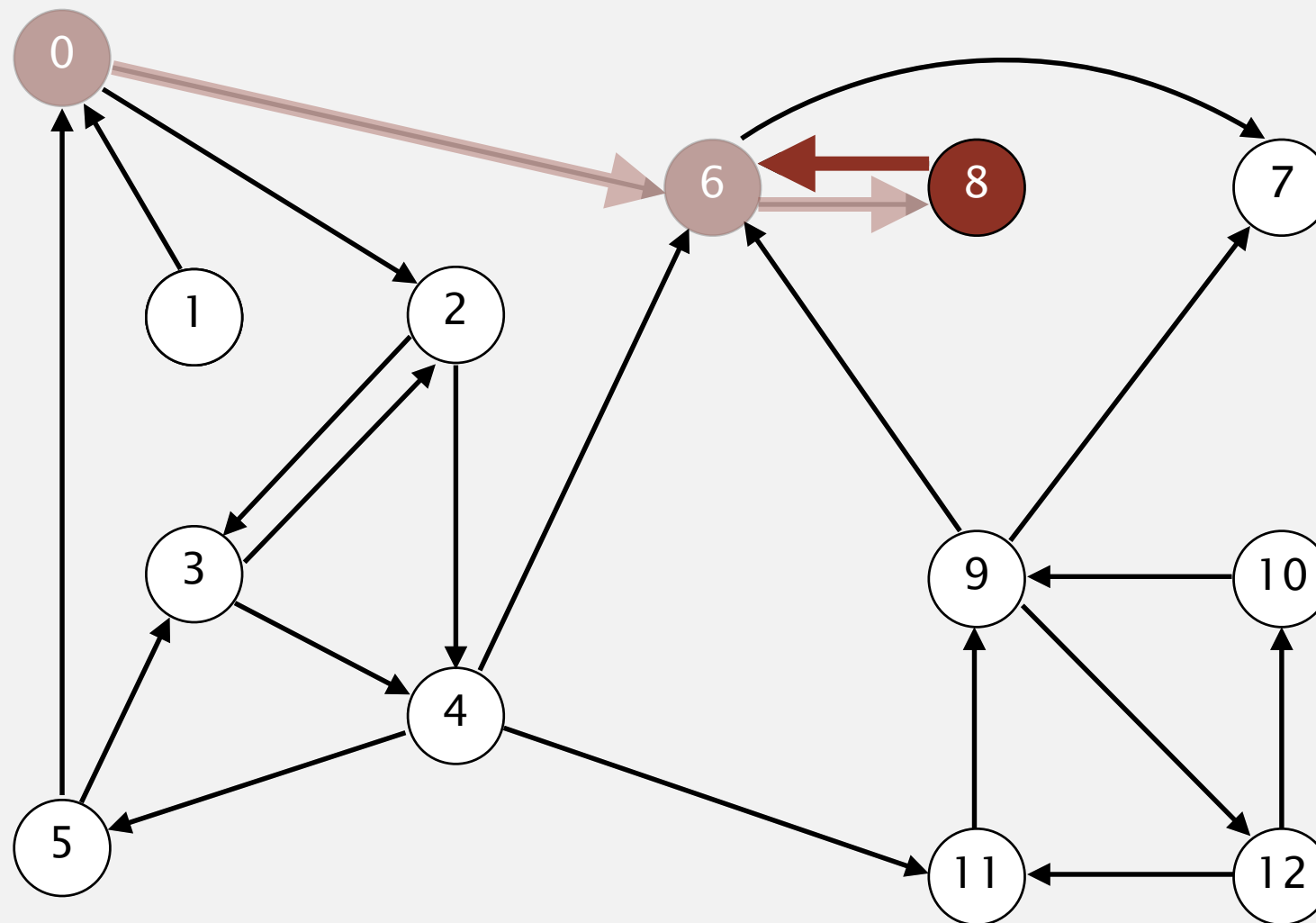


v	marked[]
0	T
1	F
2	F
3	F
4	F
5	F
6	T
7	F
8	F
9	F
10	F
11	F
12	F

visit 6: check 8 and check 7

# 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

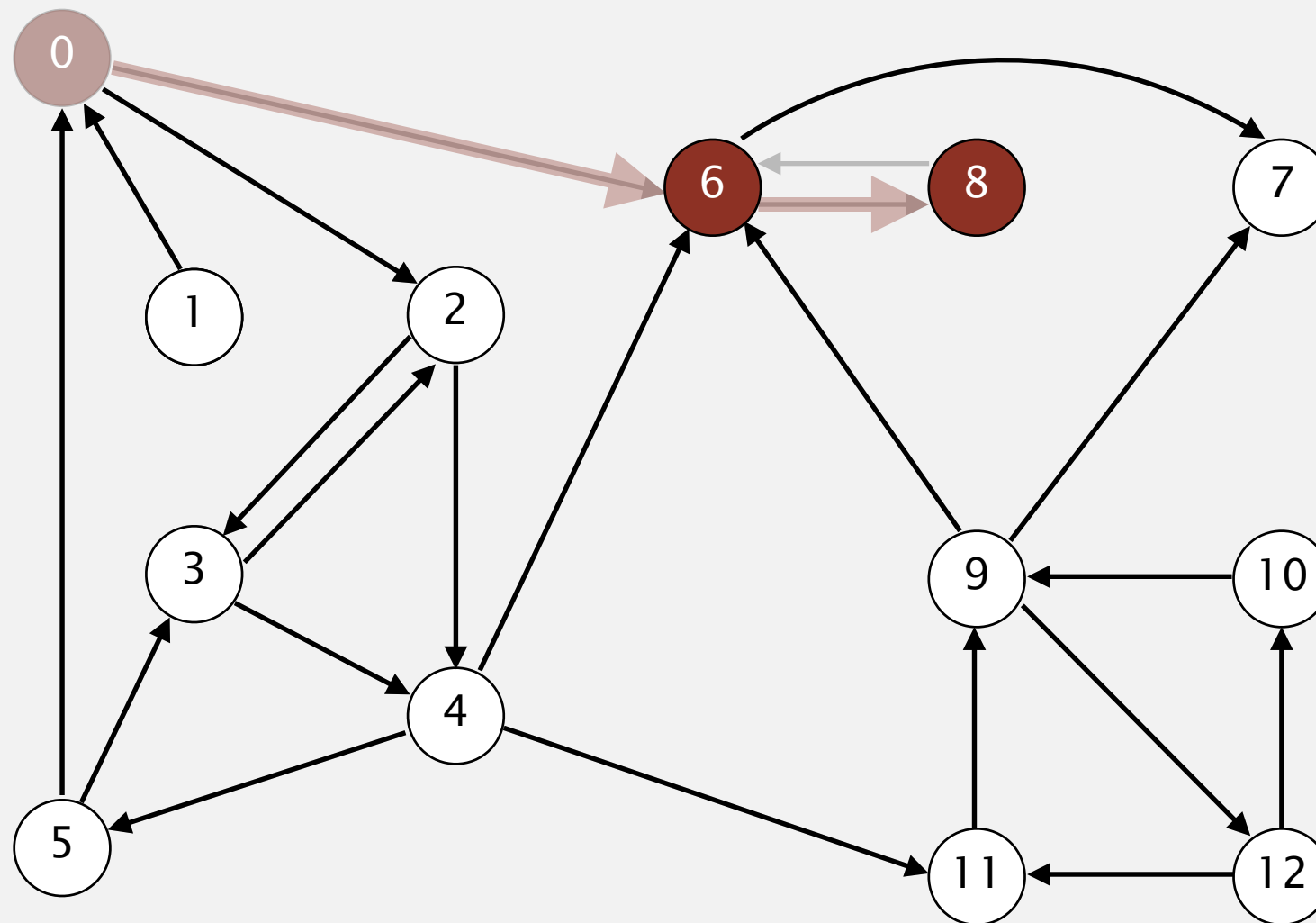
visit 8: check 6



# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

8



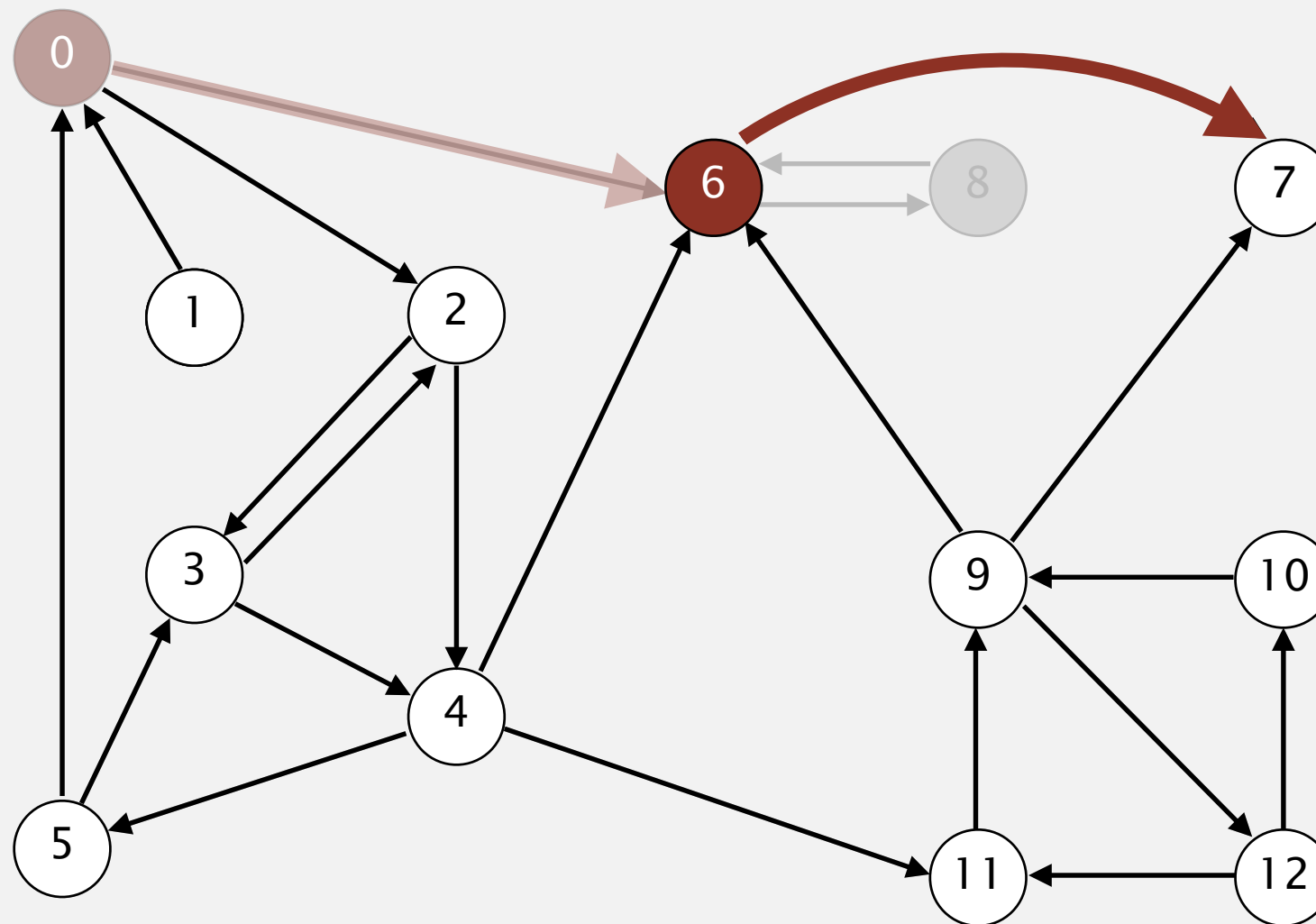
v	marked[]
0	T
1	F
2	F
3	F
4	F
5	F
6	T
7	F
8	T
9	F
10	F
11	F
12	F

8 done

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

8



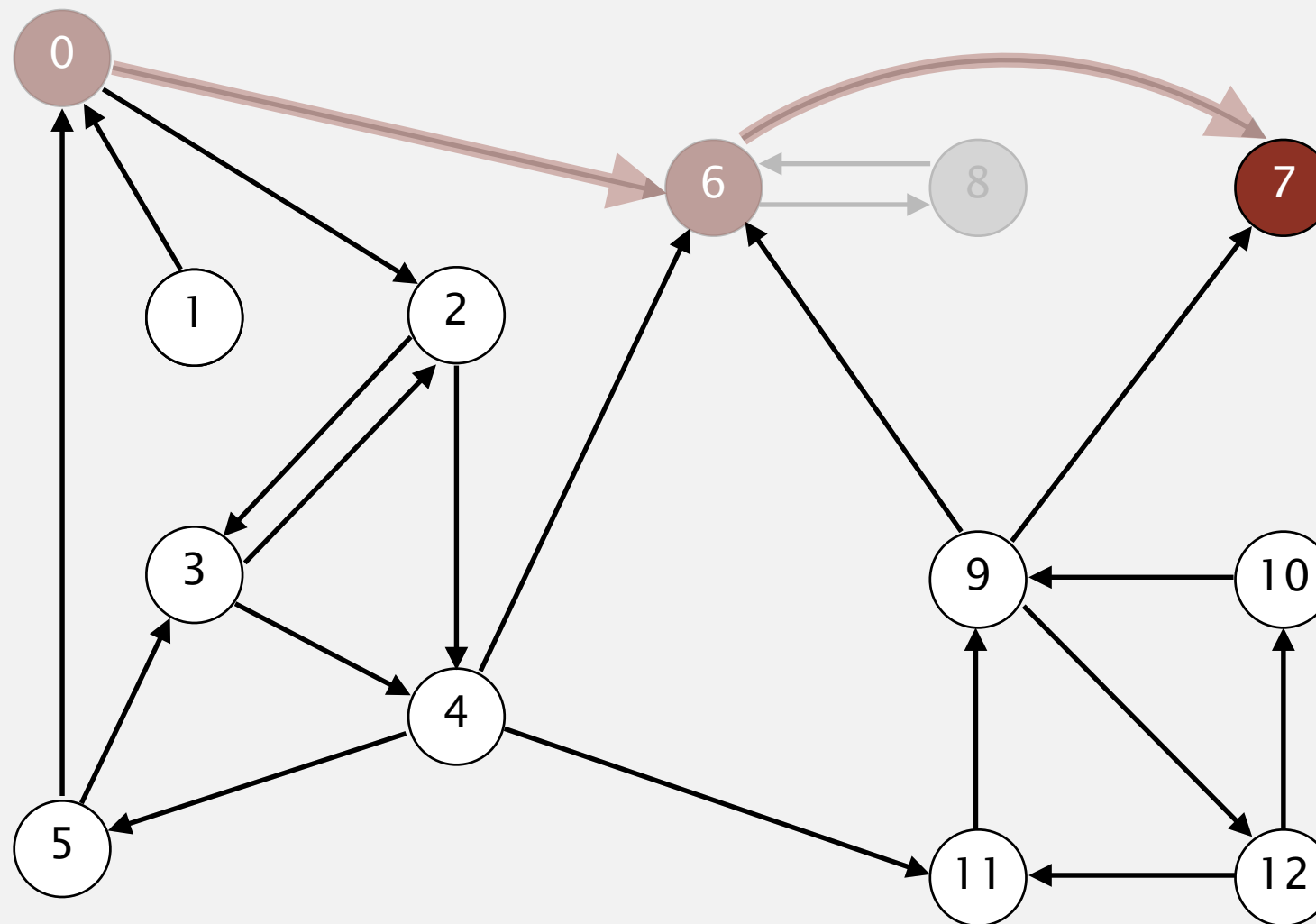
v	marked[]
0	T
1	F
2	F
3	F
4	F
5	F
6	T
7	F
8	T
9	F
10	F
11	F
12	F

visit 6: check 8 and check 7

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

8



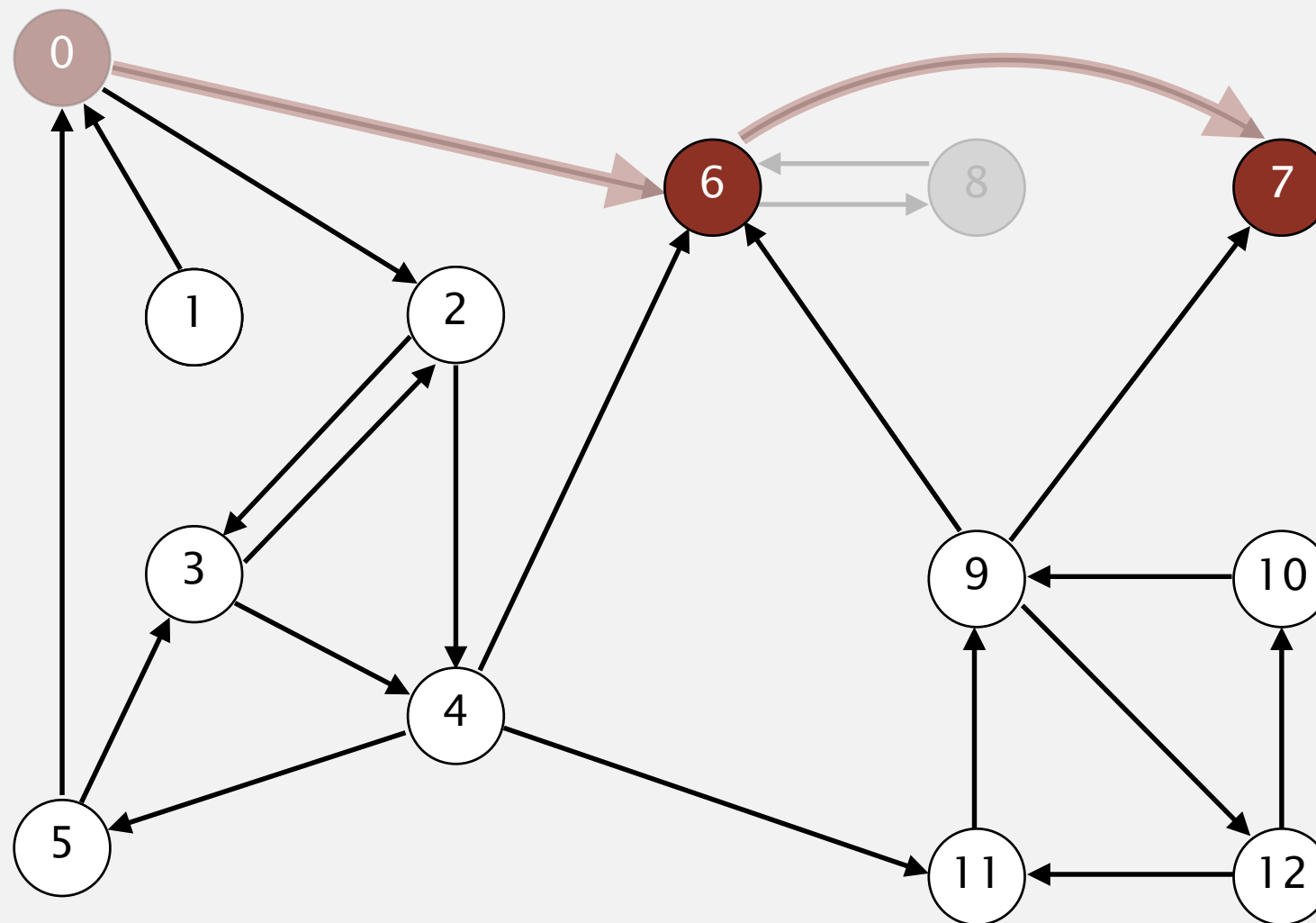
v	marked[]
0	T
1	F
2	F
3	F
4	F
5	F
6	T
7	T
8	T
9	F
10	F
11	F
12	F

visit 7

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

**7** 8



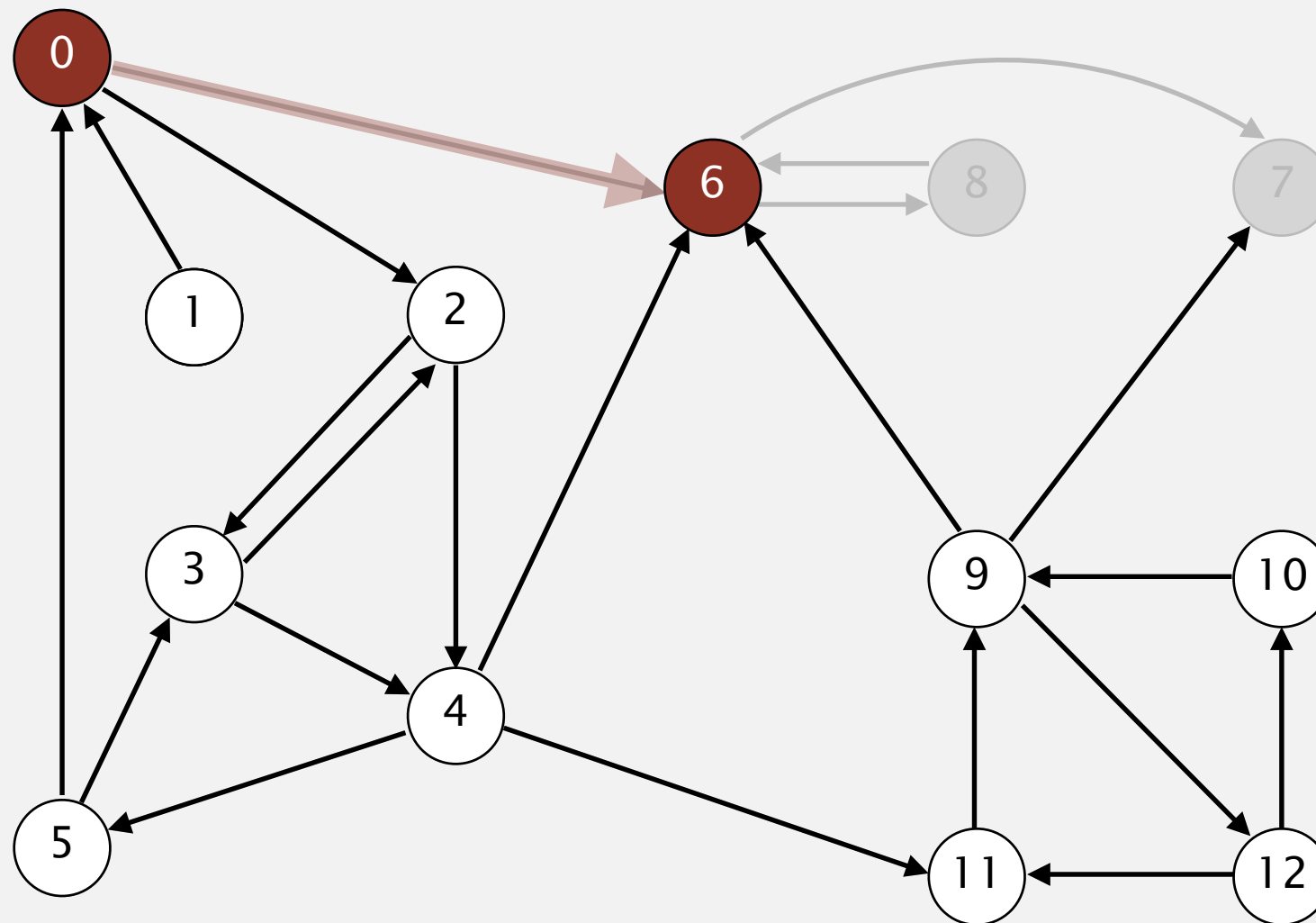
v	marked[]
0	T
1	F
2	F
3	F
4	F
5	F
6	T
7	T
8	T
9	F
10	F
11	F
12	F

**7 done**

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

**6** 7 8



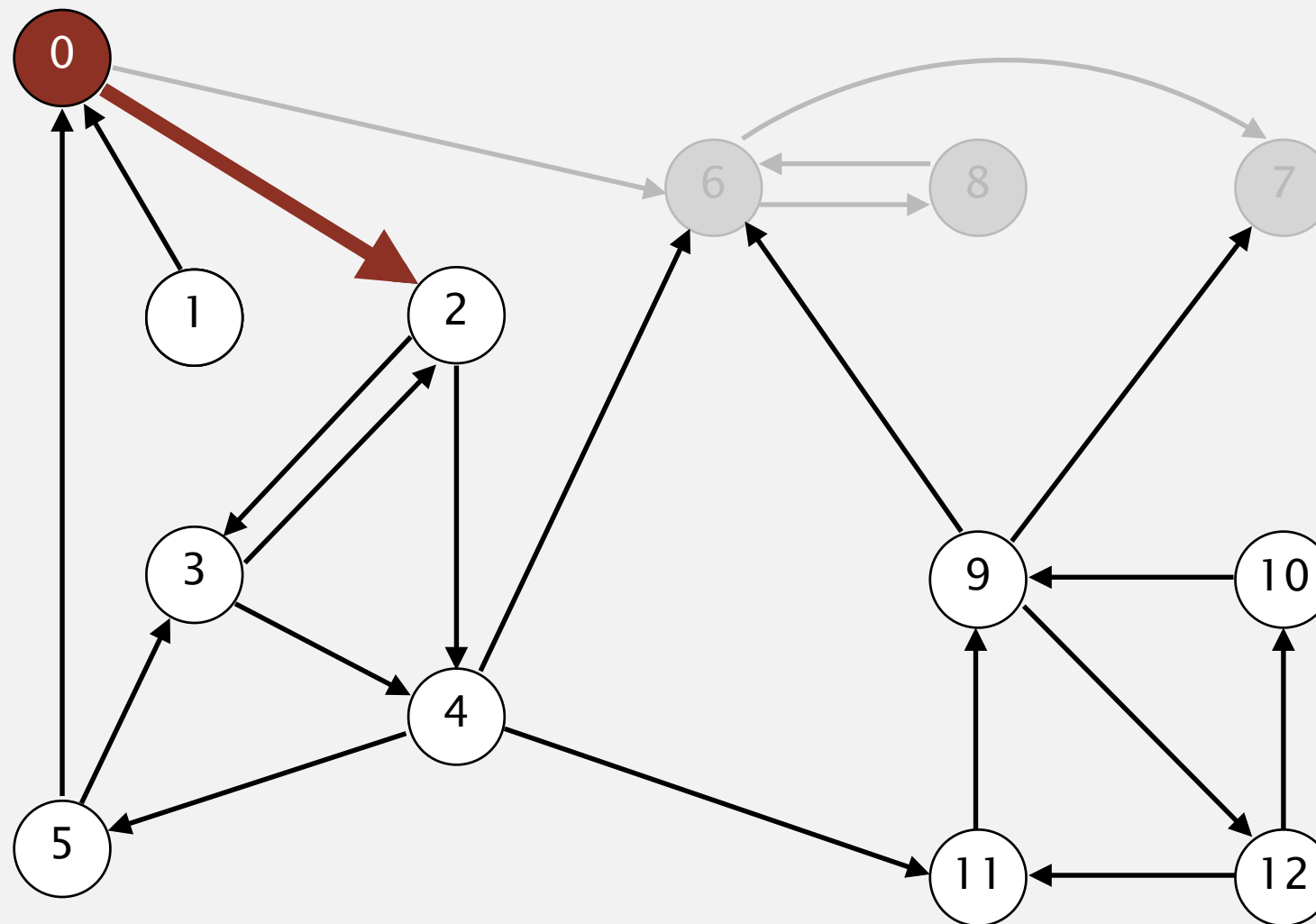
v	marked[]
0	T
1	F
2	F
3	F
4	F
5	F
6	T
7	T
8	T
9	F
10	F
11	F
12	F

**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

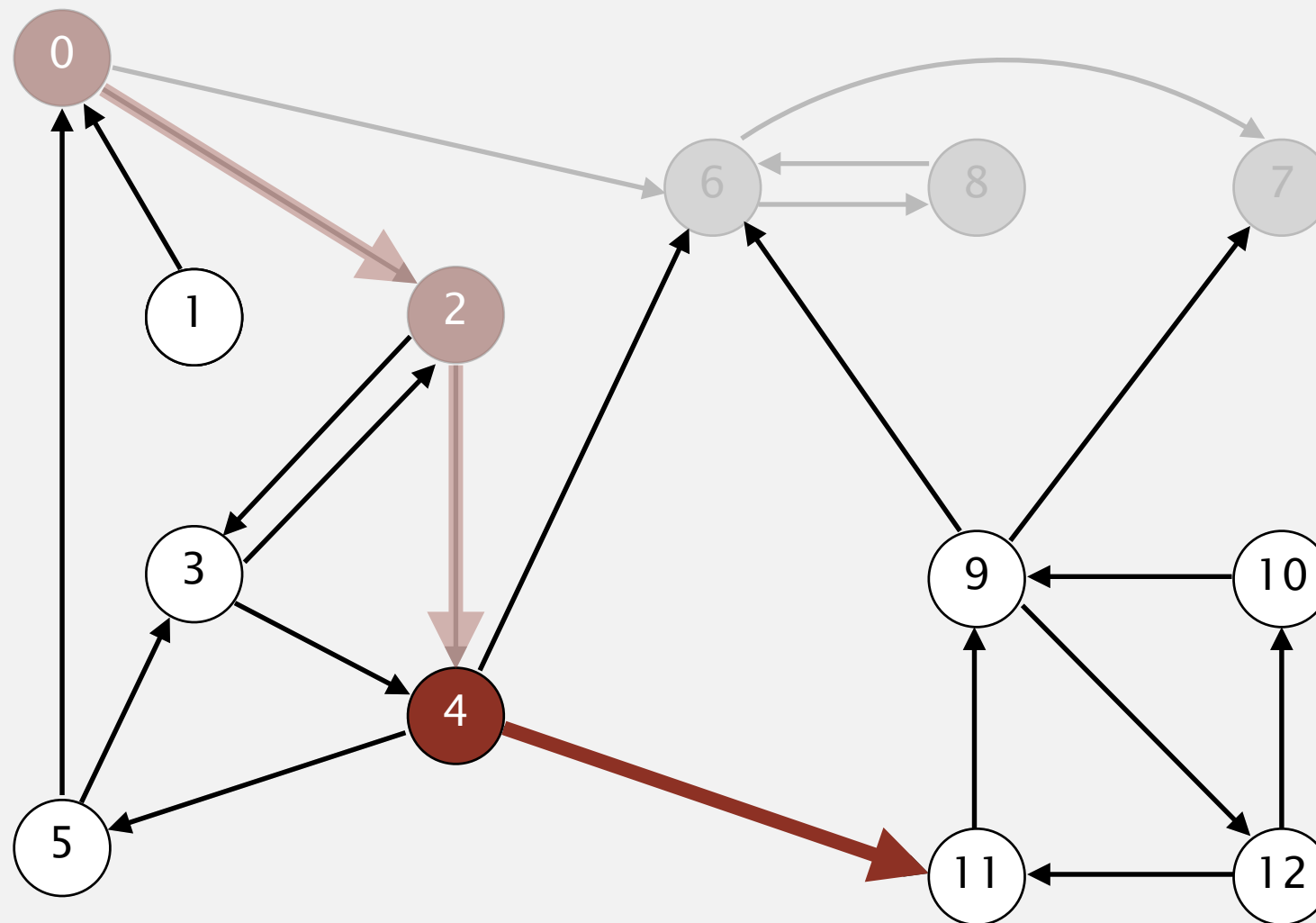
visit 0: check 6 and check 2



# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

6 7 8



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

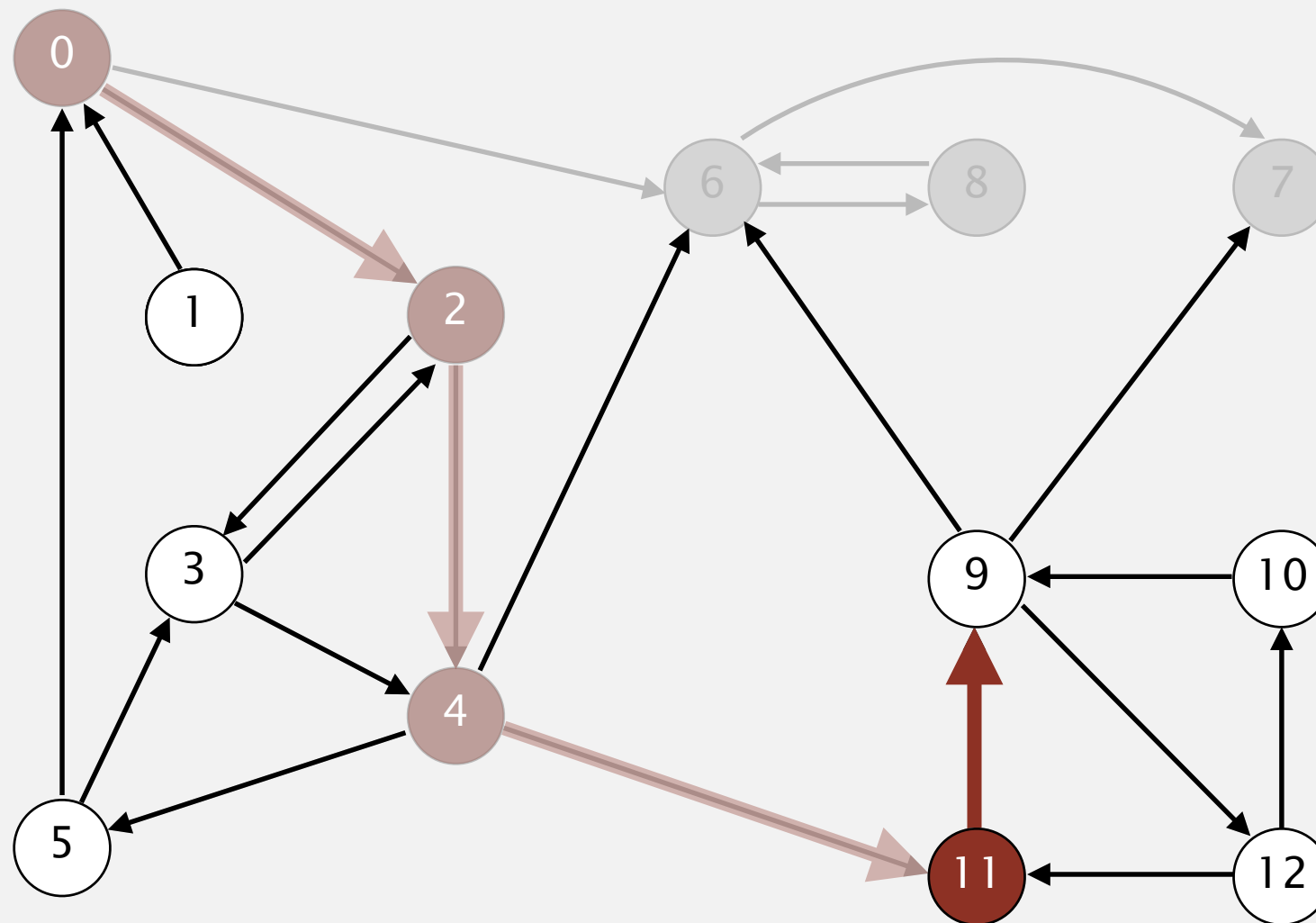
visit 4: check 11, check 6, and check 5



# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

6 7 8



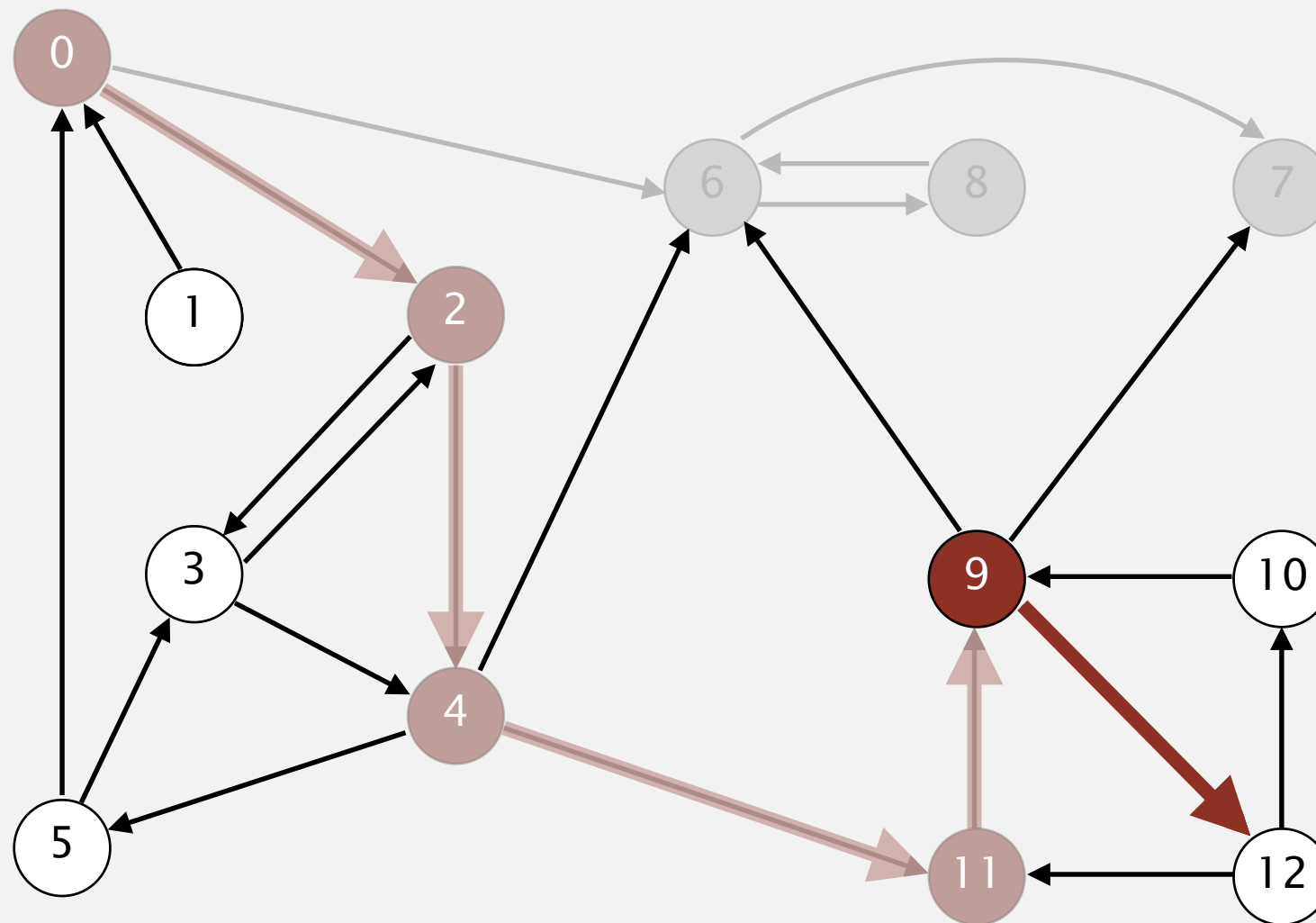
v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	F
10	F
11	T
12	F

visit 11: check 9

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

6 7 8



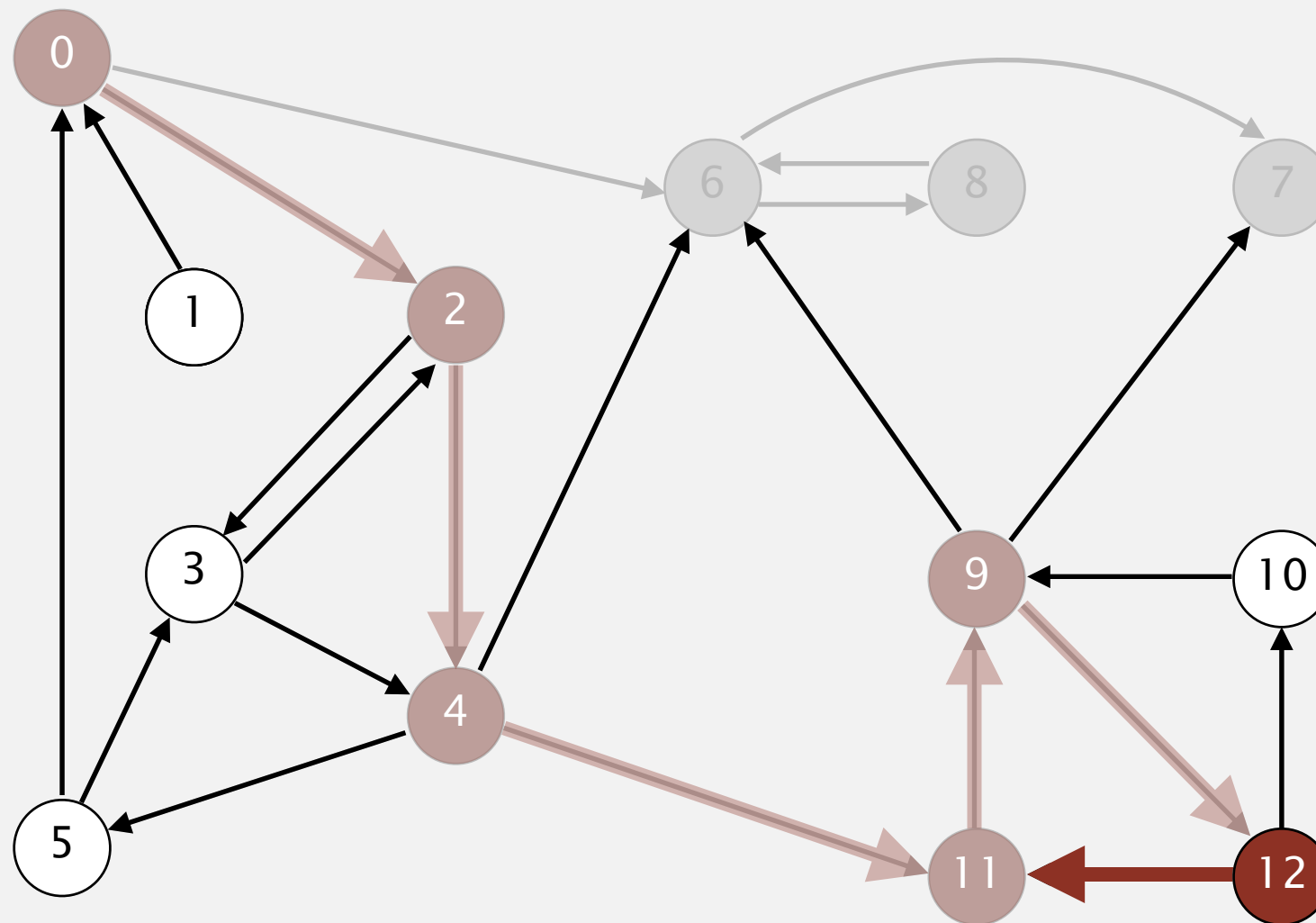
v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	F
11	T
12	F

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

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

6 7 8



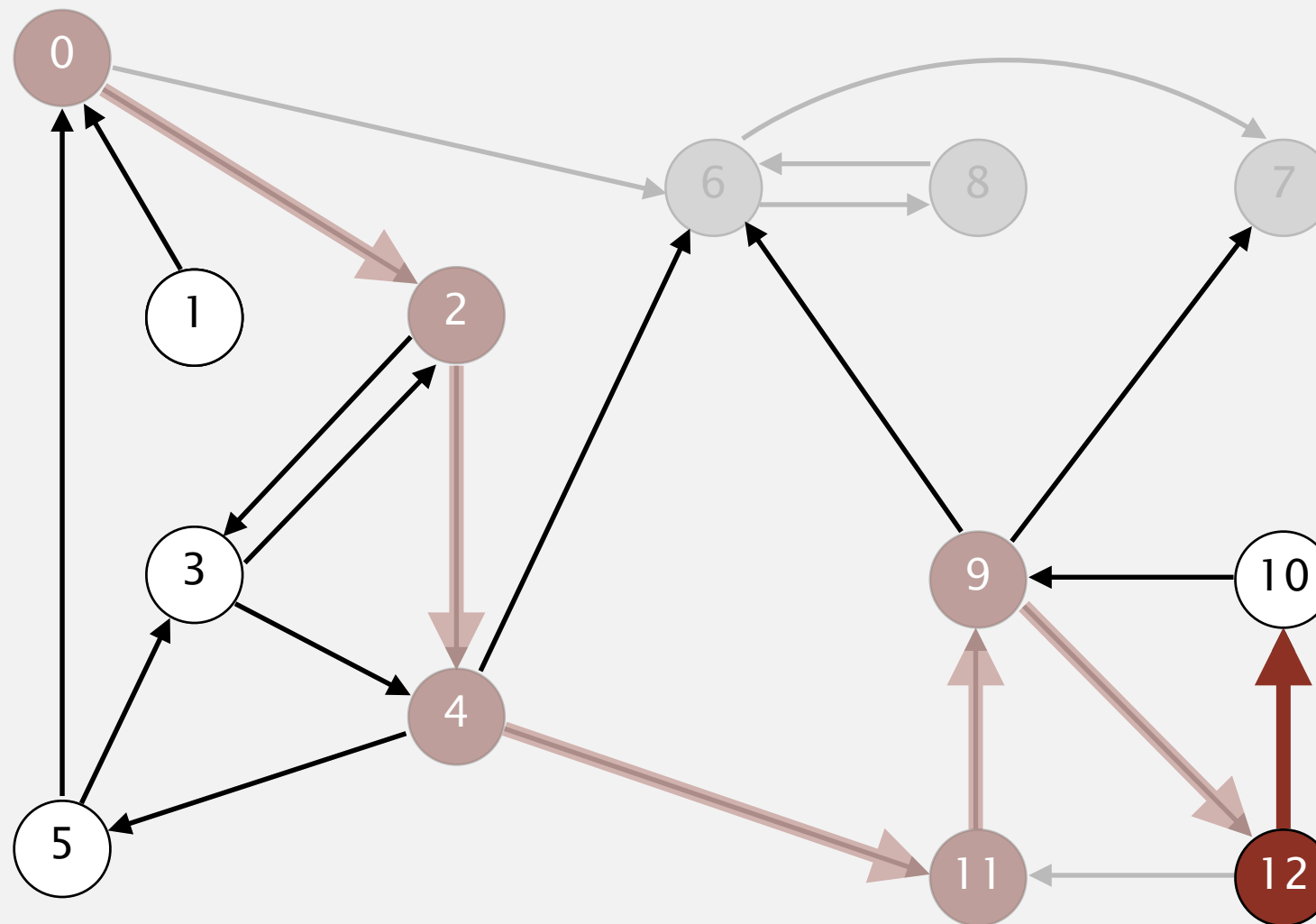
v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	F
11	T
12	T

visit 12: check 11 and check 10

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

6 7 8



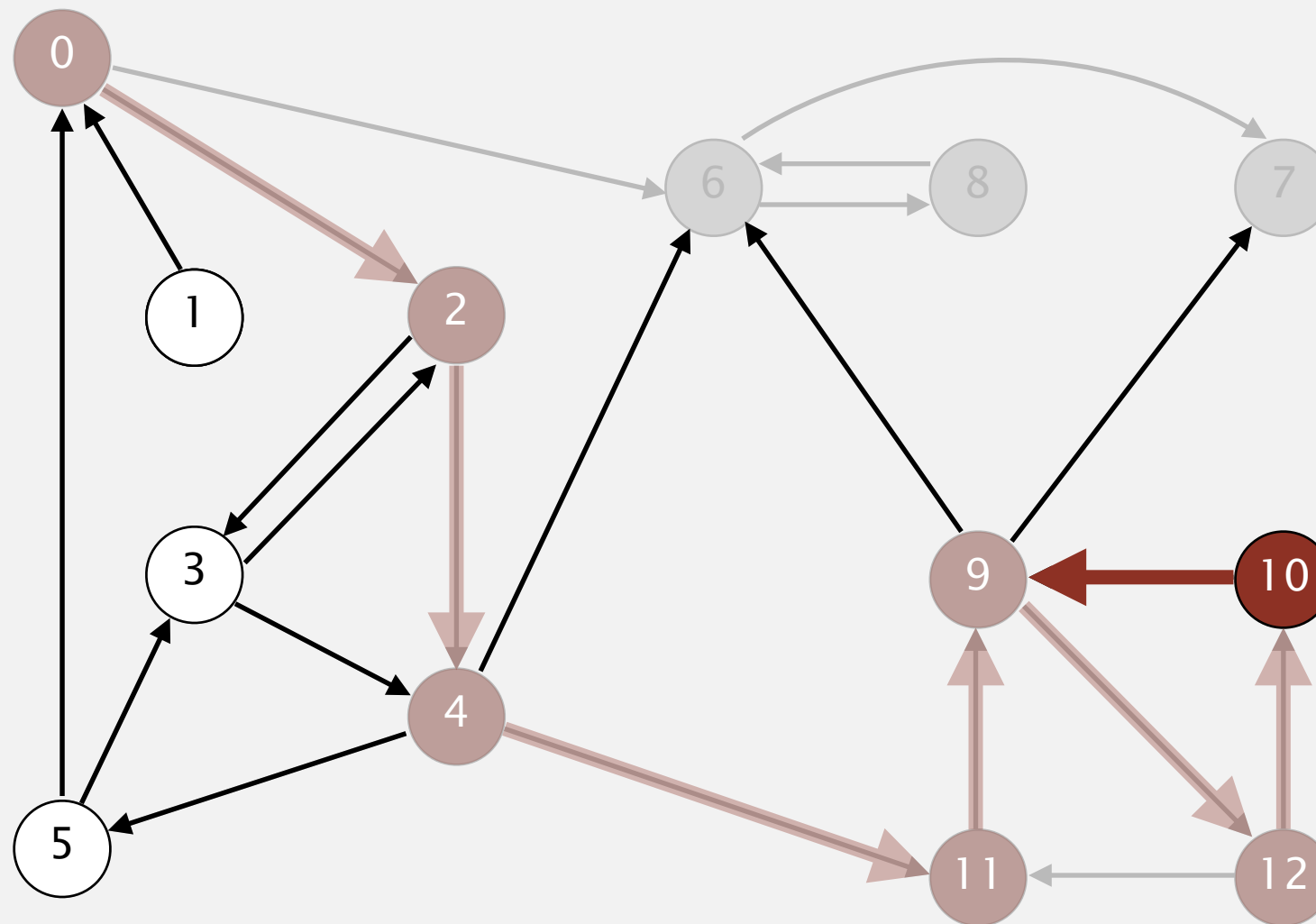
v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	F
11	T
12	T

visit 12: check 11 and check 10

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

6 7 8



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 10: check 9

Phase 1. Compute reverse postorder in  $G^R$ .

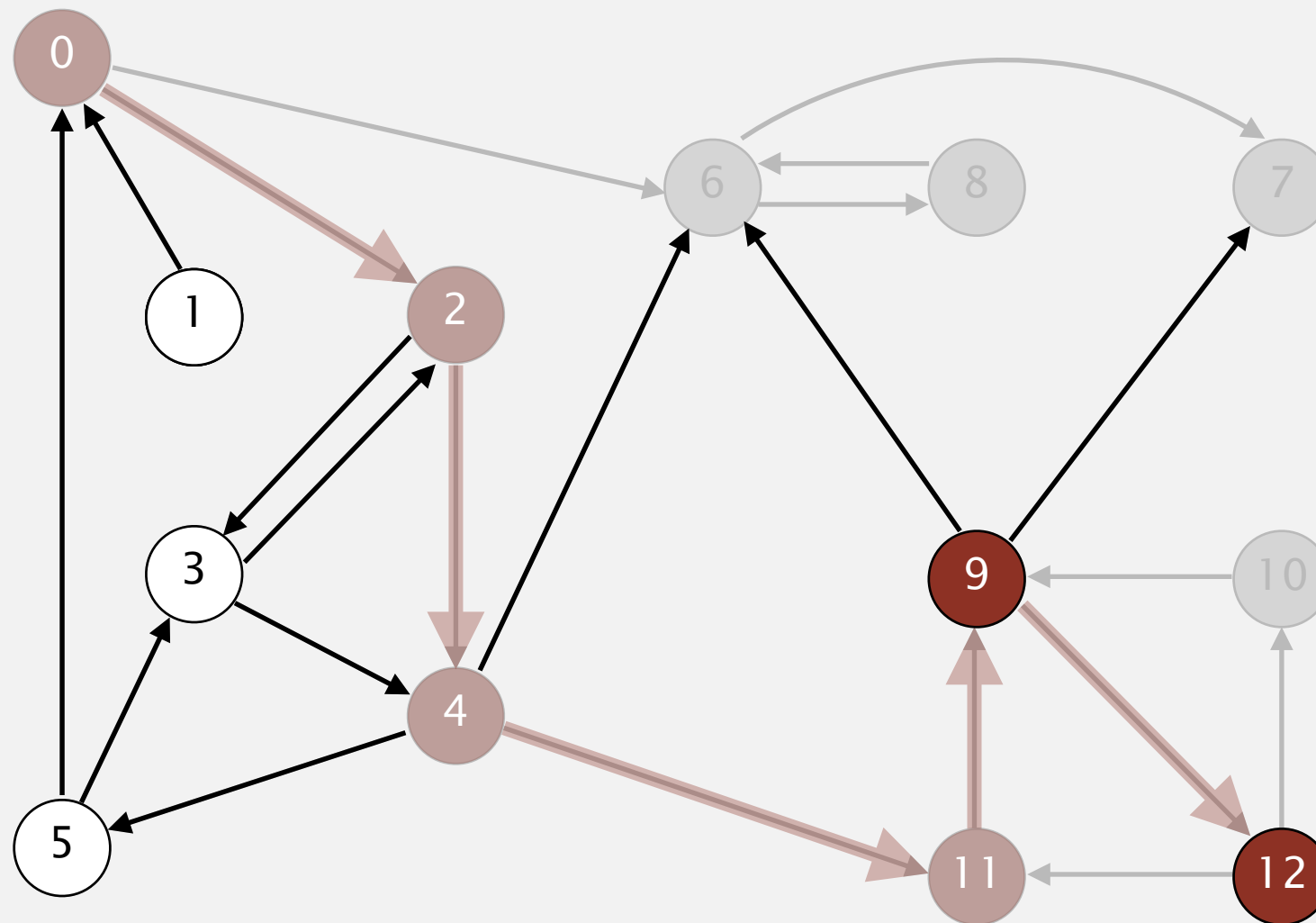
The graph consists of 13 nodes labeled 0 through 12. The nodes are colored as follows: 0, 2, 4, 6, 8, and 10 are red; 1, 3, 5, 7, 9, and 11 are white; and 12 and 13 are dark red. The edges are directed and have varying thicknesses and colors. The flow path is highlighted with thick red arrows: 0 → 2 → 4 → 11 → 9 → 7 → 12. Other edges include: 0 → 6 (thin grey), 1 → 0 (black), 2 → 3 (black), 3 → 2 (black), 3 → 4 (black), 4 → 3 (black), 4 → 6 (black), 5 → 0 (black), 5 → 3 (black), 5 → 4 (black), 6 → 8 (black), 8 → 6 (black), 7 → 6 (black), 9 → 6 (black), 9 → 7 (black), 10 → 9 (black), 11 → 9 (black), 12 → 11 (black), and 12 → 10 (black). There are also curved grey arrows from 0 to 6 and from 10 to 12.

10 done

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

**12** 10 6 7 8



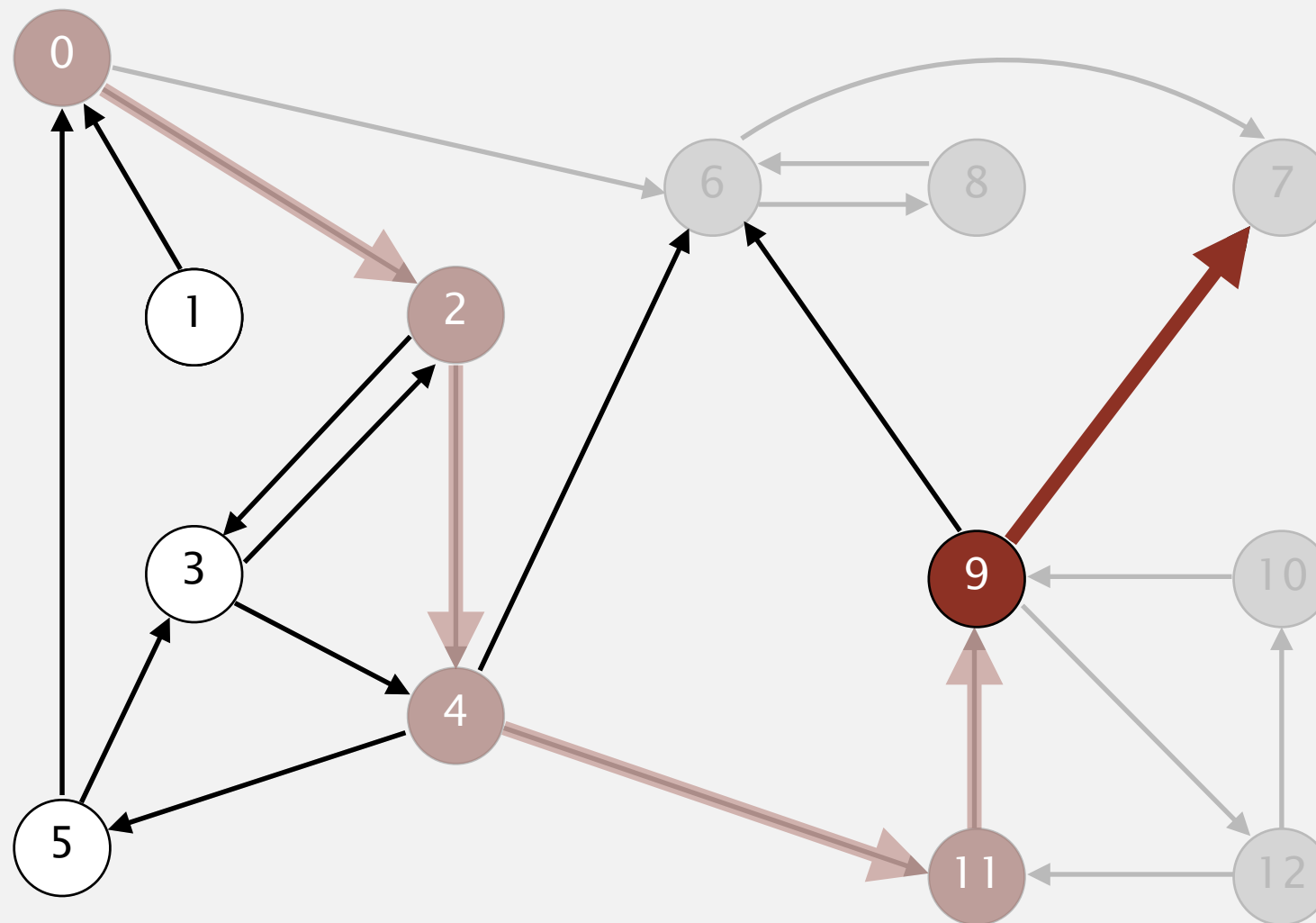
v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	T
11	T
12	T

**12 done**

# 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

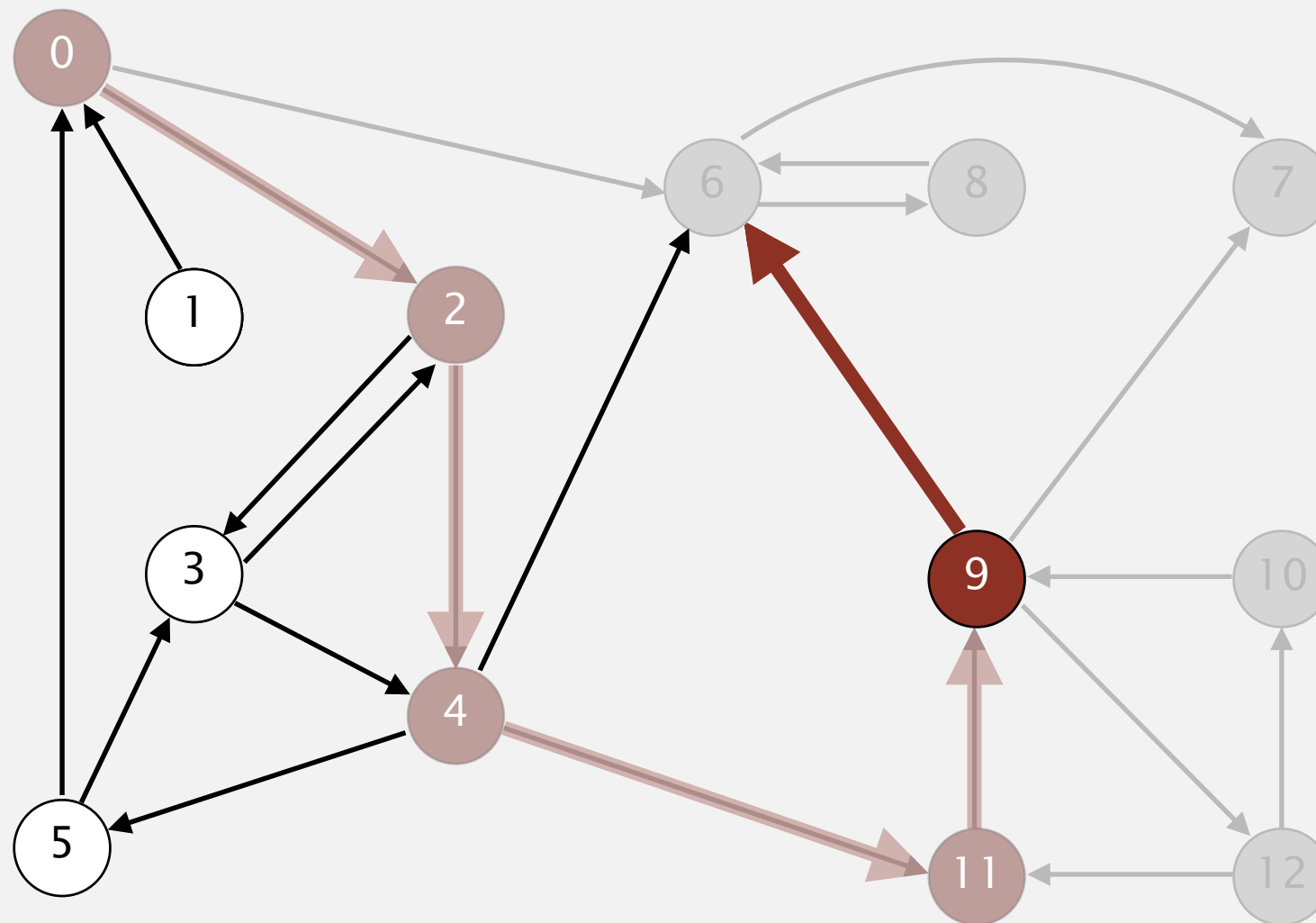
visit 9: check 12, check 7 and check 6



# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

12 10 6 7 8



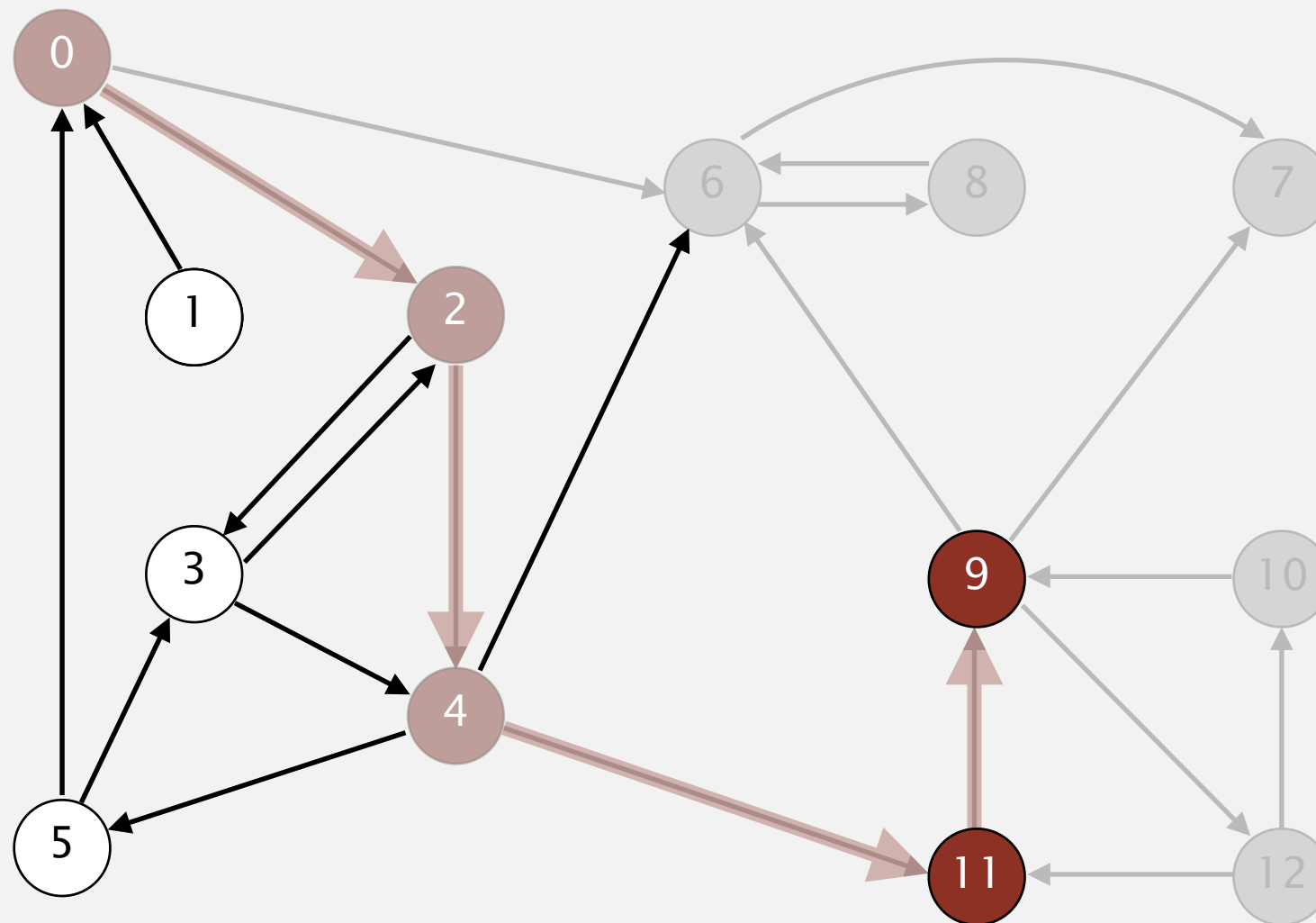
v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	T
11	T
12	T

**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



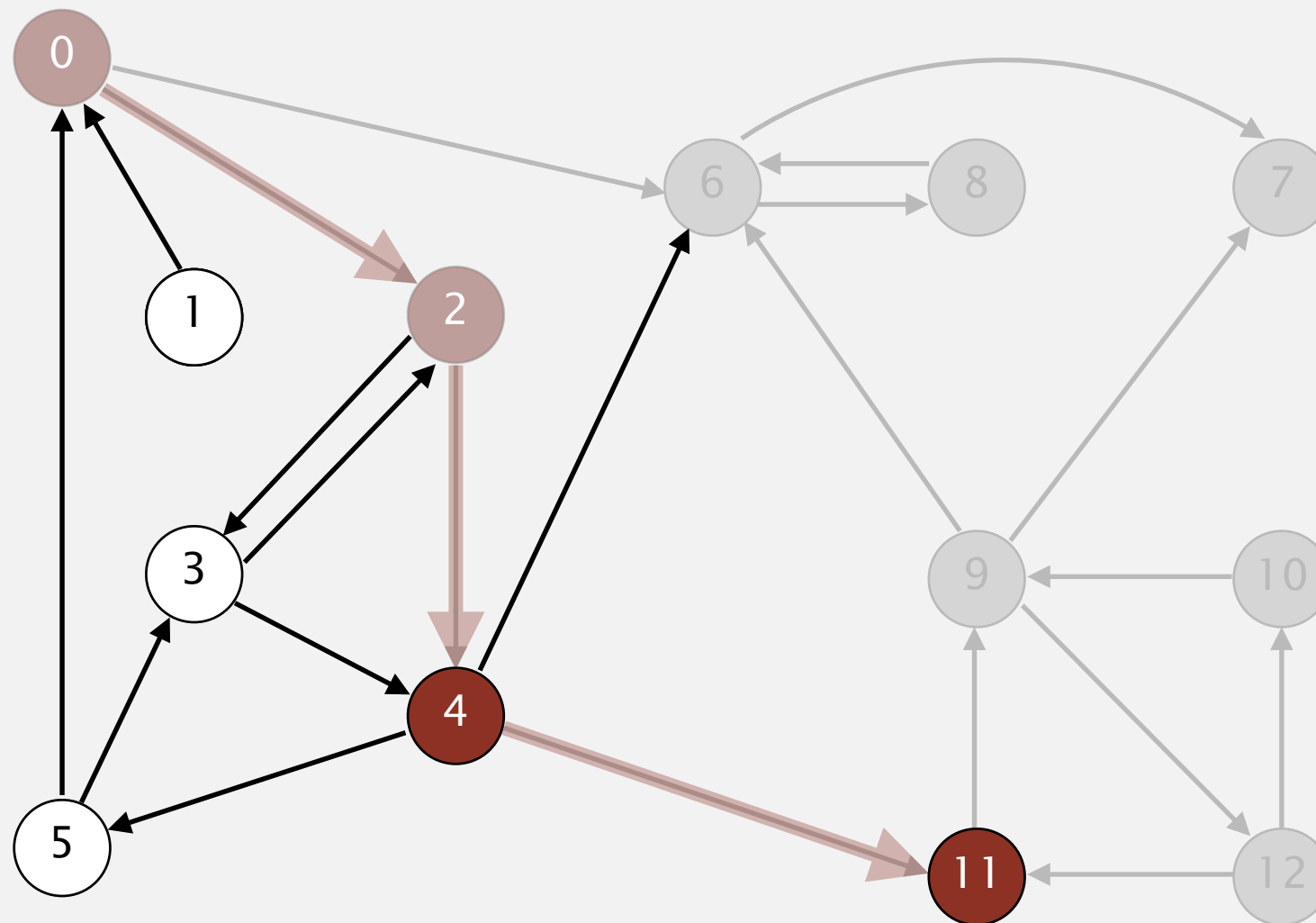
v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	T
11	T
12	T

**9 done**

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

**11** 9 12 10 6 7 8



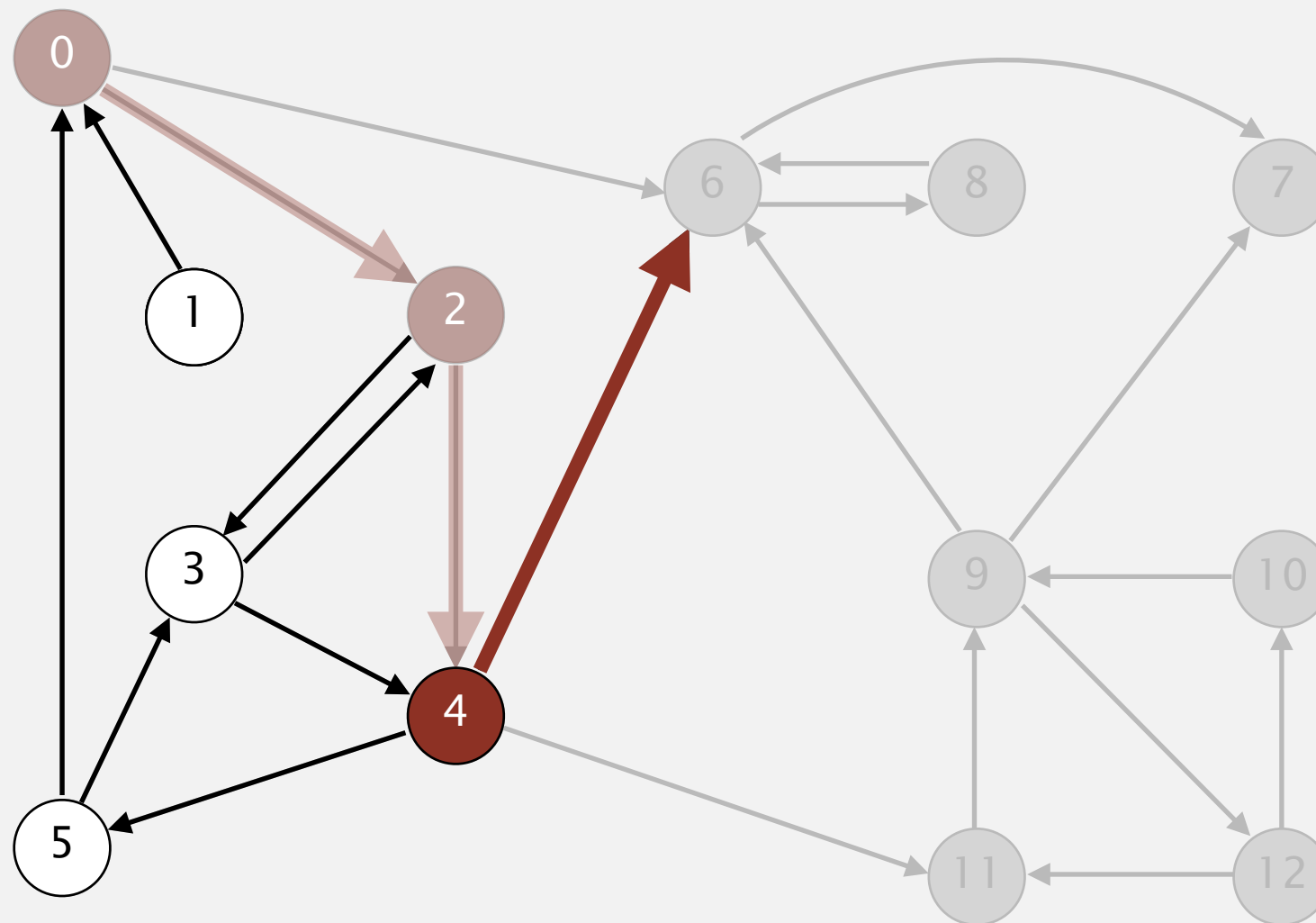
v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	T
11	T
12	T

**11 done**

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

11 9 12 10 6 7 8



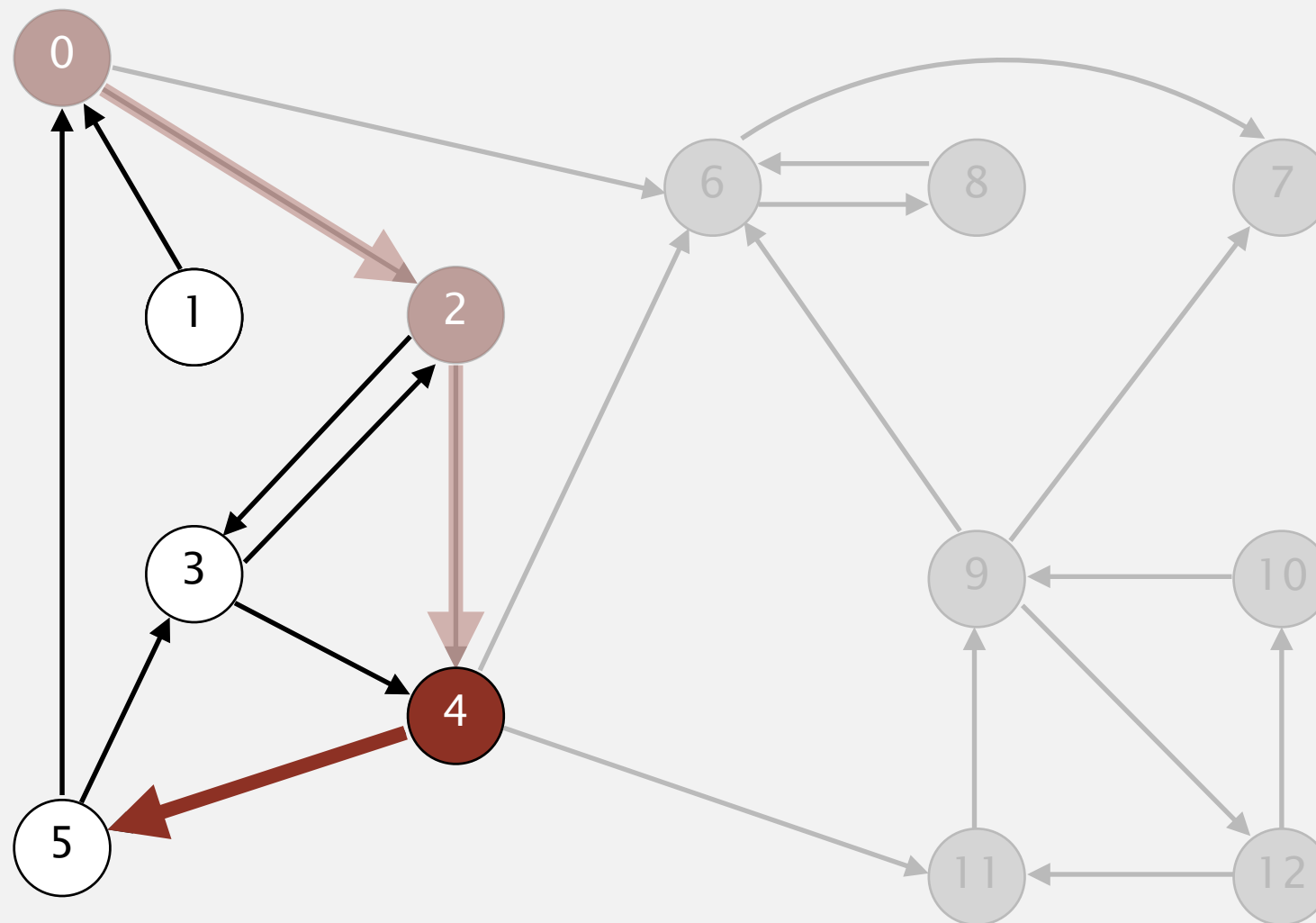
v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	T
11	T
12	T

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

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

11 9 12 10 6 7 8



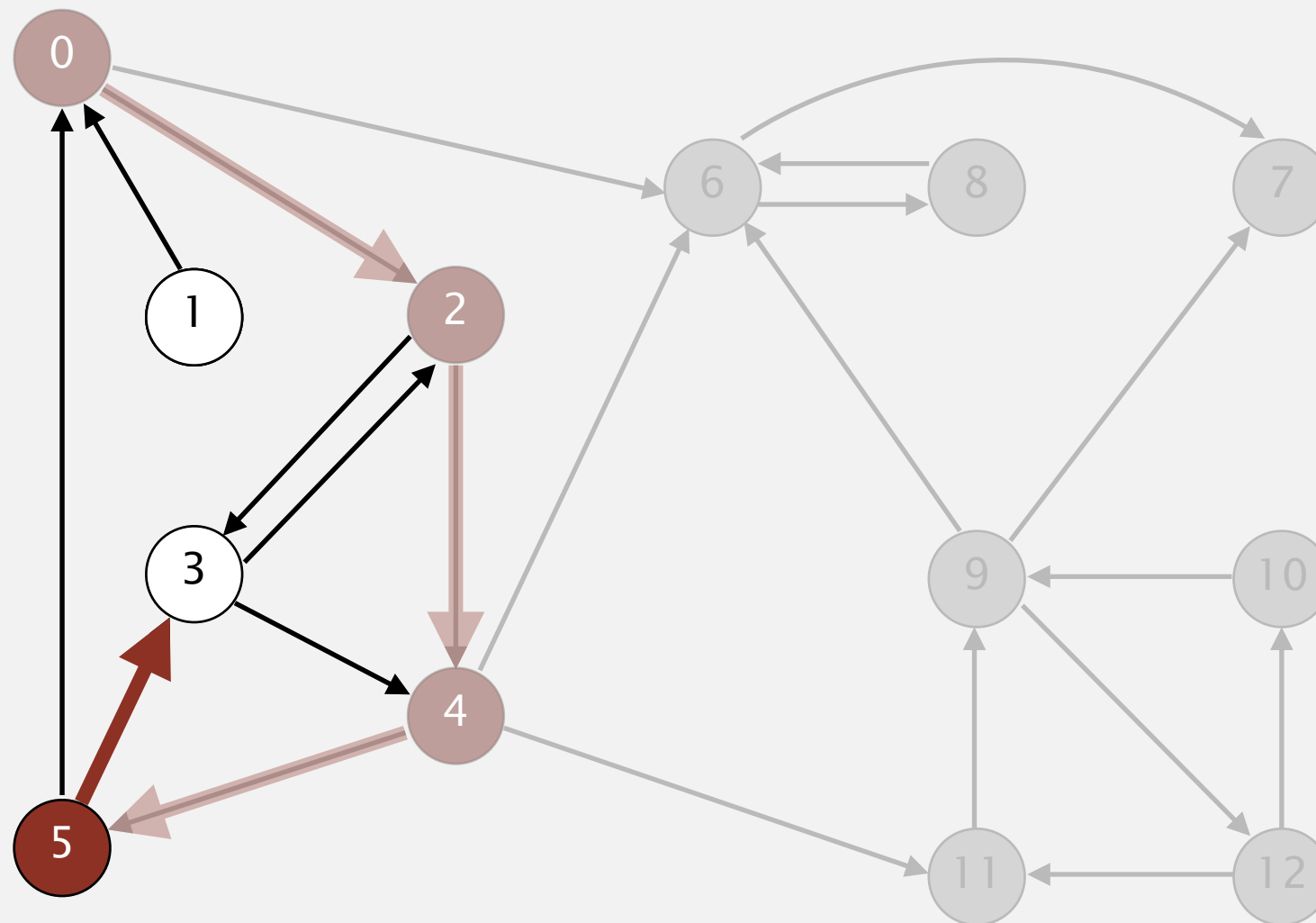
v	marked[]
0	T
1	F
2	T
3	F
4	T
5	F
6	T
7	T
8	T
9	T
10	T
11	T
12	T

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

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

11 9 12 10 6 7 8



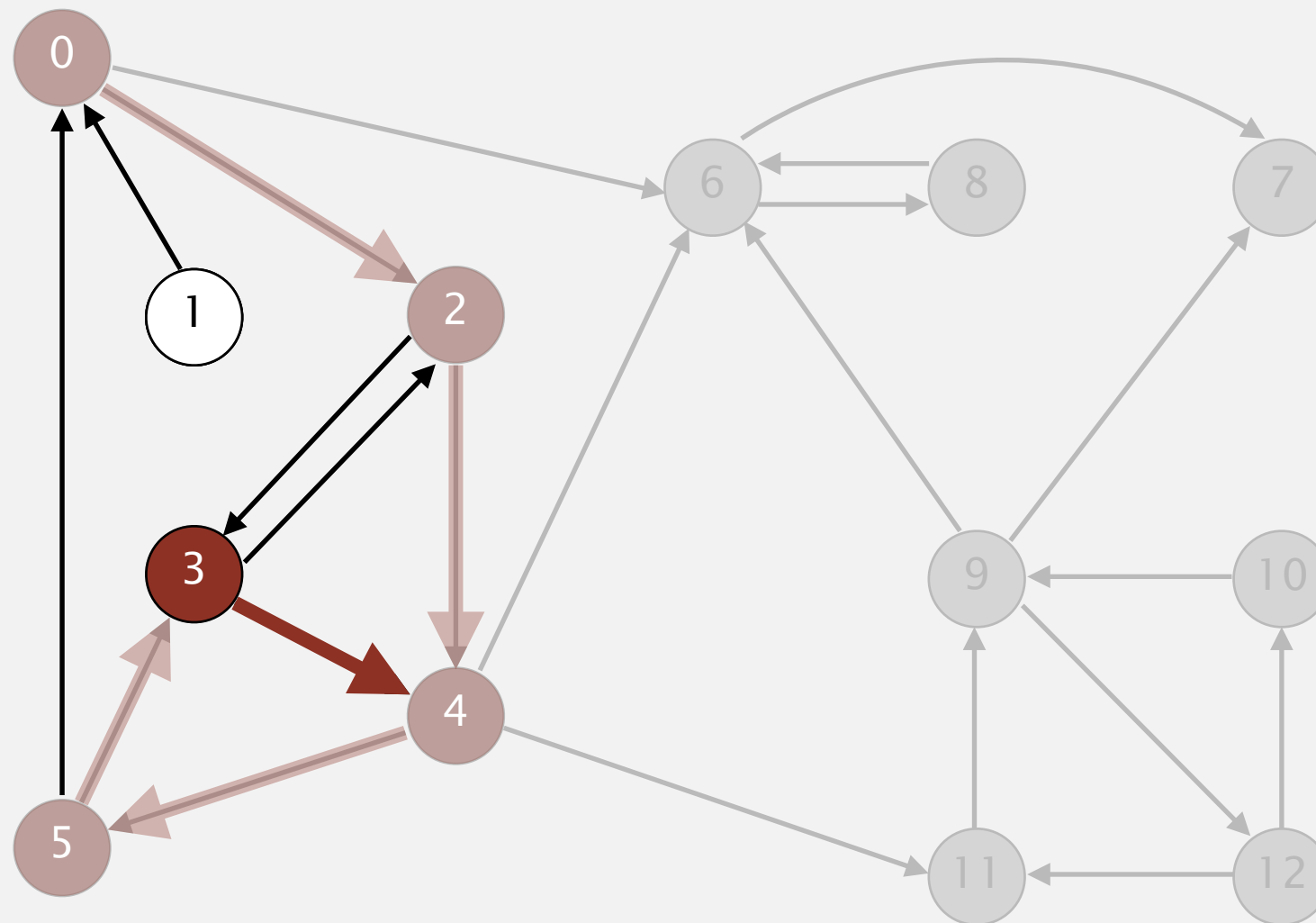
v	marked[]
0	T
1	F
2	T
3	F
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

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



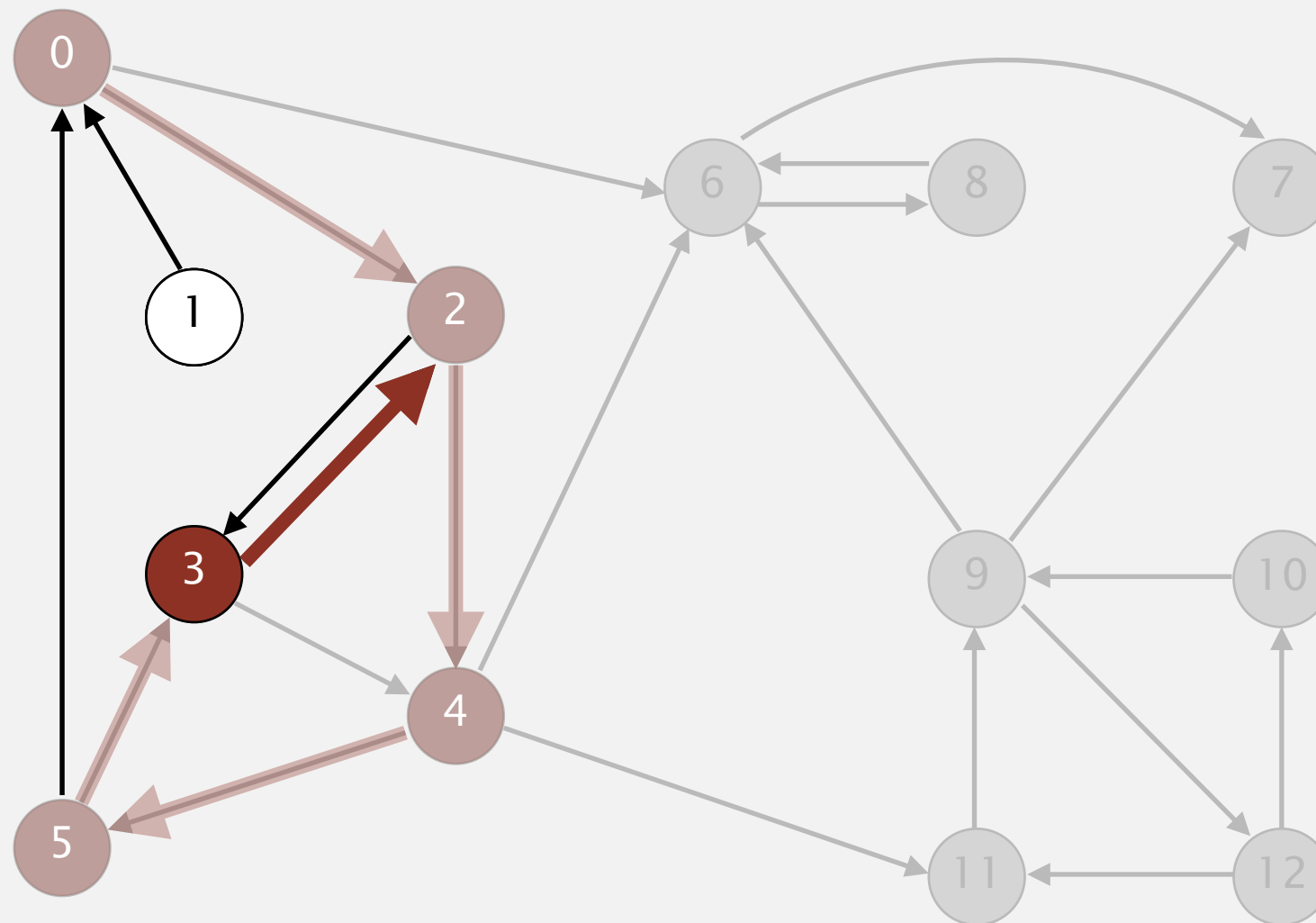
v	marked[]
0	T
1	F
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

visit 3: check 4 and check 2

# 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	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

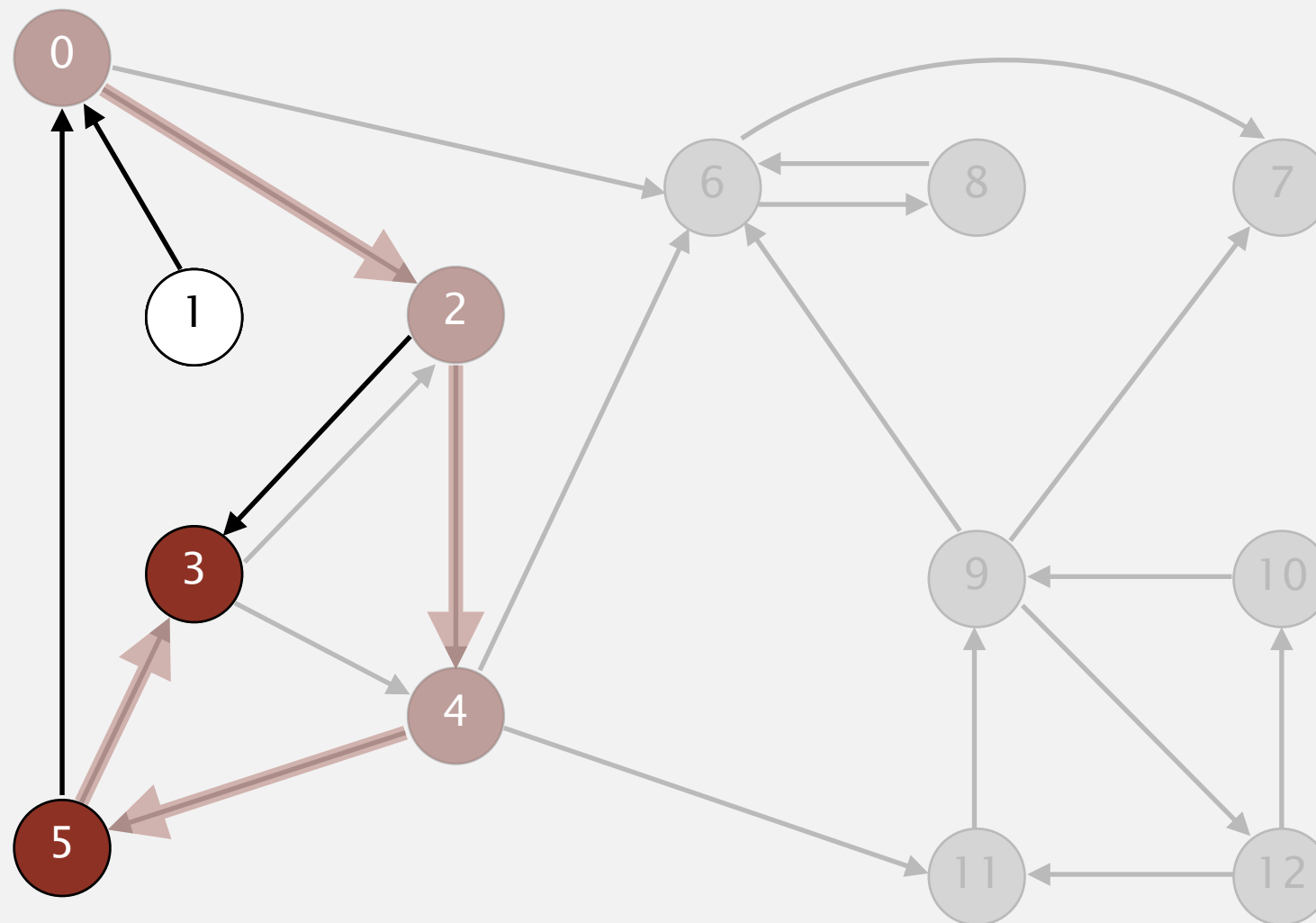
visit 3: check 4 and check 2



# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

**3** 11 9 12 10 6 7 8



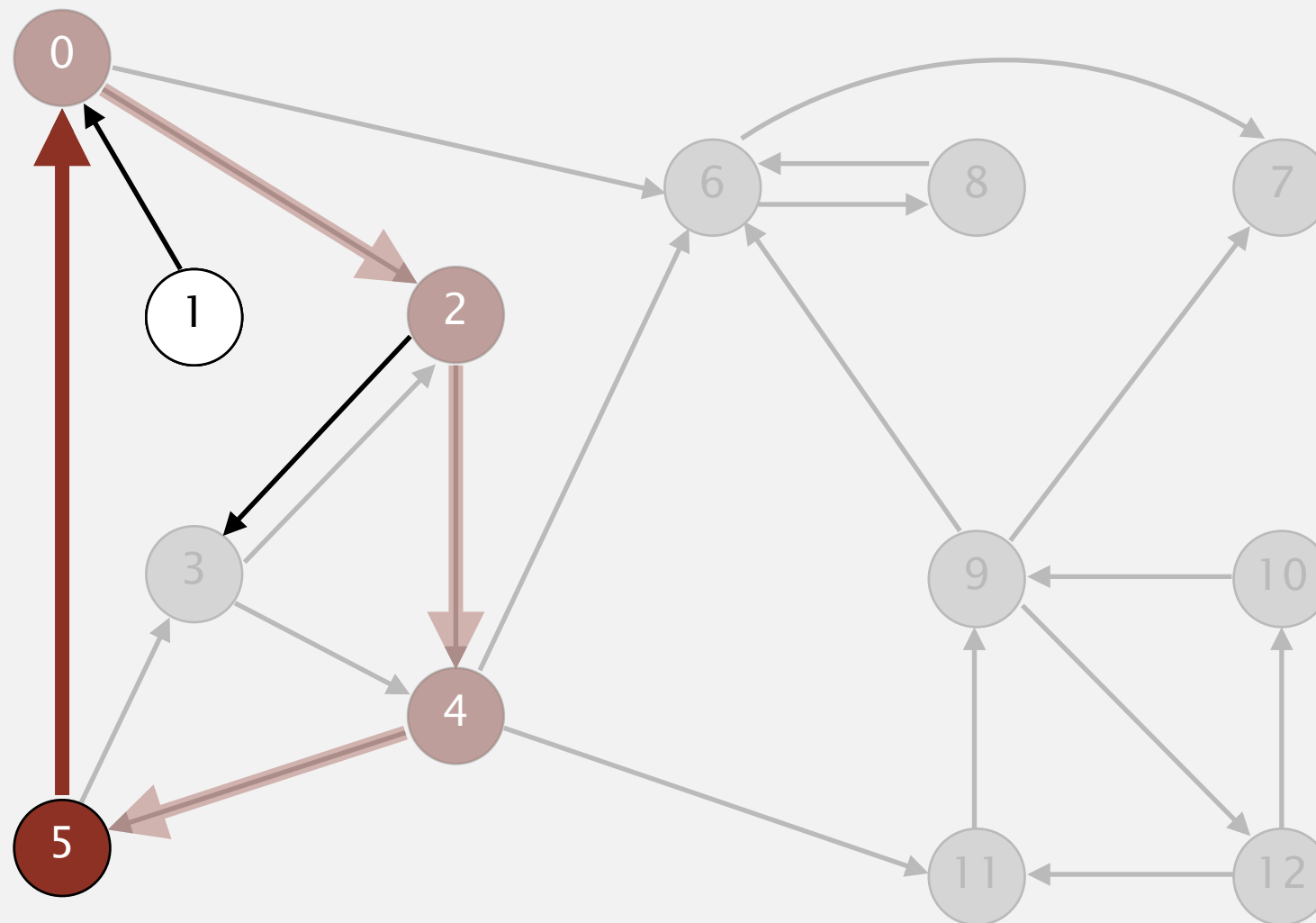
v	marked[]
0	T
1	F
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

**3 done**

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

3 11 9 12 10 6 7 8



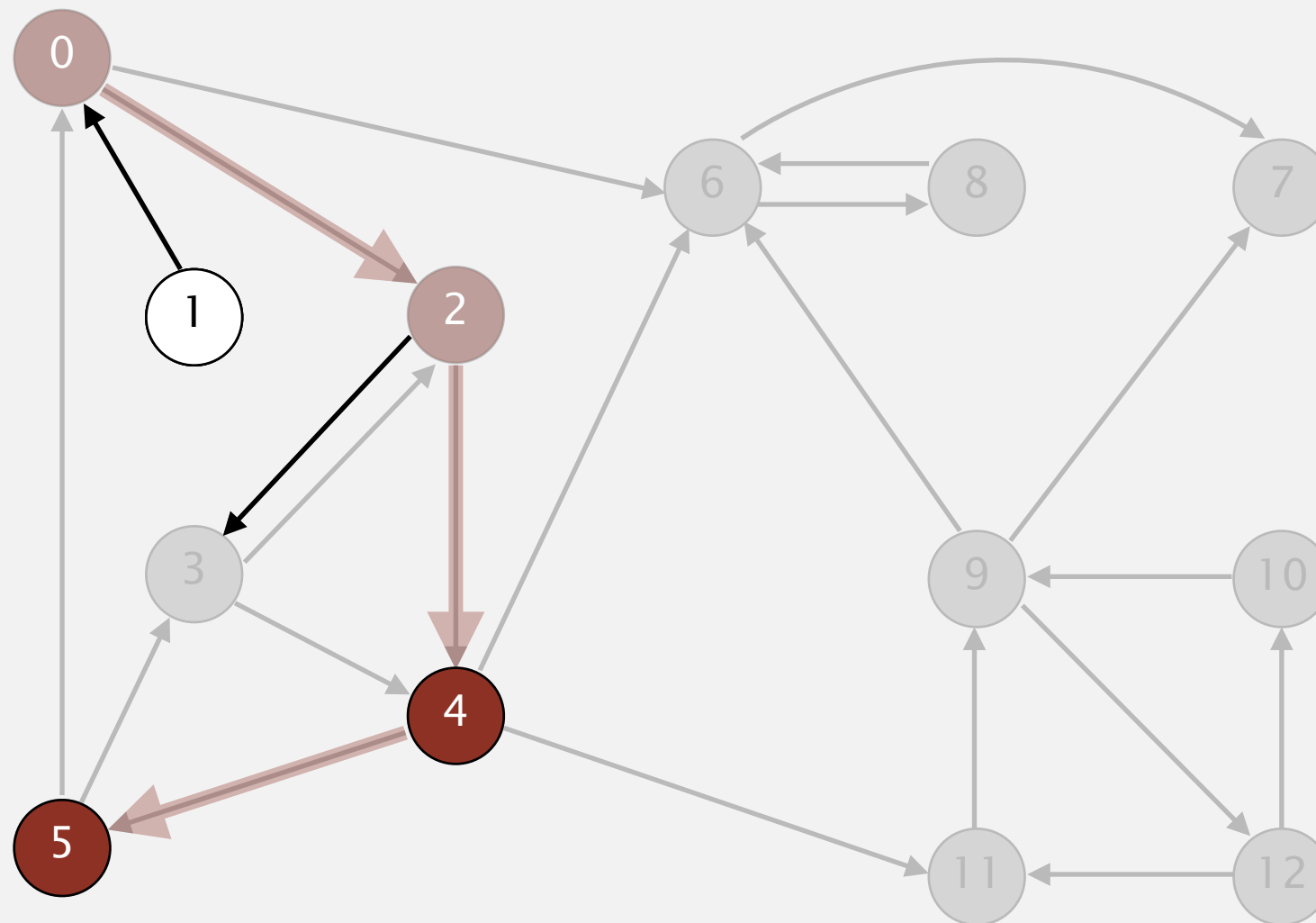
v	marked[]
0	T
1	F
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

## 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



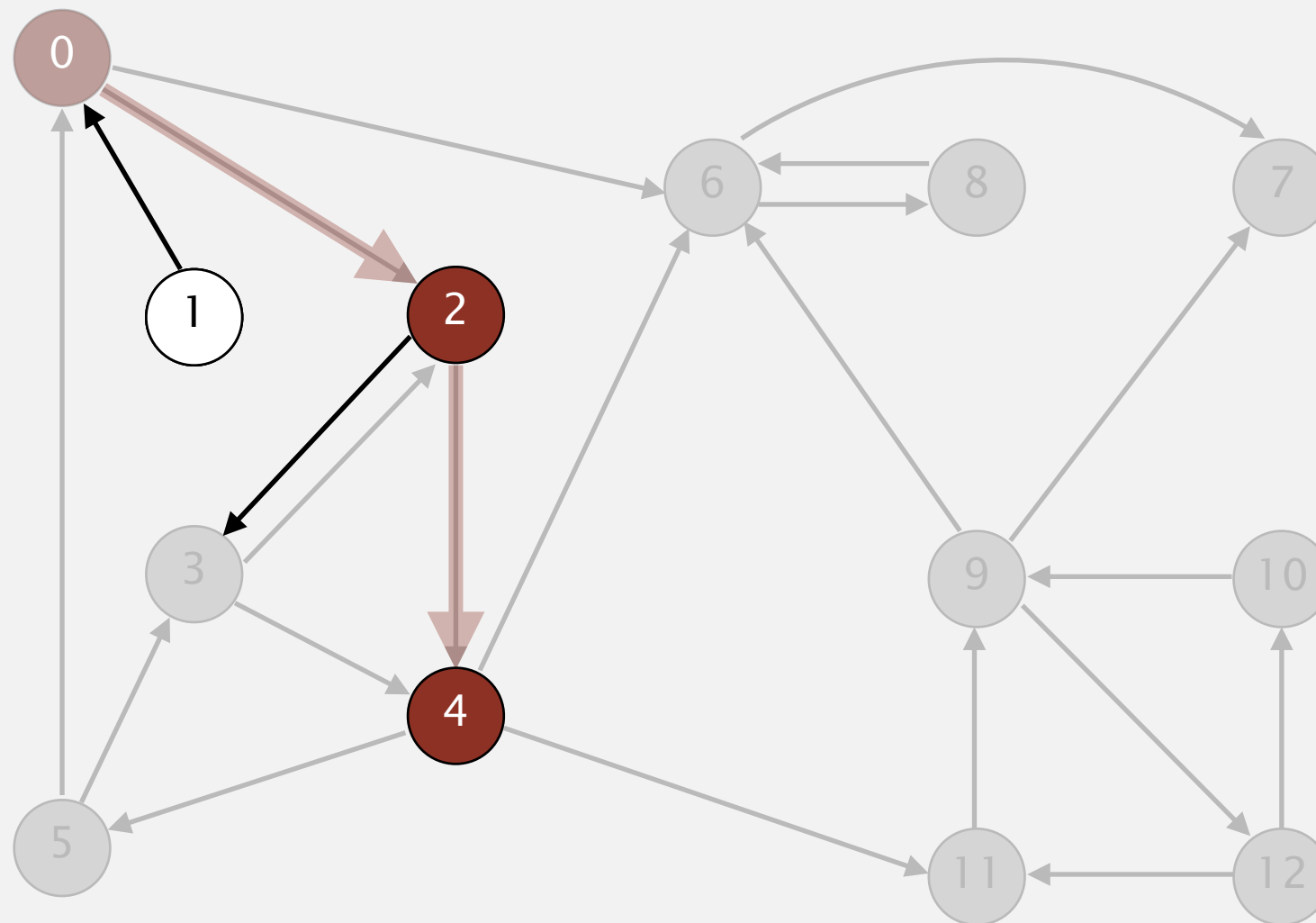
v	marked[]
0	T
1	F
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

**5 done**

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

4 5 3 11 9 12 10 6 7 8



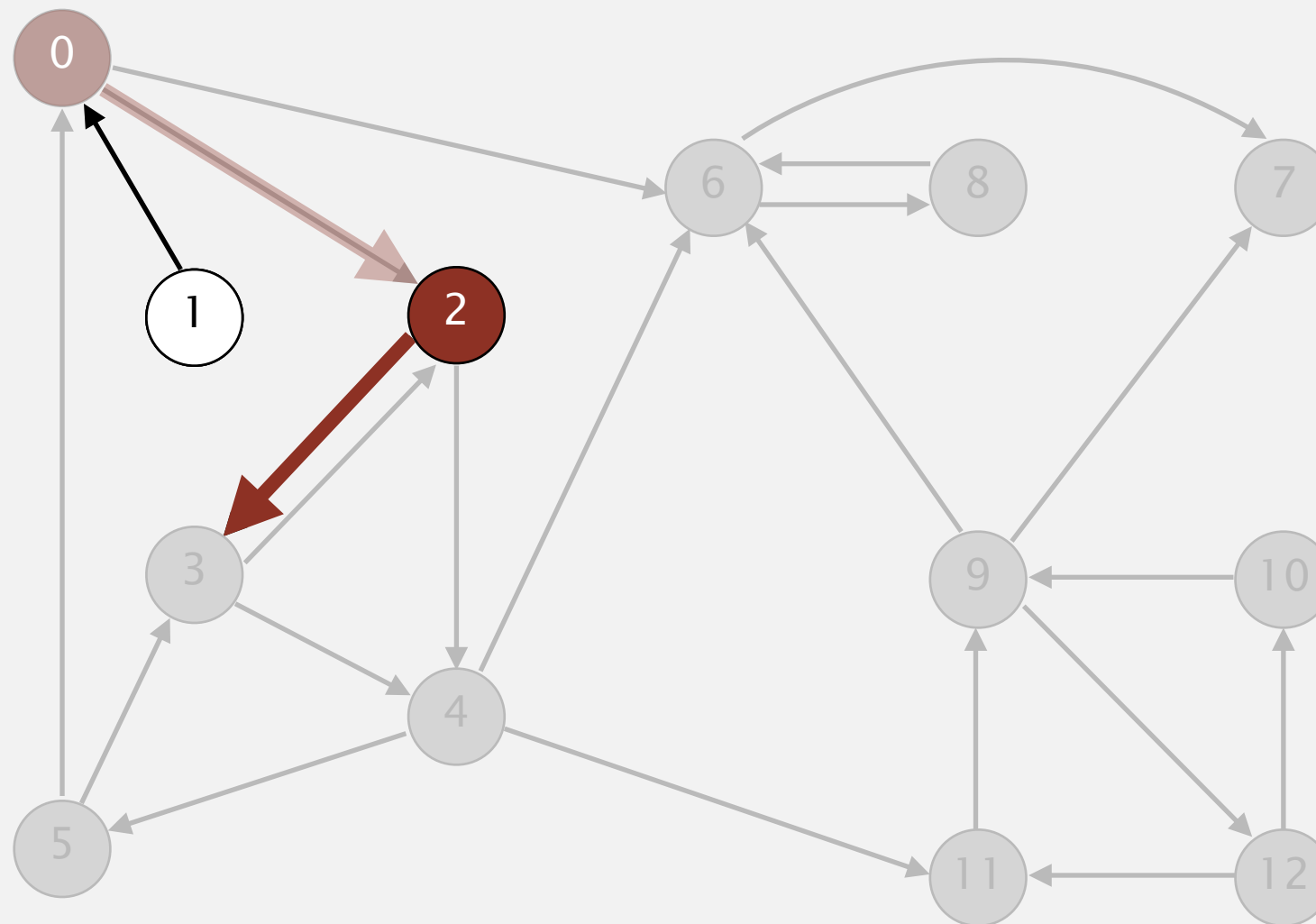
v	marked[]
0	T
1	F
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

4 done

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

4 5 3 11 9 12 10 6 7 8



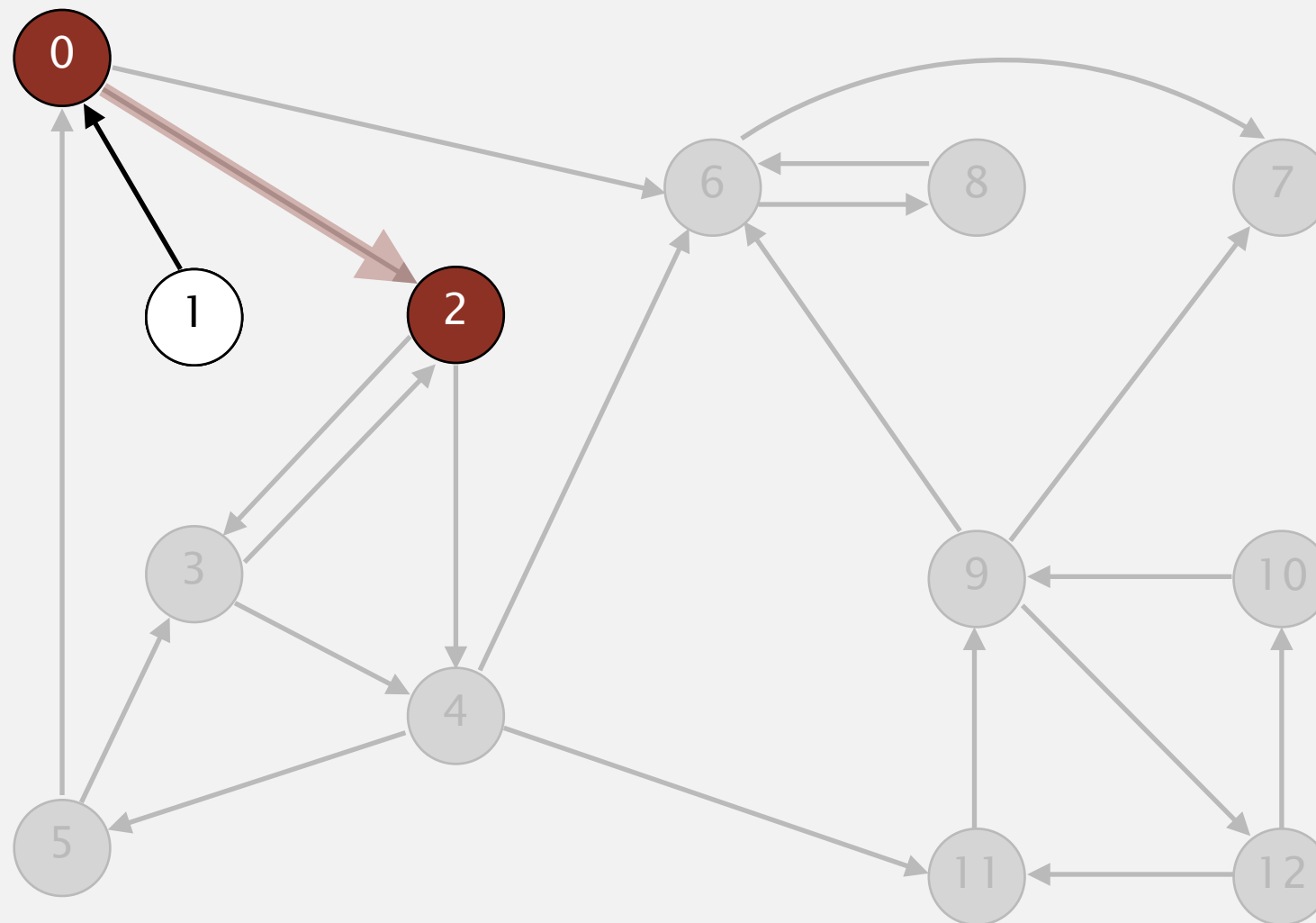
v	marked[]
0	T
1	F
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

visit 2: check 4 and check 3

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

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



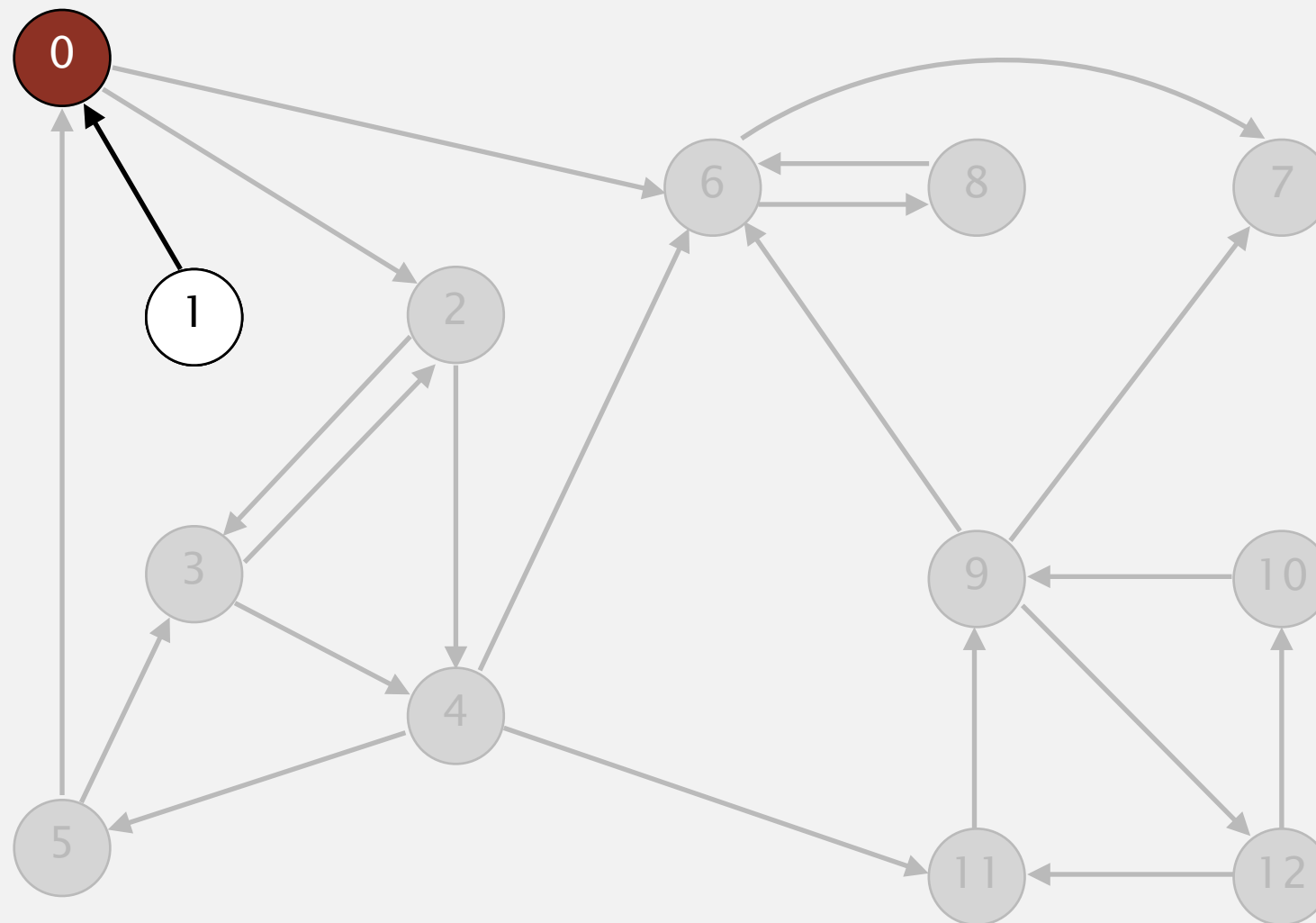
v	marked[]
0	T
1	F
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

**2 done**

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

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



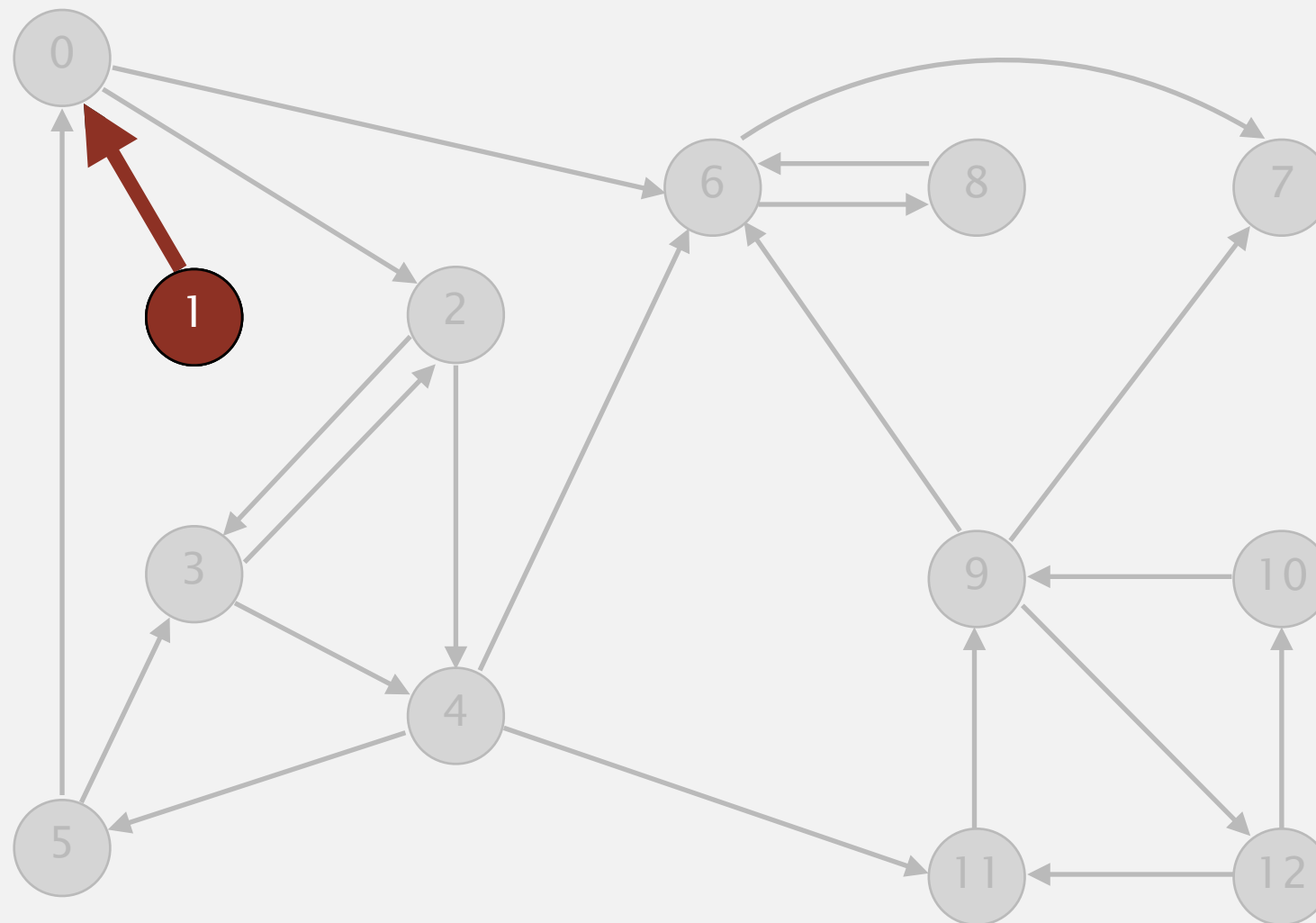
v	marked[]
0	T
1	F
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

**0 done**

# Kosaraju-Sharir algorithm demo

Phase 1. Compute reverse postorder in  $G^R$ .

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



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

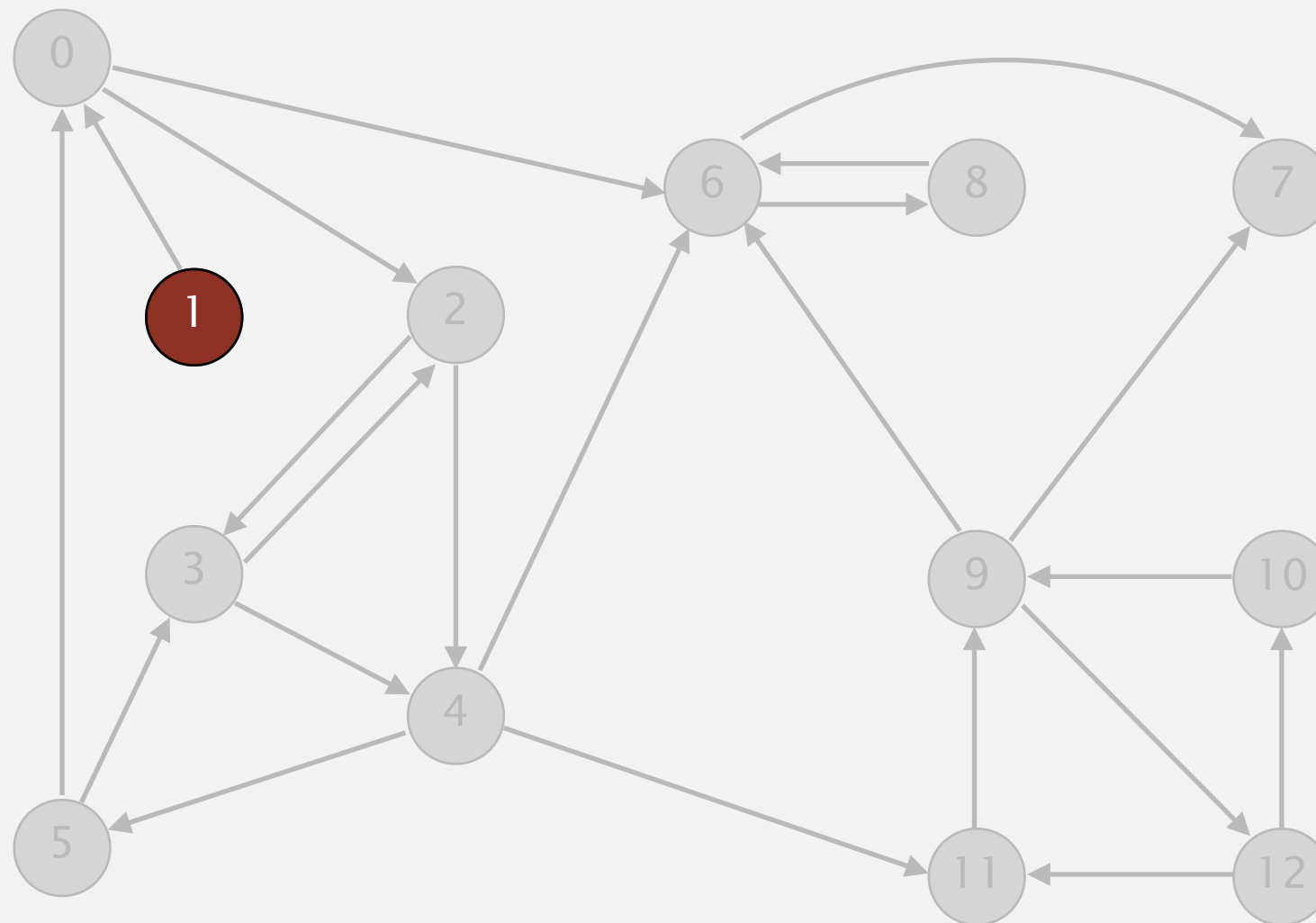
visit 1: check 0



# 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



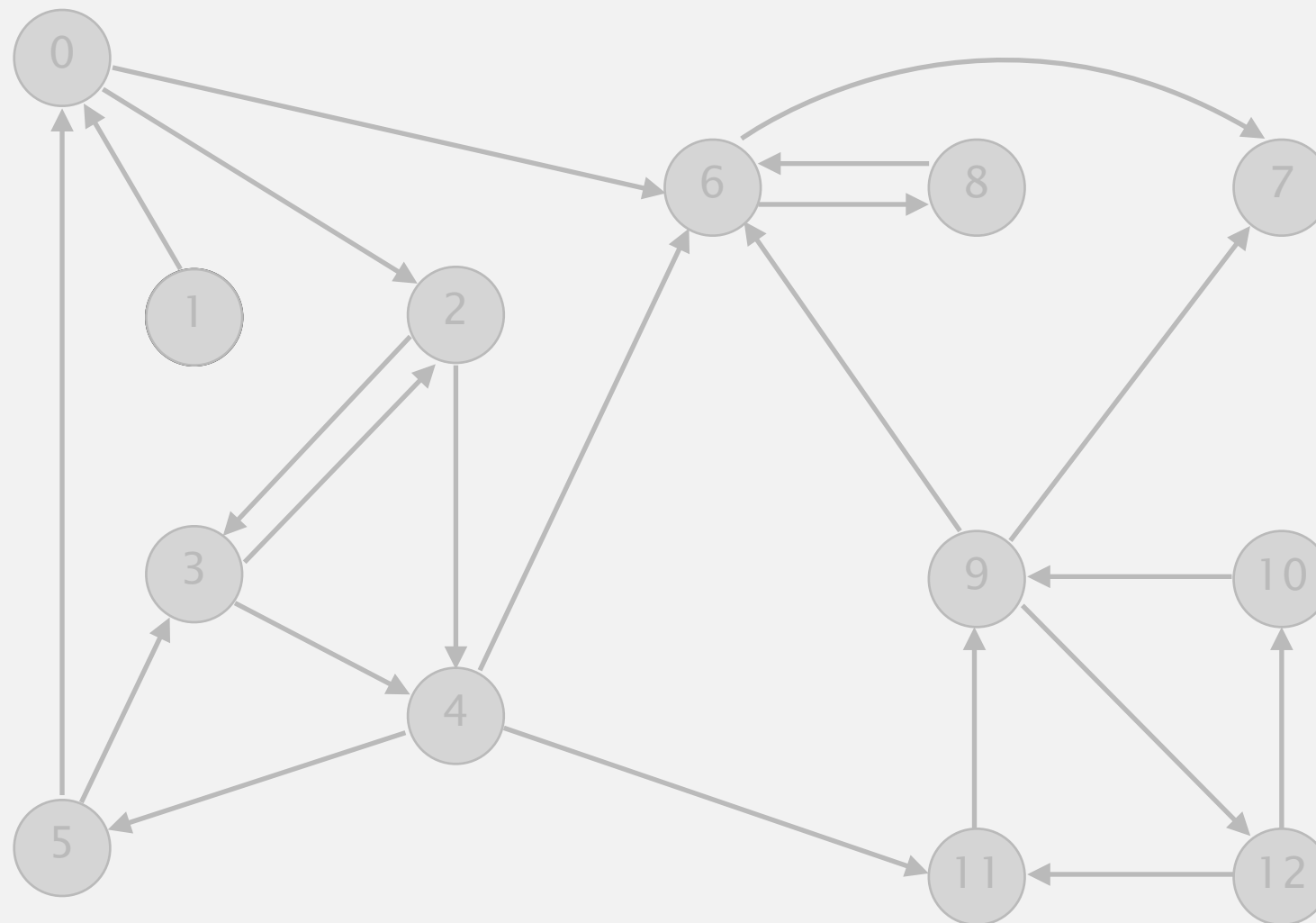
v	marked[]
0	T
1	T
2	T
3	T
4	T
5	T
6	T
7	T
8	T
9	T
10	T
11	T
12	T

**1 done**

# 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



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

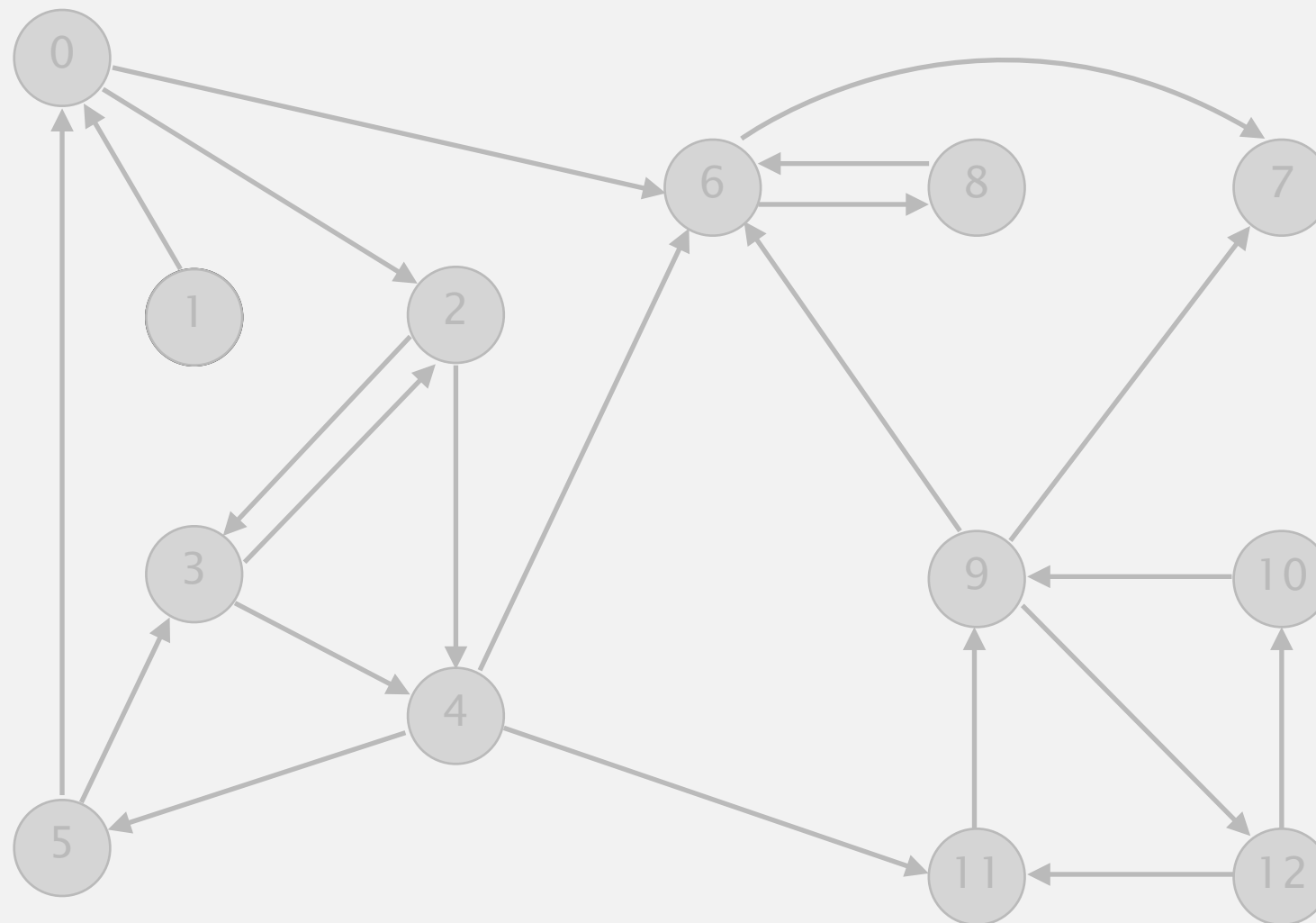
check 2 3 4 5 6 7 8 9 10 11 12

# 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  $G^R$**



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

## 4.2 KOSARAJU-SHARIR DEMO

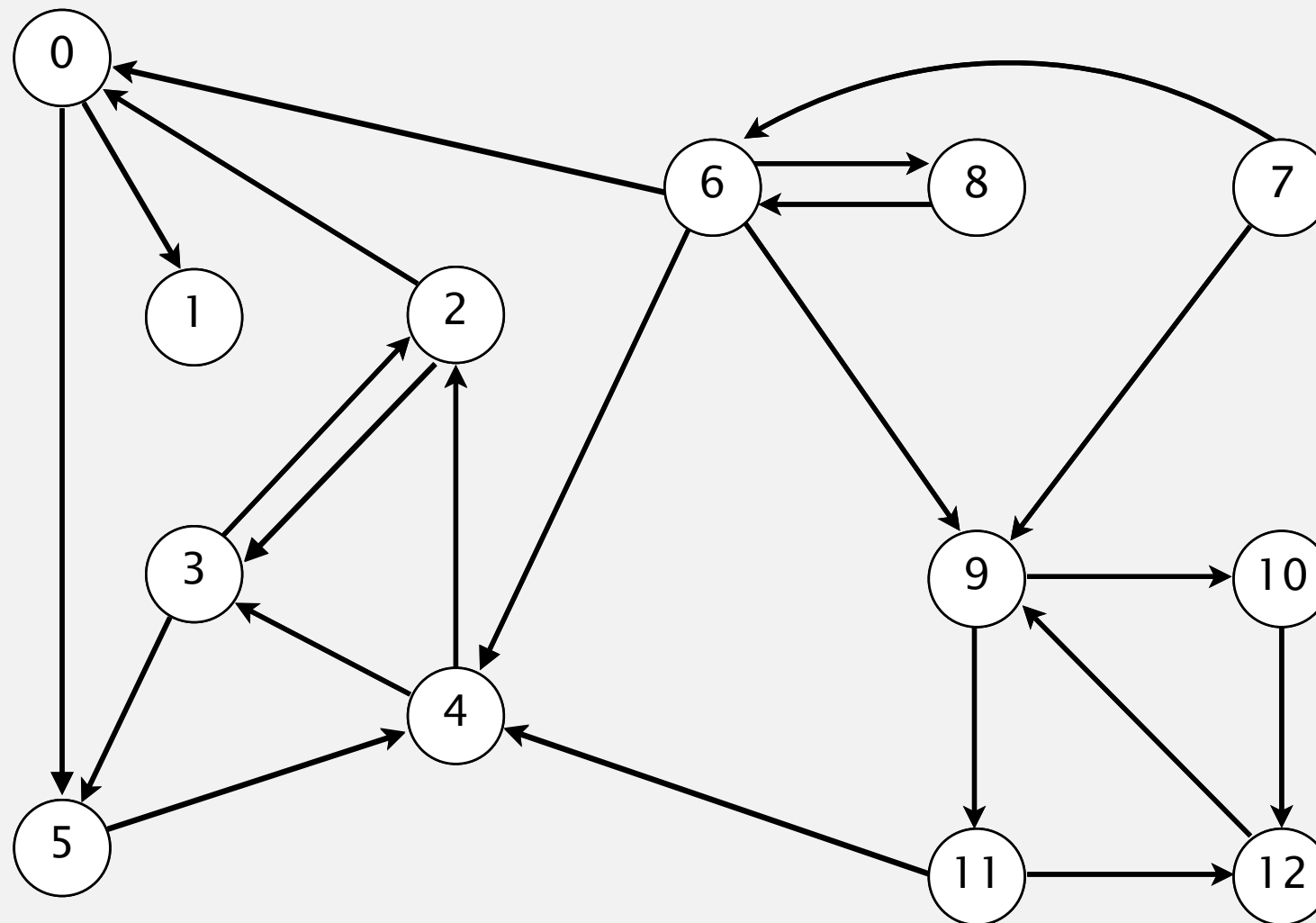
---

- *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



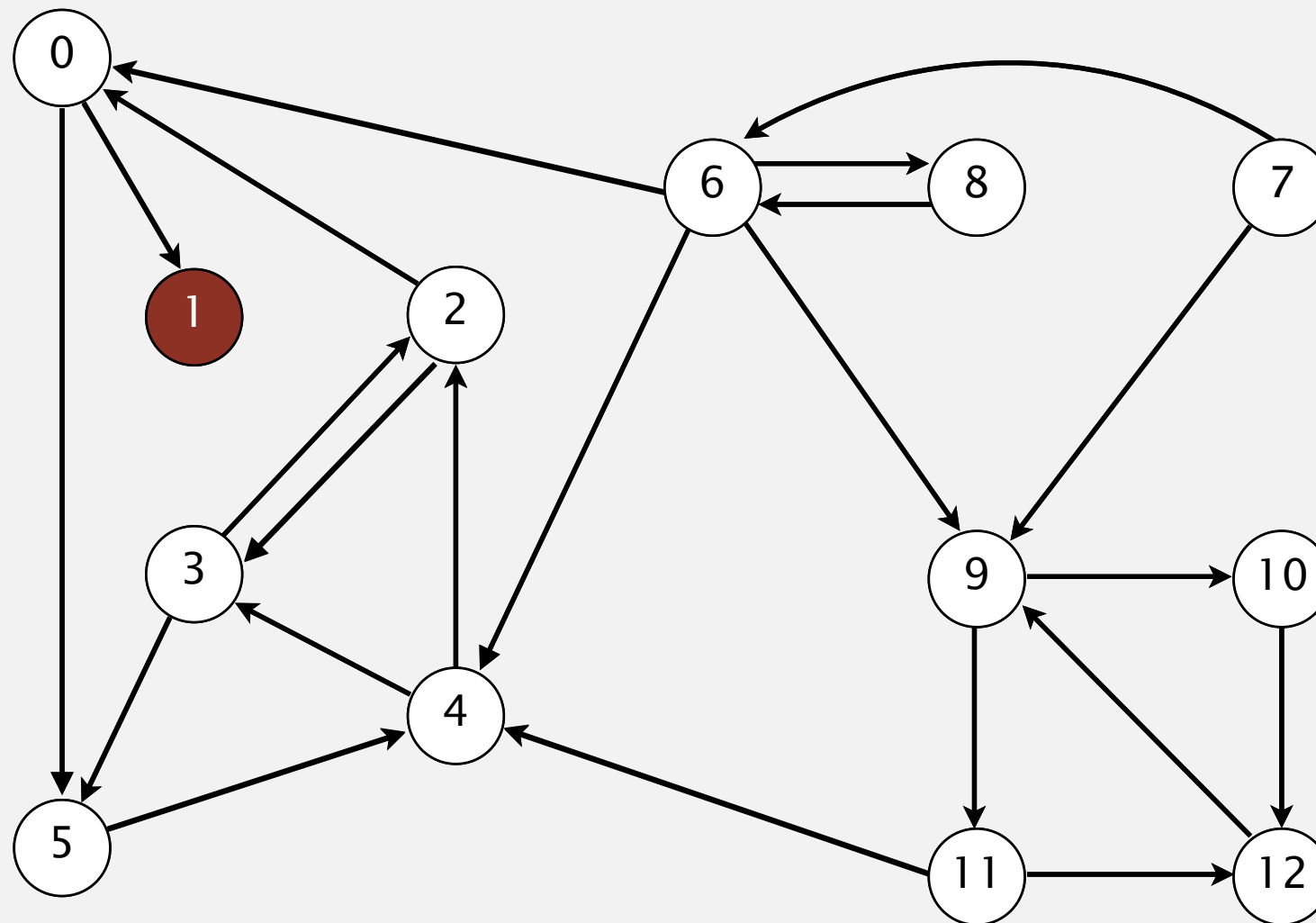
v	id[]
0	—
1	—
2	—
3	—
4	—
5	—
6	—
7	—
8	—
9	—
10	—
11	—
12	—

original digraph G

# 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



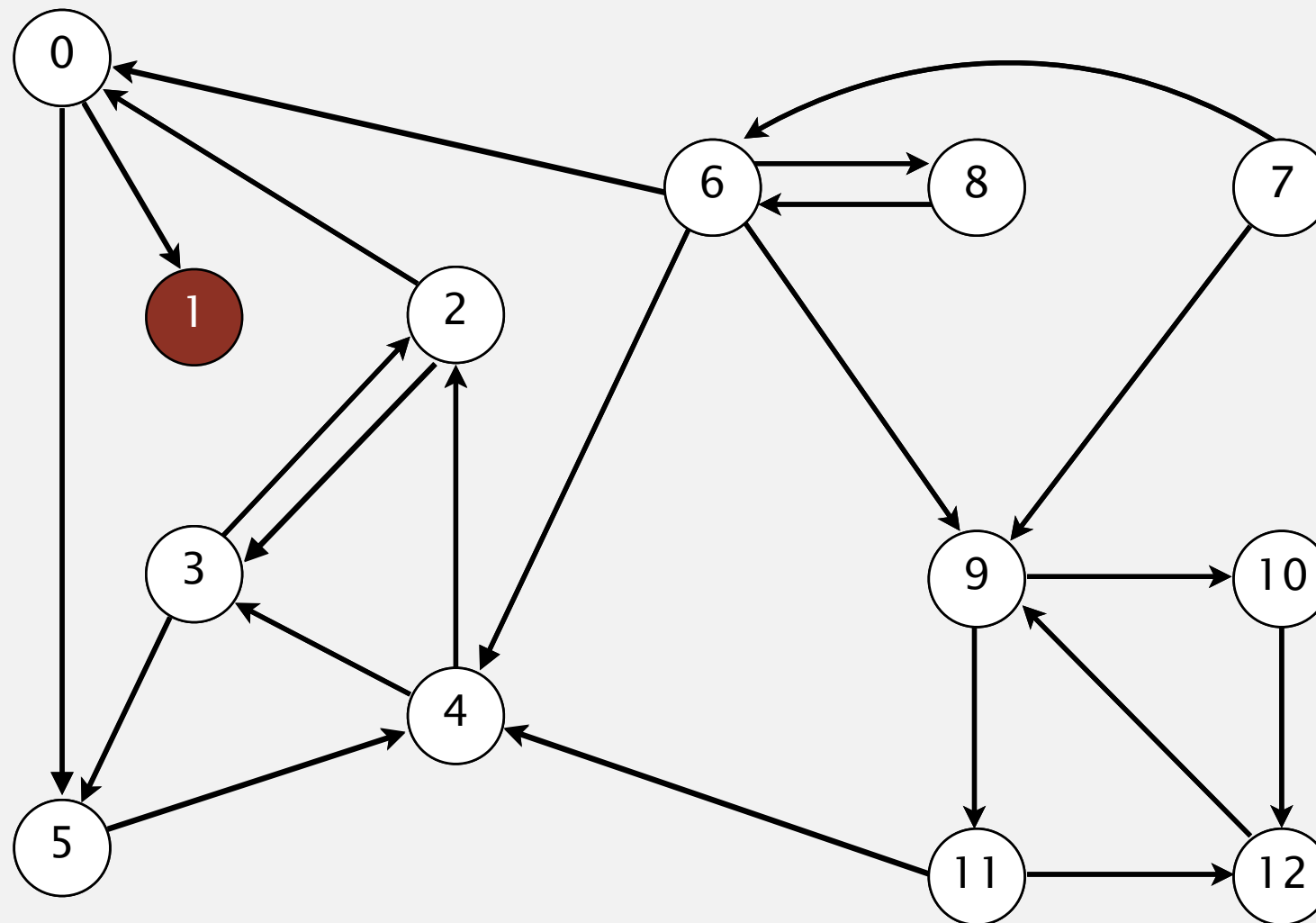
v	id[]
0	—
1	<b>0</b>
2	—
3	—
4	—
5	—
6	—
7	—
8	—
9	—
10	—
11	—
12	—

**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



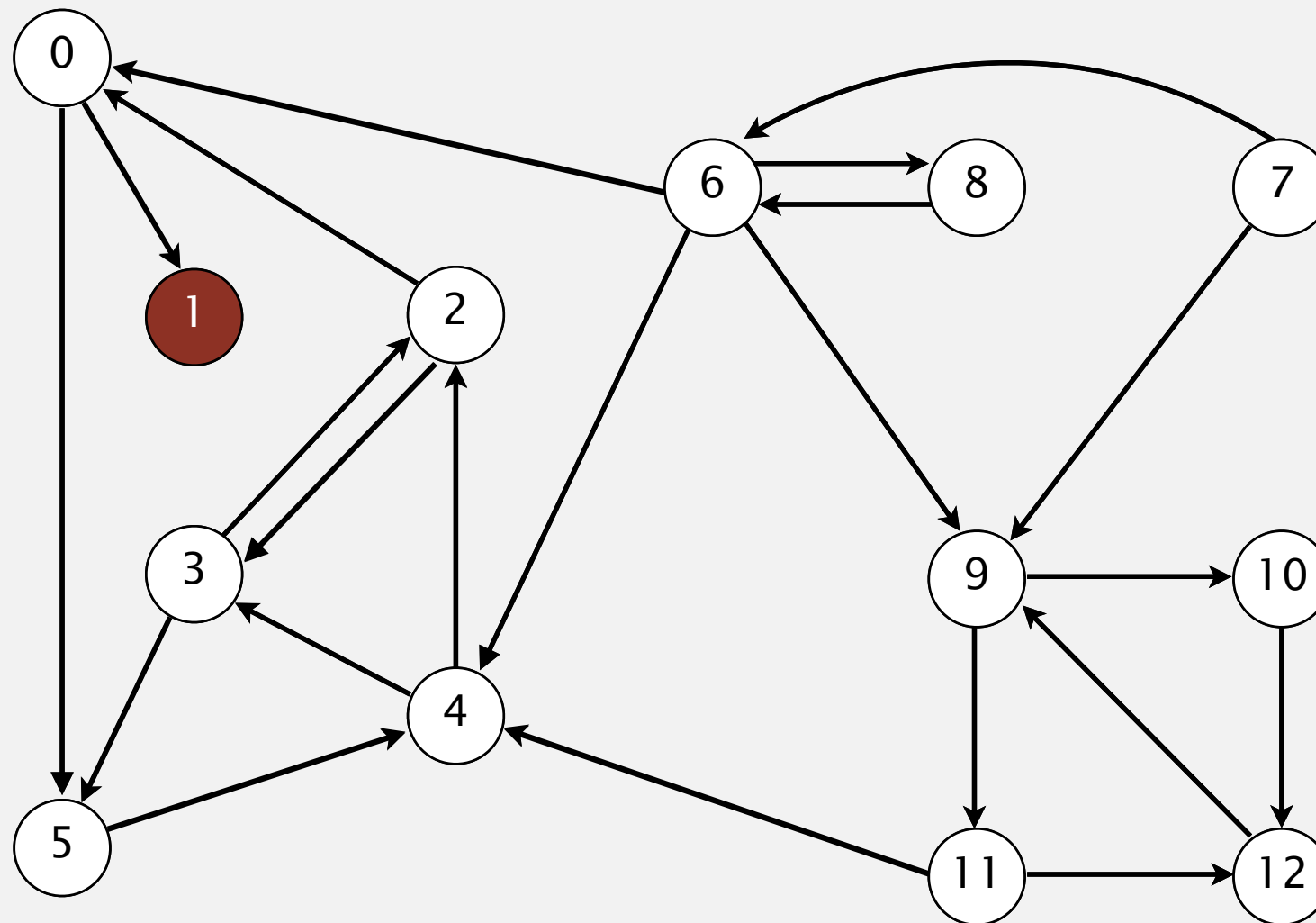
v	id[]
0	—
1	<b>0</b>
2	—
3	—
4	—
5	—
6	—
7	—
8	—
9	—
10	—
11	—
12	—

**1 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	<b>0</b>
2	—
3	—
4	—
5	—
6	—
7	—
8	—
9	—
10	—
11	—
12	—

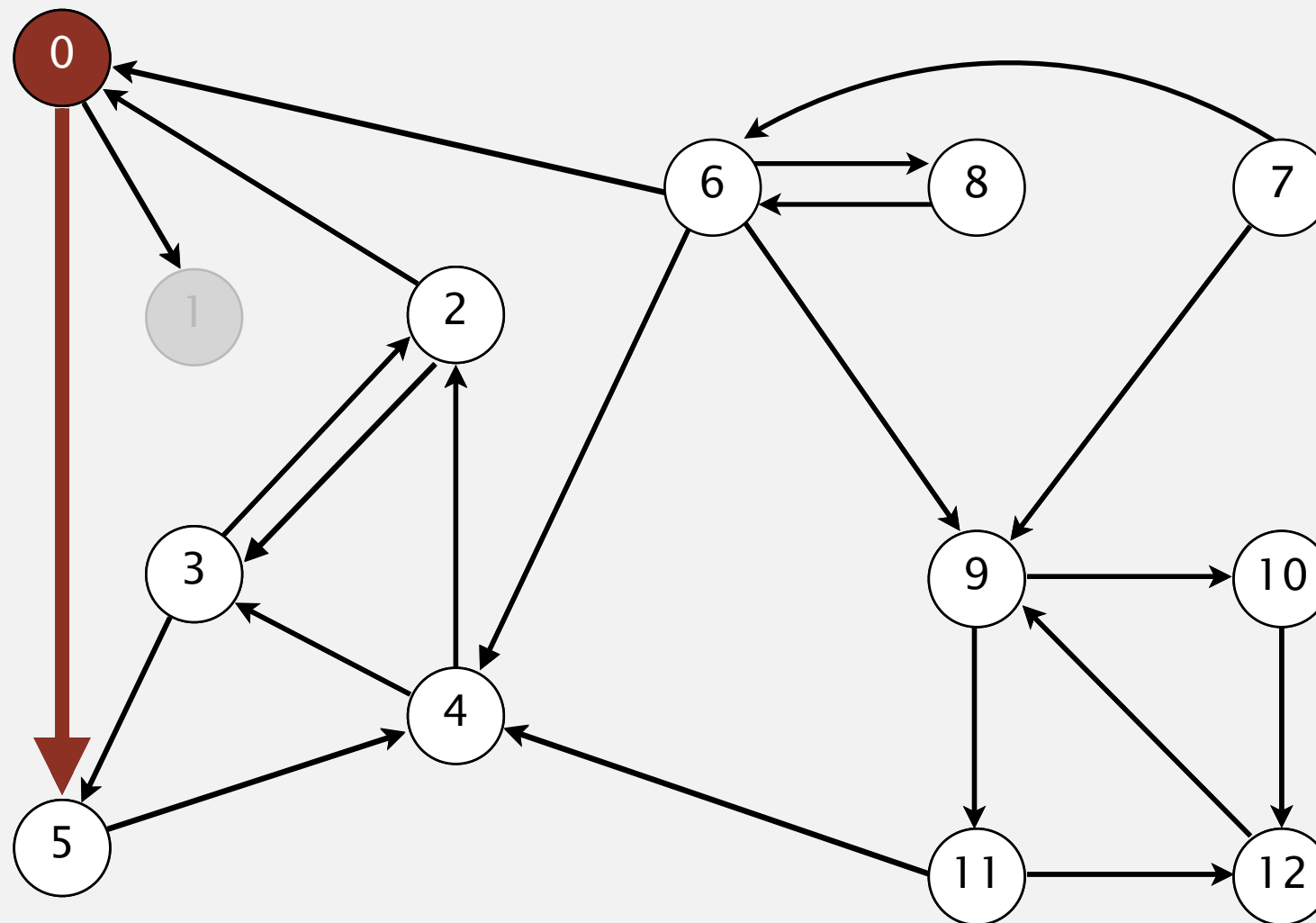
**strong component: 1**



# 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



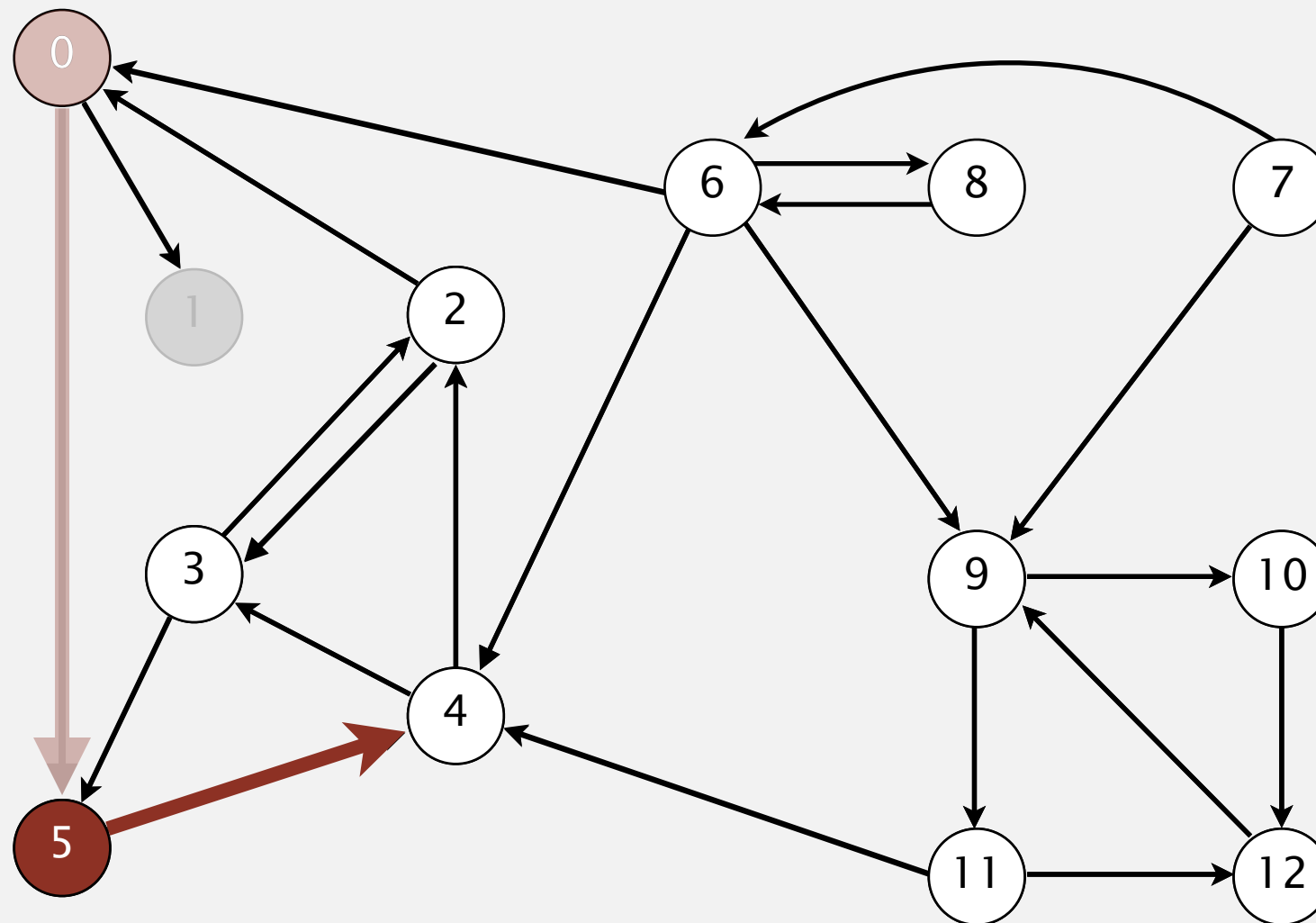
v	id[]
0	1
1	0
2	—
3	—
4	—
5	—
6	—
7	—
8	—
9	—
10	—
11	—
12	—

**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



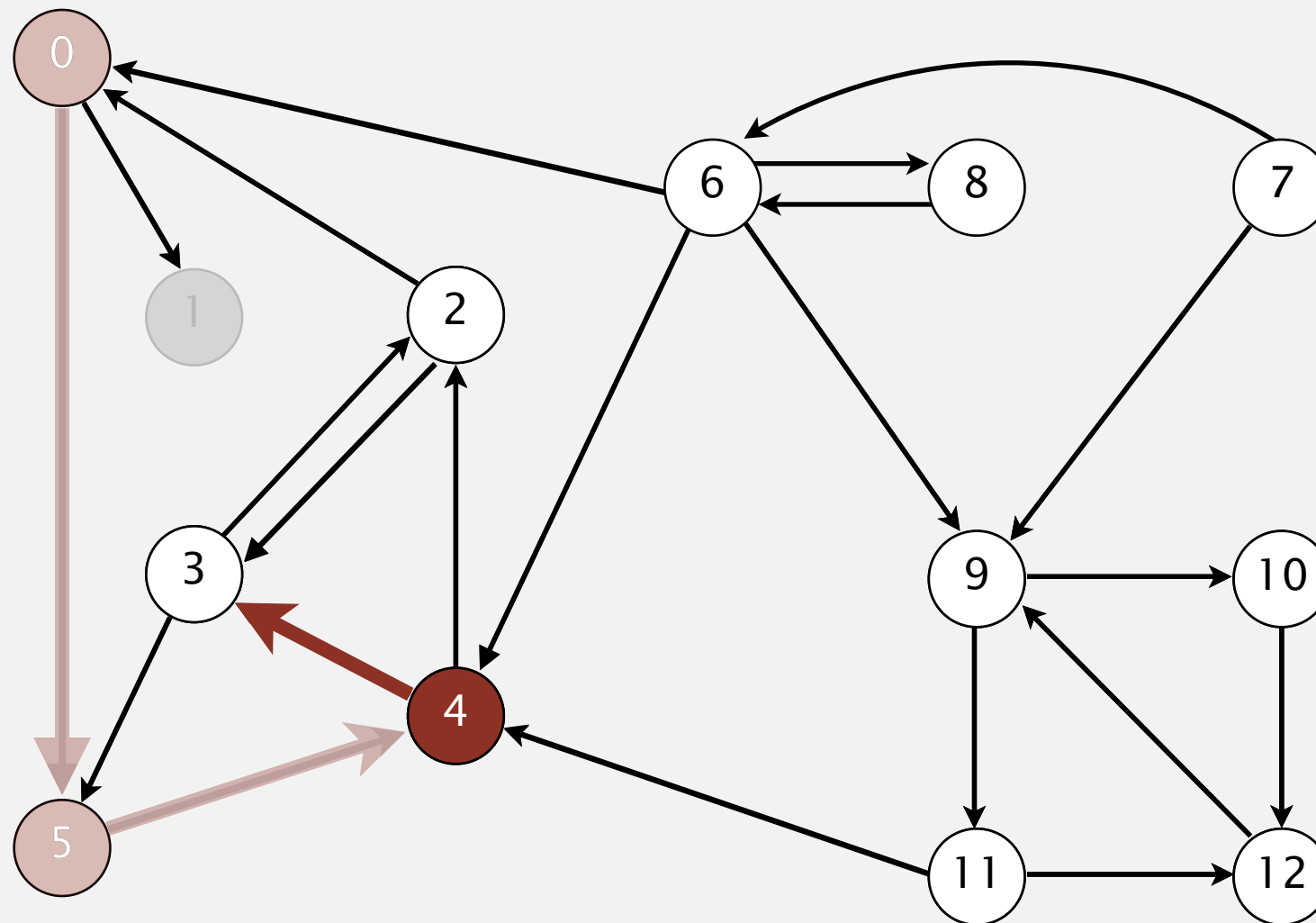
v	id[]
0	1
1	0
2	—
3	—
4	—
5	1
6	—
7	—
8	—
9	—
10	—
11	—
12	—

visit 5: 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



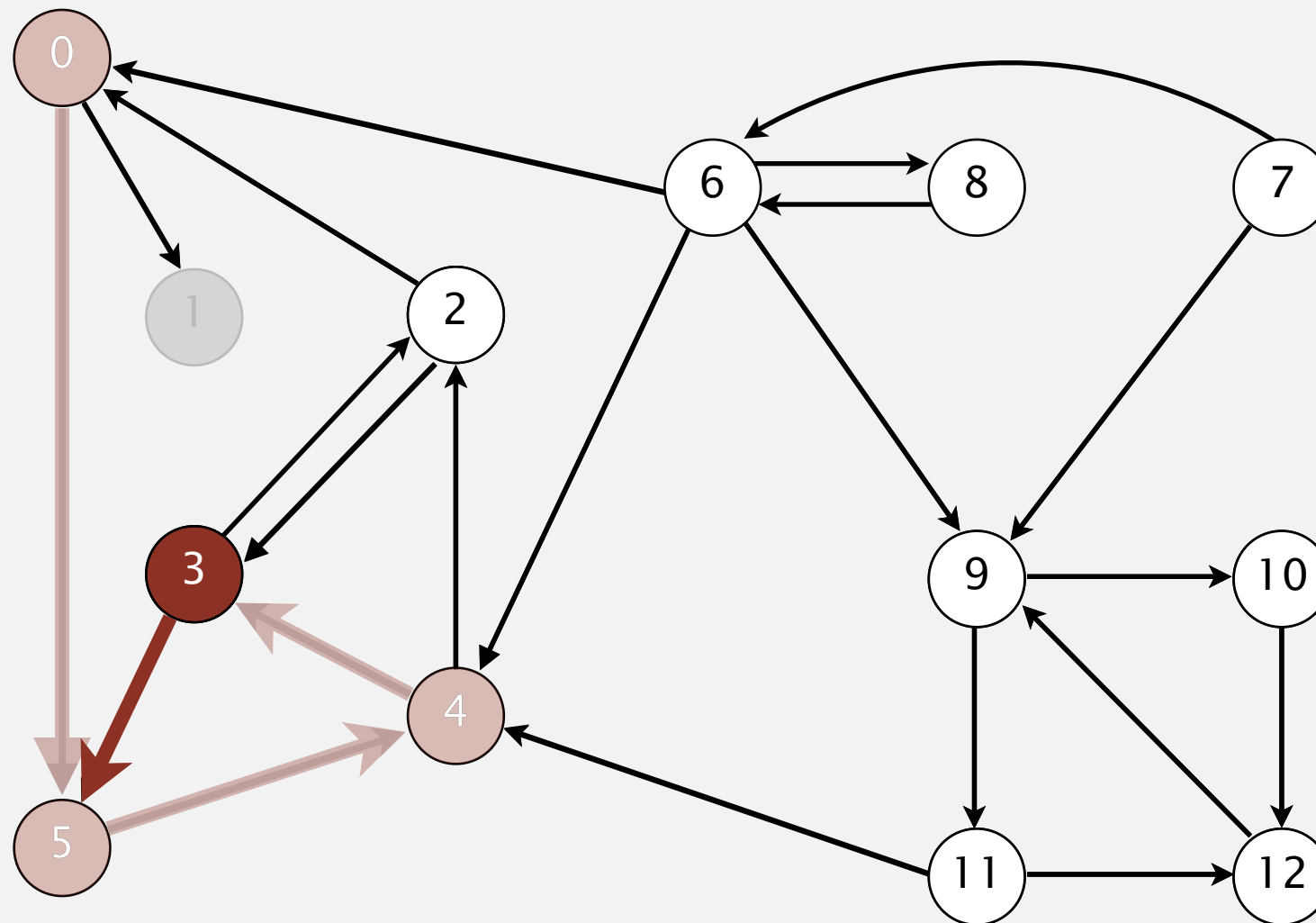
v	id[]
0	1
1	0
2	-
3	-
4	1
5	1
6	-
7	-
8	-
9	-
10	-
11	-
12	-

**visit 4: check 3 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



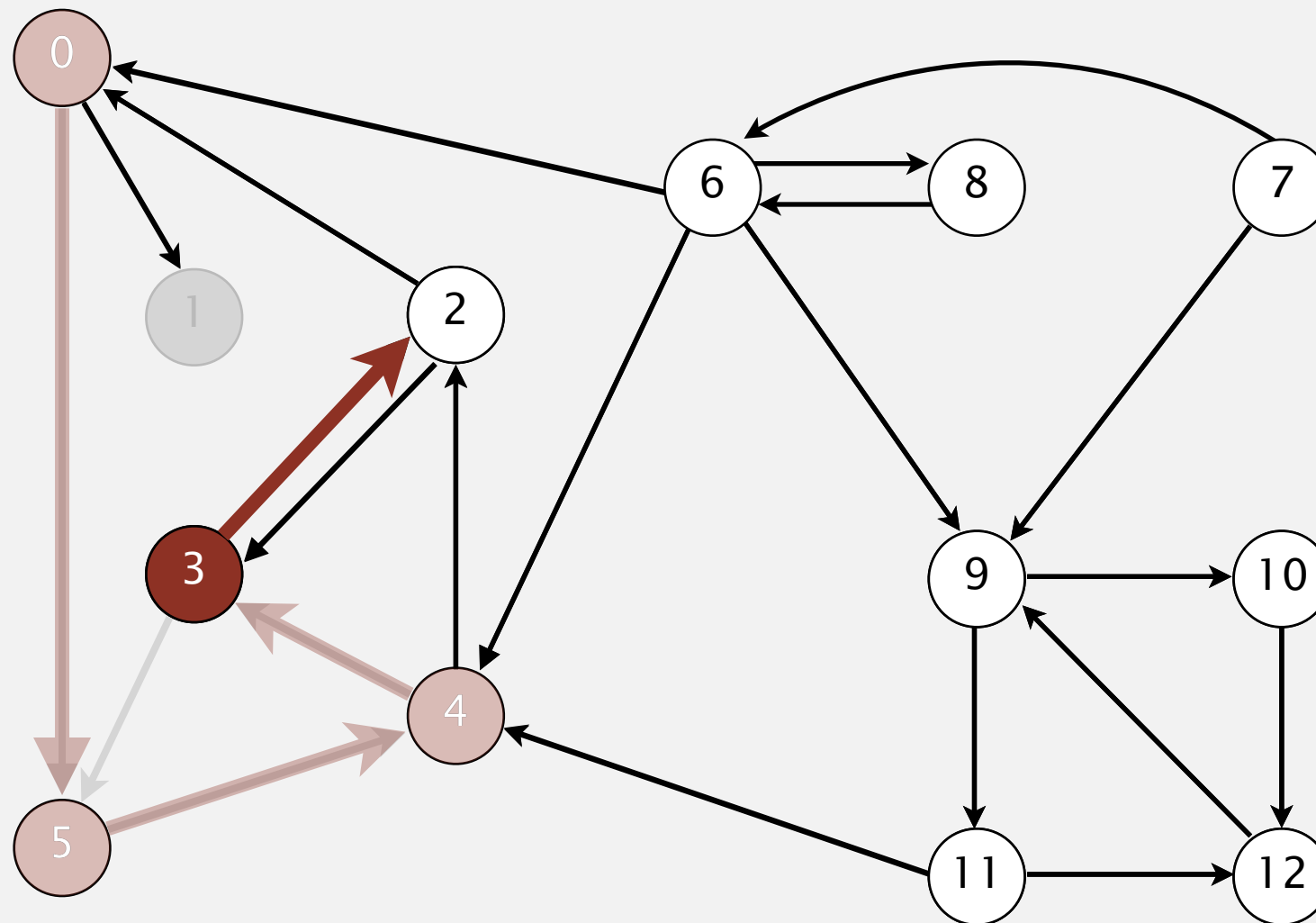
v	id[]
0	1
1	0
2	-
3	1
4	1
5	1
6	-
7	-
8	-
9	-
10	-
11	-
12	-

**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



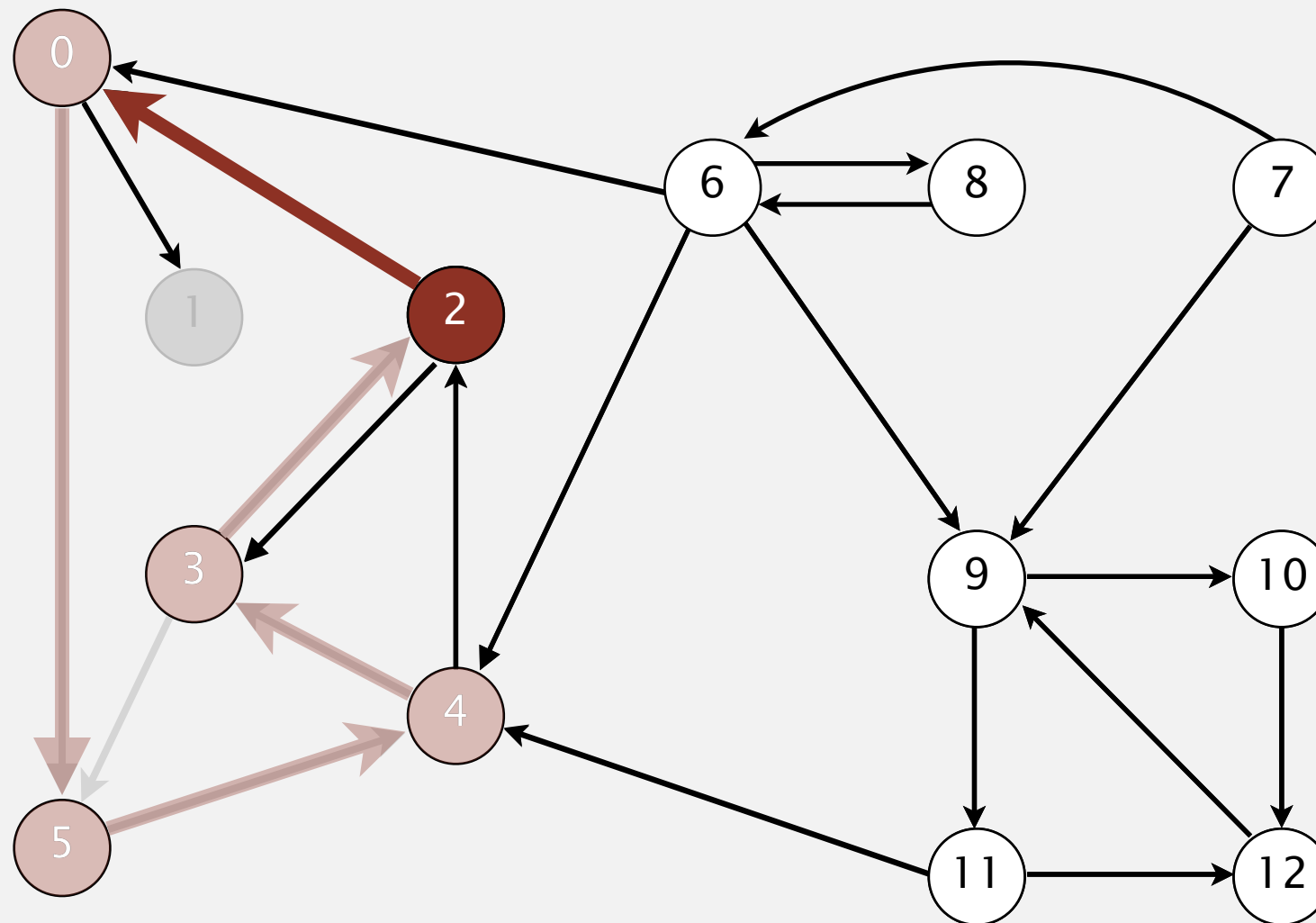
v	id[]
0	1
1	0
2	—
3	1
4	1
5	1
6	—
7	—
8	—
9	—
10	—
11	—
12	—

**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



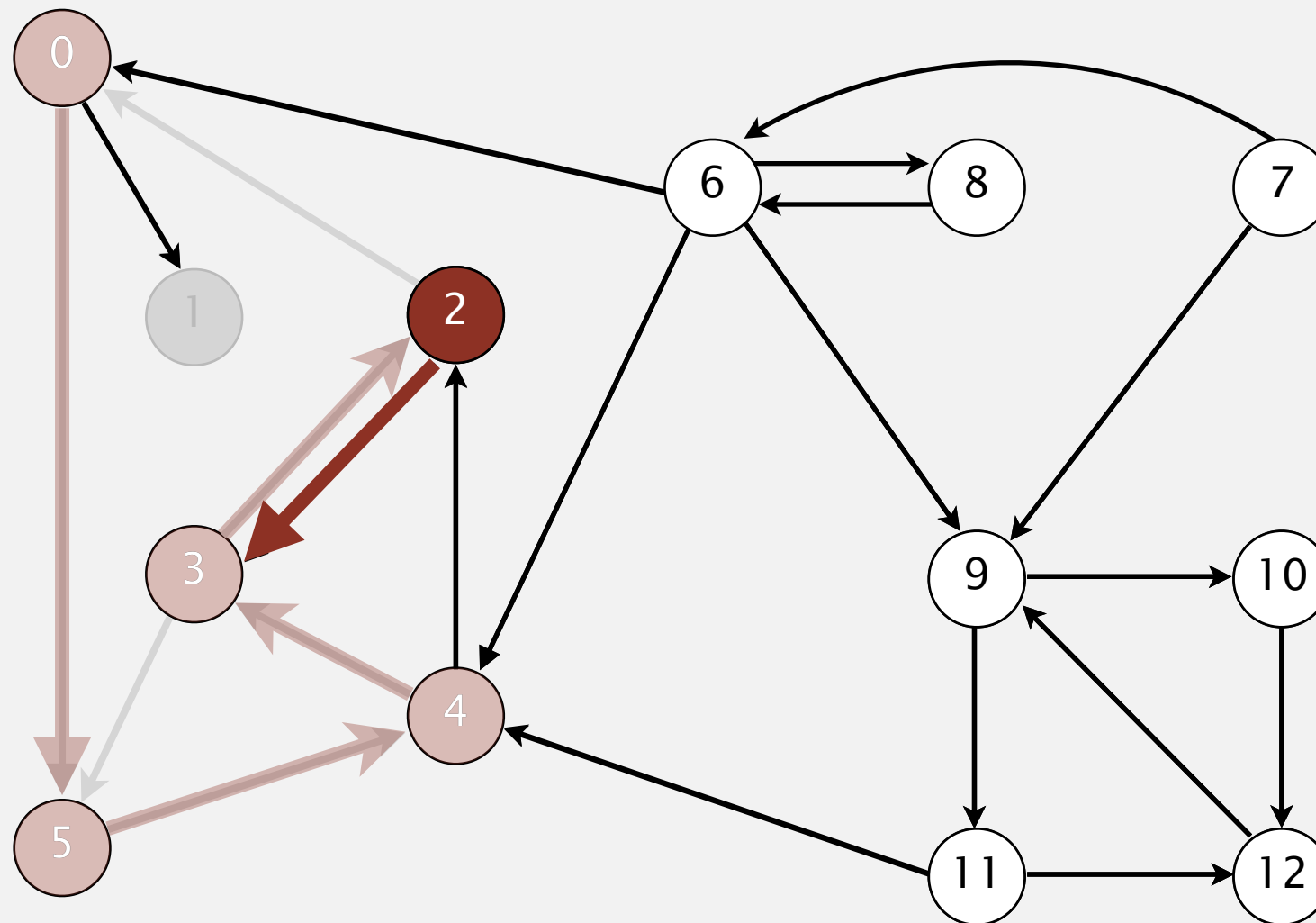
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	—
7	—
8	—
9	—
10	—
11	—
12	—

**visit 2: check 0** and check 3

# 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



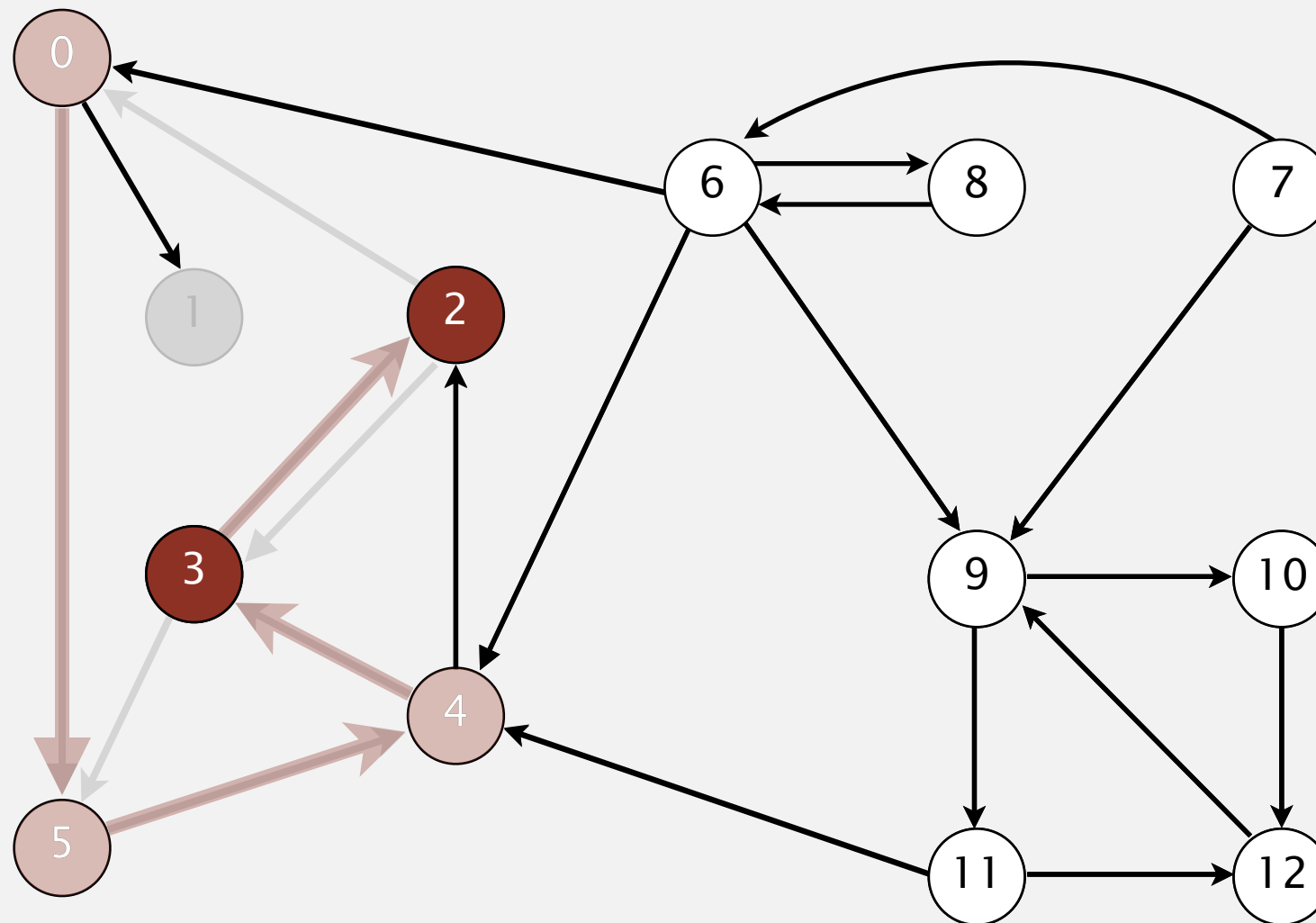
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	—
7	—
8	—
9	—
10	—
11	—
12	—

**visit 2: check 0 and check 3**

# 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	—

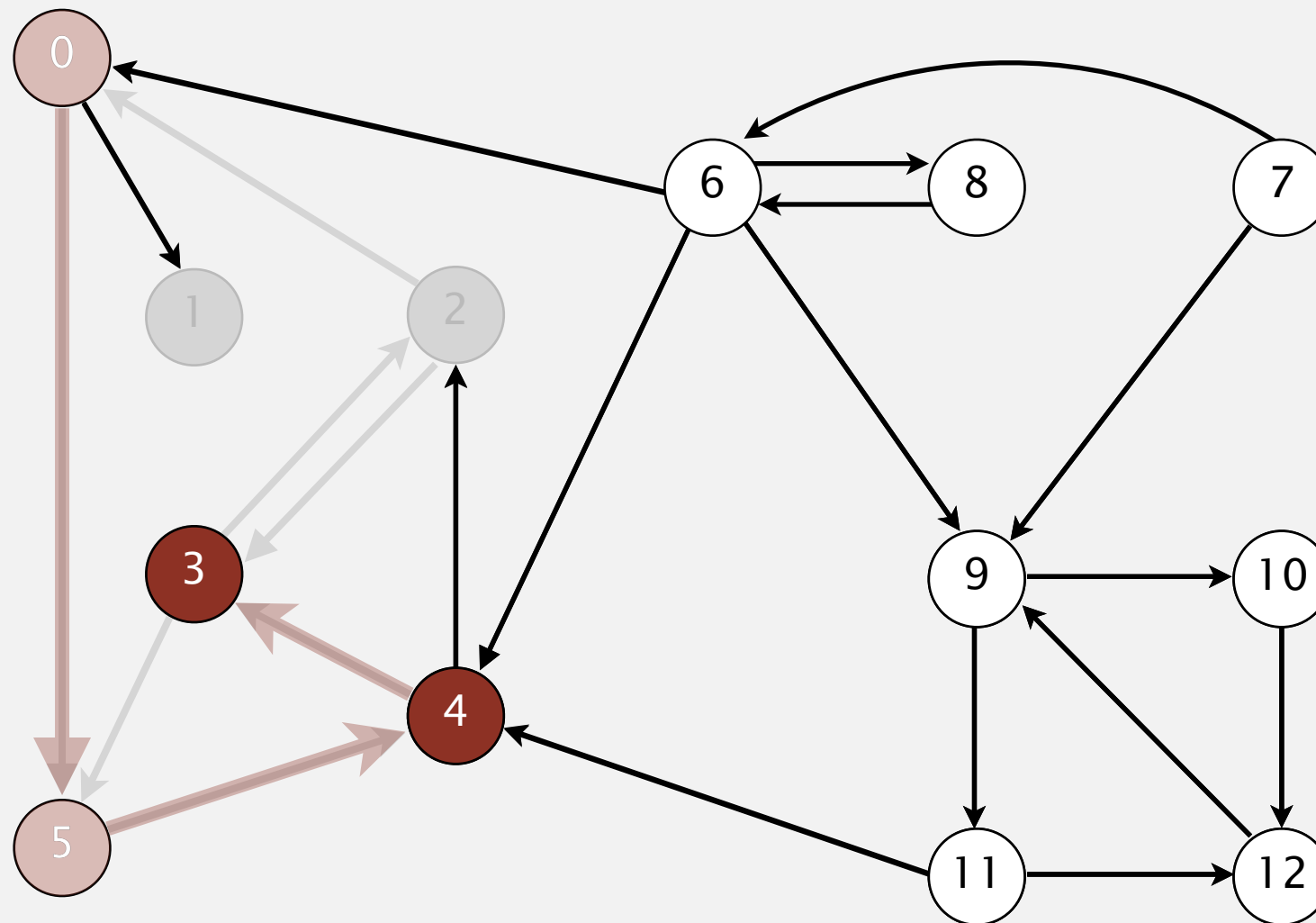
**2 done**



# 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



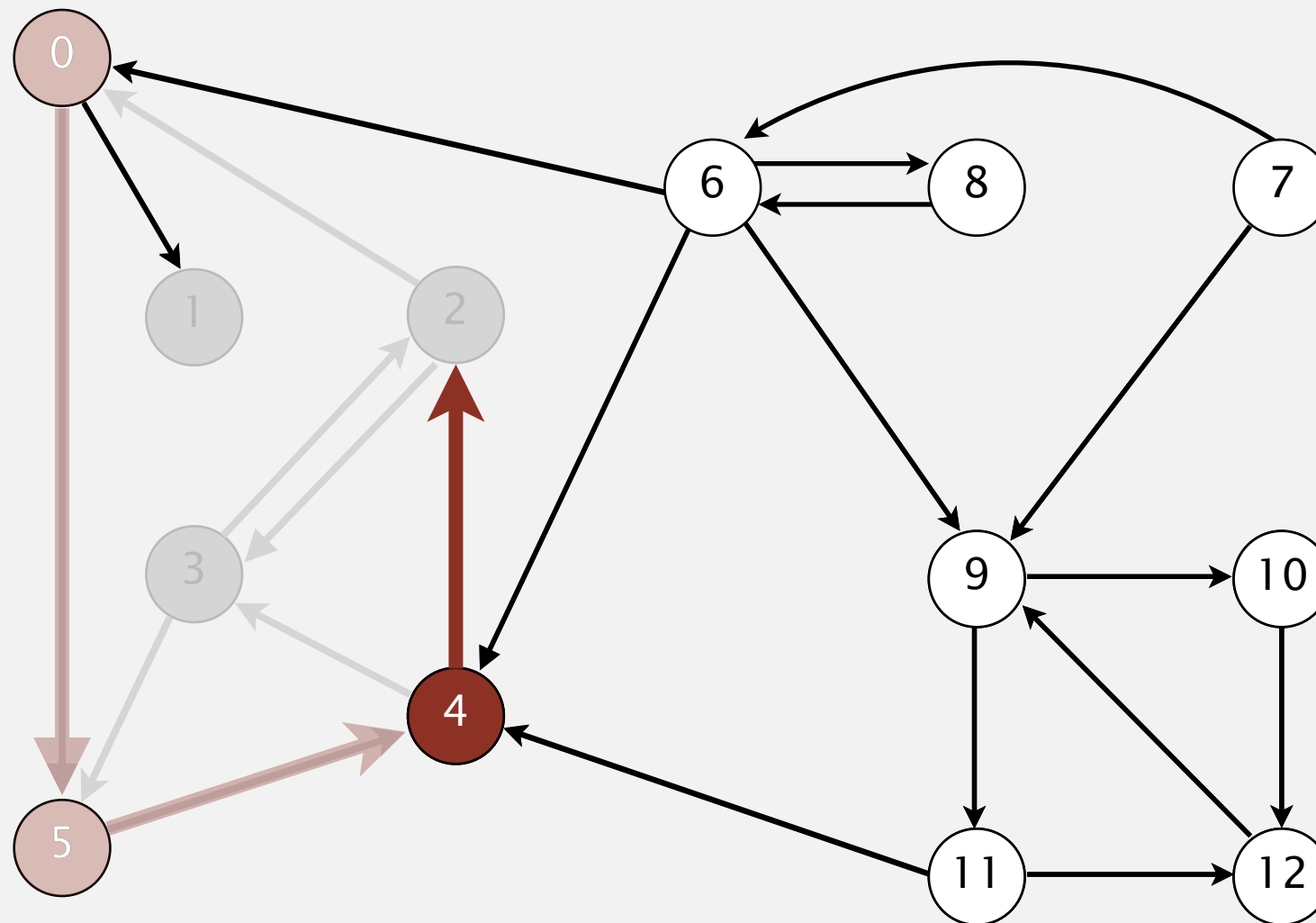
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	—
7	—
8	—
9	—
10	—
11	—
12	—

**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



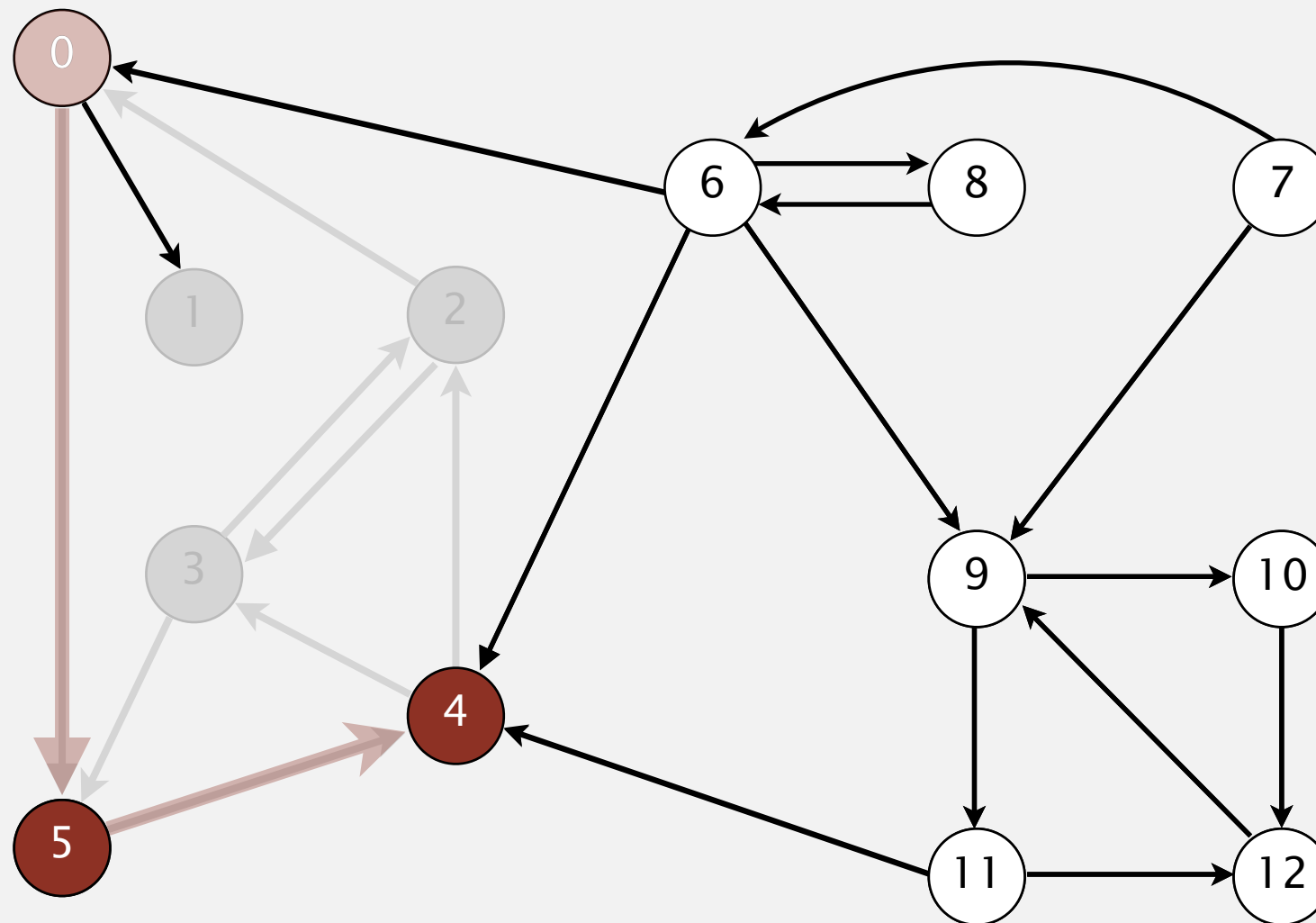
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	—
7	—
8	—
9	—
10	—
11	—
12	—

**visit 4:** check 3 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



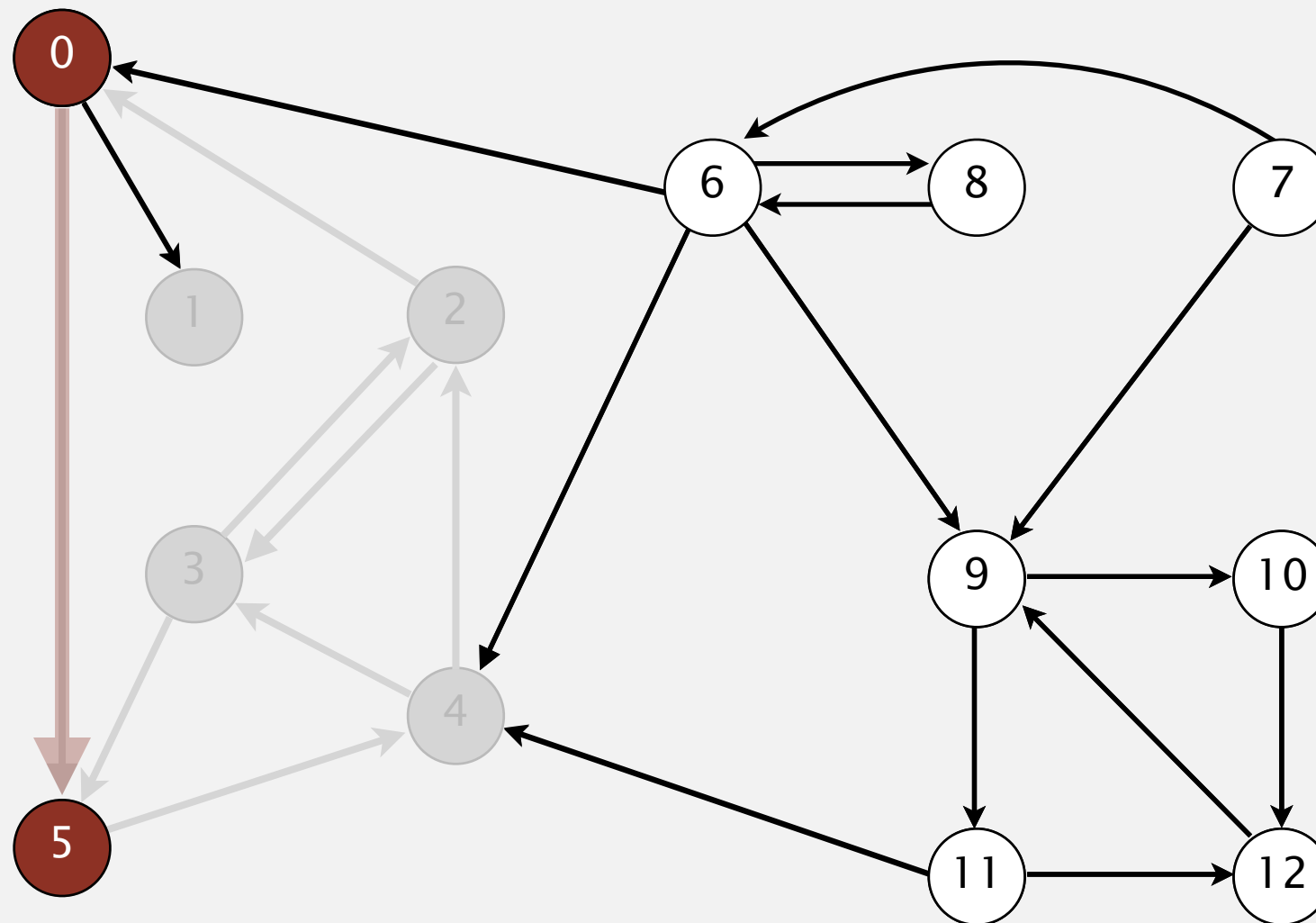
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	—
7	—
8	—
9	—
10	—
11	—
12	—

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



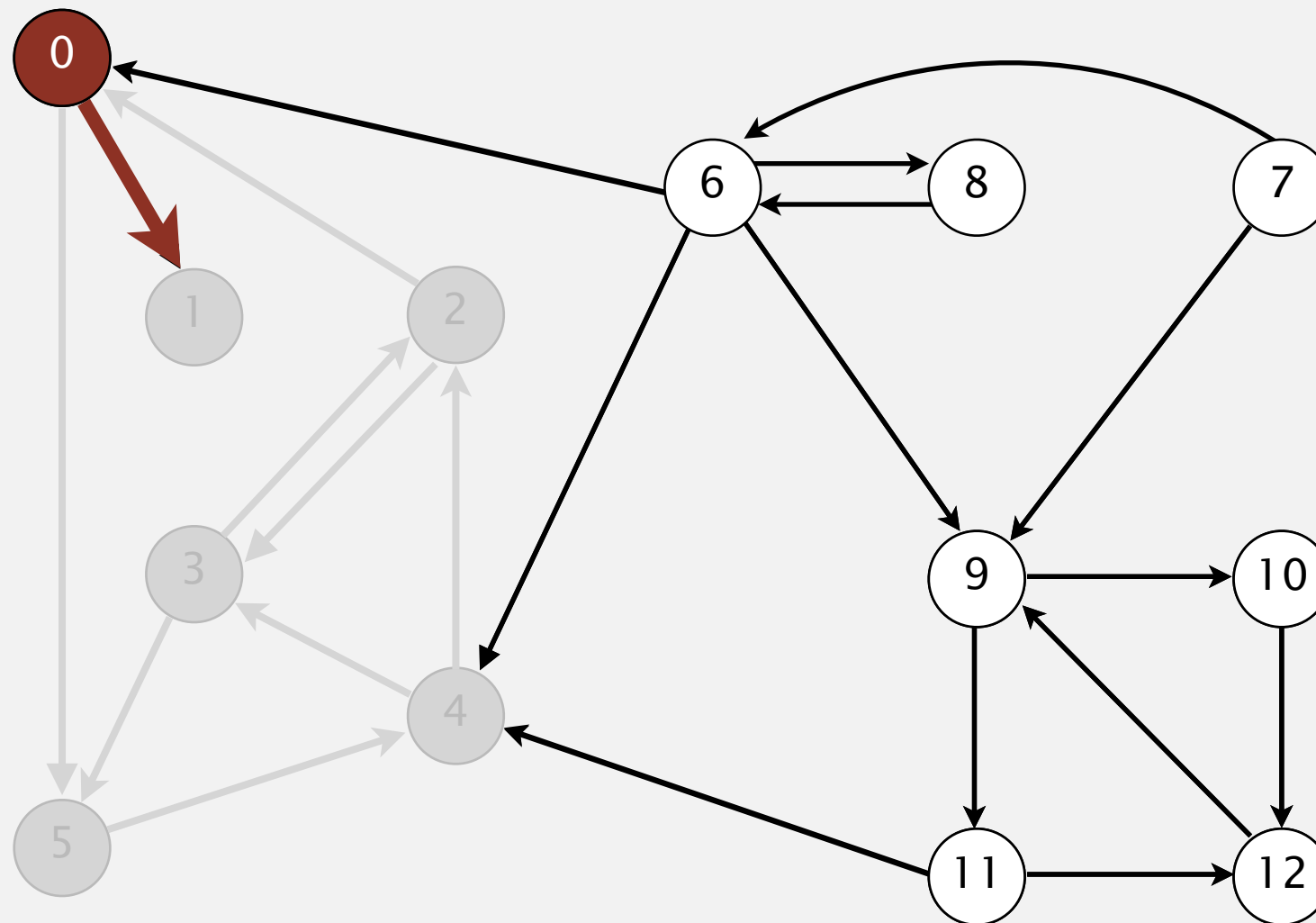
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	—
7	—
8	—
9	—
10	—
11	—
12	—

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



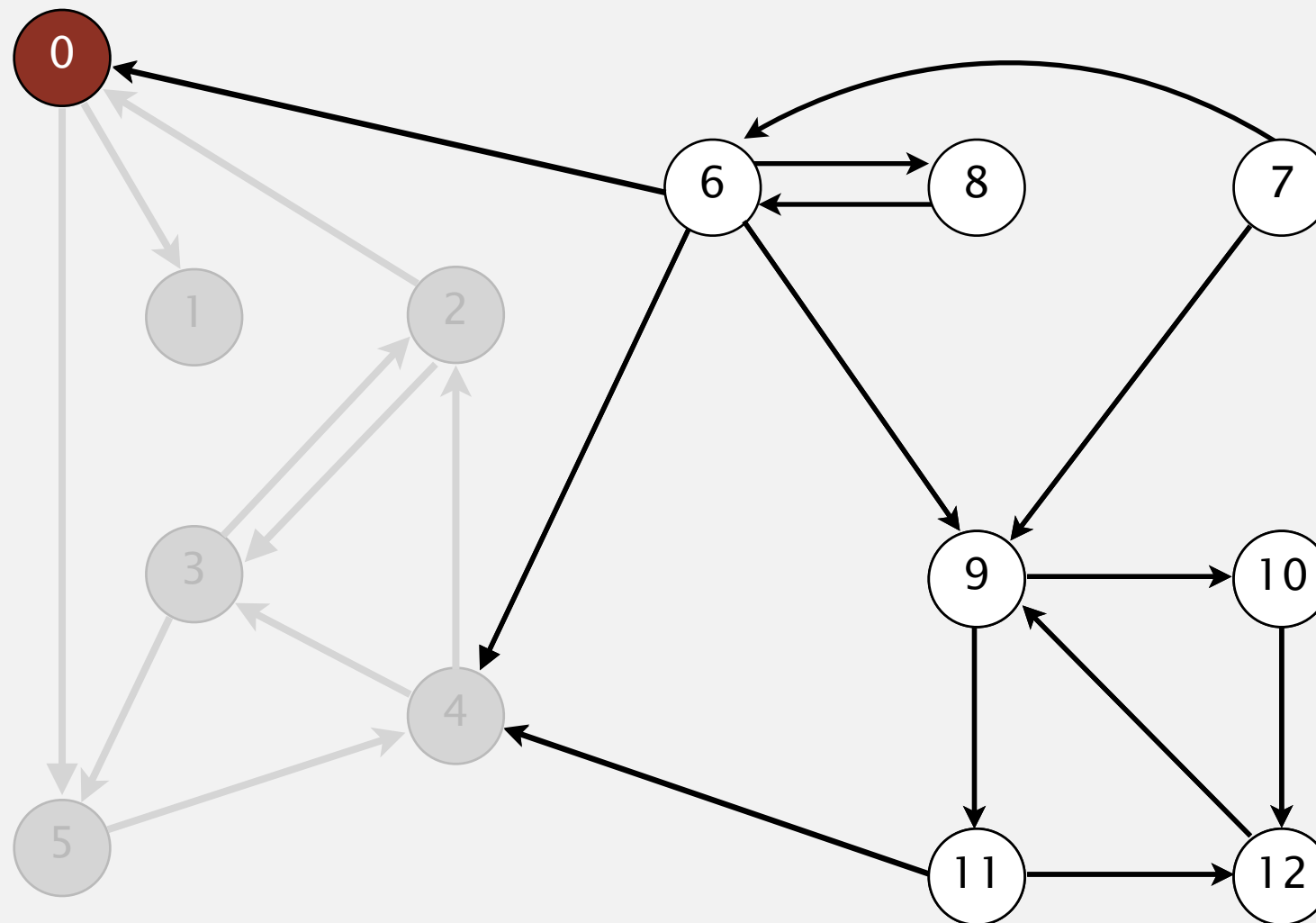
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	—
7	—
8	—
9	—
10	—
11	—
12	—

**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



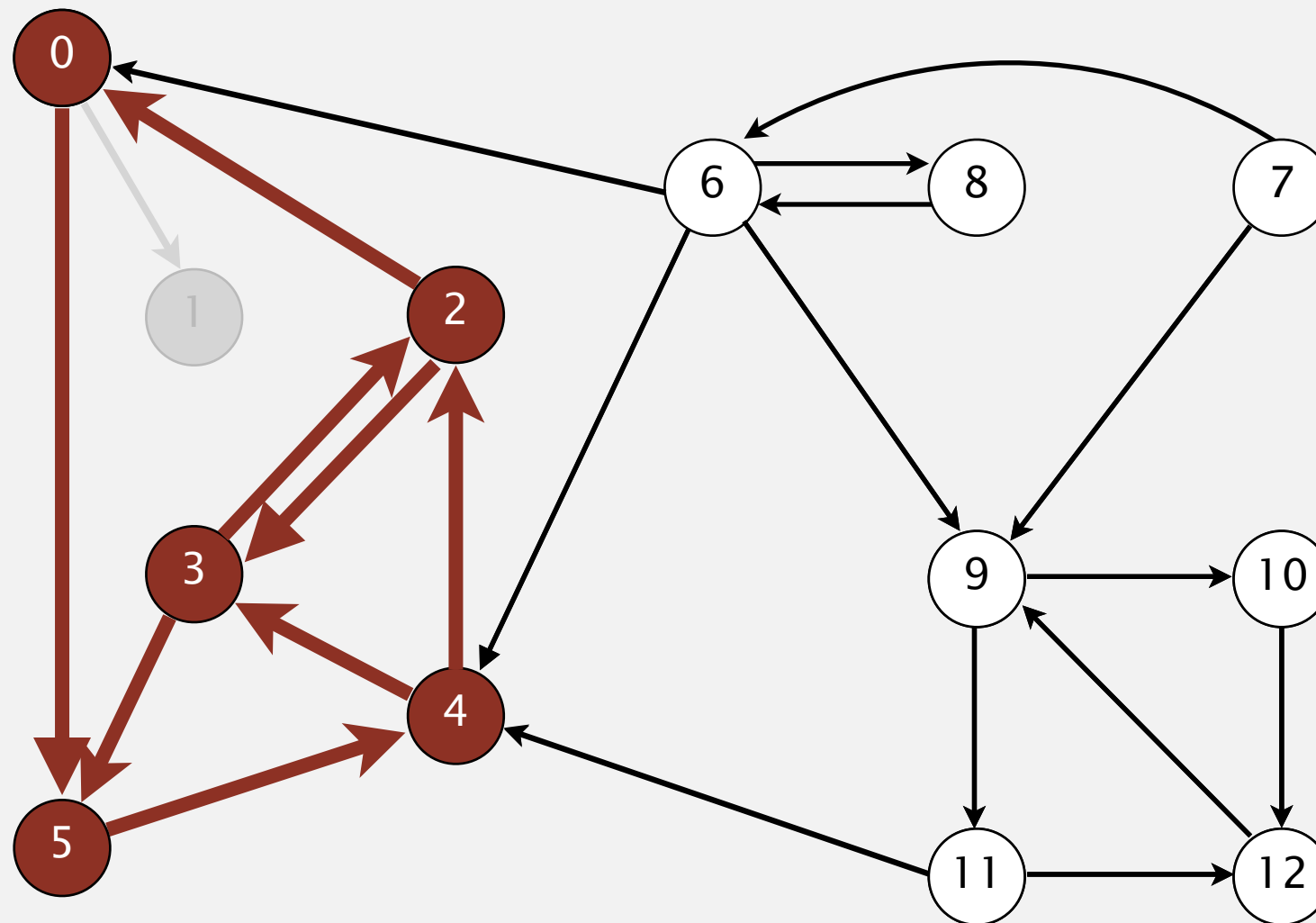
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	—
7	—
8	—
9	—
10	—
11	—
12	—

**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



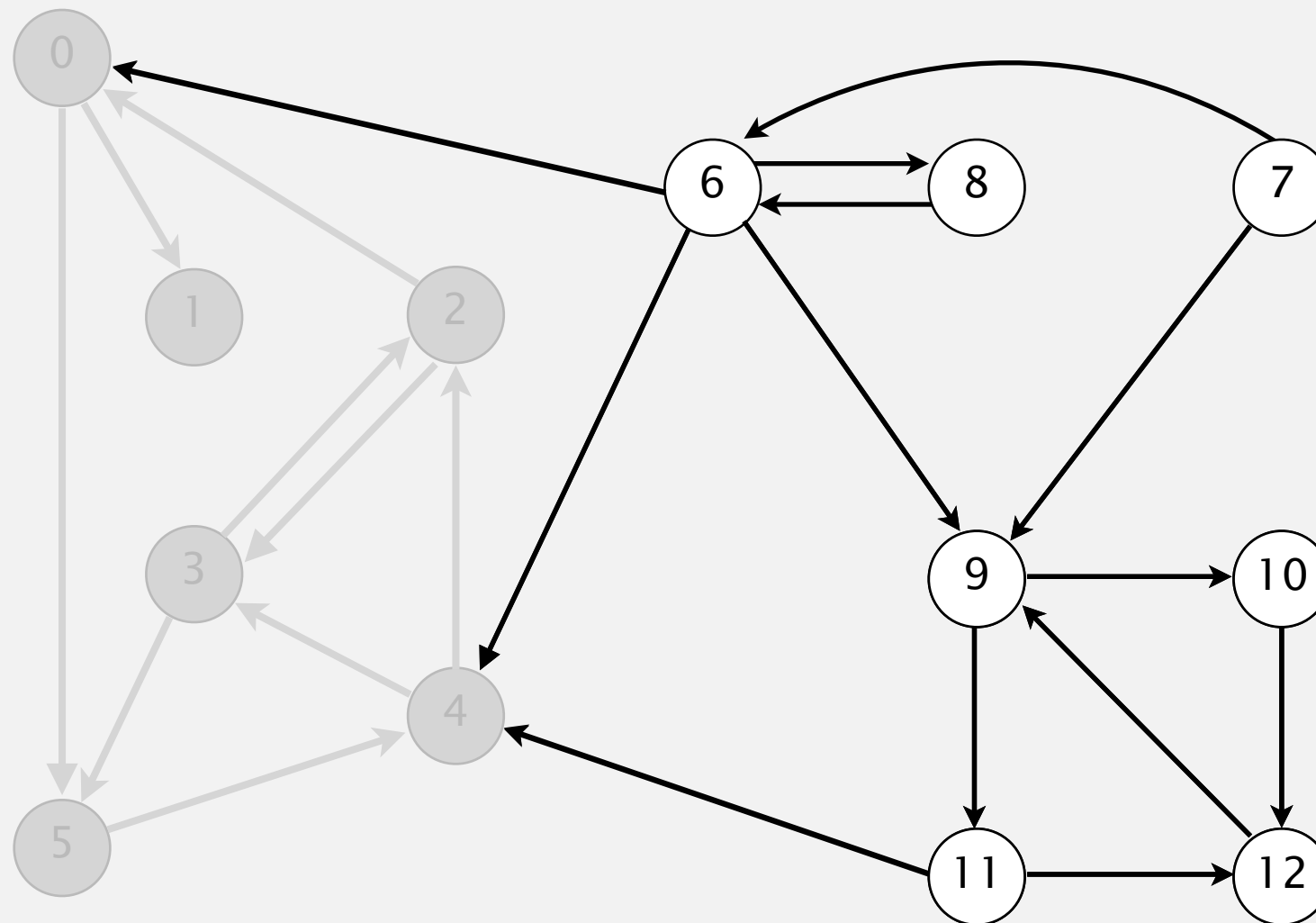
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	—
7	—
8	—
9	—
10	—
11	—
12	—

**strong component: 0 2 3 4 5**

# 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	–

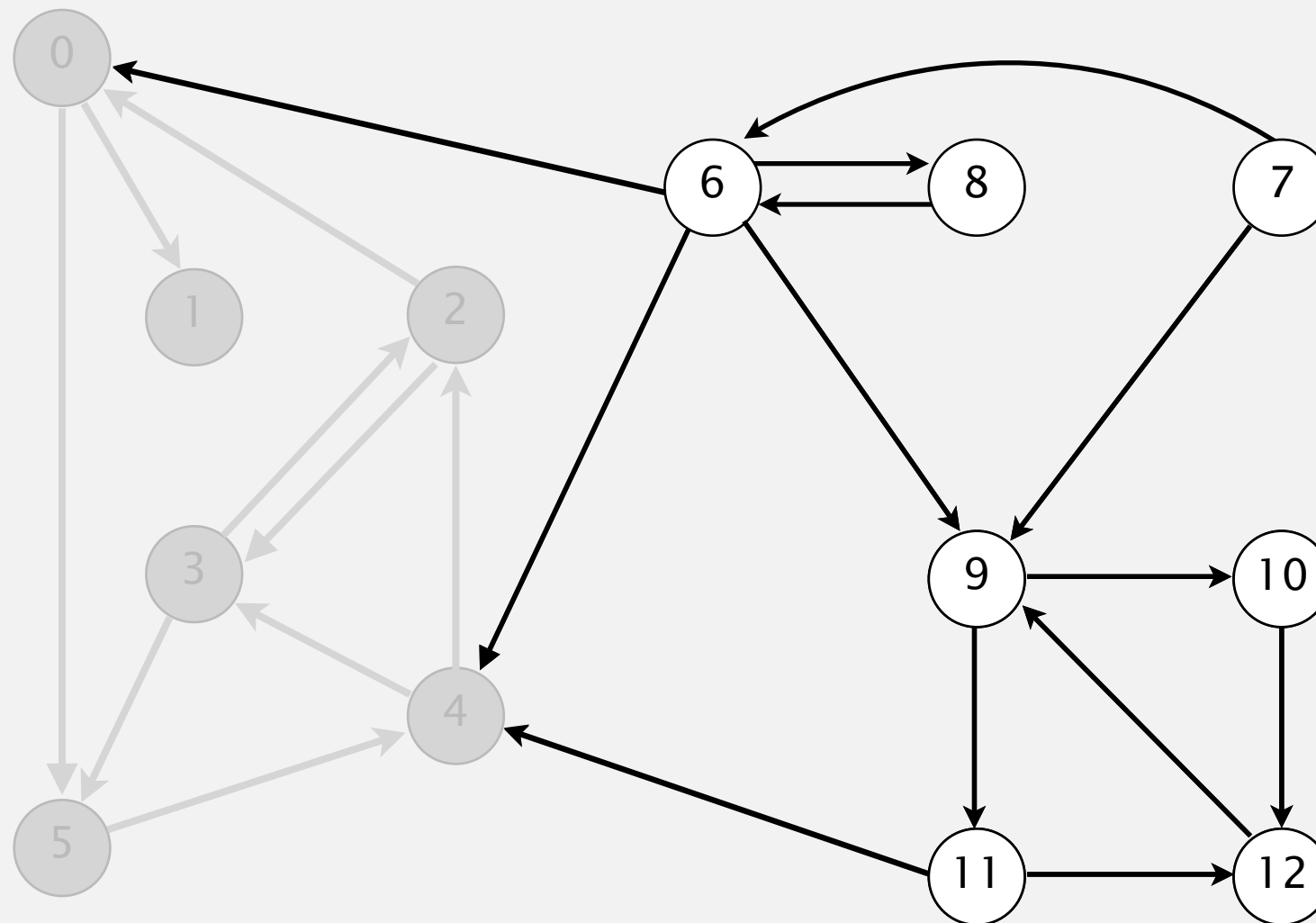
check 2



# 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



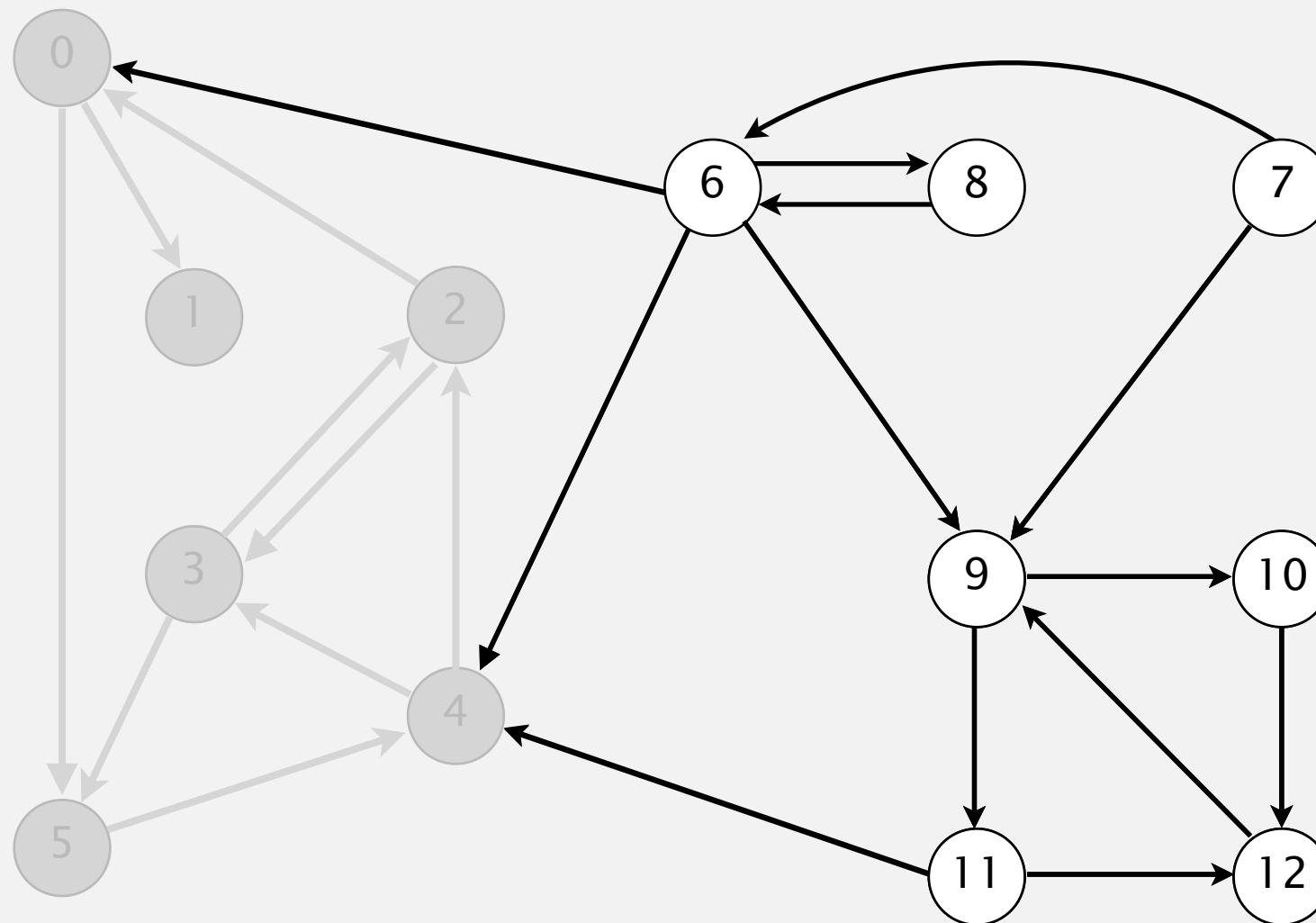
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	—
7	—
8	—
9	—
10	—
11	—
12	—

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



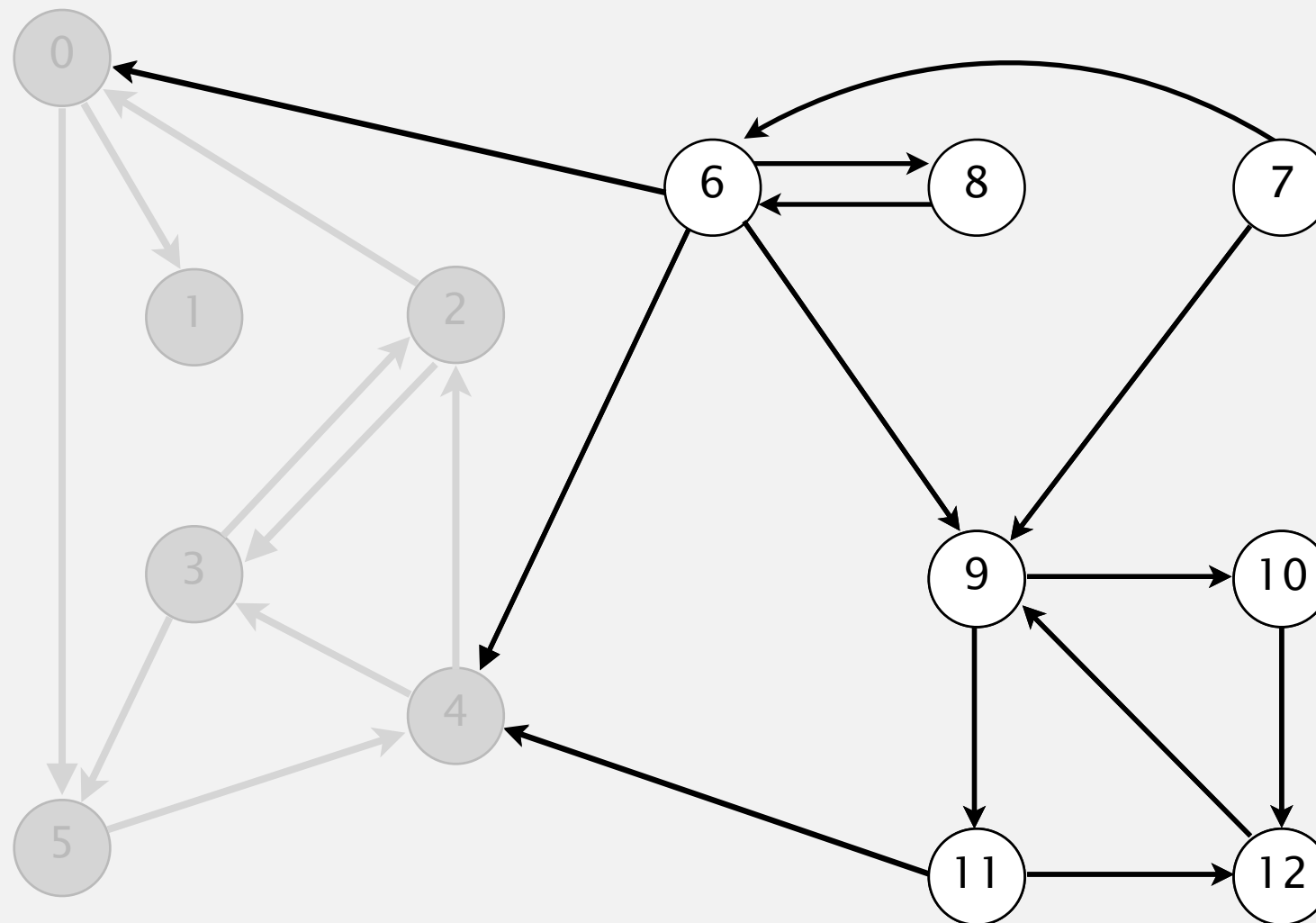
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

check 5

# 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



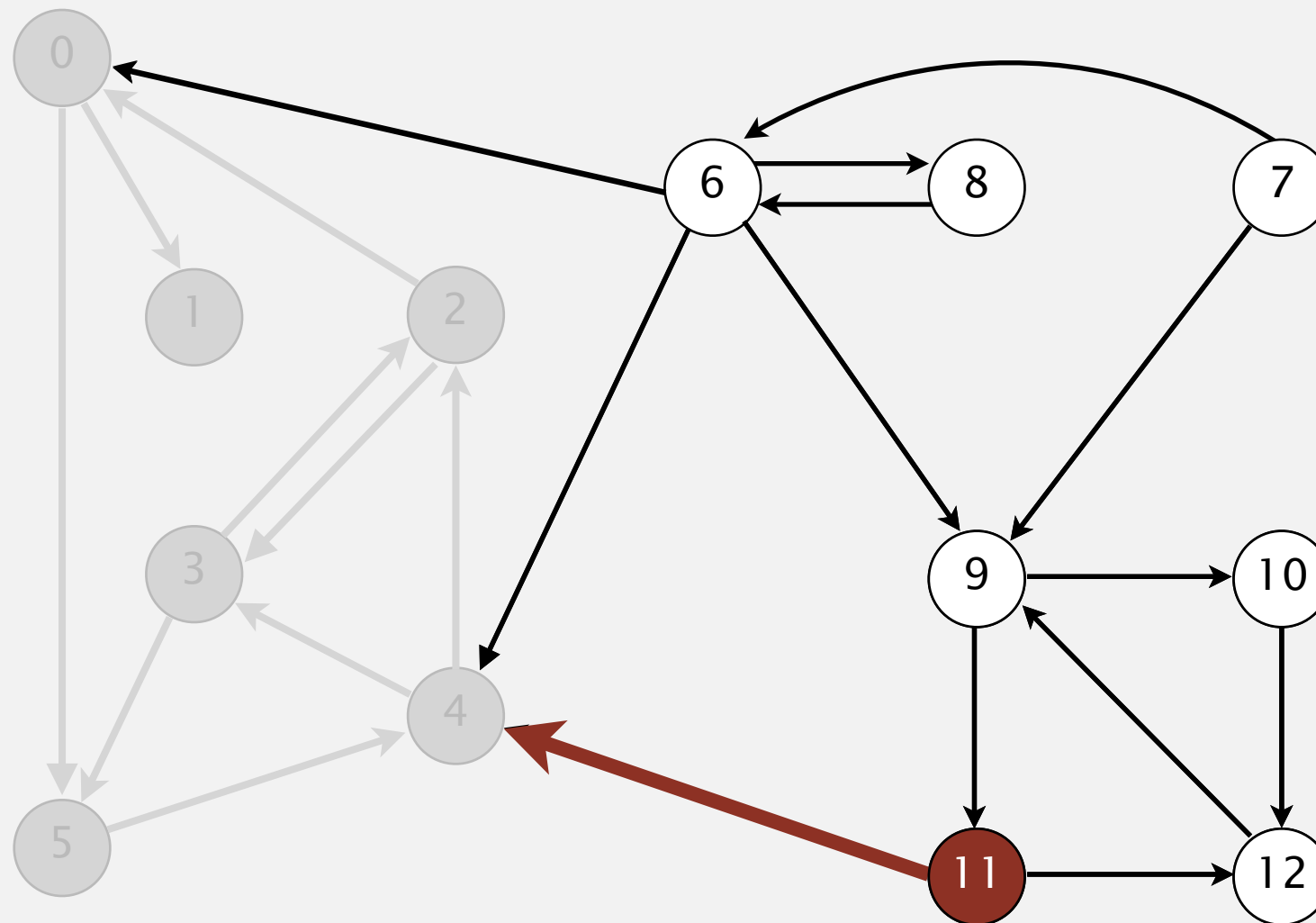
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	–
12	–

check 3

# 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



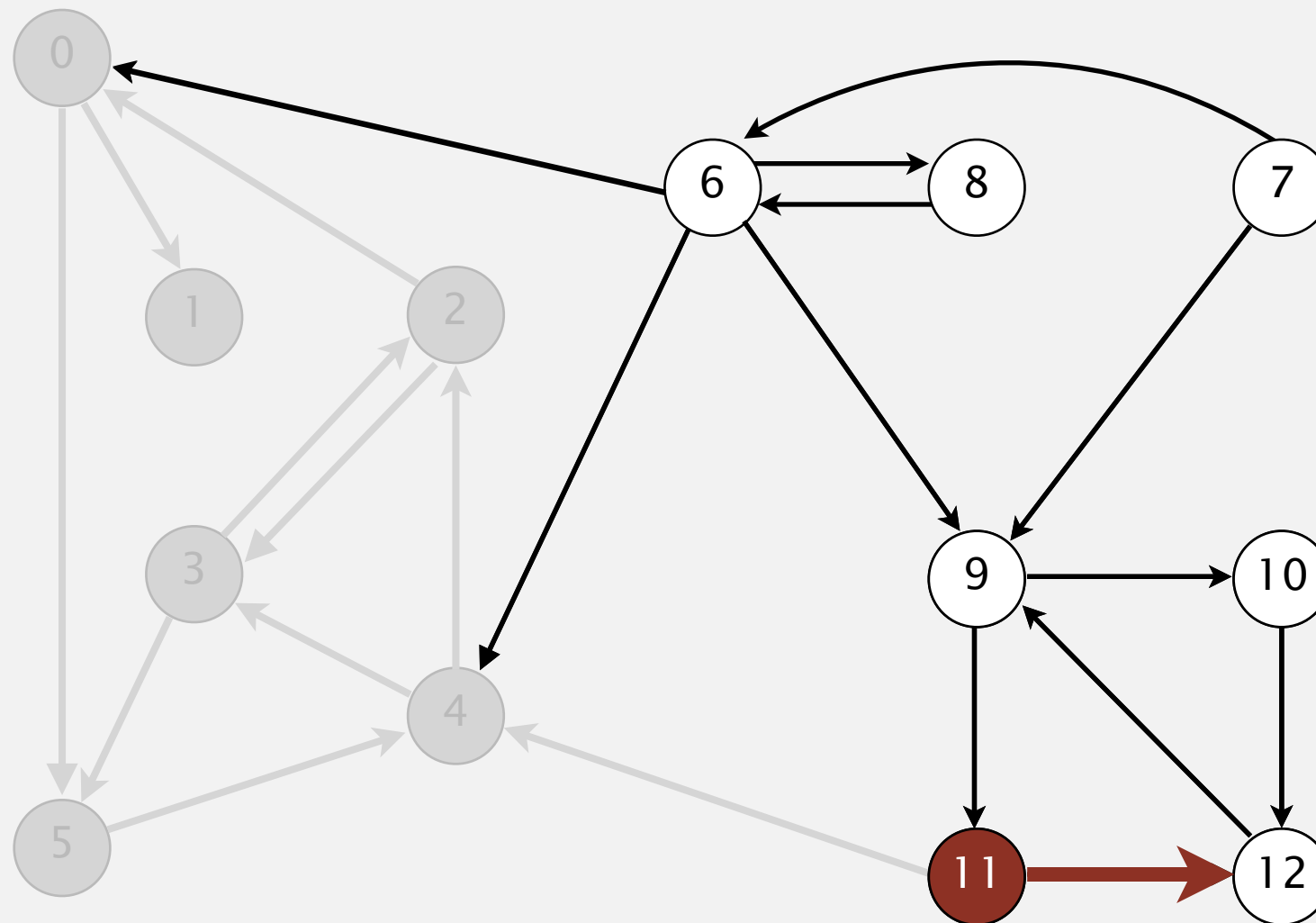
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	<b>2</b>
12	–

**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



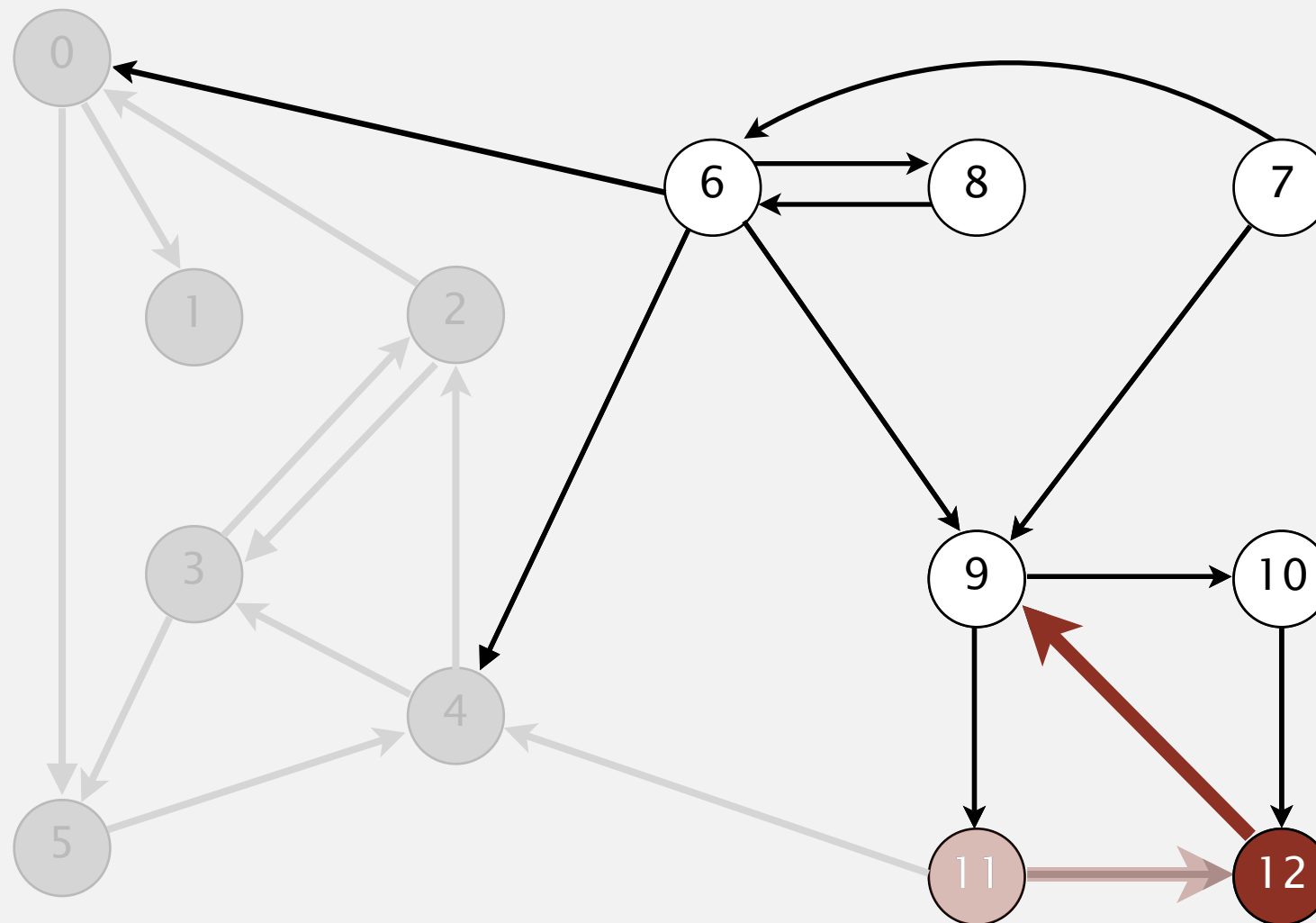
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	<b>2</b>
12	–

**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



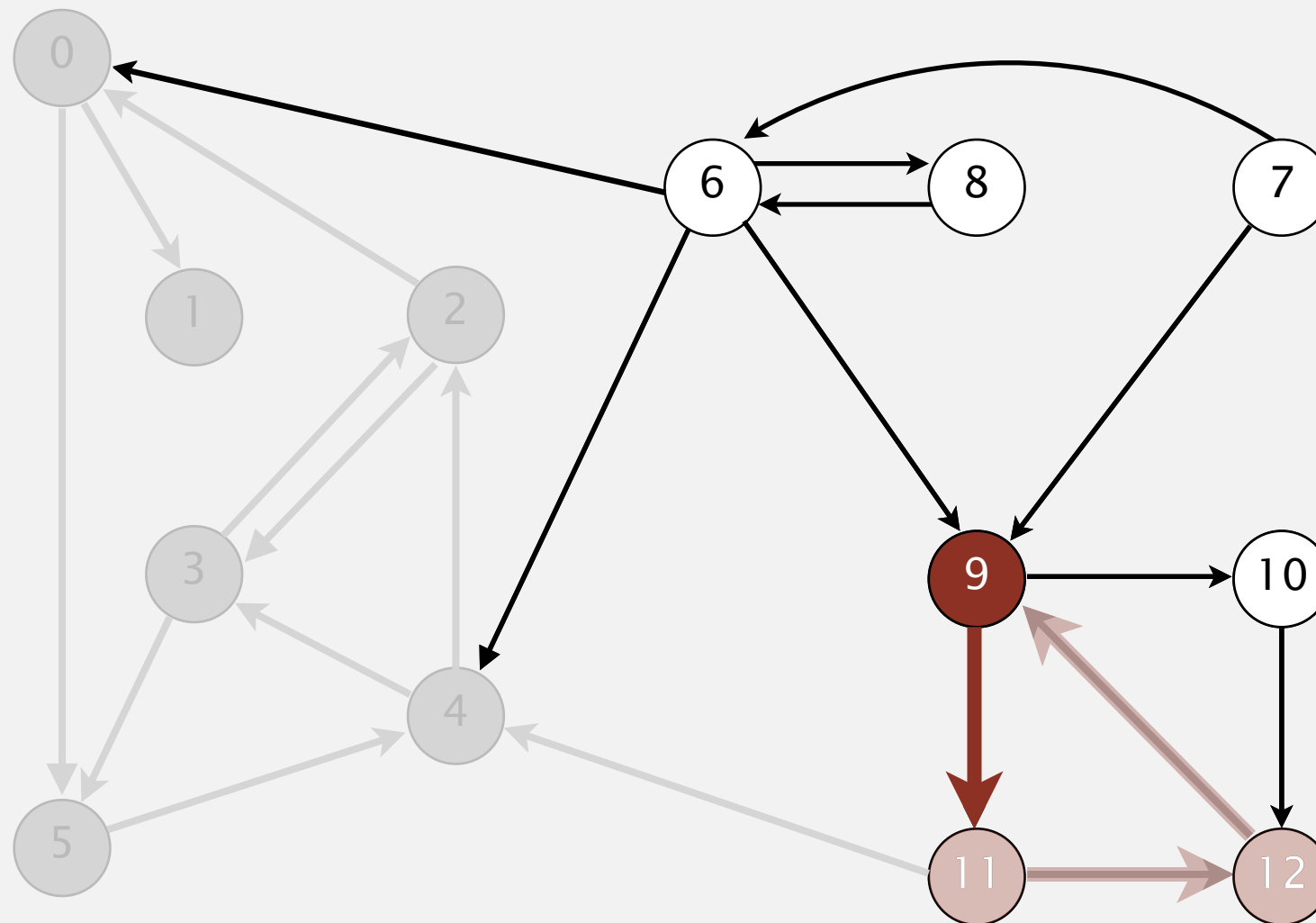
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	–
10	–
11	2
12	2

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



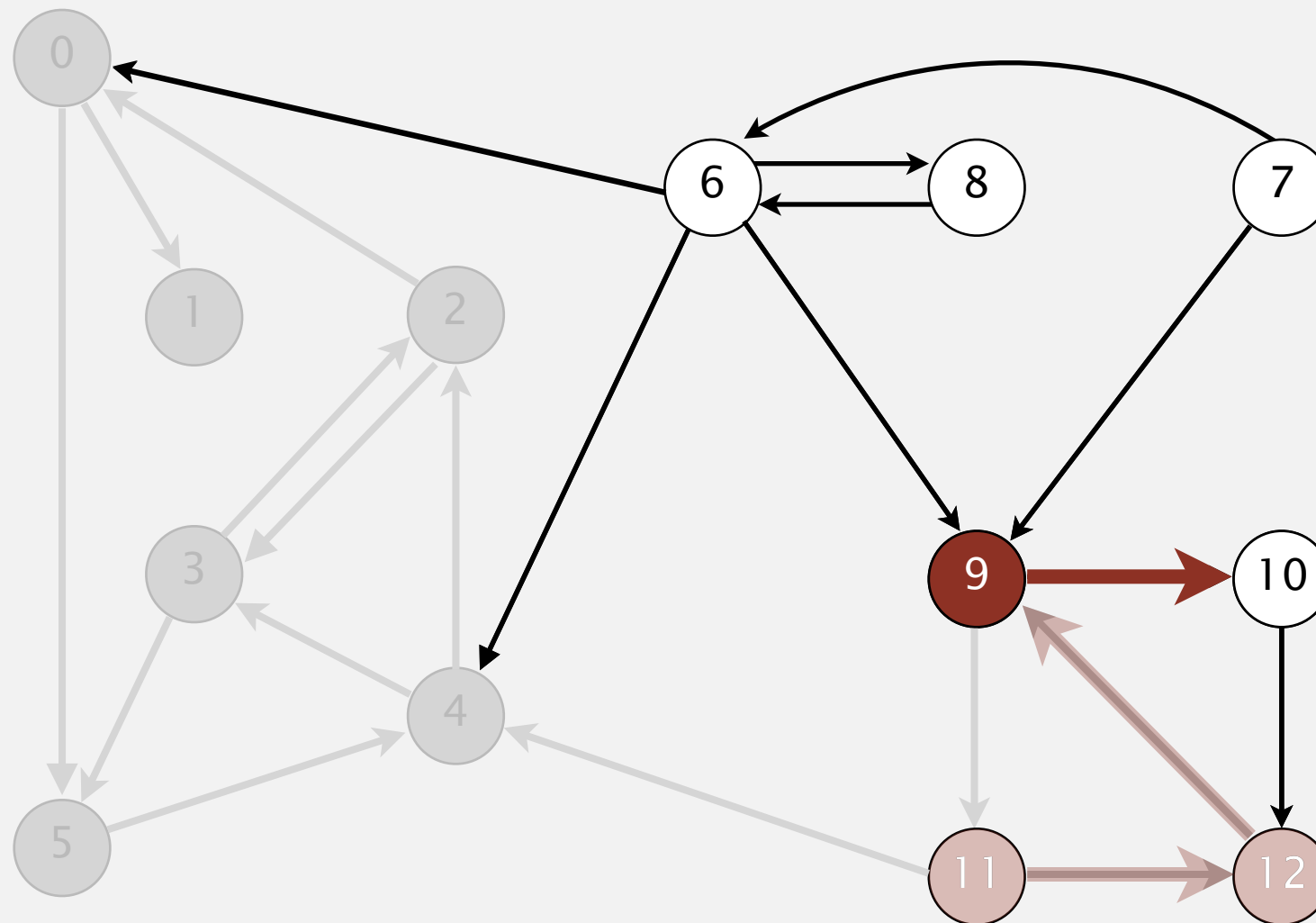
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	<b>2</b>
10	–
11	<b>2</b>
12	<b>2</b>

**visit 9: check 11** and check 10

# 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	–
11	2
12	2

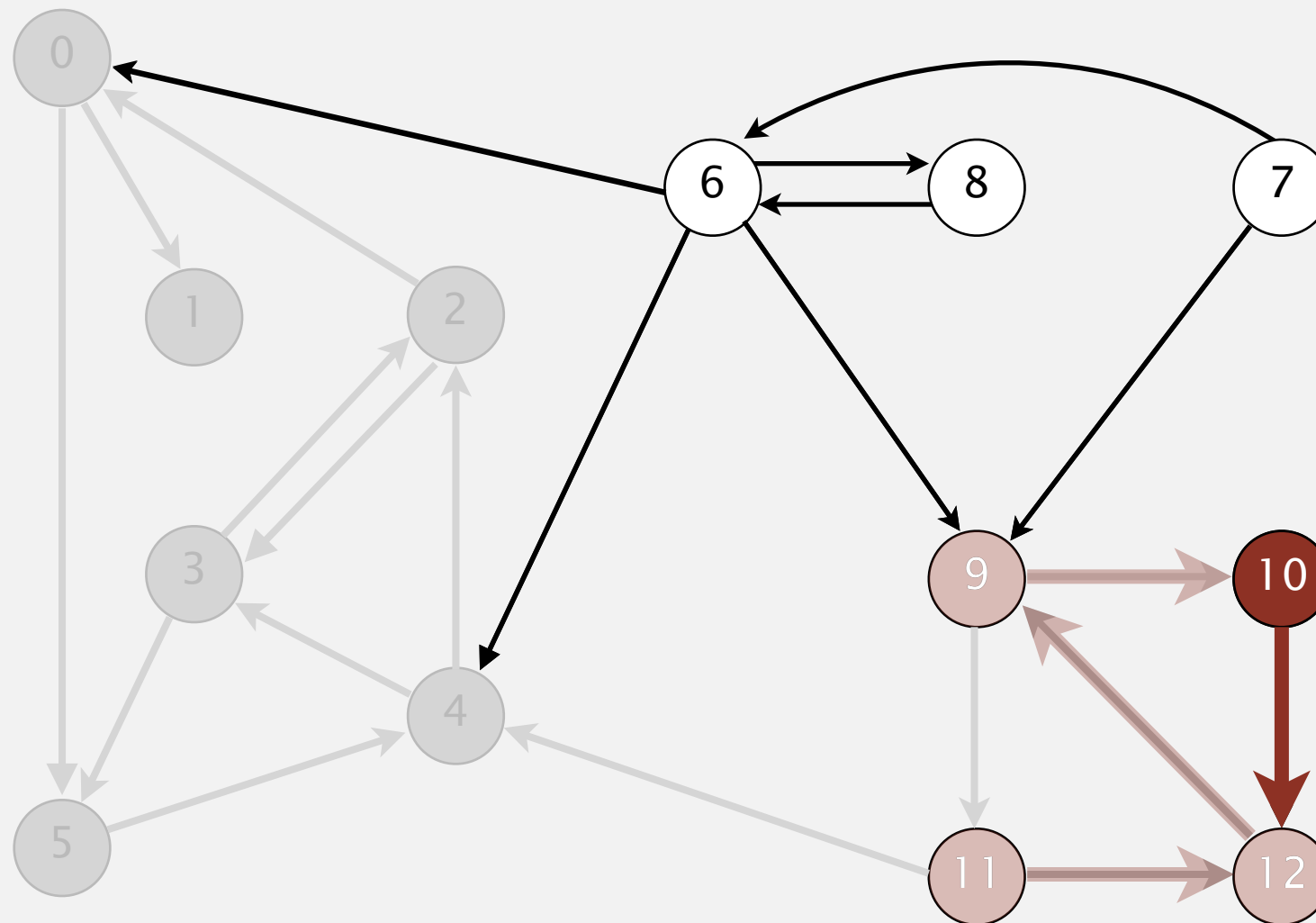
**visit 9:** check 11 and **check 10**



# 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



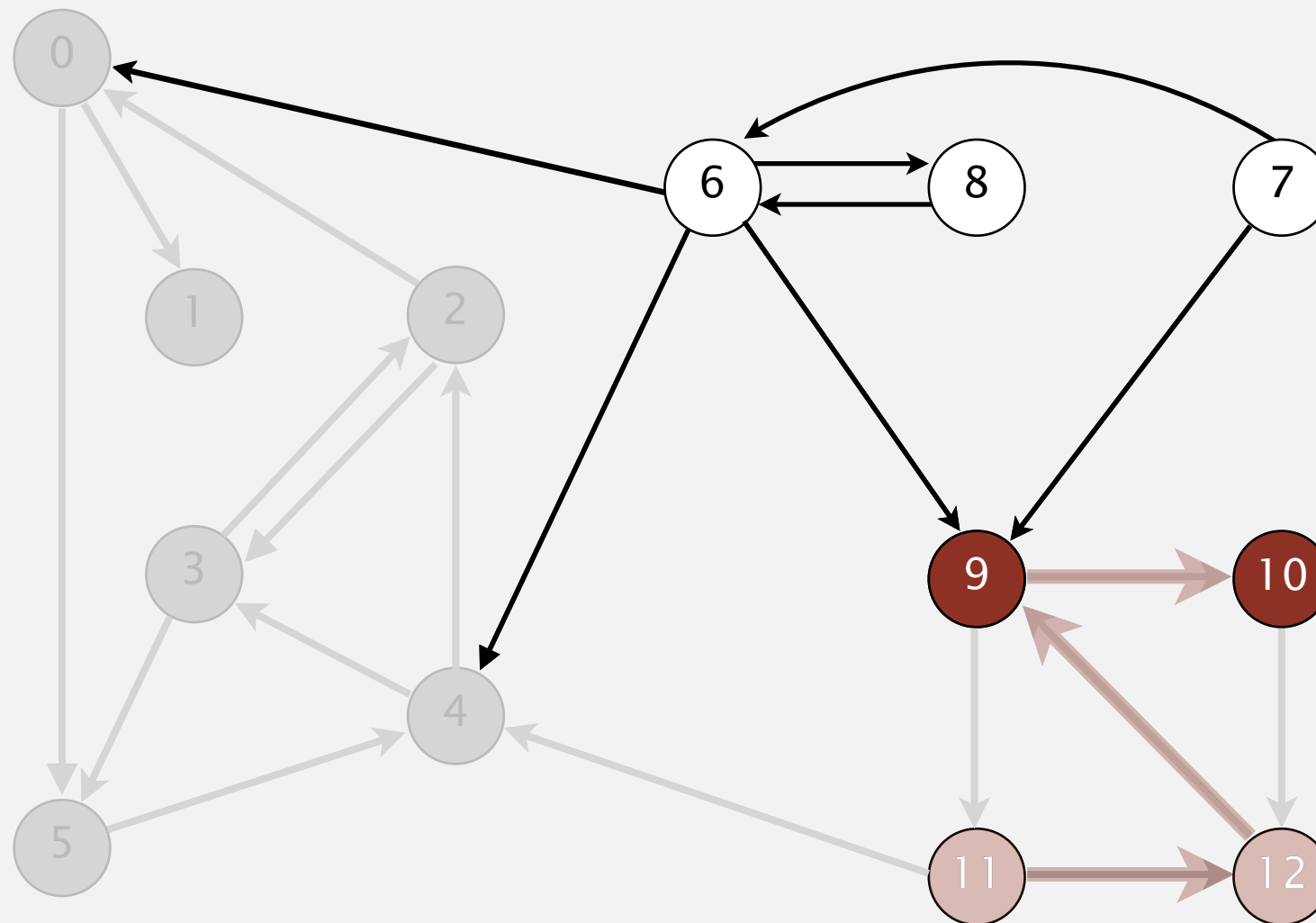
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	2
11	2
12	2

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



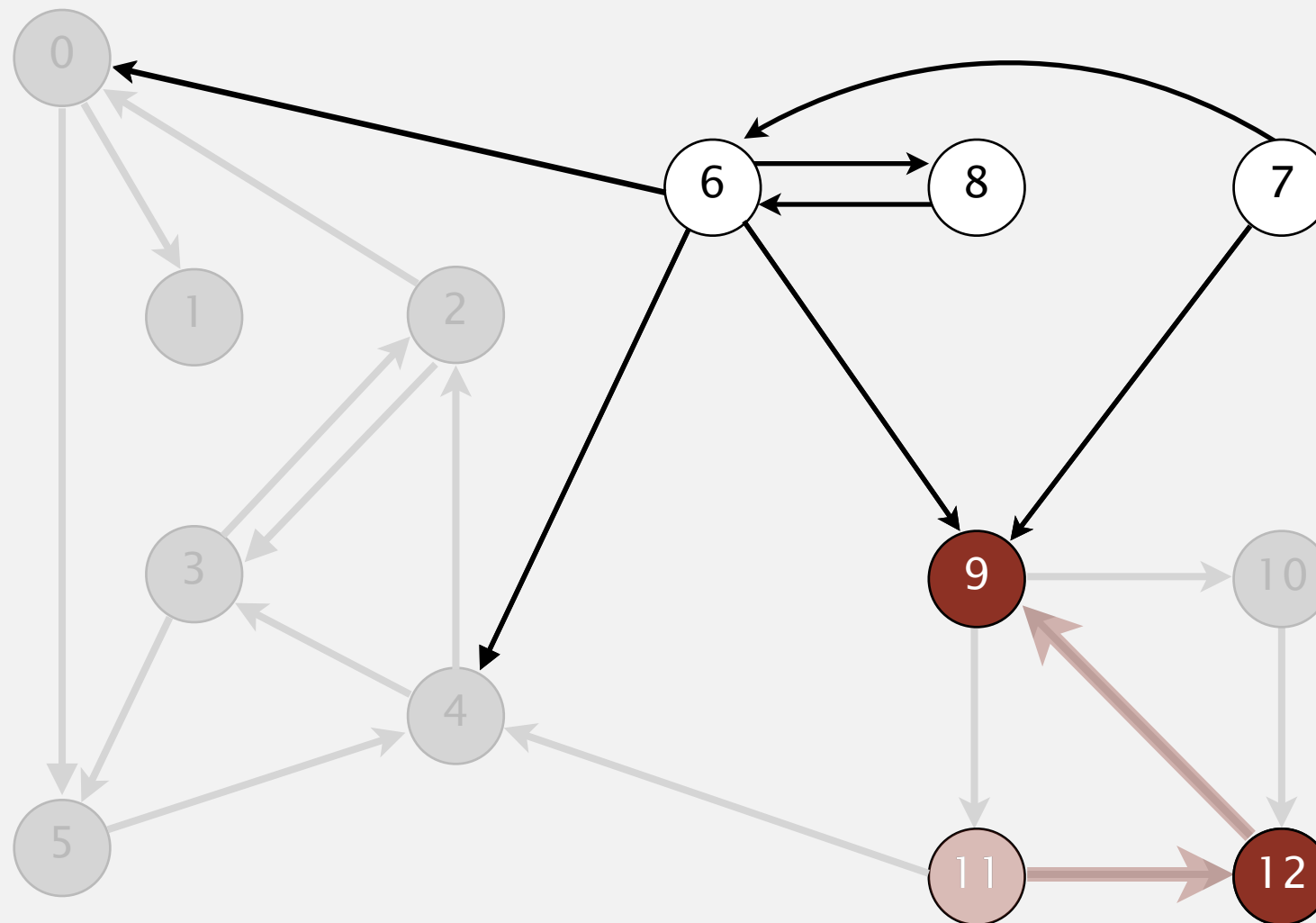
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	2
11	2
12	2

**10 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



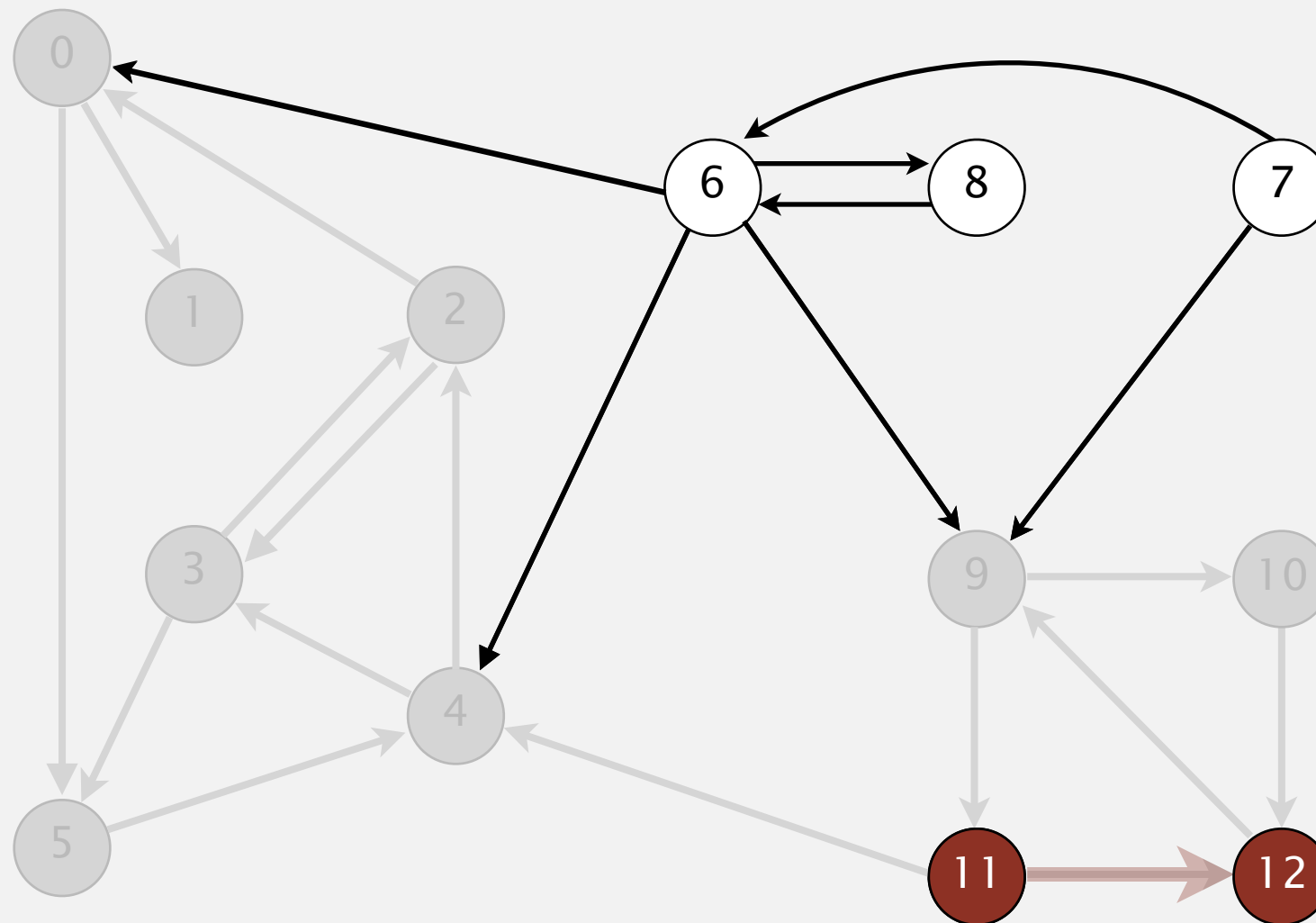
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	—
7	—
8	—
9	2
10	2
11	2
12	2

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



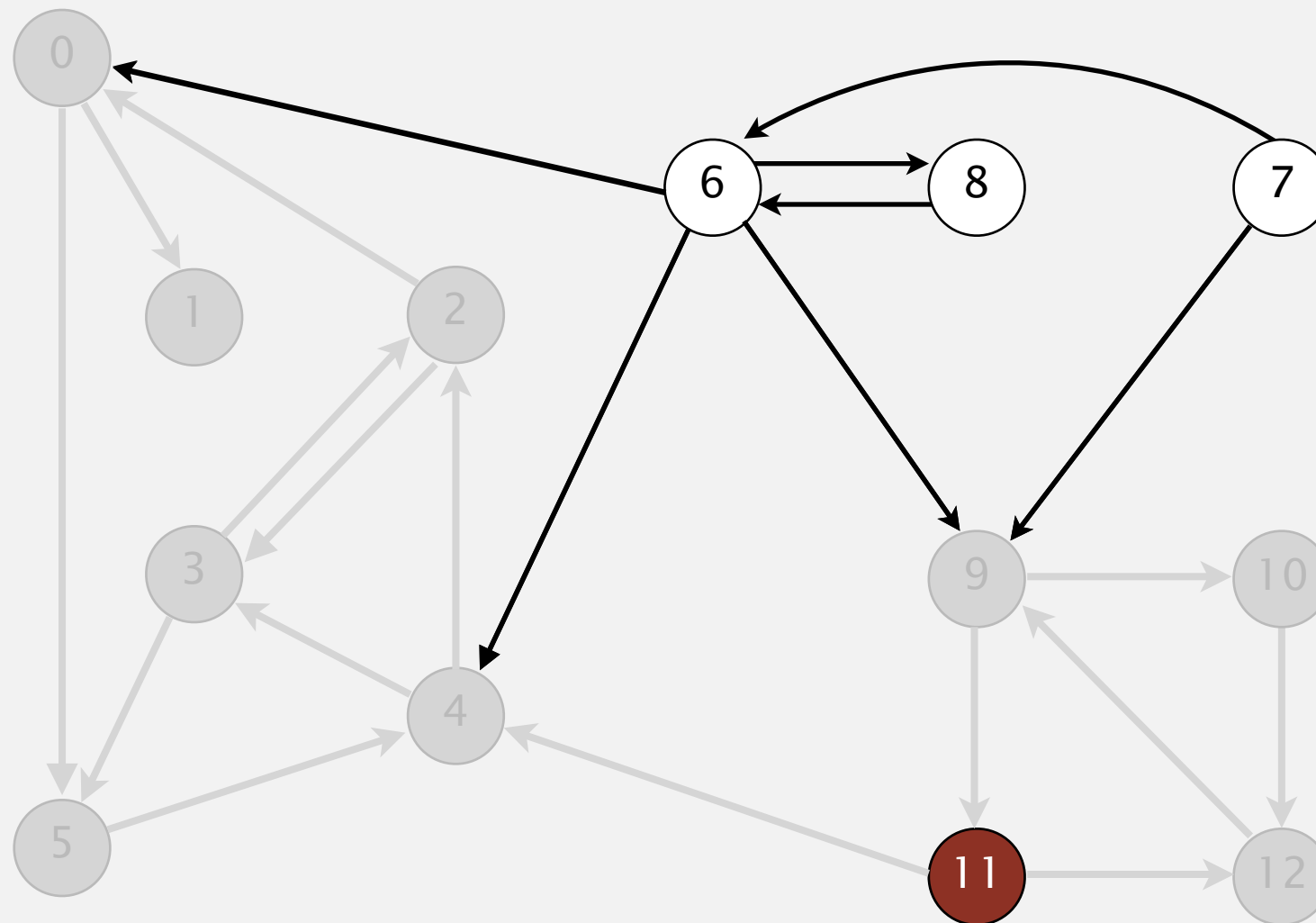
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	2
11	2
12	2

**12 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



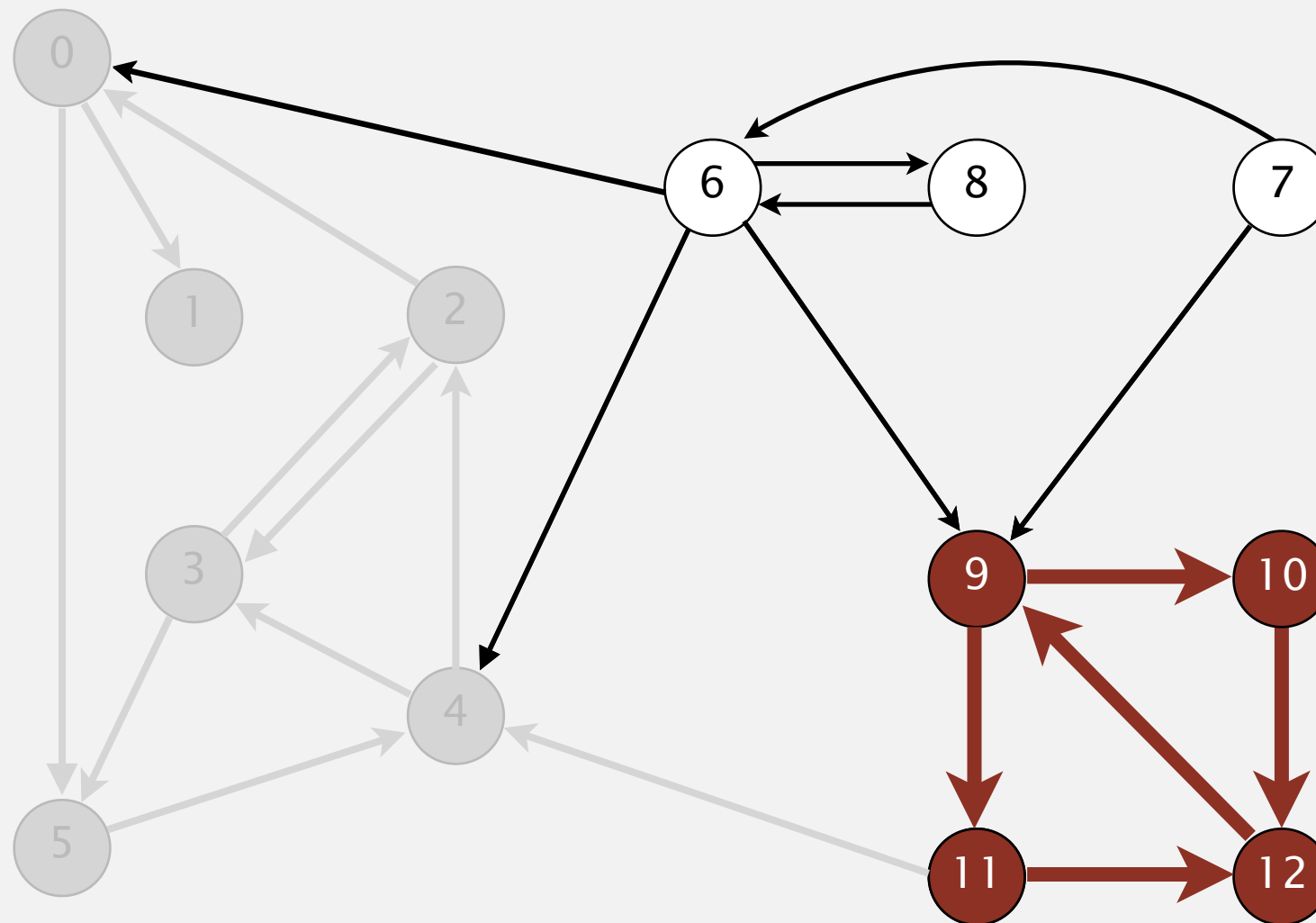
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	2
11	2
12	2

**11 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



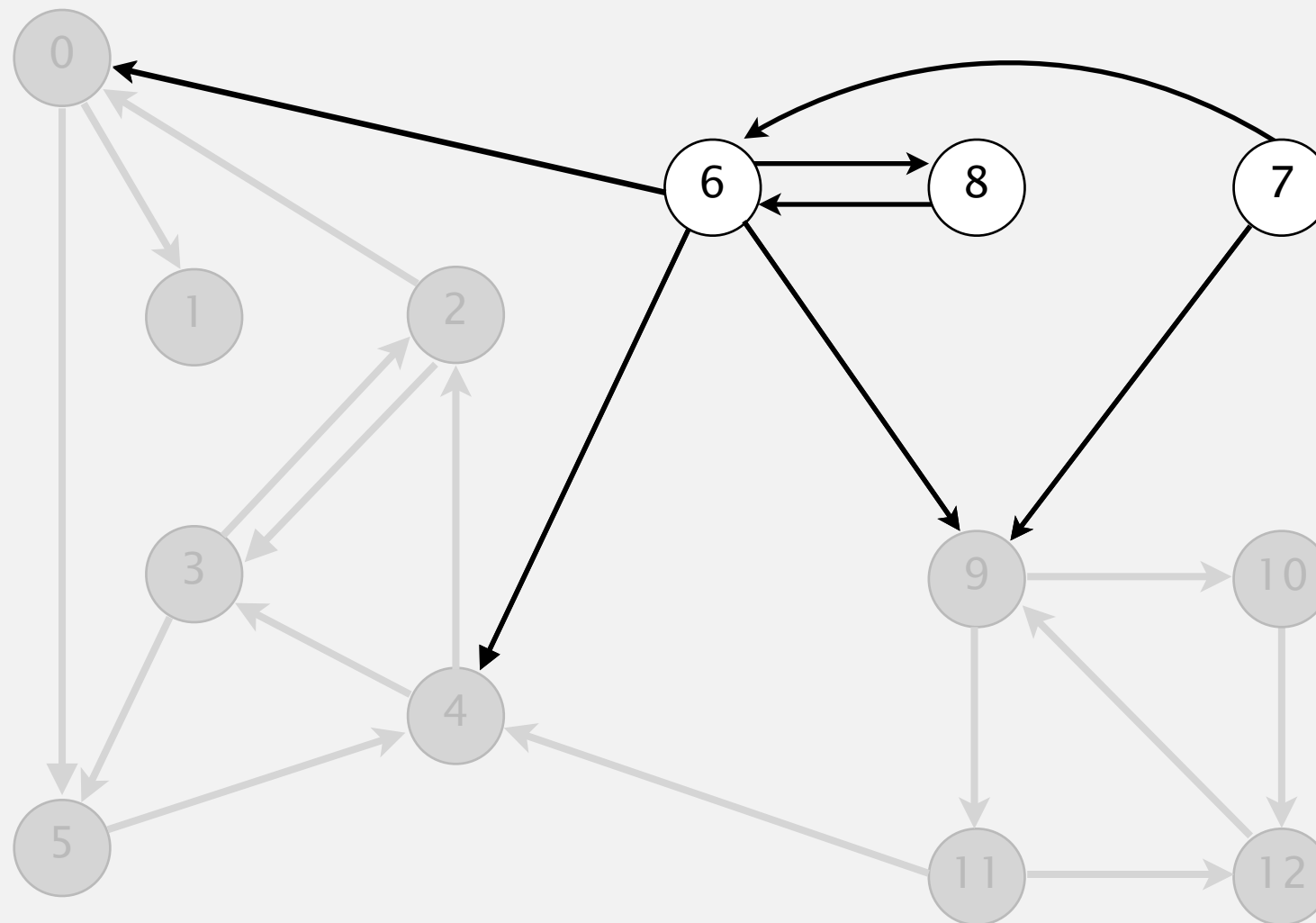
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	2
11	2
12	2

**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



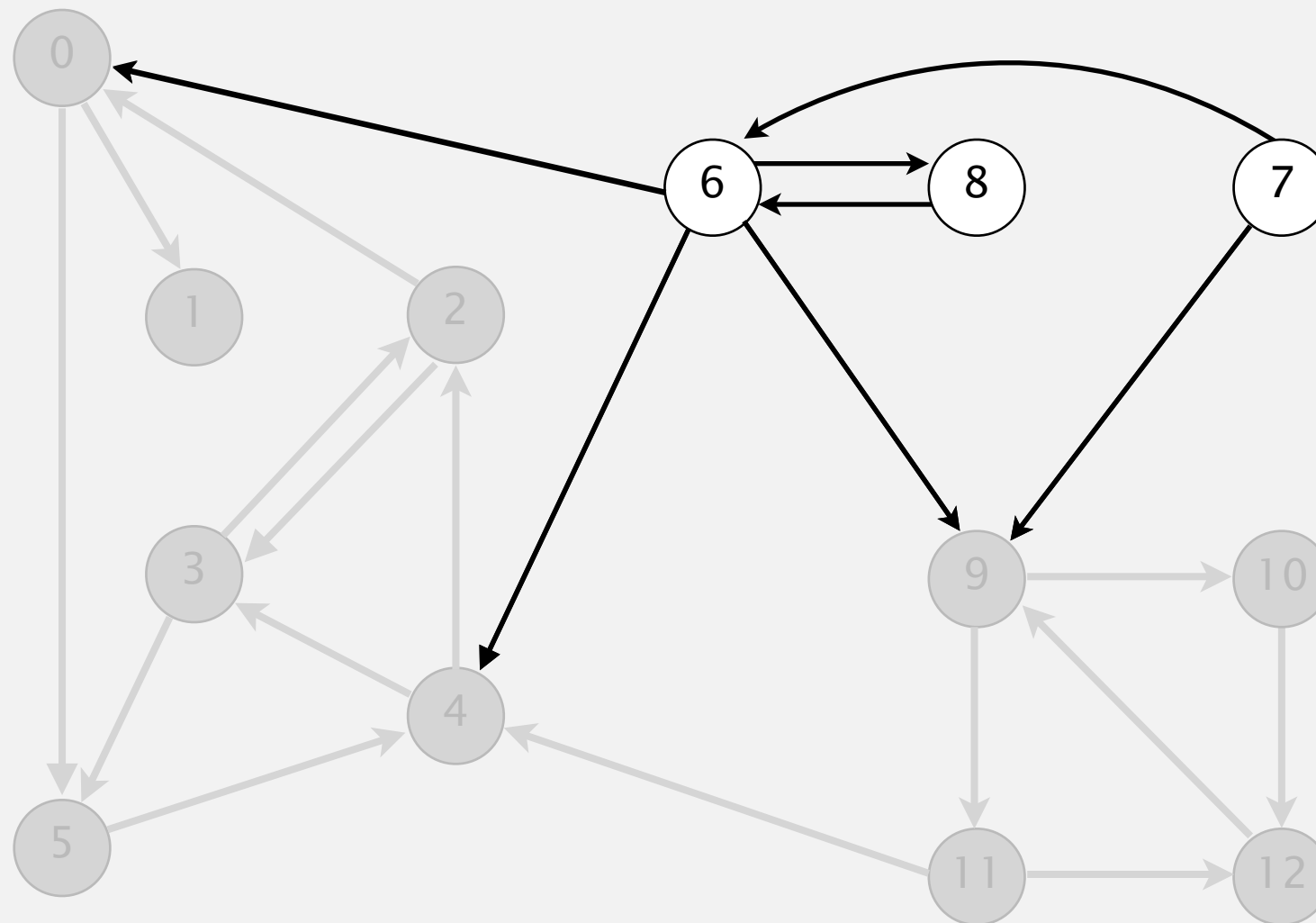
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	2
11	2
12	2

check 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



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

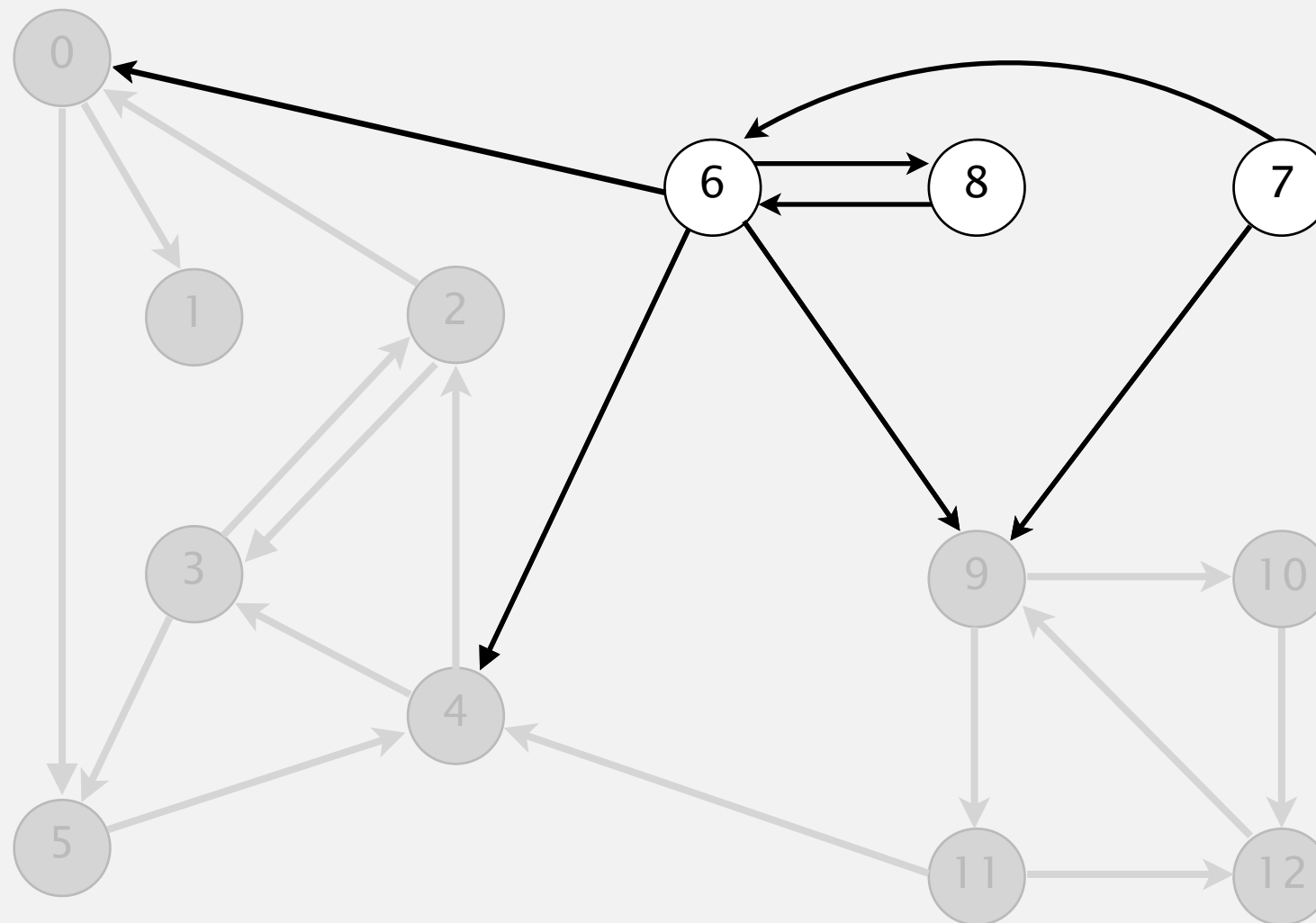
check 12



# 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



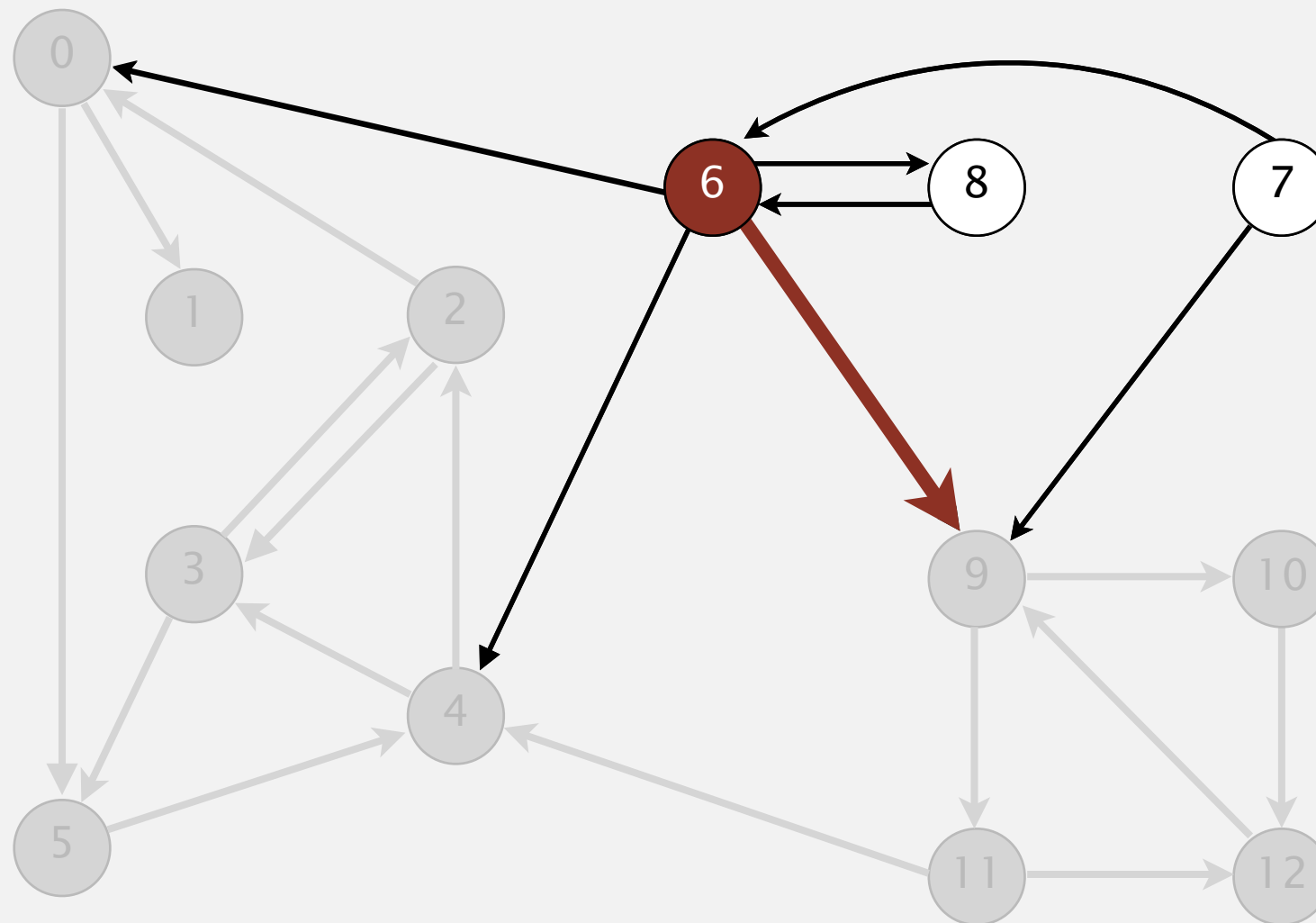
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	–
7	–
8	–
9	2
10	2
11	2
12	2

check 10

# 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



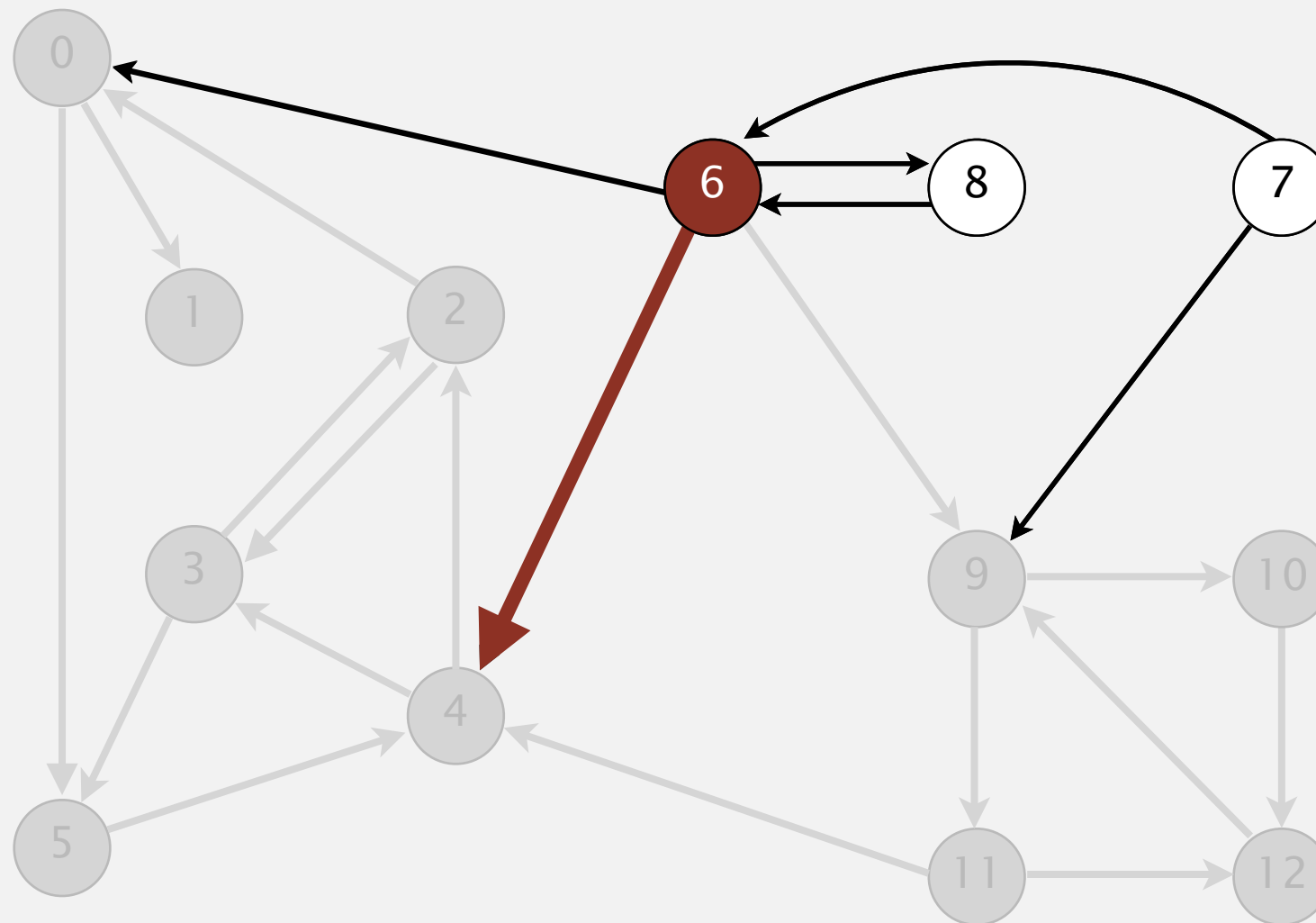
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	<b>3</b>
7	–
8	–
9	2
10	2
11	2
12	2

**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



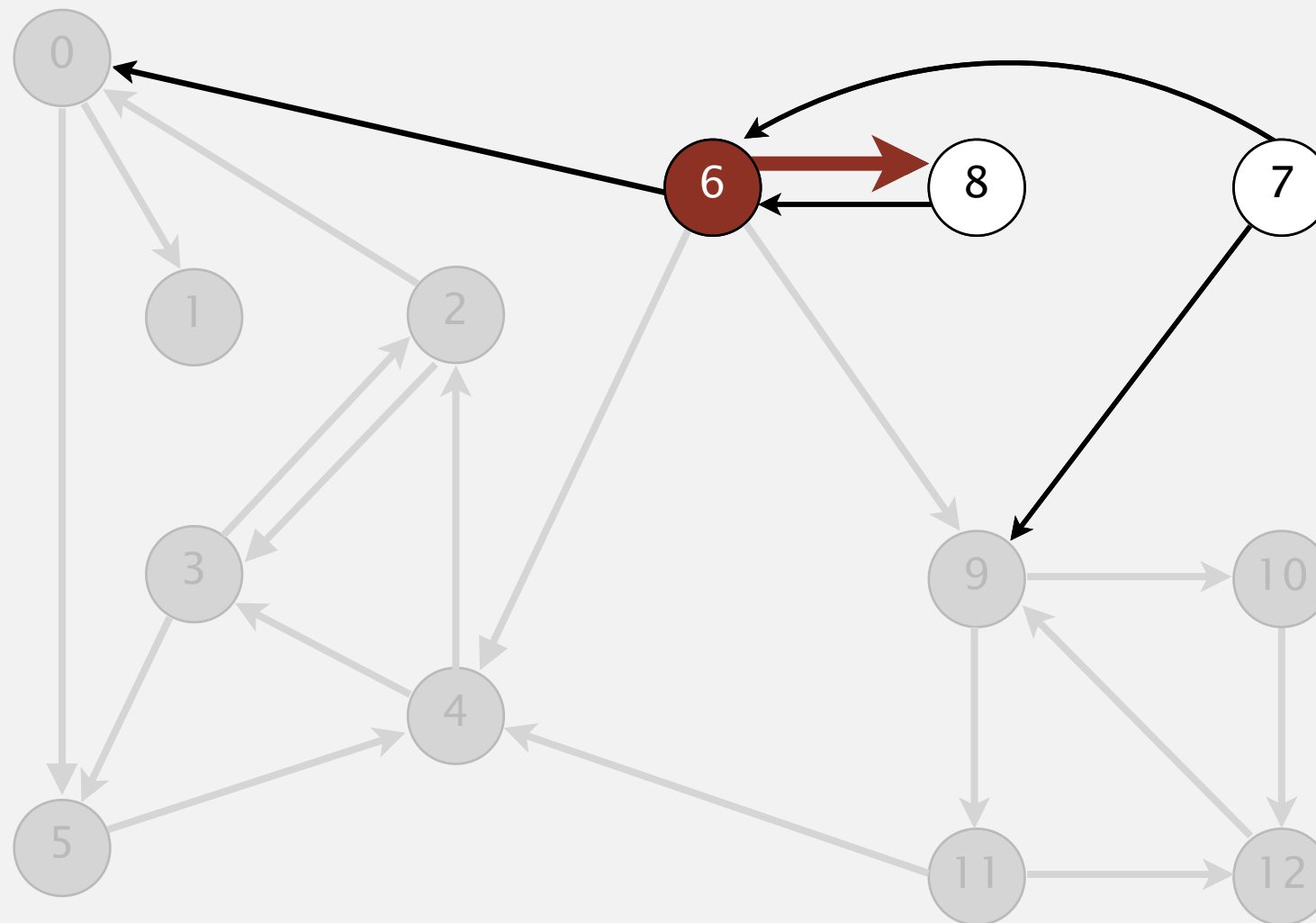
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	–
8	–
9	2
10	2
11	2
12	2

**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



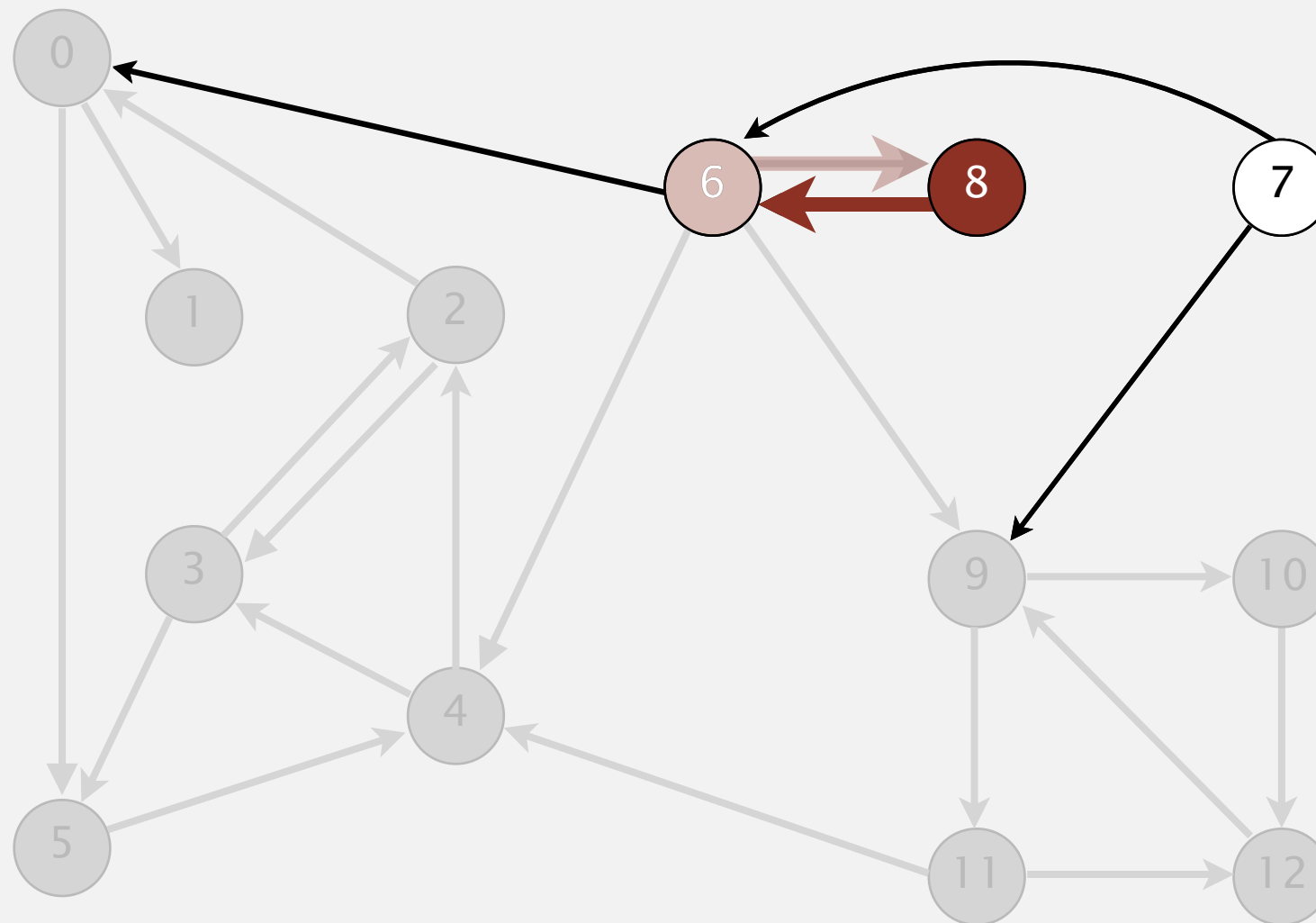
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	–
8	–
9	2
10	2
11	2
12	2

**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



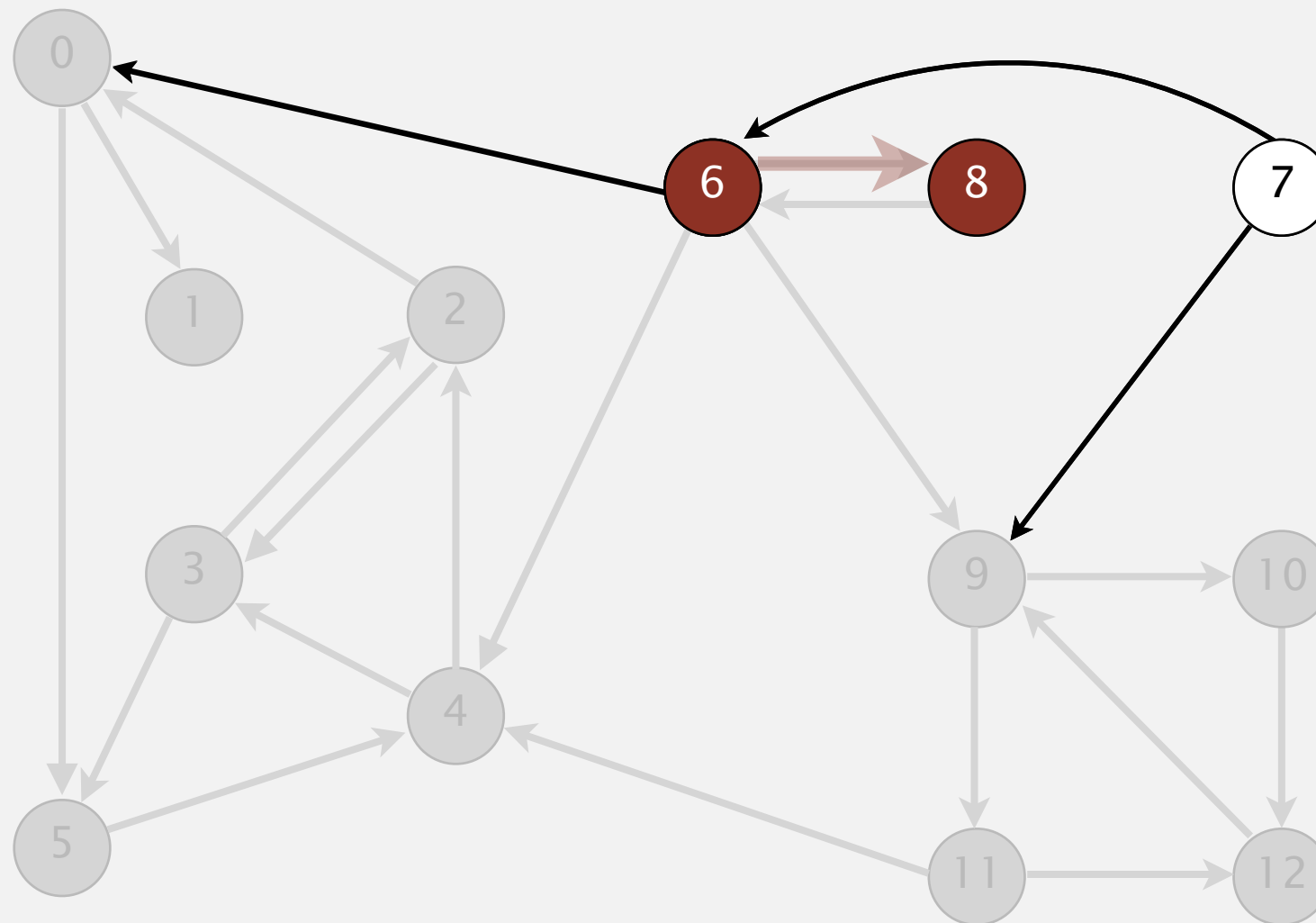
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	–
8	3
9	2
10	2
11	2
12	2

visit 8: check 6

# 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



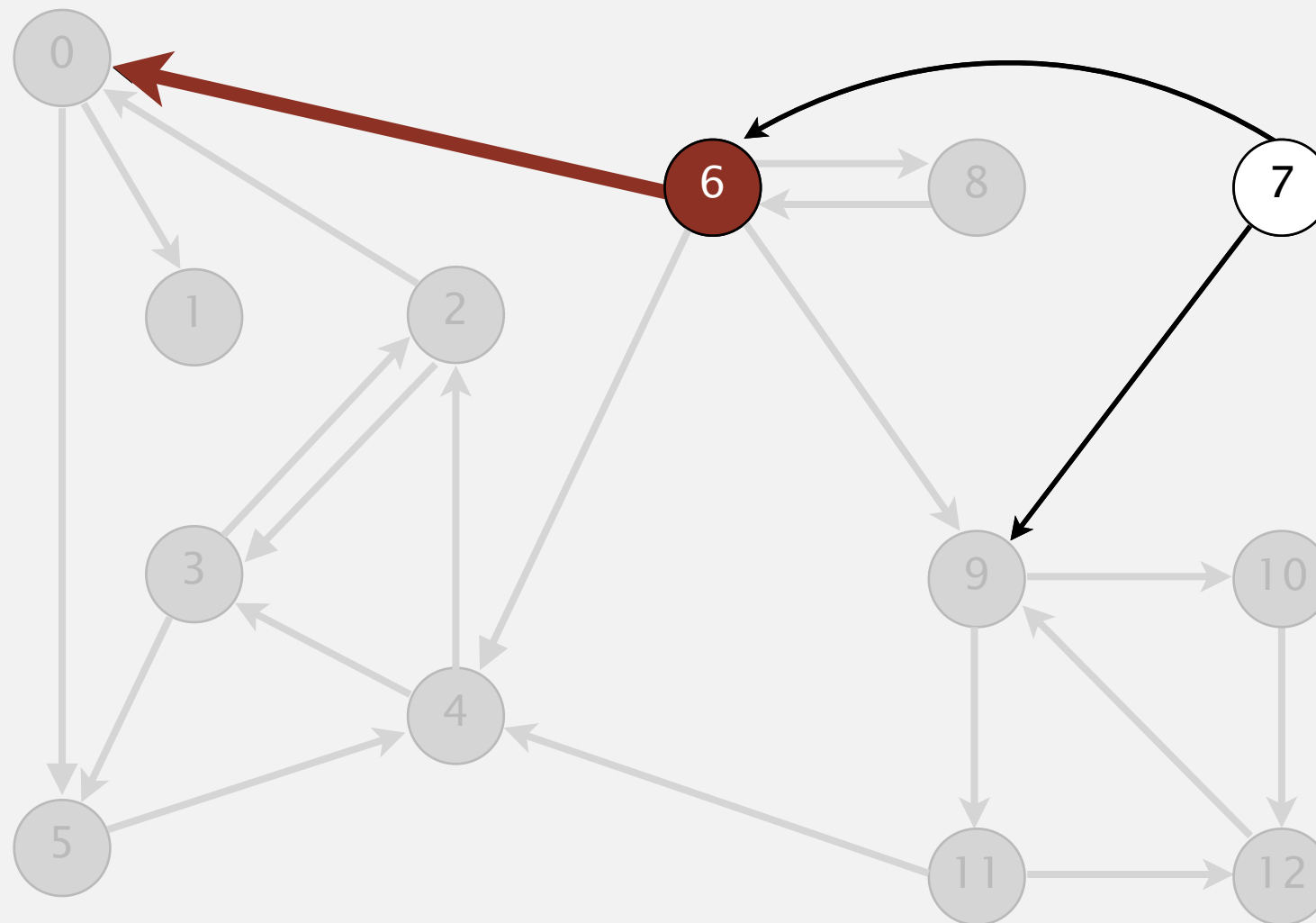
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	—
8	3
9	2
10	2
11	2
12	2

**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



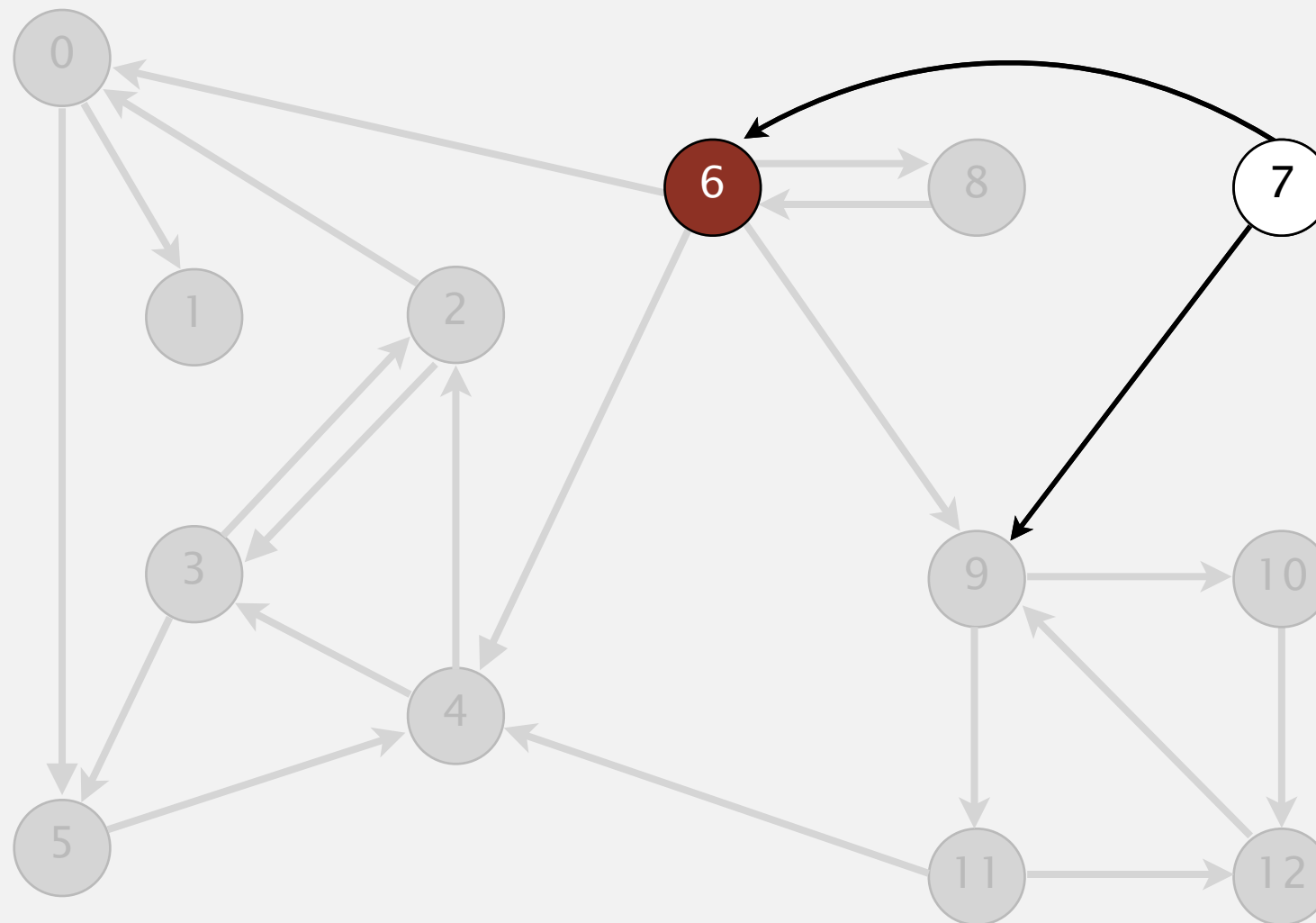
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	–
8	3
9	2
10	2
11	2
12	2

**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

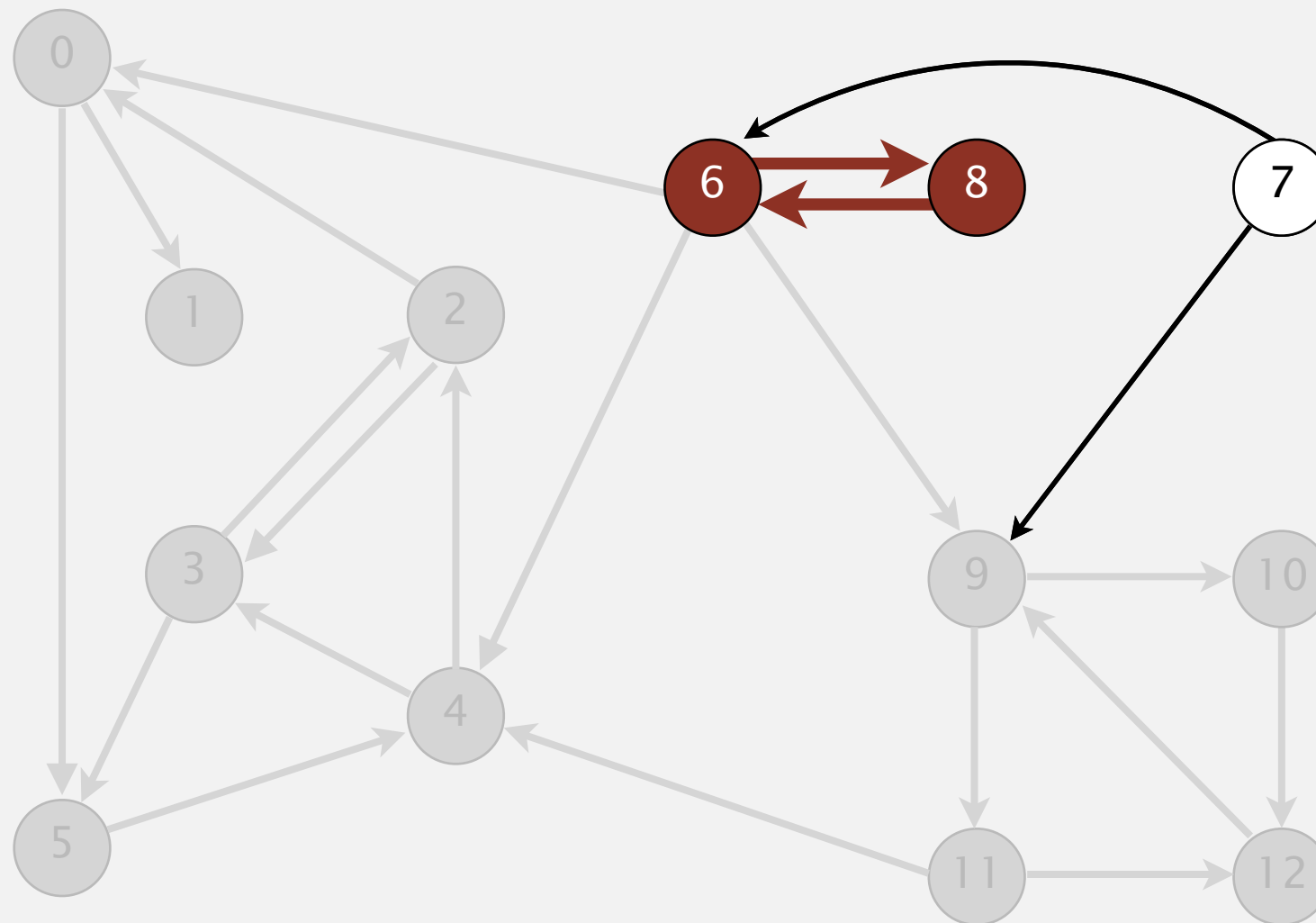
**6 done**



# 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



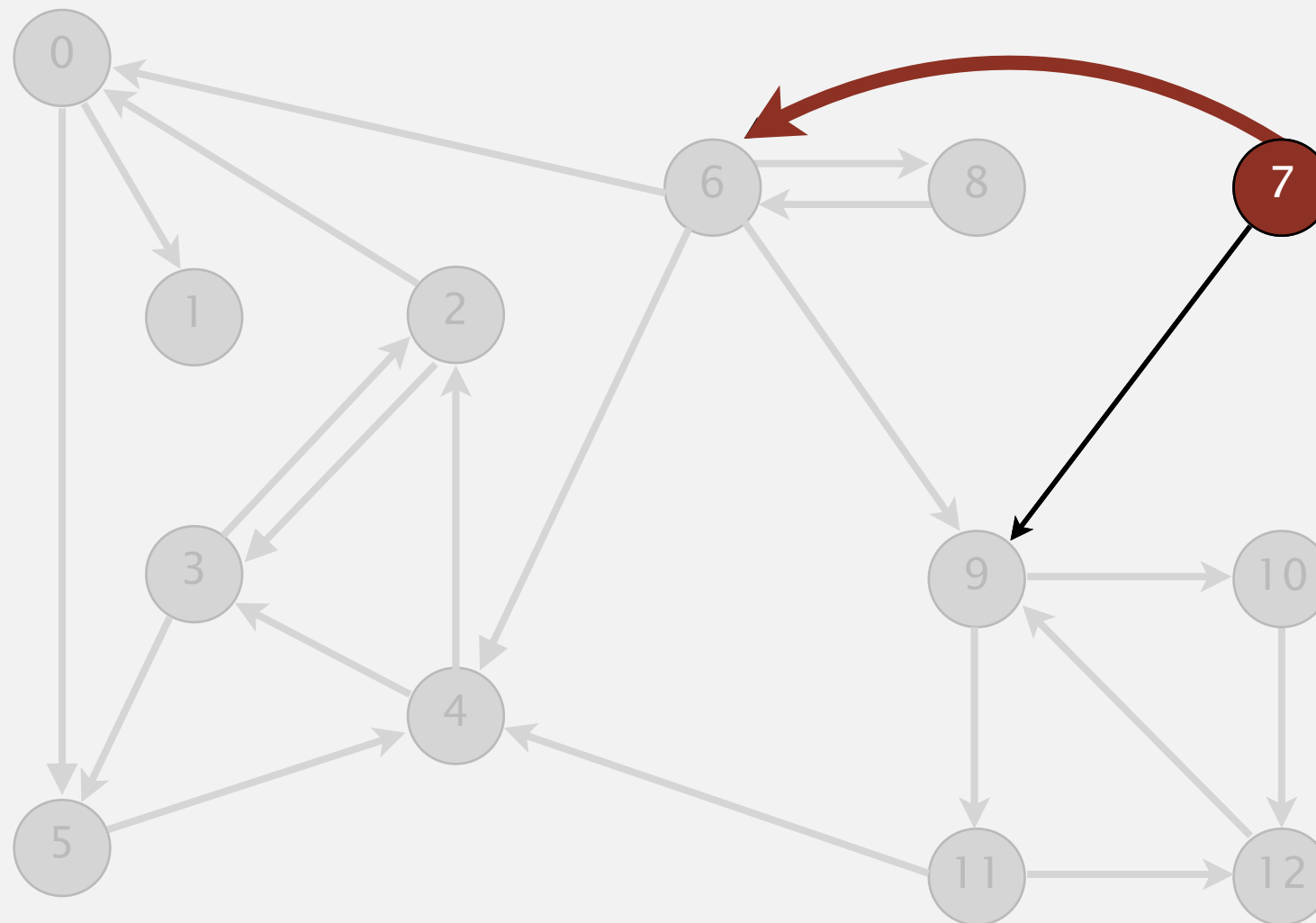
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	<b>3</b>
7	-
8	<b>3</b>
9	2
10	2
11	2
12	2

**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



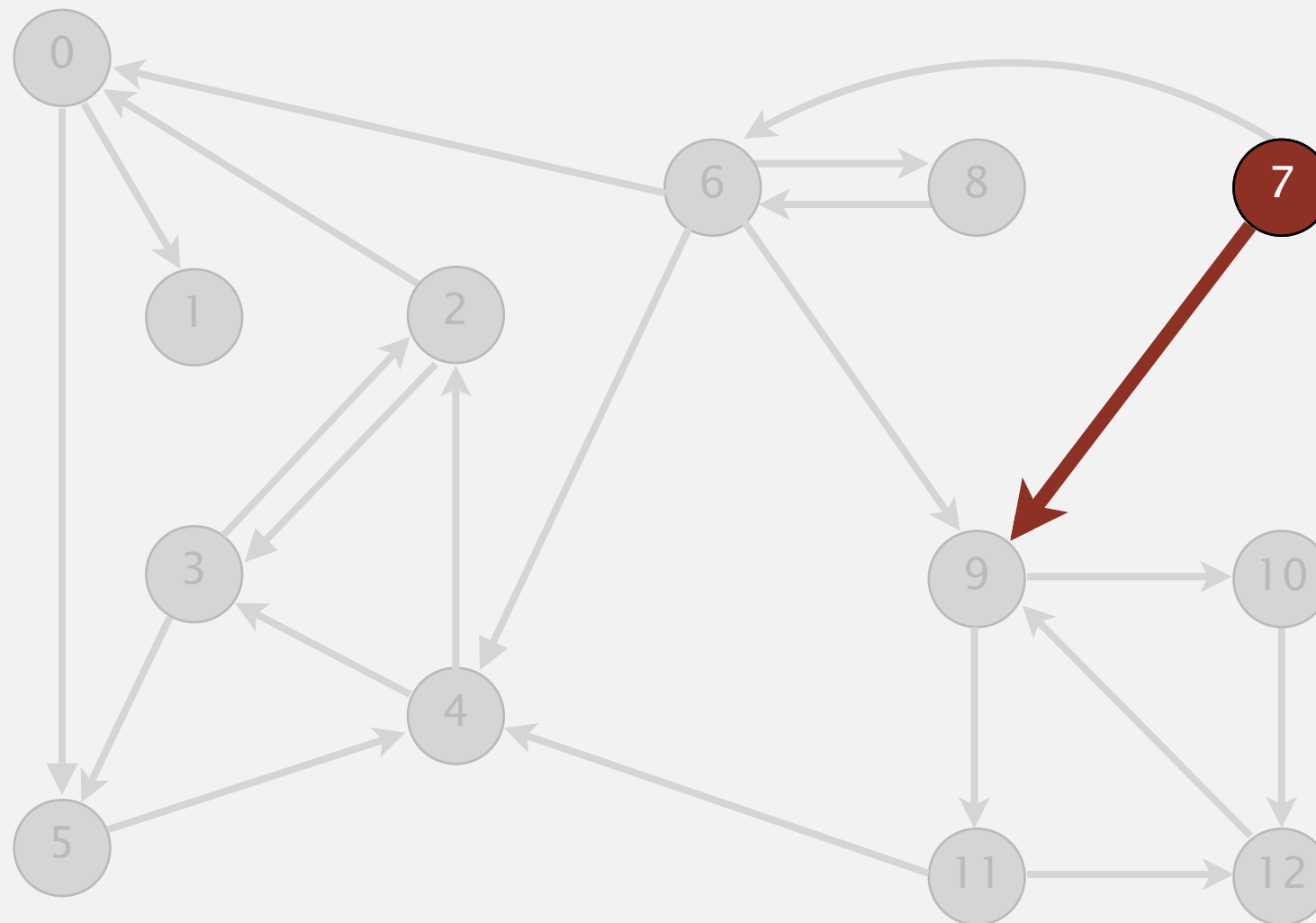
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	<b>4</b>
8	3
9	2
10	2
11	2
12	2

**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



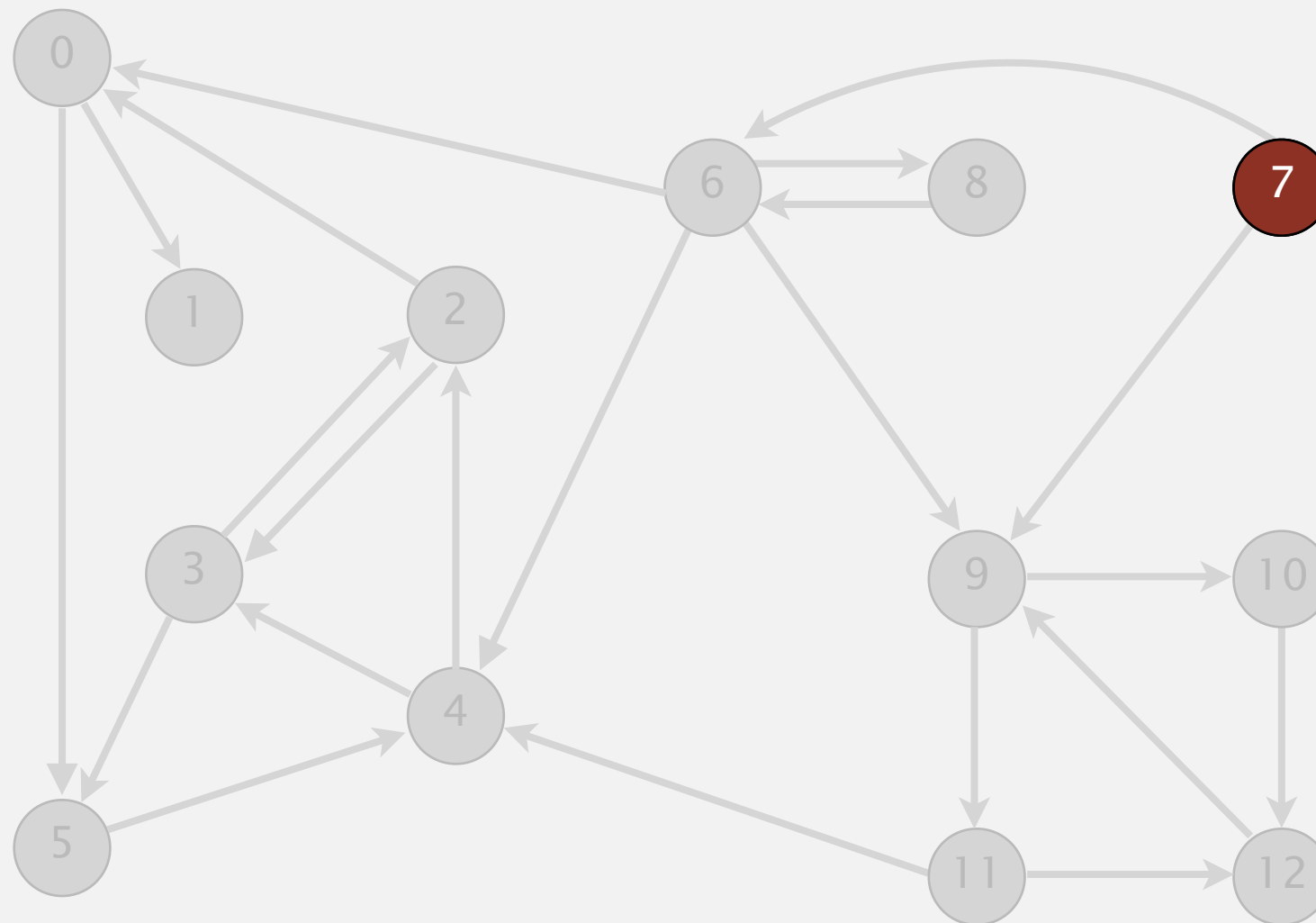
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	4
8	3
9	2
10	2
11	2
12	2

**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



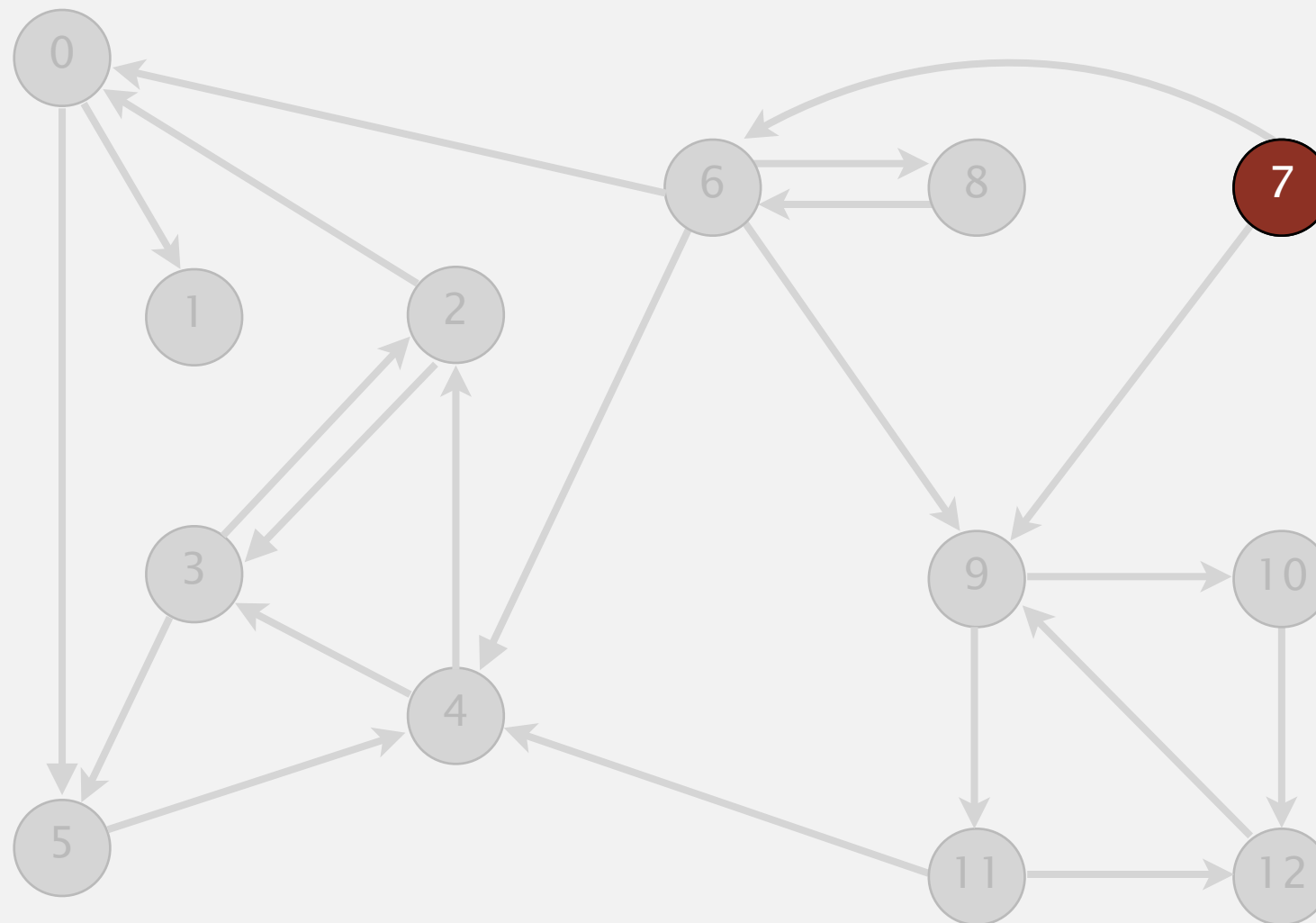
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	4
8	3
9	2
10	2
11	2
12	2

**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



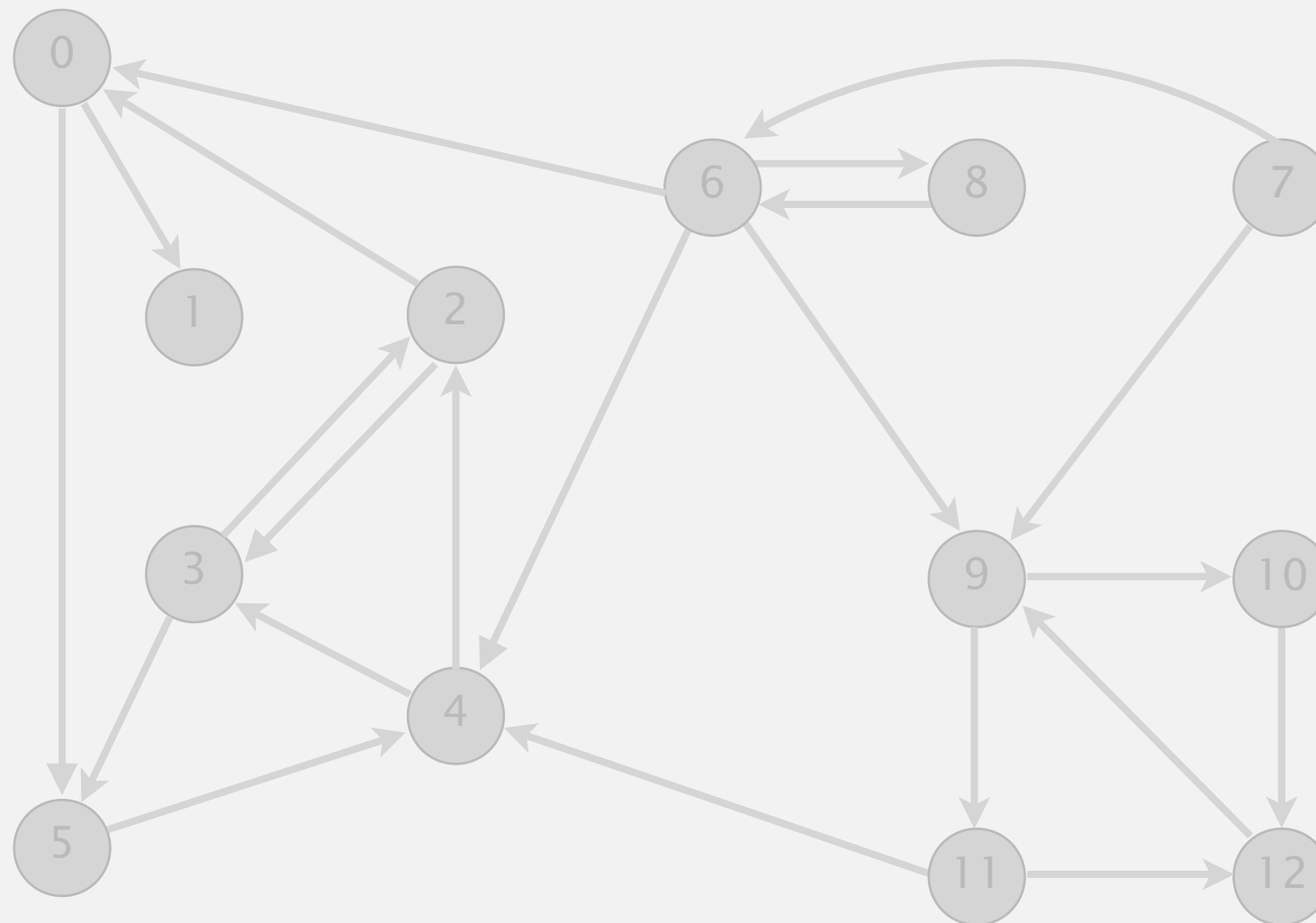
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	<b>4</b>
8	3
9	2
10	2
11	2
12	2

**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**



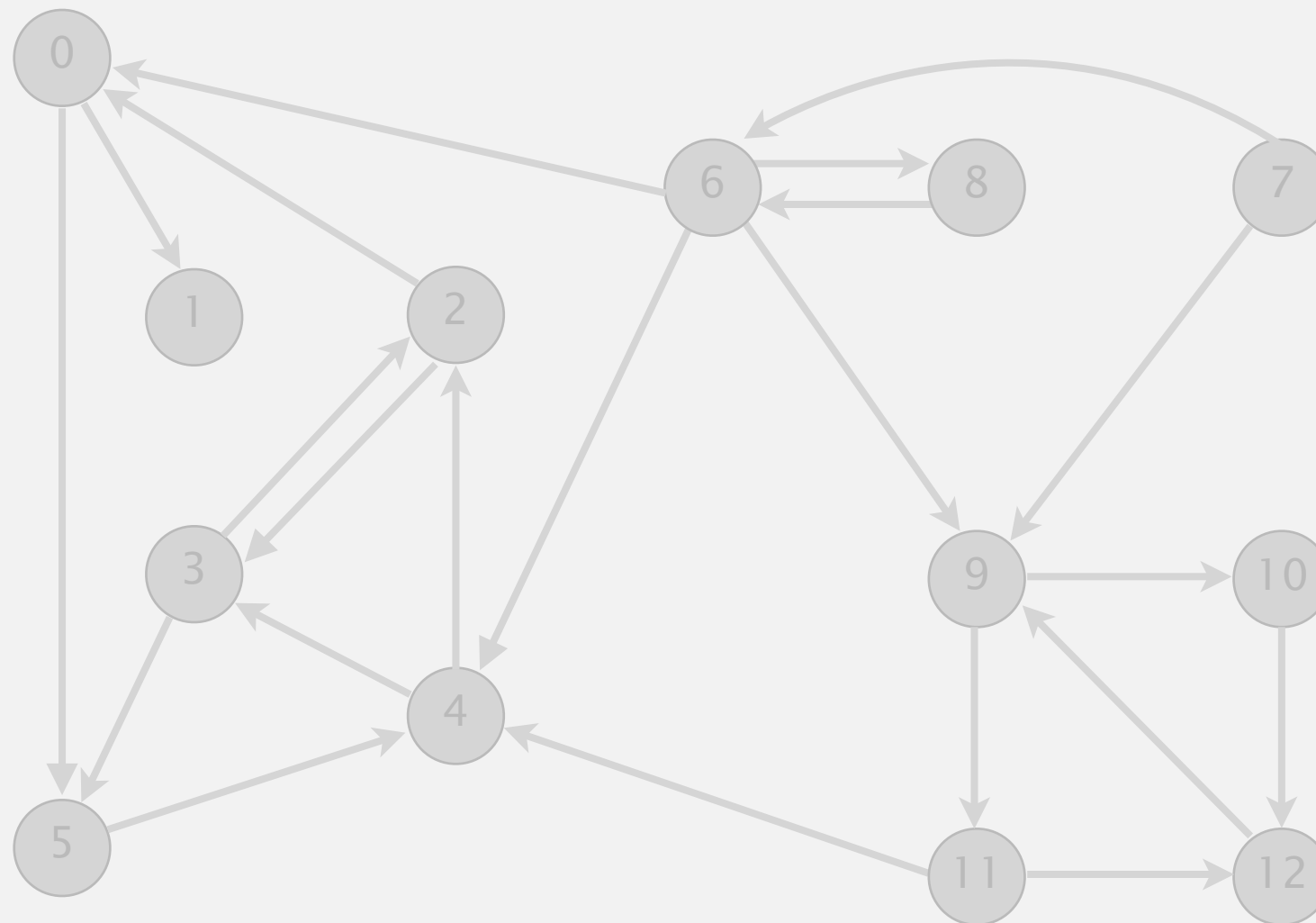
v	id[]
0	1
1	0
2	1
3	1
4	1
5	1
6	3
7	4
8	3
9	2
10	2
11	2
12	2

**check 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

done