Algorithms

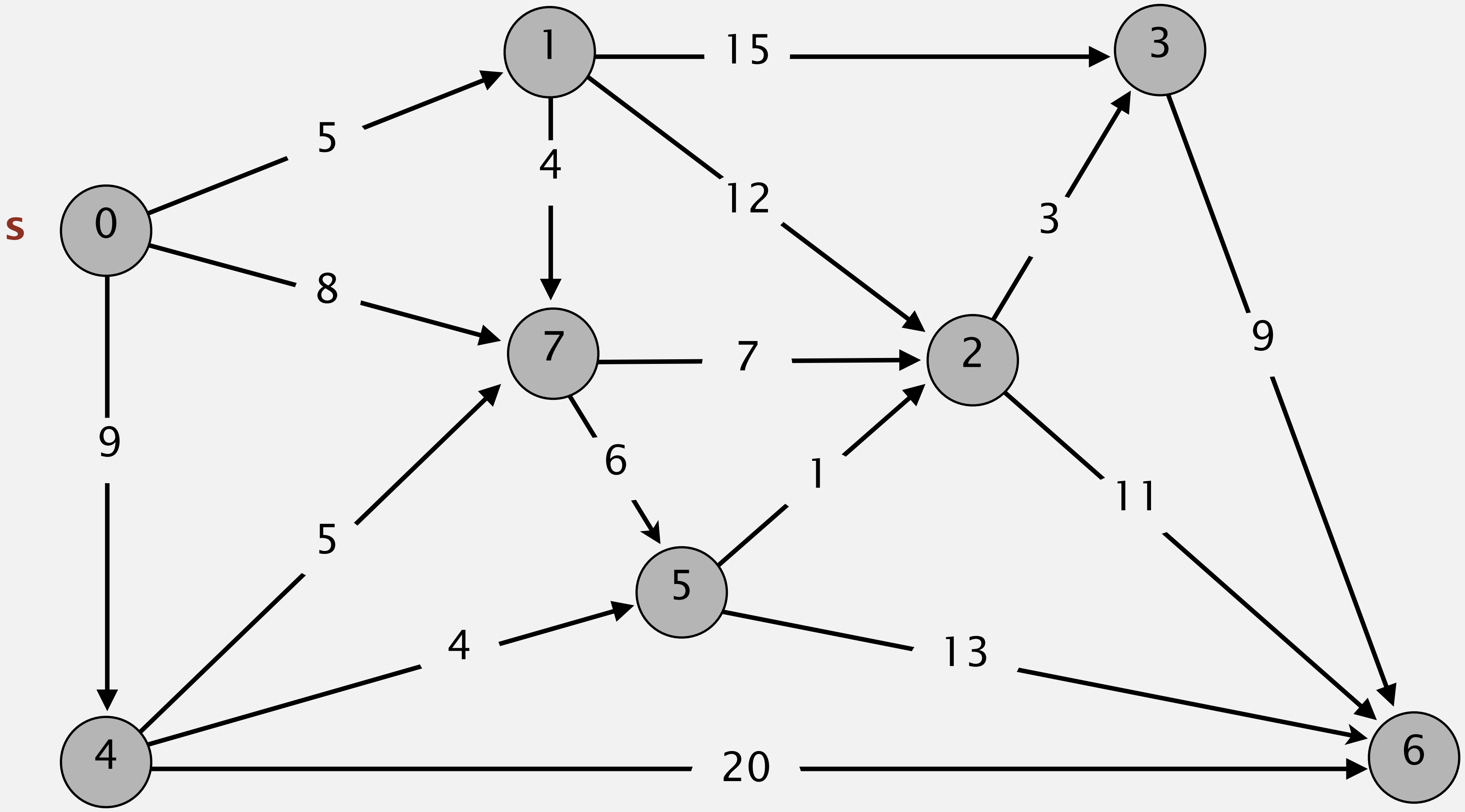
Robert Sedgewick | Kevin Wayne

http://algs4.cs.princeton.edu

Acyclic Shortest Paths Demo

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.

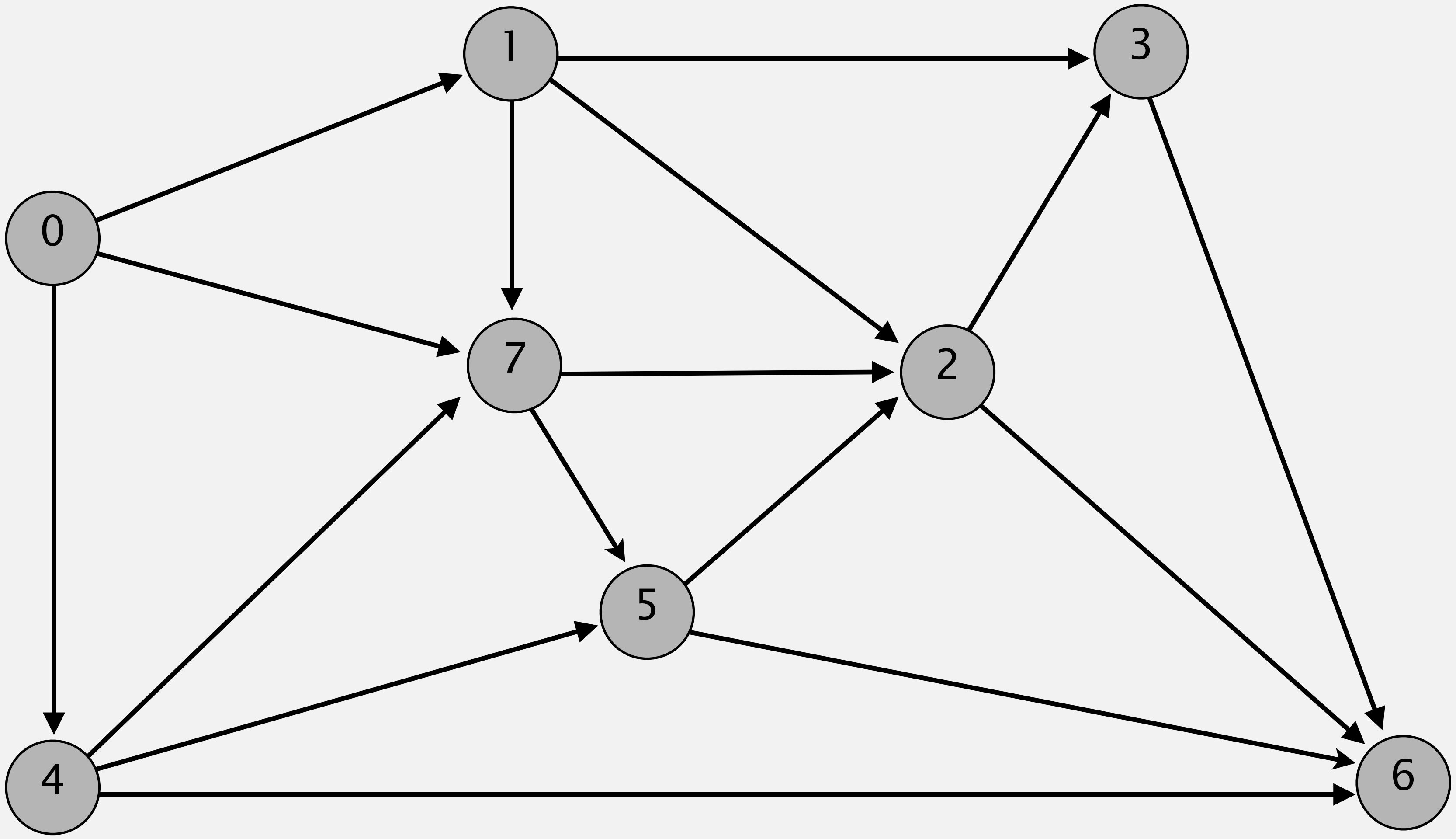


an edge-weighted DAG

0→1	5.0
0→4	9.0
0→7	8.0
1→2	12.0
1→3	15.0
1→7	4.0
2→3	3.0
2→6	11.0
3→6	9.0
4→5	4.0
4→6	20.0
4→7	5.0
5→2	1.0
5→6	13.0
7→5	6.0
7→2	7.0

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.

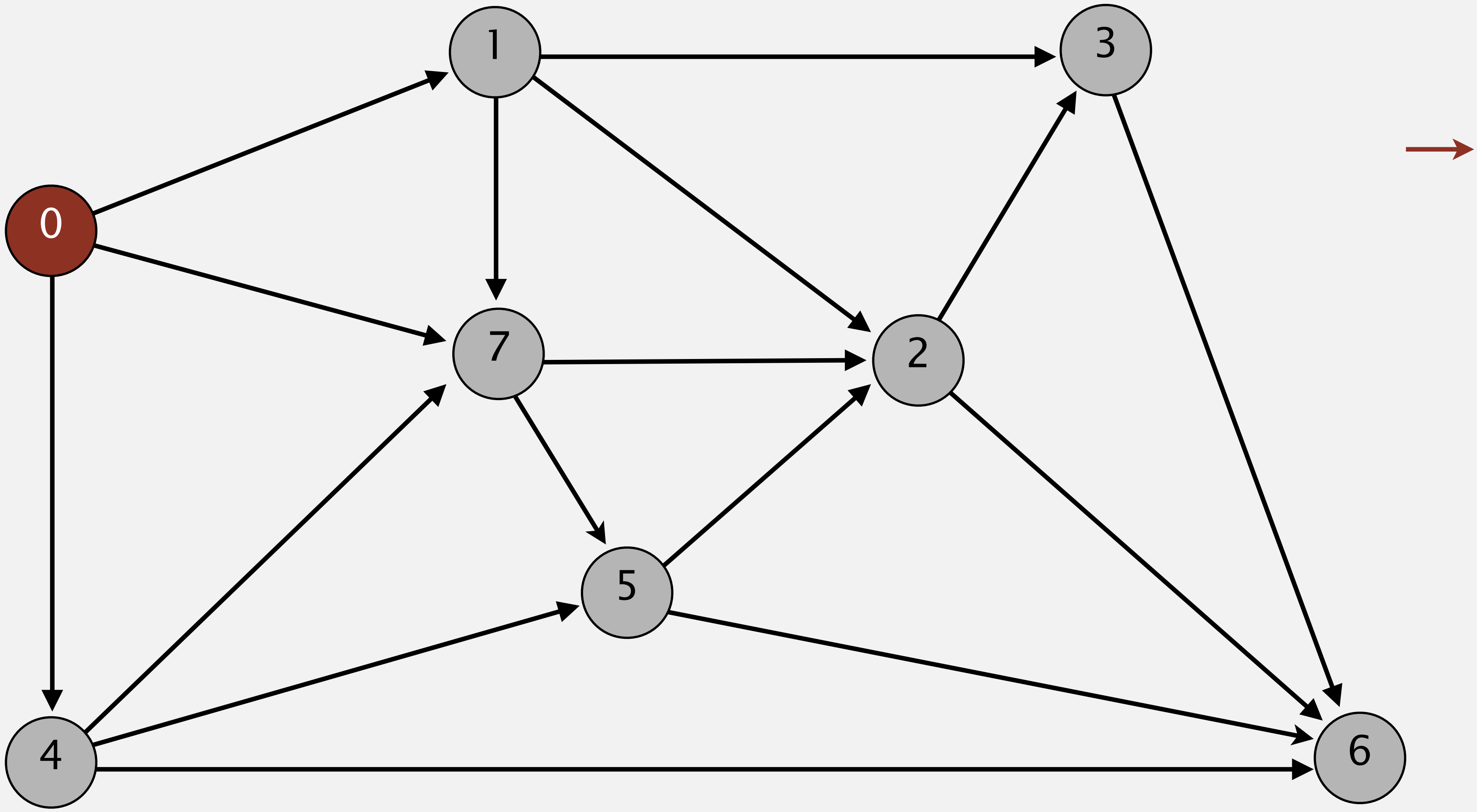


topological order: 0 1 4 7 5 2 3 6

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.

0 1 4 7 5 2 3 6

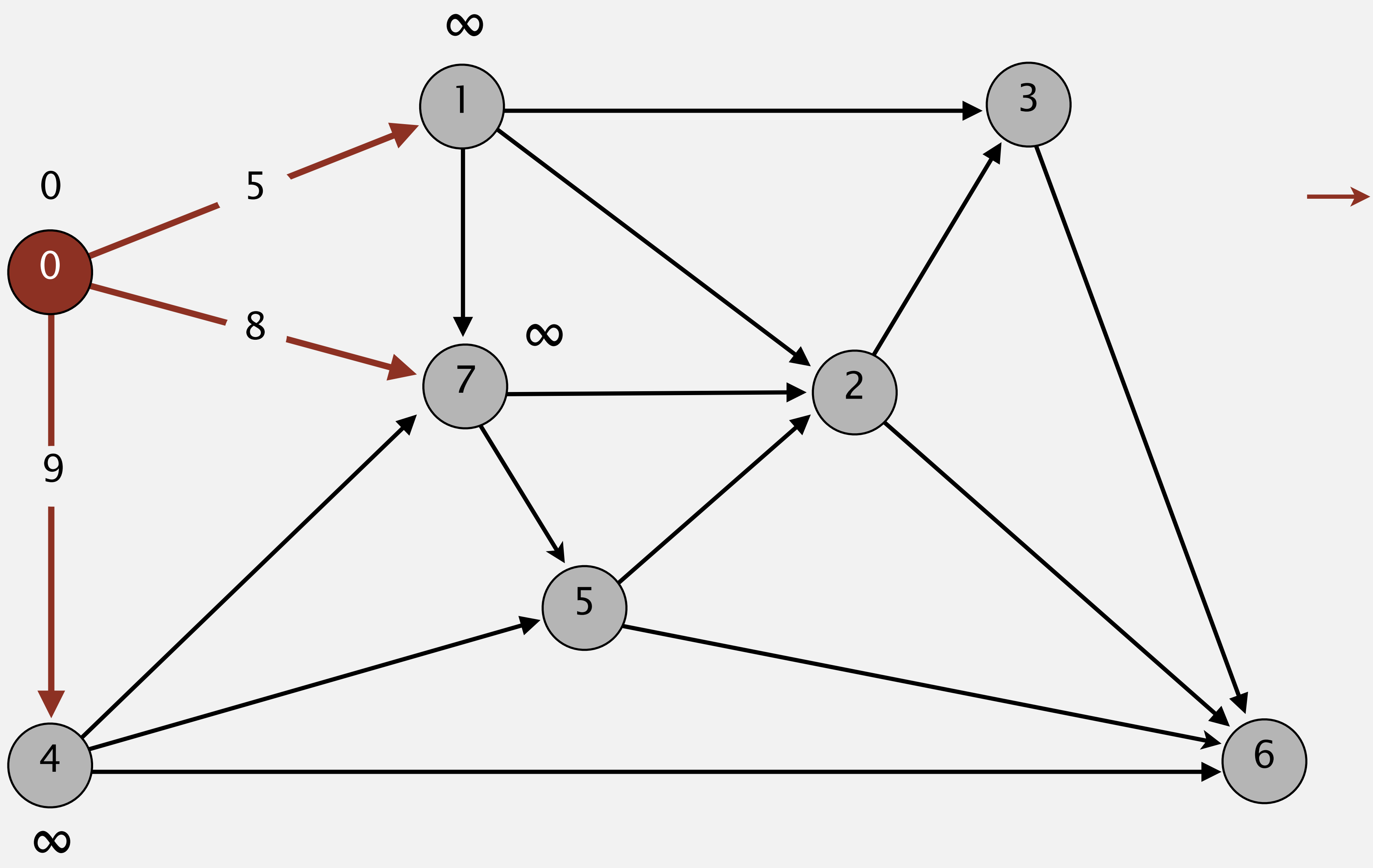


v	distTo[]	edgeTo[]
0	0.0	-
1		
2		
3		
4		
5		
6		
7		

choose vertex 0

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



↓

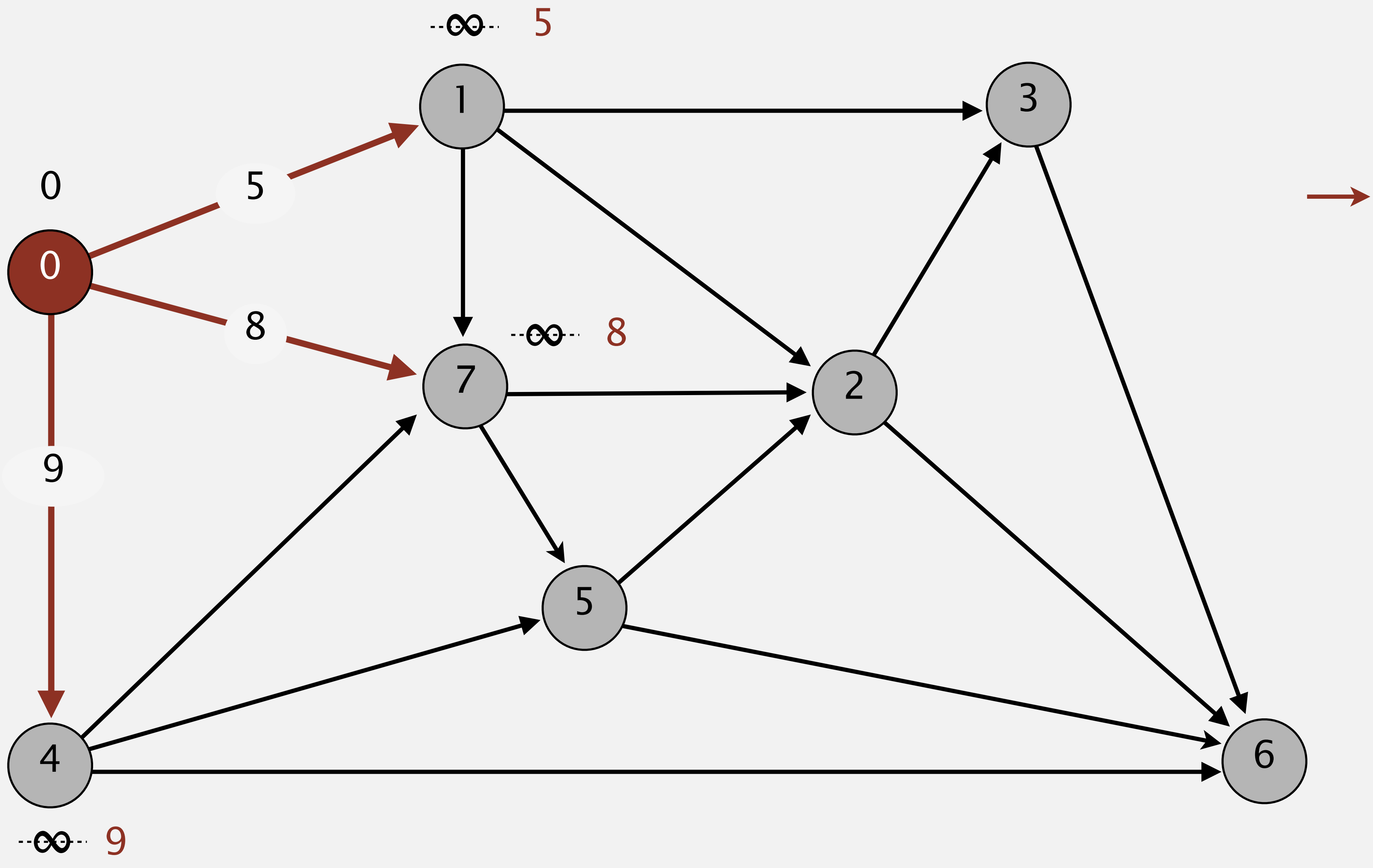
0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
→ 0	0.0	-
1		
2		
3		
4		
5		
6		
7		

relax all edges adjacent from 0

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



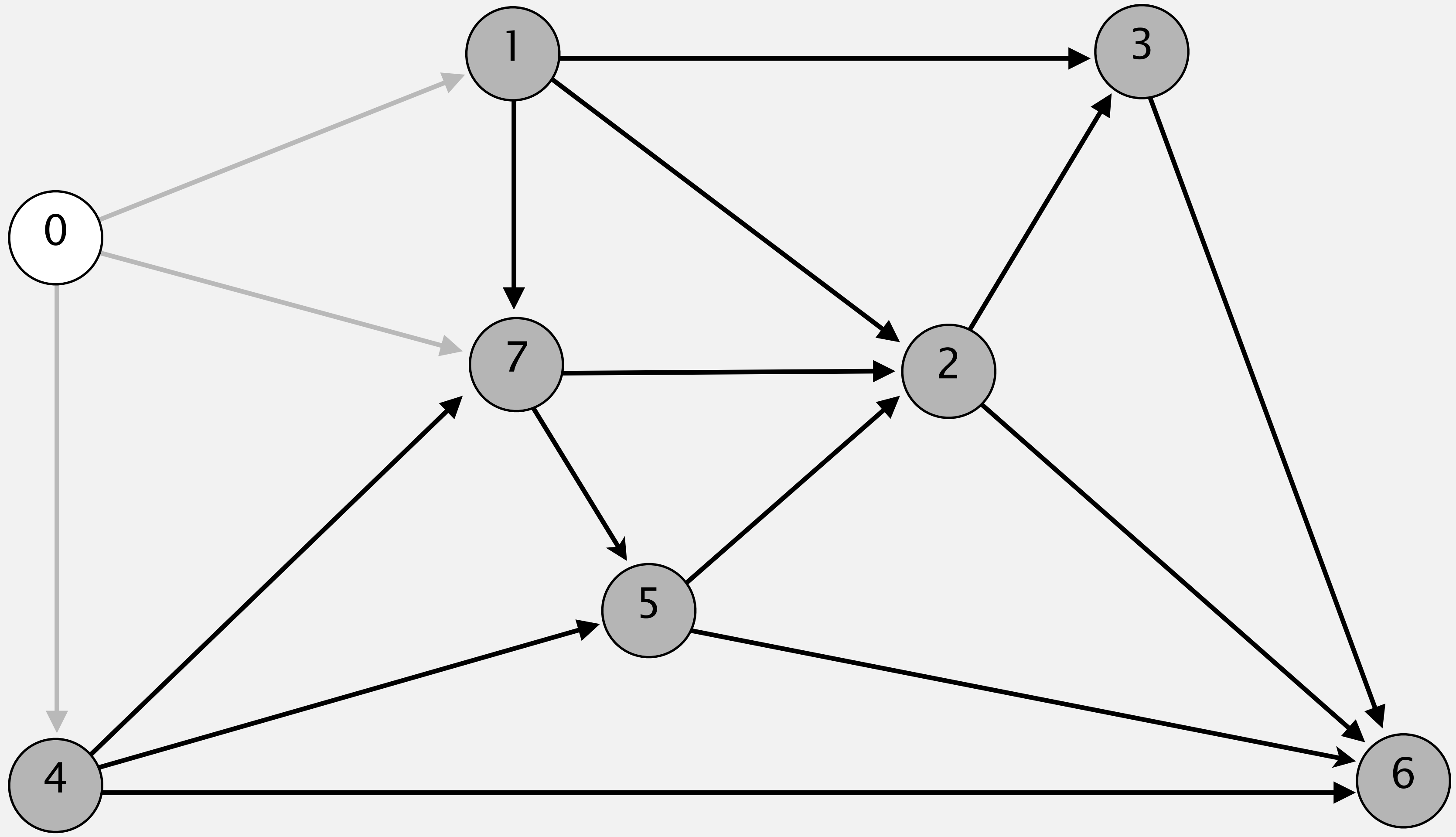
0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2		
3		
4	9.0	0→4
5		
6		
7	8.0	0→7

relax all edges adjacent from 0

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.

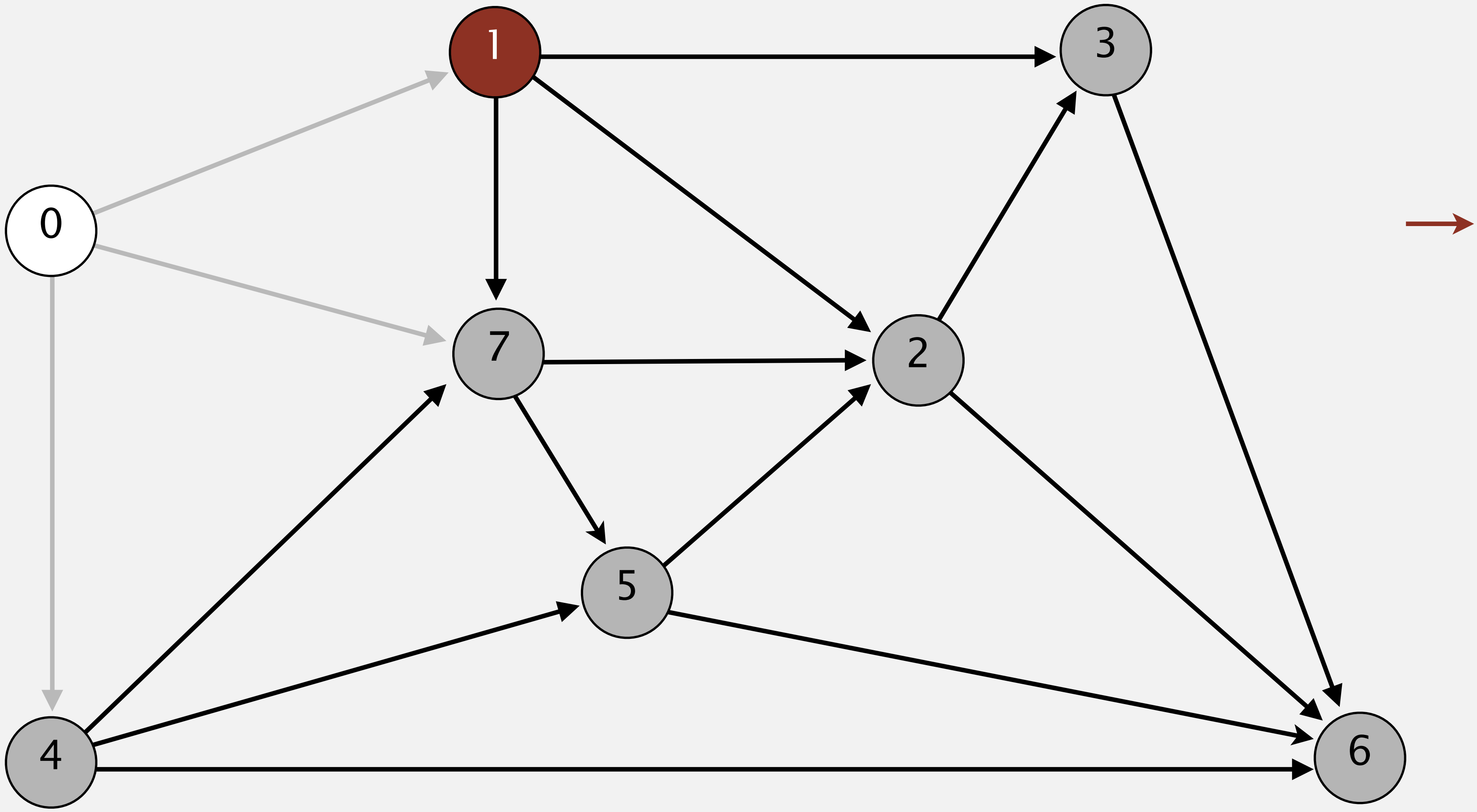


0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2		
3		
4	9.0	0→4
5		
6		
7	8.0	0→7

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



↓

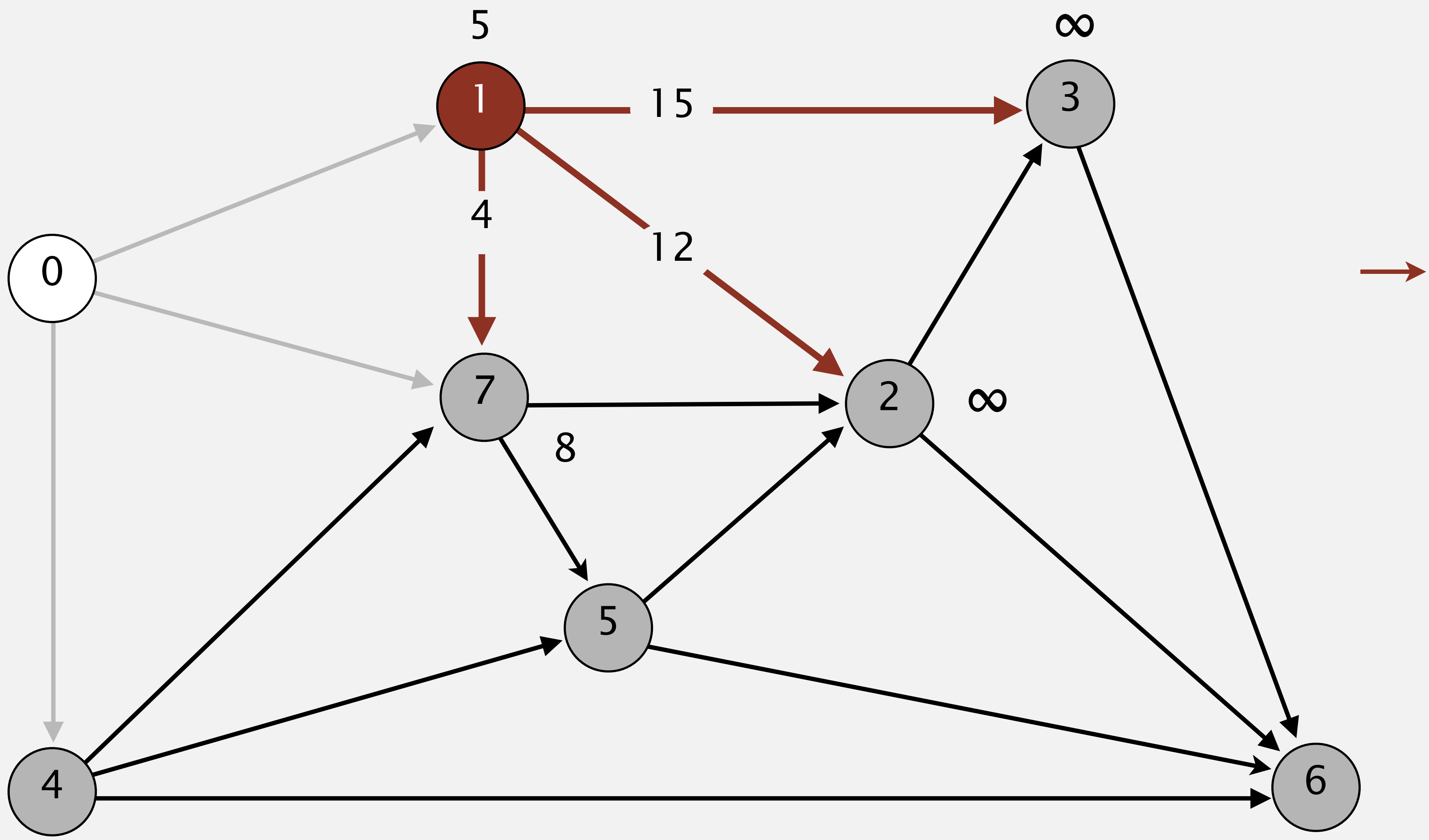
0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
→ 1	5.0	0→1
2		
3		
4	9.0	0→4
5		
6		
7	8.0	0→7

choose vertex 1

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



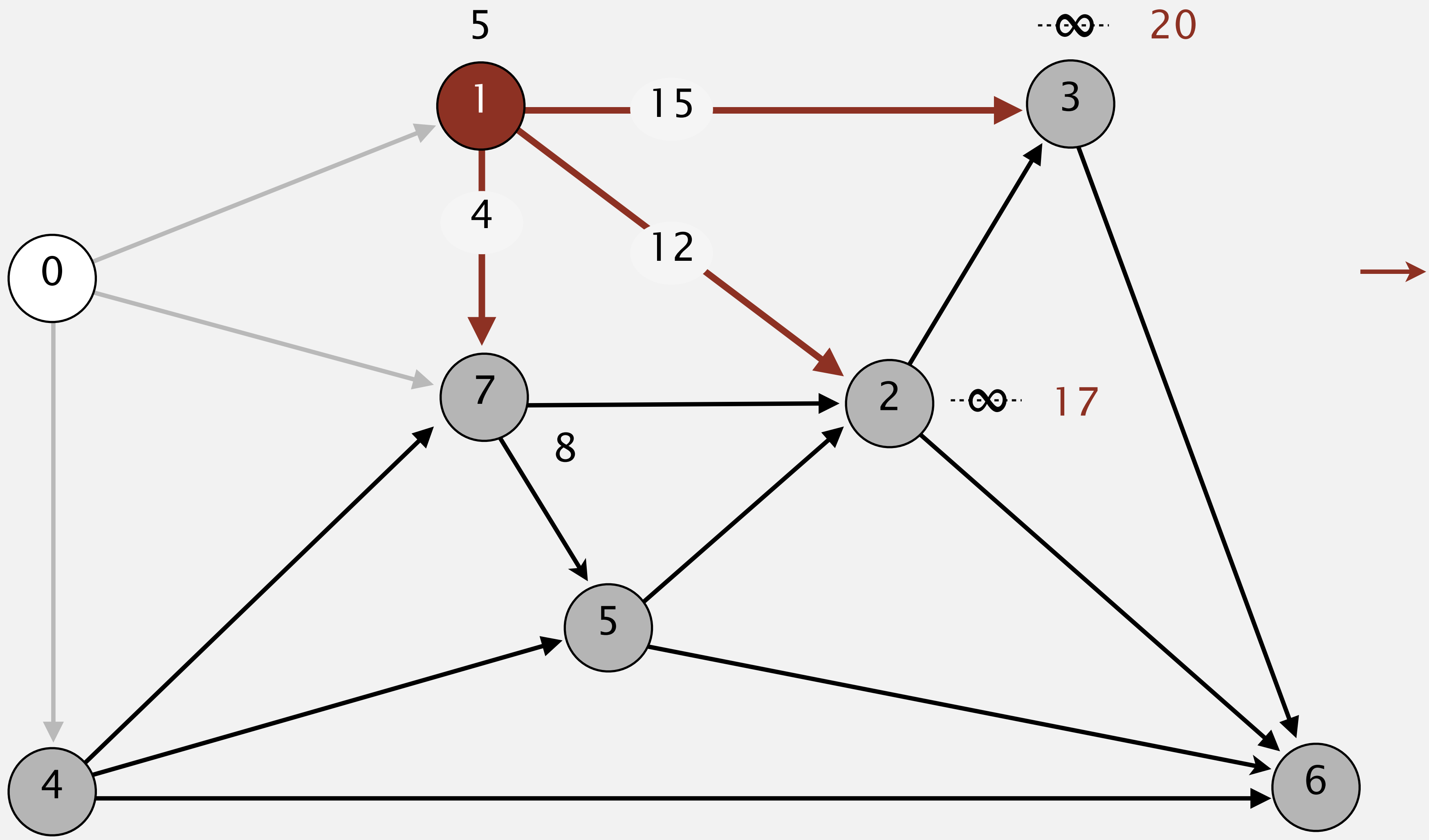
0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2		
3		
4	9.0	0→4
5		
6		
7	8.0	0→7

relax all edges adjacent from 1

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



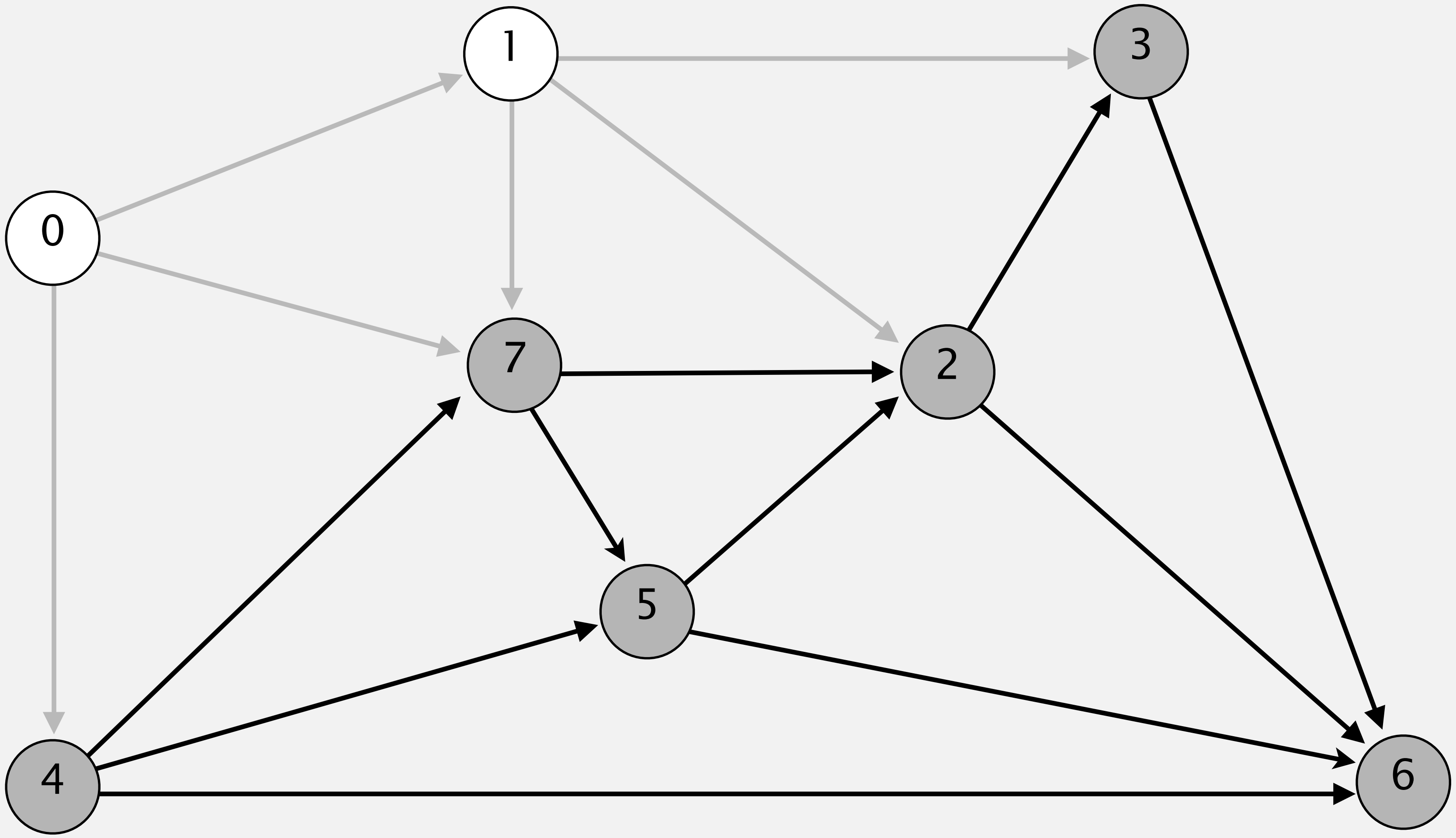
0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	17.0	1→2
3	20.0	1→3
4	9.0	0→4
5		
6		
7	8.0 ✓	0→7

relax all edges adjacent from 1

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



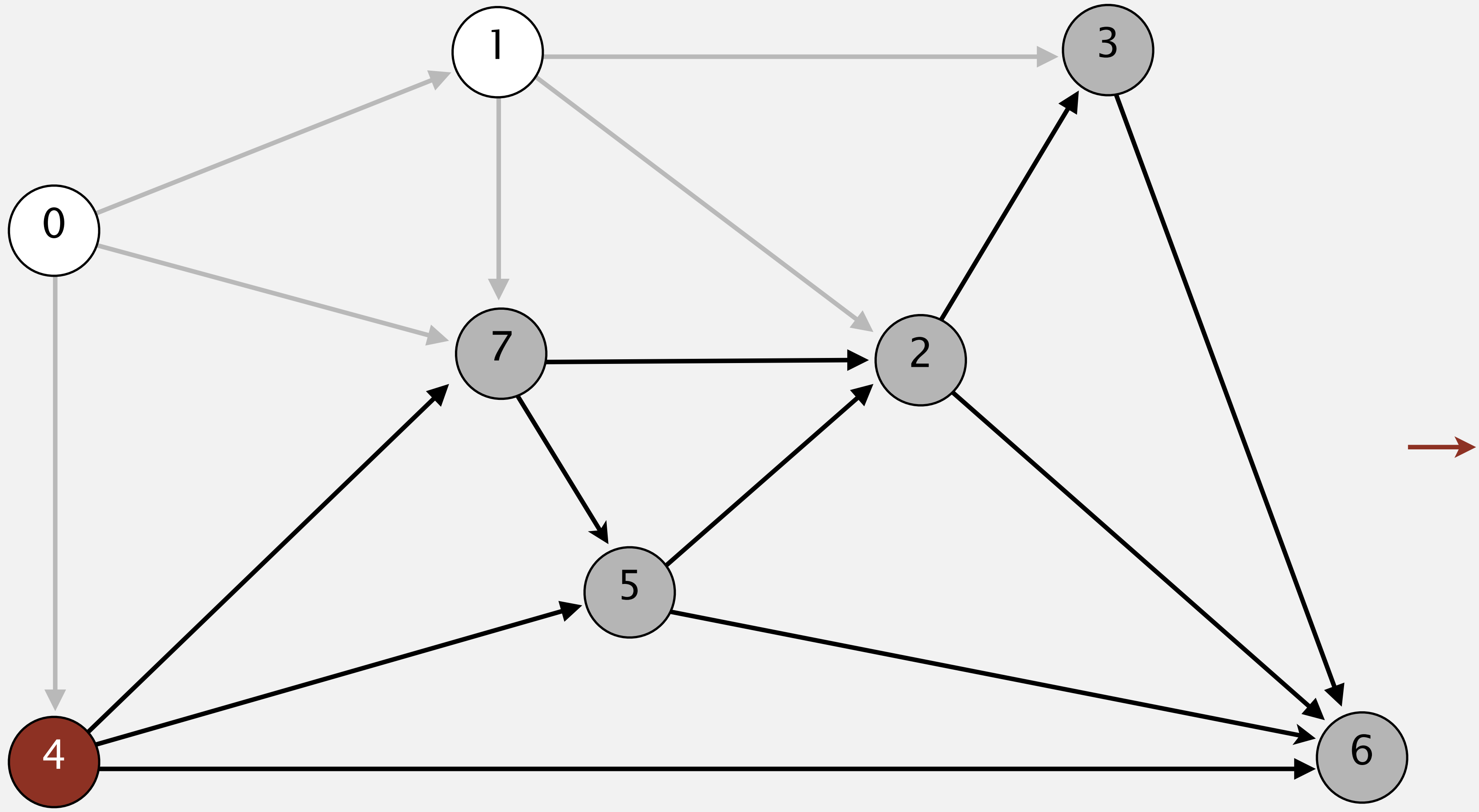
↓

0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	17.0	1→2
3	20.0	1→3
4	9.0	0→4
5		
6		
7	8.0	0→7

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



↓

0 1 4 7 5 2 3 6

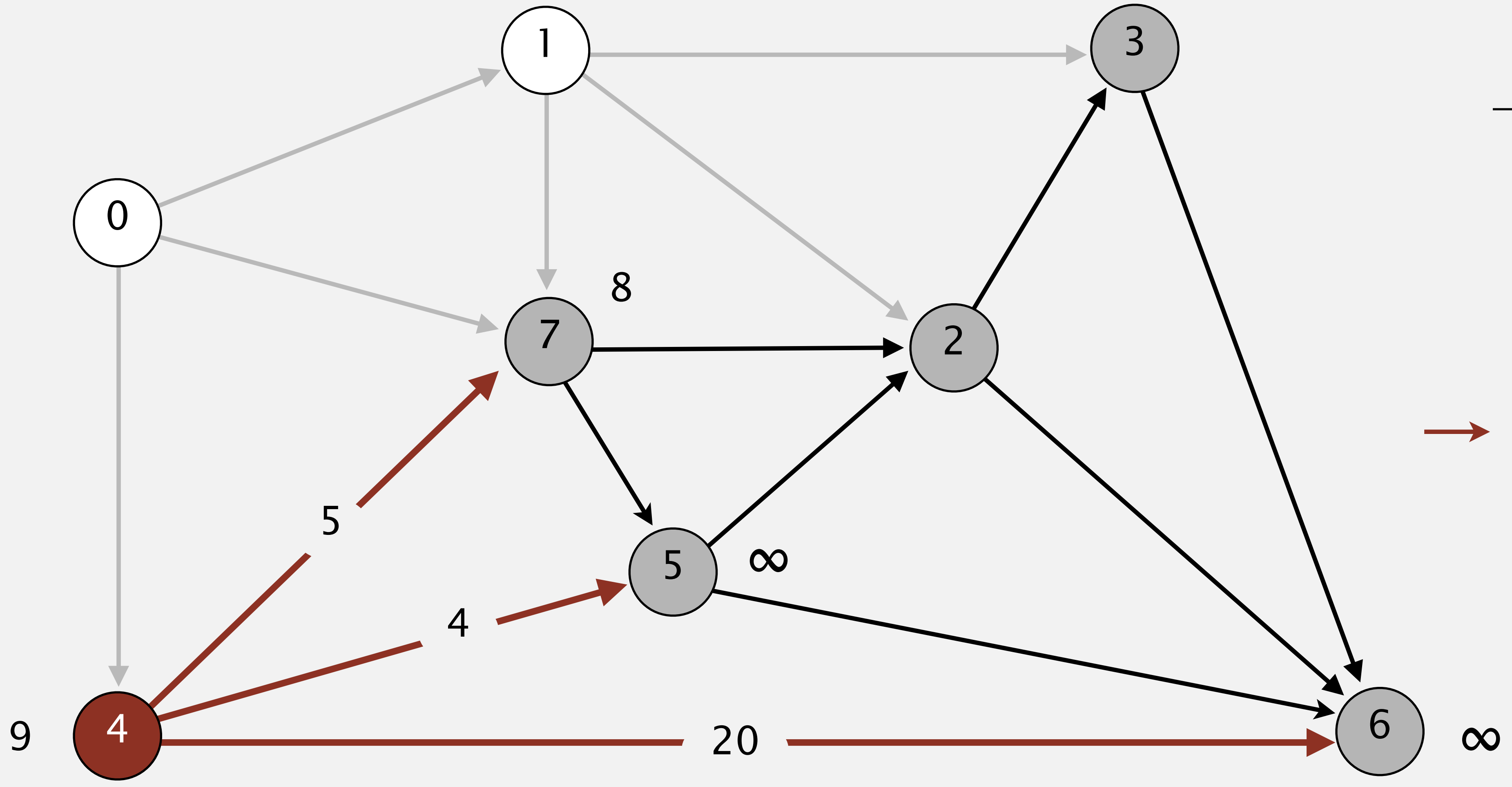
	v	distTo[]	edgeTo[]
	0	0.0	-
	1	5.0	0→1
	2	17.0	1→2
	3	20.0	1→3
→	4	9.0	0→4
	5		
	6		
	7	8.0	0→7

select vertex 4

(Dijkstra would have selected vertex 7)

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



0 1 4 7 5 2 3 6

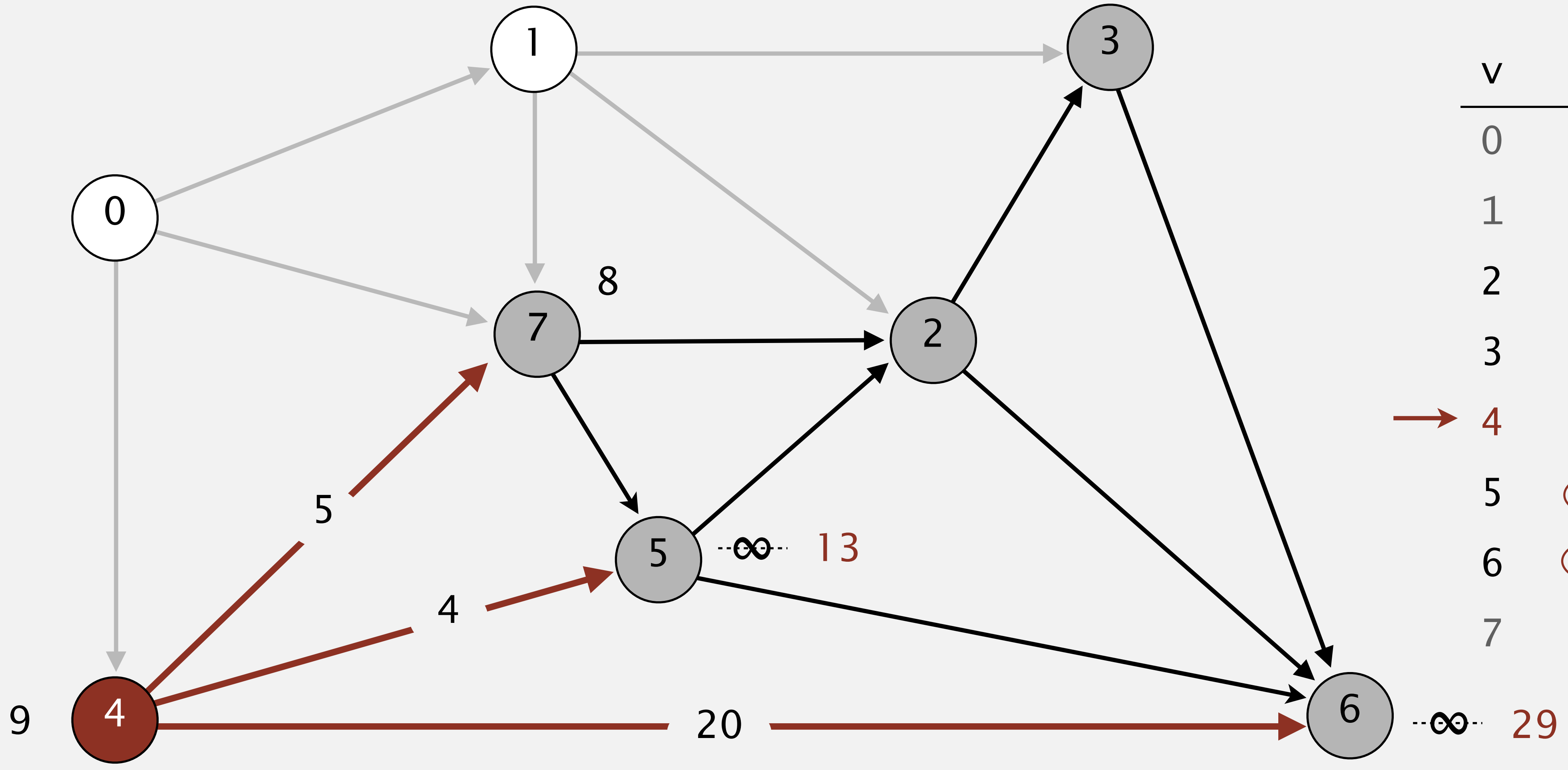
v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	17.0	1→2
3	20.0	1→3
→ 4	9.0	0→4
5		
6		
7	8.0	0→7

relax all edges adjacent from 4

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.

0 1 4 7 5 2 3 6

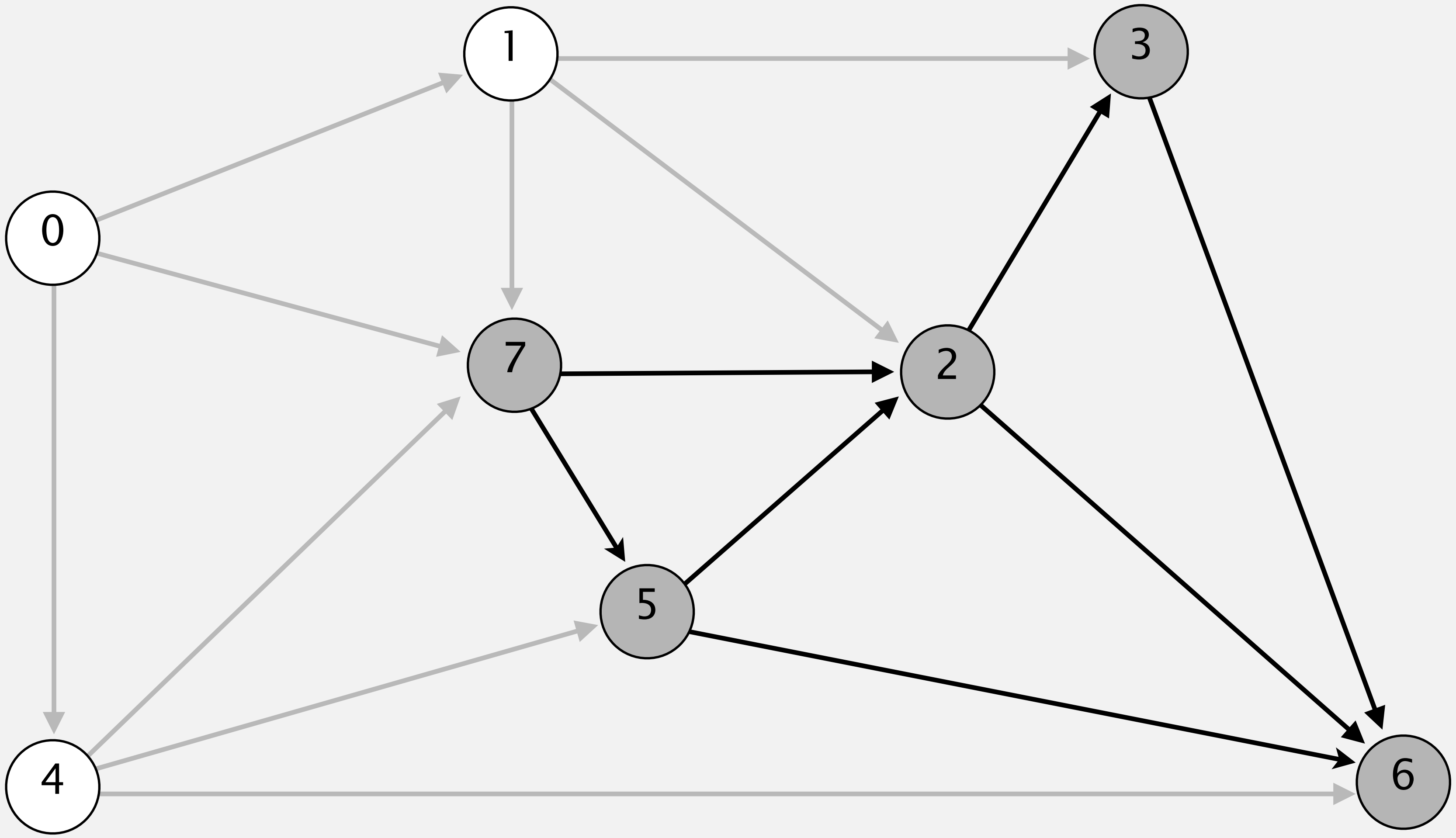


v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	17.0	1→2
3	20.0	1→3
4	9.0	0→4
5	13.0	4→5
6	29.0	4→6
7	8.0 ✔	0→7

relax all edges adjacent from 4

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



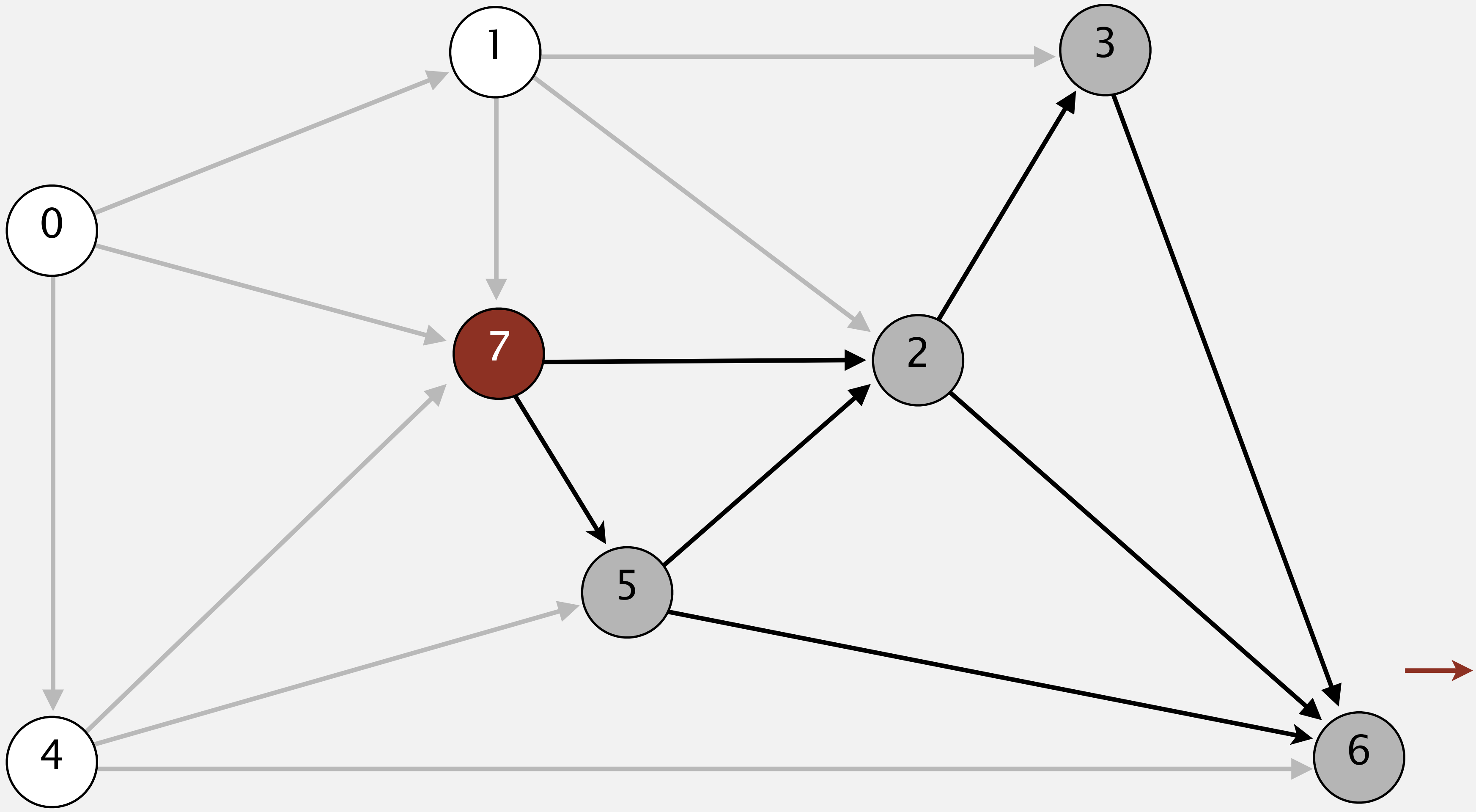
↓

0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	17.0	1→2
3	20.0	1→3
4	9.0	0→4
5	13.0	4→5
6	29.0	4→6
7	8.0	0→7

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



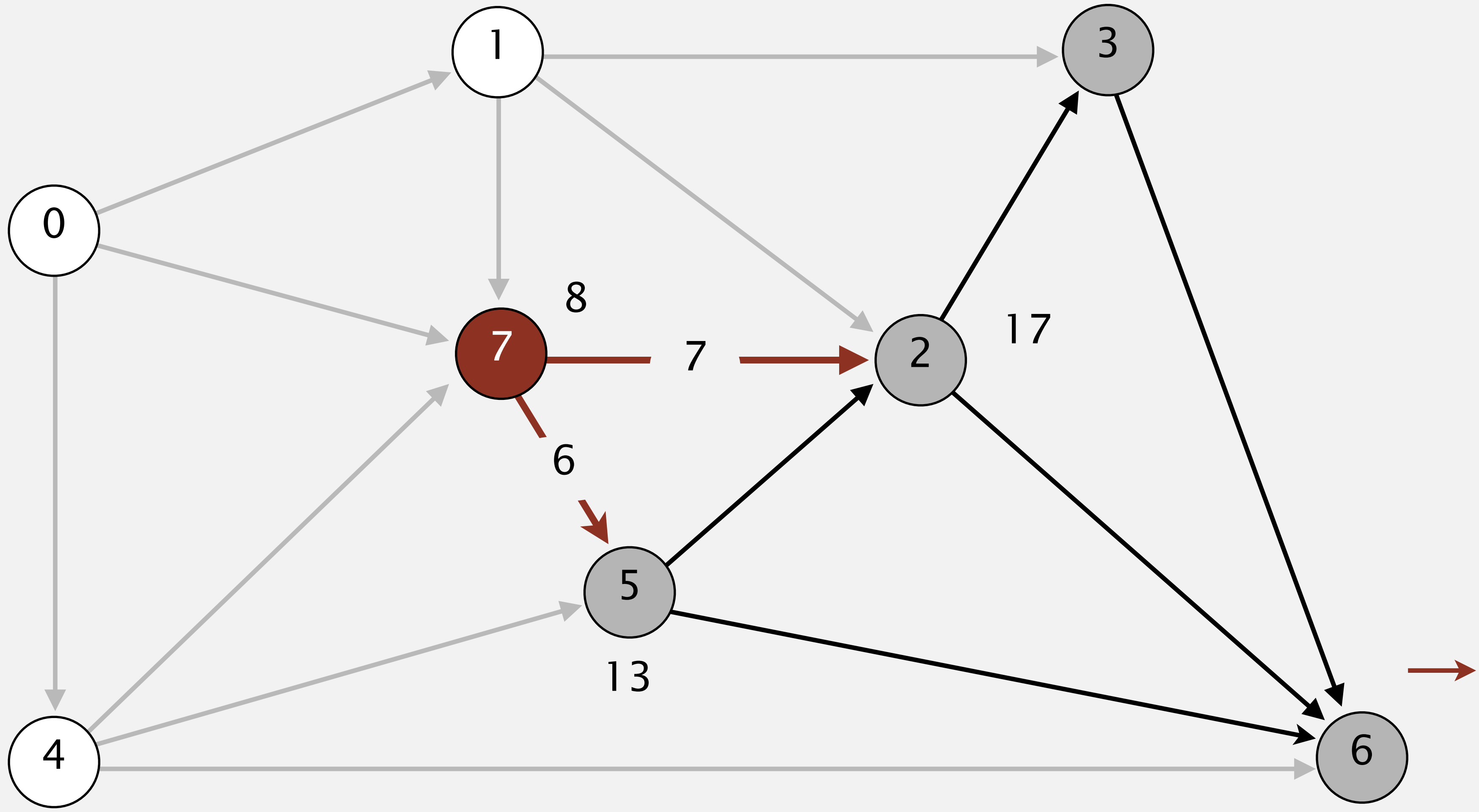
0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	17.0	1→2
3	20.0	1→3
4	9.0	0→4
5	13.0	4→5
6	29.0	4→6
7	8.0	0→7

choose vertex 7

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



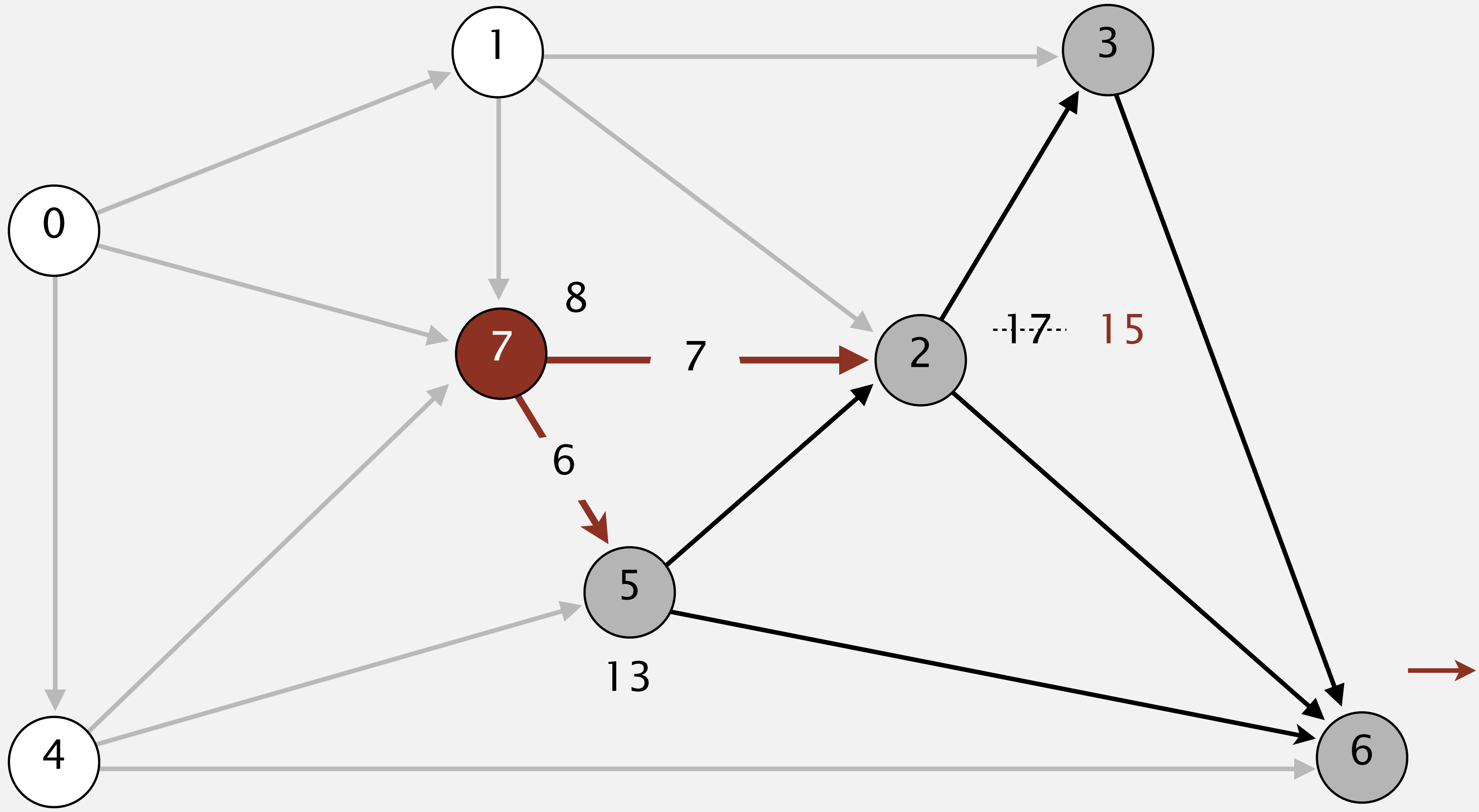
0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	17.0	1→2
3	20.0	1→3
4	9.0	0→4
5	13.0	4→5
6	29.0	4→6
7	8.0	0→7

relax all edges adjacent from 7

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



↓

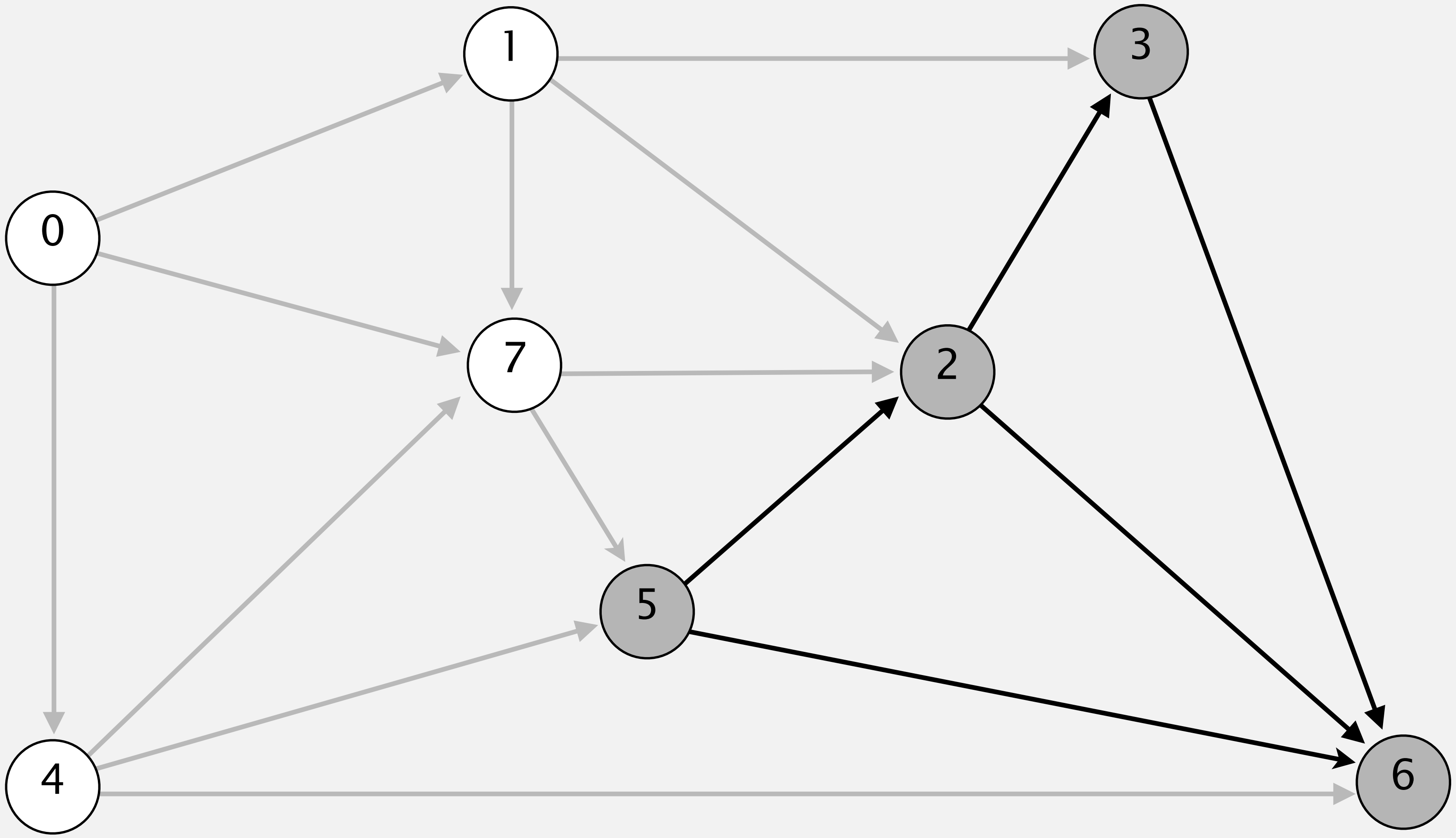
0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	15.0	7→2
3	20.0	1→3
4	9.0	0→4
5	13.0 ✔	4→5
6	29.0	4→6
→ 7	8.0	0→7

relax all edges adjacent from 7

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.

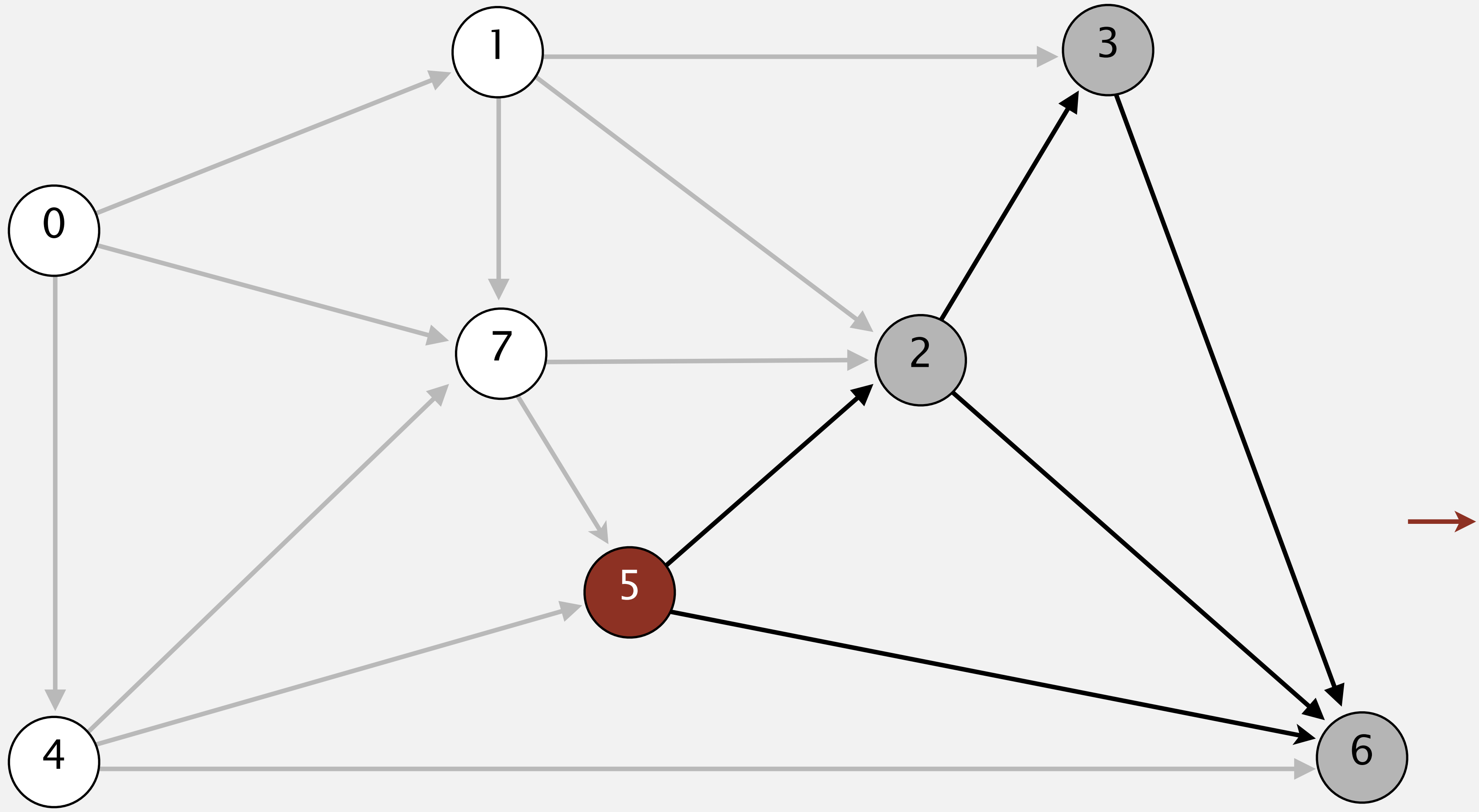


0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	15.0	7→2
3	20.0	1→3
4	9.0	0→4
5	13.0	4→5
6	29.0	4→6
7	8.0	0→7

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



↓

0 1 4 7 5 2 3 6

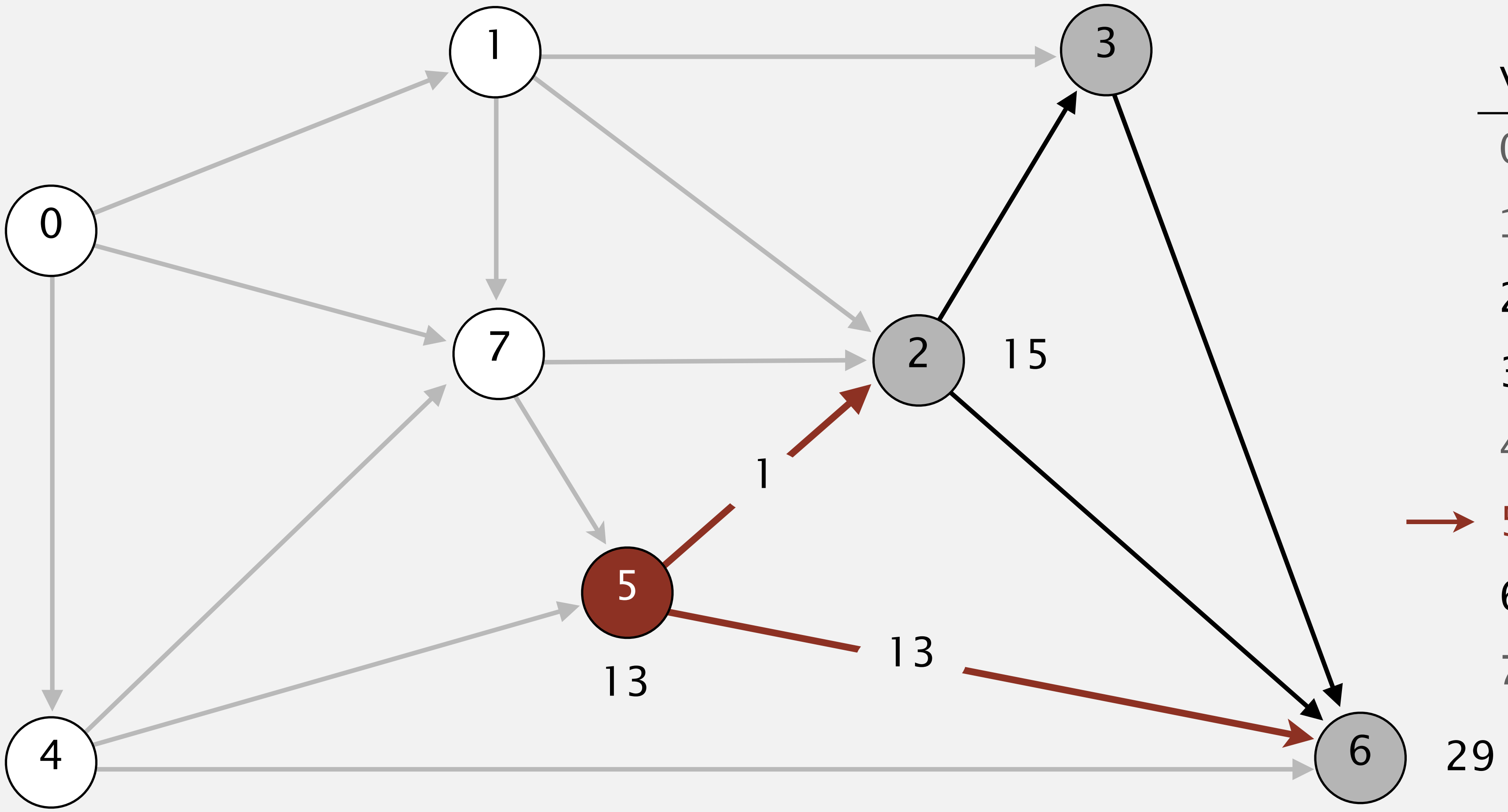
v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	15.0	7→2
3	20.0	1→3
4	9.0	0→4
→ 5	13.0	4→5
6	29.0	4→6
7	8.0	0→7

select vertex 5

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.

0 1 4 7 5 2 3 6

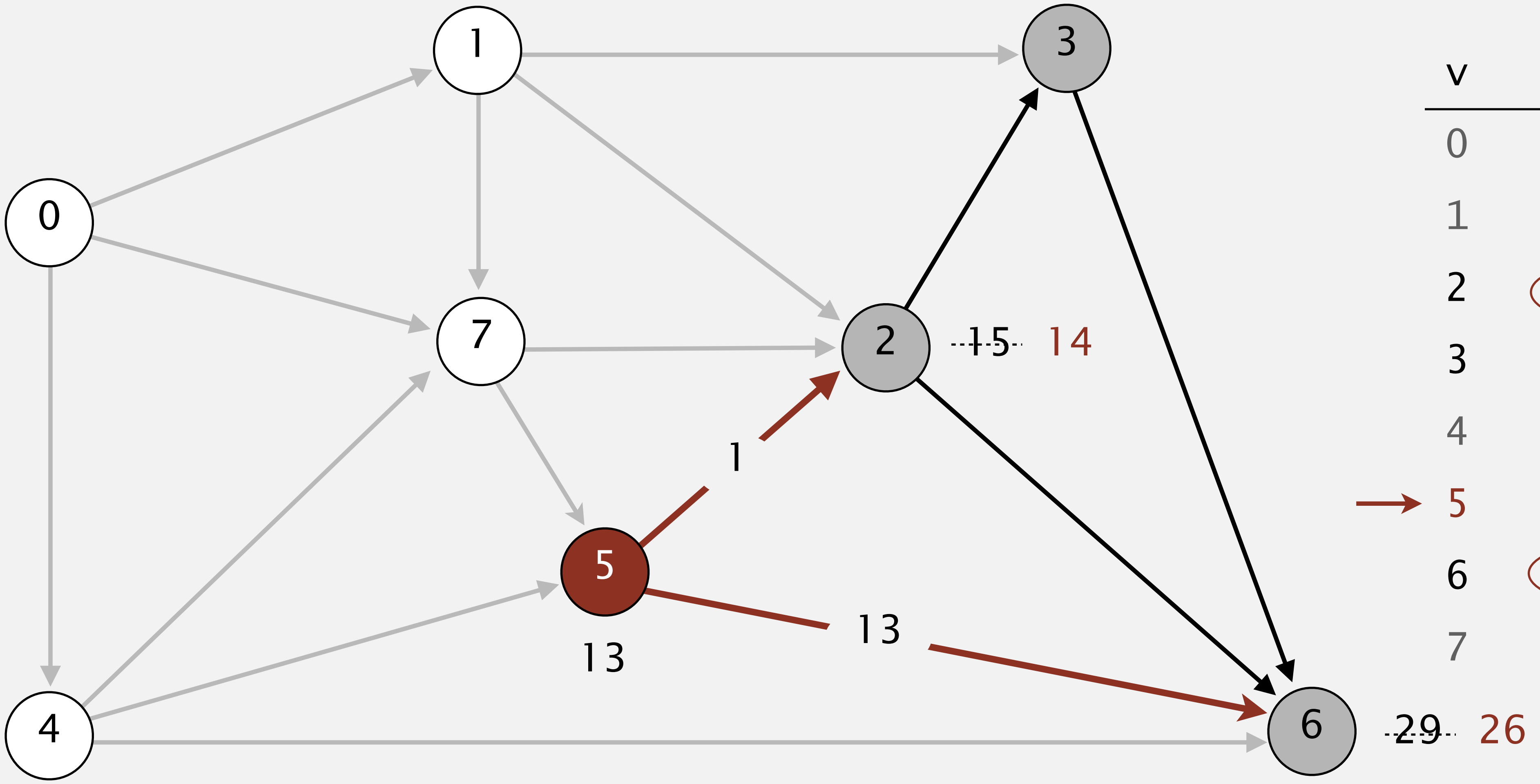


v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	15.0	7→2
3	20.0	1→3
4	9.0	0→4
5	13.0	4→5
6	29.0	4→6
7	8.0	0→7

relax all edges adjacent from 5

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



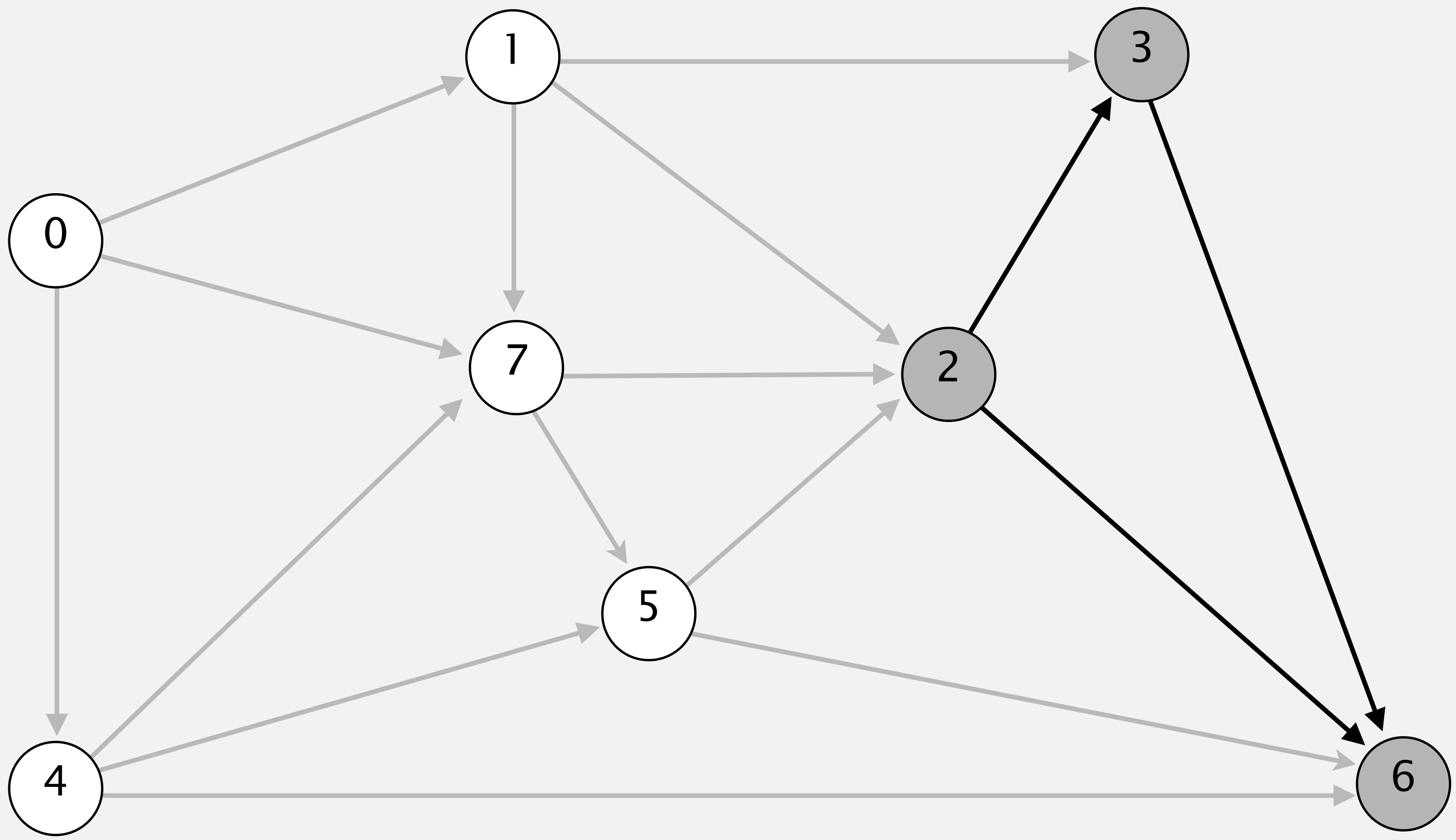
0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	14.0	5→2
3	20.0	1→3
4	9.0	0→4
5	13.0	4→5
6	26.0	5→6
7	8.0	0→7

relax all edges adjacent from 5

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.

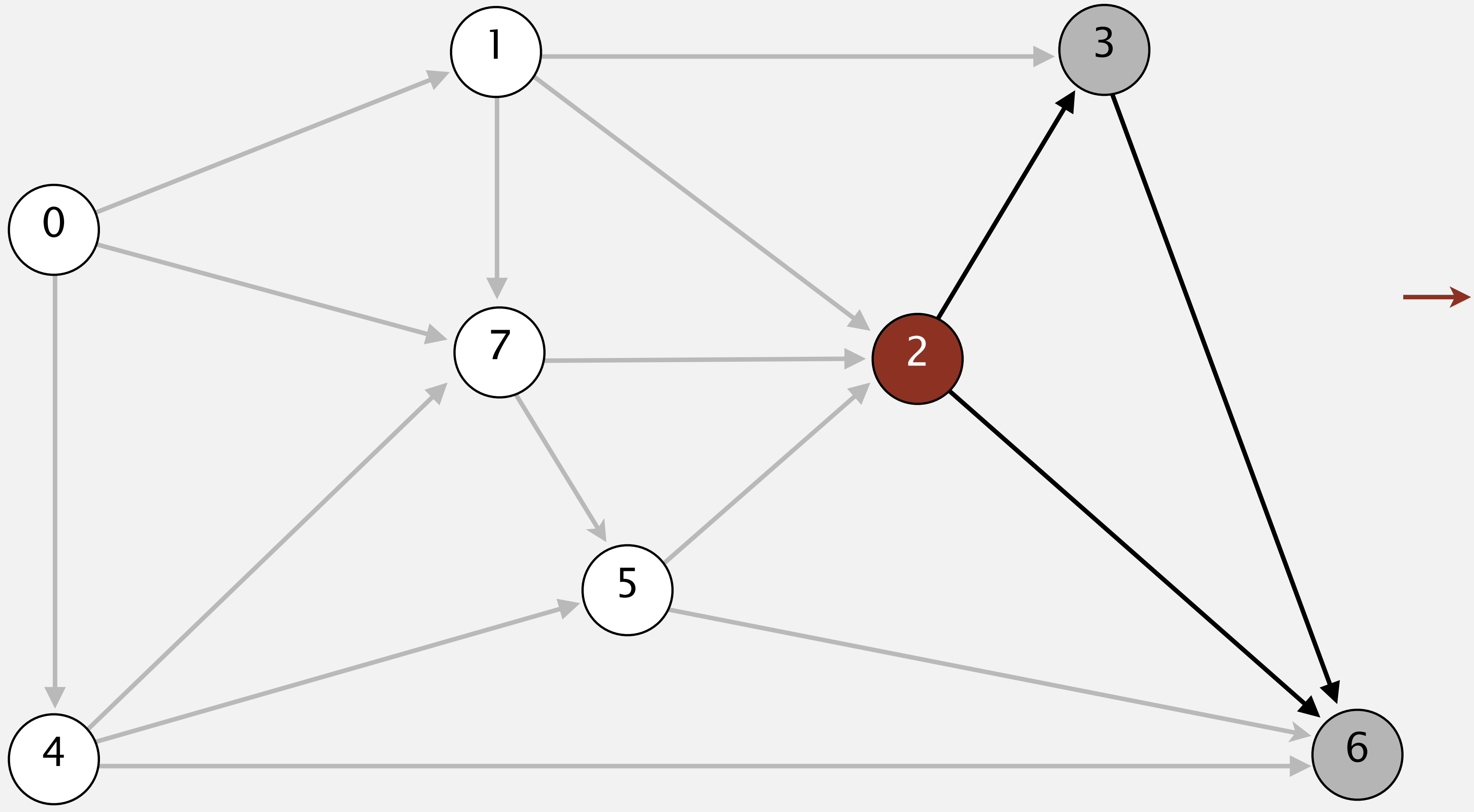


0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	14.0	5→2
3	20.0	1→3
4	9.0	0→4
5	13.0	4→5
6	26.0	5→6
7	8.0	0→7

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



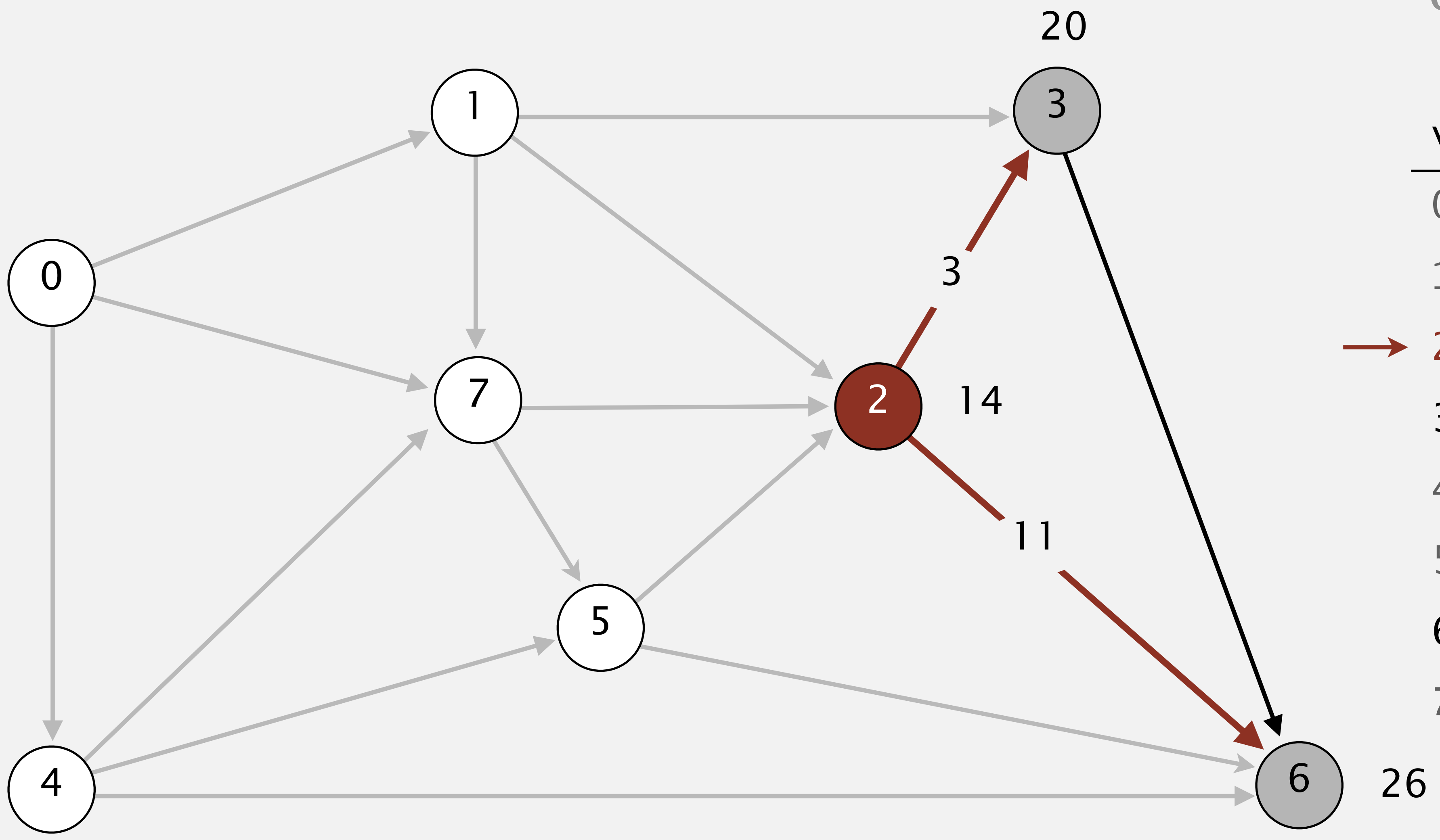
0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	14.0	5→2
3	20.0	1→3
4	9.0	0→4
5	13.0	4→5
6	26.0	5→6
7	8.0	0→7

select vertex 2

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



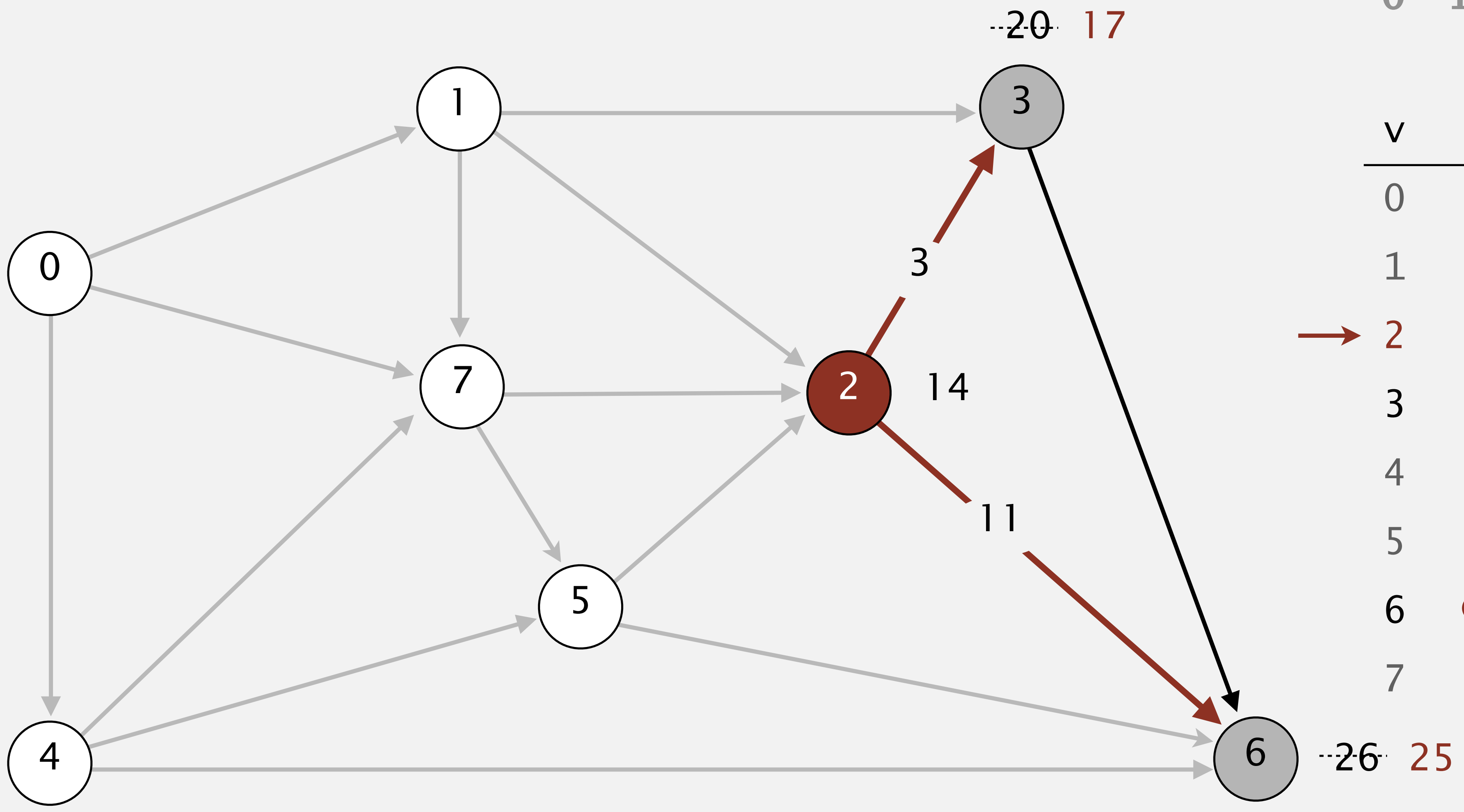
0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	14.0	5→2
3	20.0	1→3
4	9.0	0→4
5	13.0	4→5
6	26.0	5→6
7	8.0	0→7

relax all edges adjacent from 2

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



↓
0 1 4 7 5 2 3 6

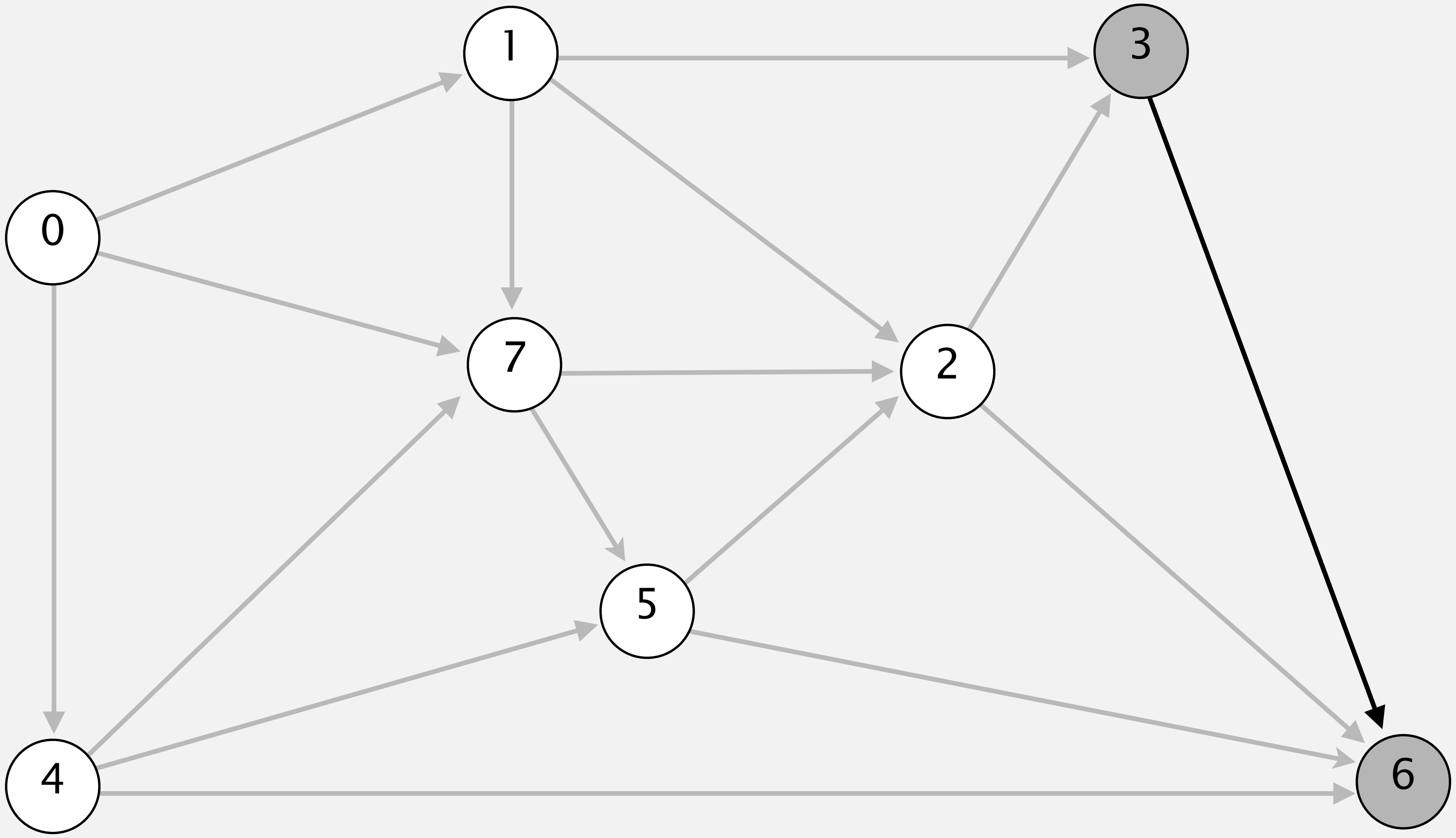
v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
→ 2	14.0	5→2
3	17.0	2→3
4	9.0	0→4
5	13.0	4→5
6	25.0	2→6
7	8.0	0→7

relax all edges adjacent from 2

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.

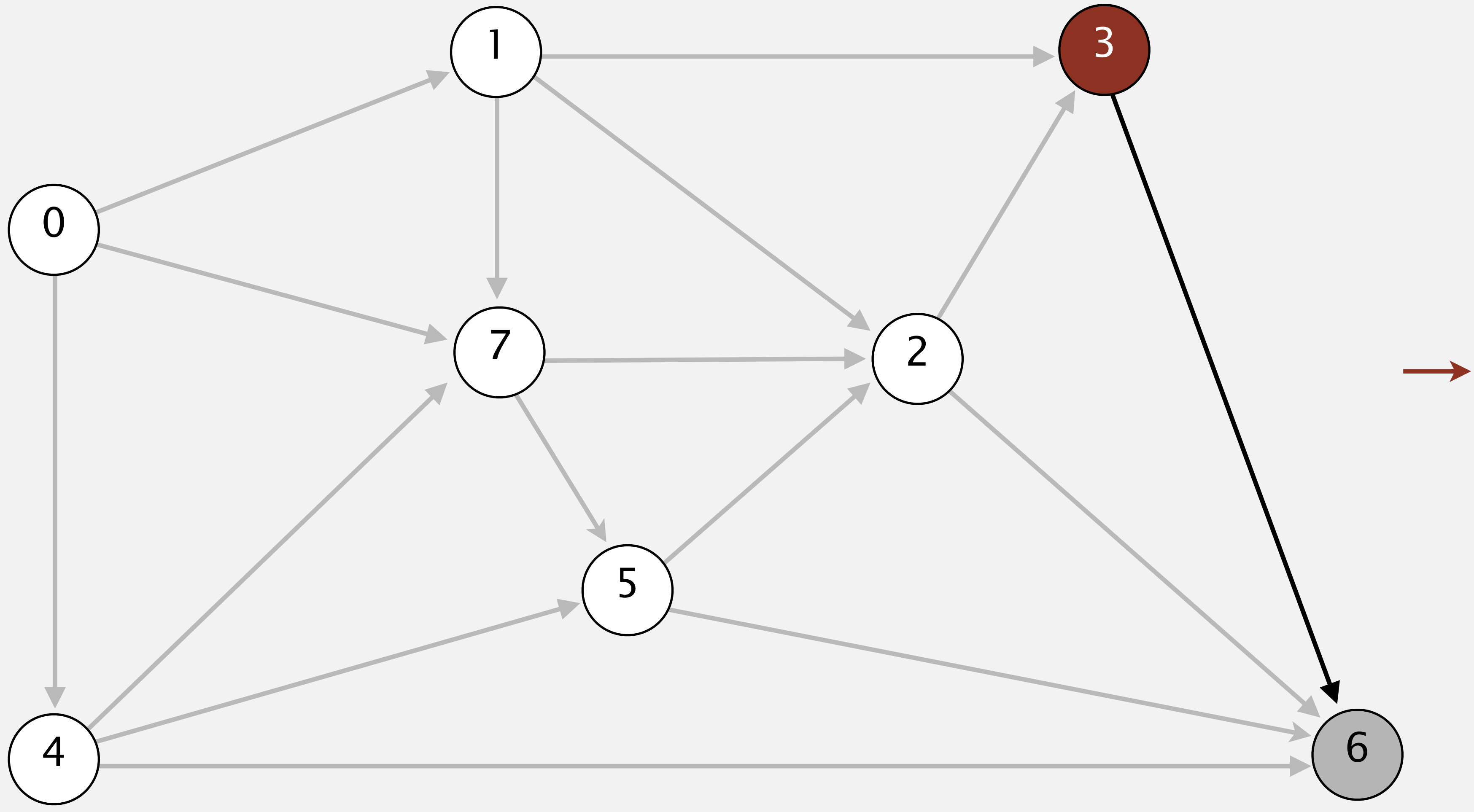
↓
0 1 4 7 5 2 3 6



v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	14.0	5→2
3	17.0	2→3
4	9.0	0→4
5	13.0	4→5
6	25.0	2→6
7	8.0	0→7

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



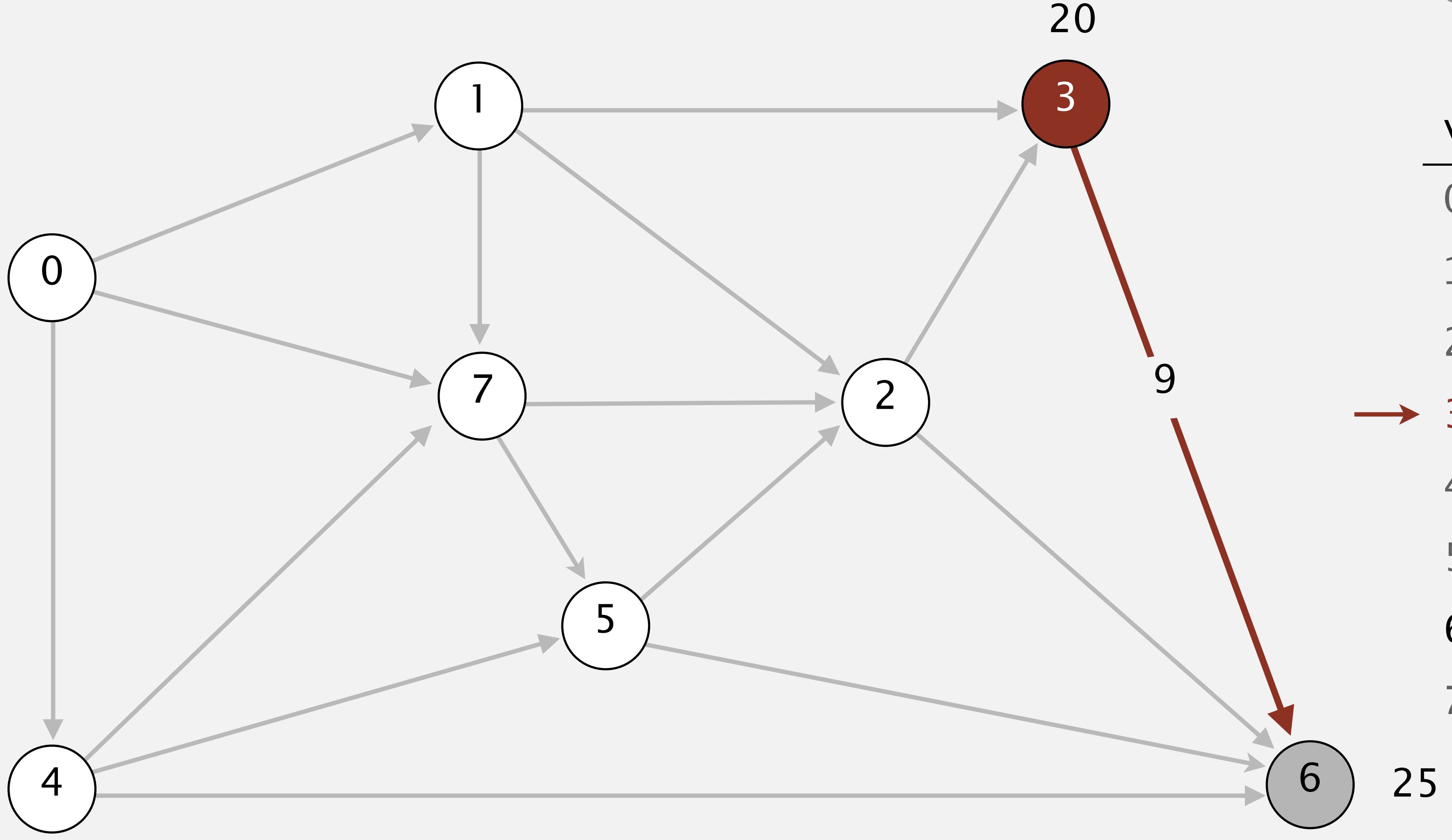
0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	14.0	5→2
3	17.0	2→3
4	9.0	0→4
5	13.0	4→5
6	25.0	2→6
7	8.0	0→7

select vertex 3

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



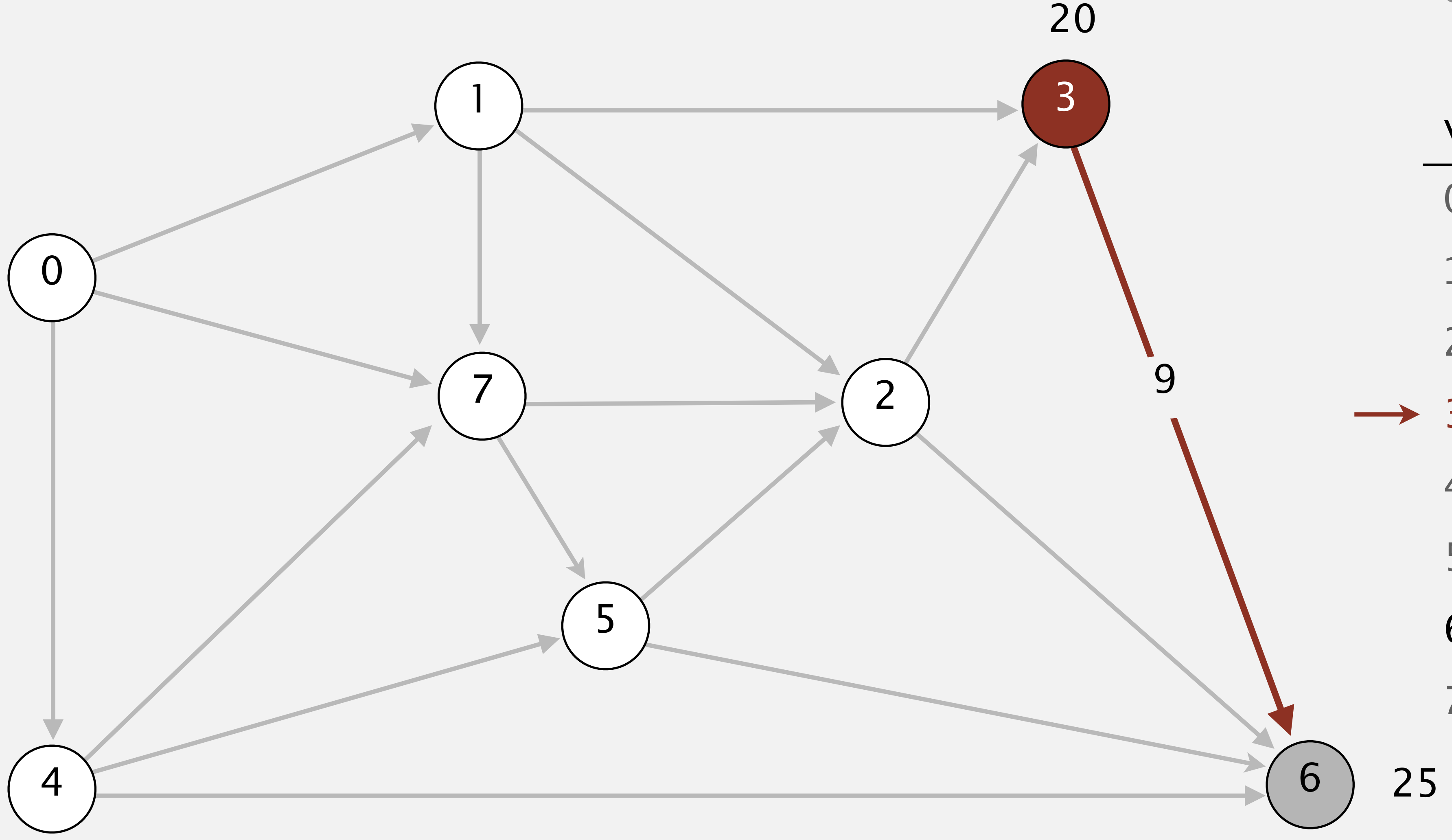
0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	14.0	5→2
3	17.0	2→3
4	9.0	0→4
5	13.0	4→5
6	25.0	2→6
7	8.0	0→7

relax all edges adjacent from 3

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



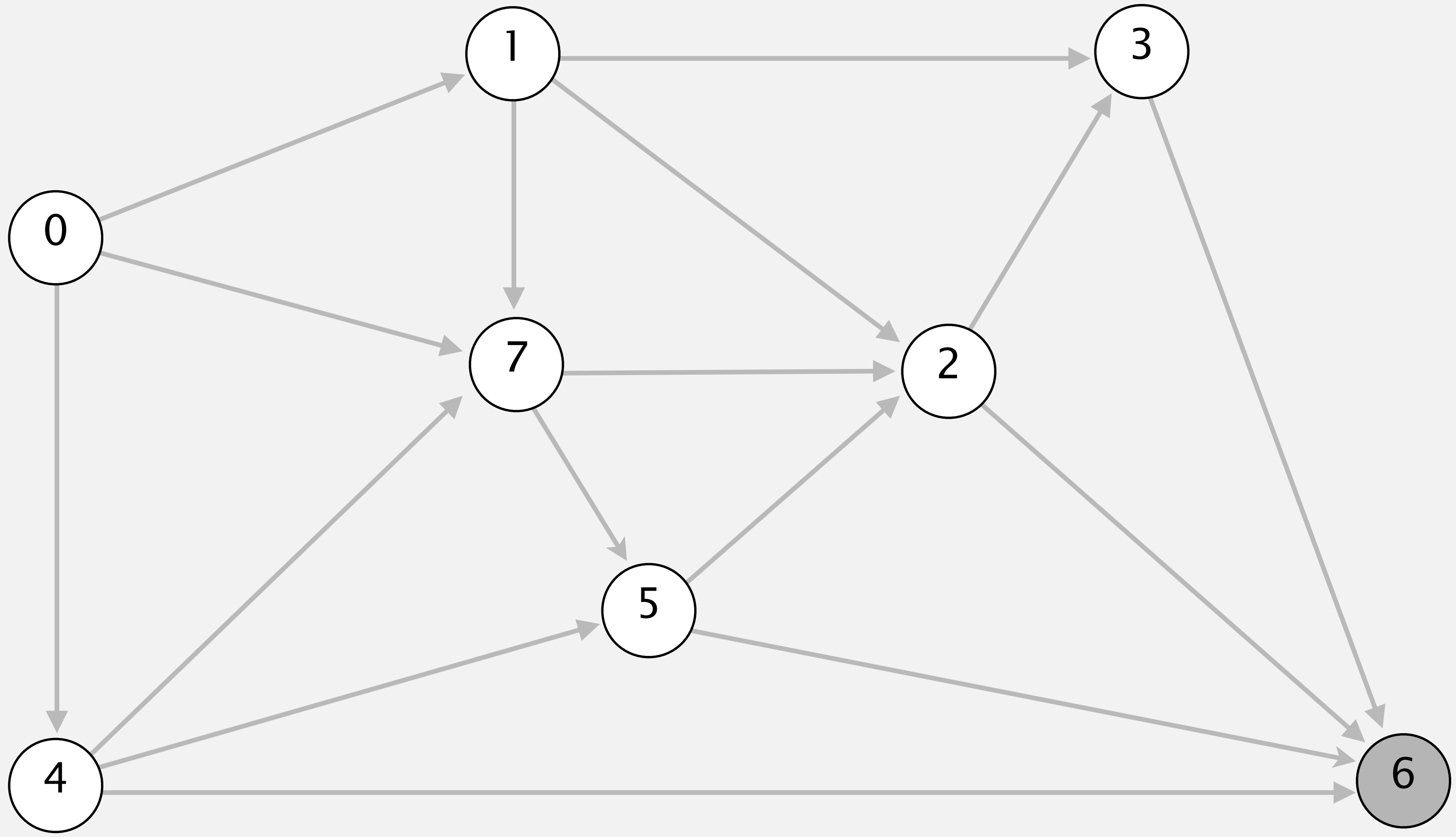
0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	14.0	5→2
3	17.0	2→3
4	9.0	0→4
5	13.0	4→5
6	25.0 ✓	2→6
7	8.0	0→7

relax all edges adjacent from 3

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



↓
0 1 4 7 5 2 3 **6**

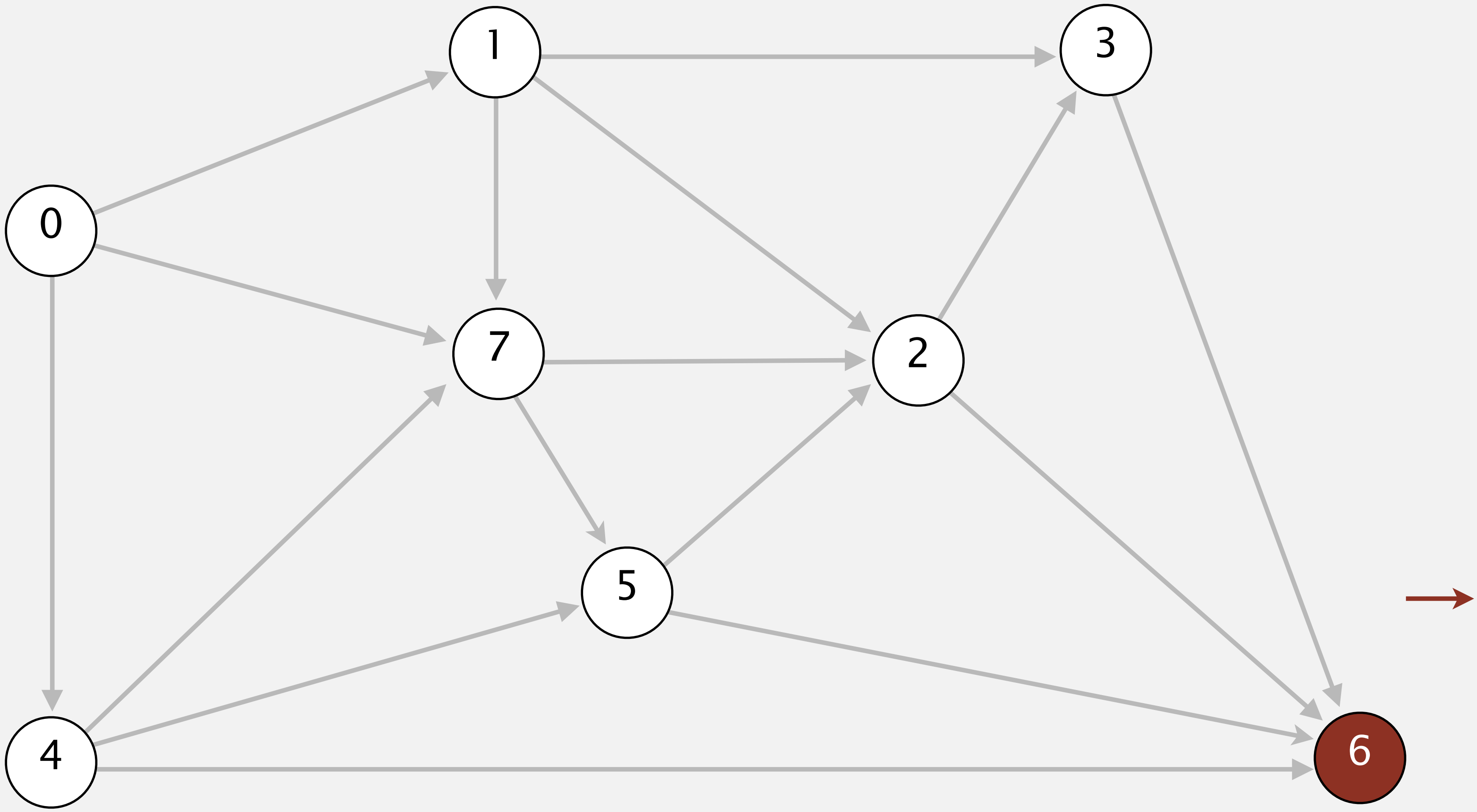
v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	14.0	5→2
3	17.0	2→3
4	9.0	0→4
5	13.0	4→5
6	25.0	2→6
7	8.0	0→7

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.

↓

0 1 4 7 5 2 3 **6**

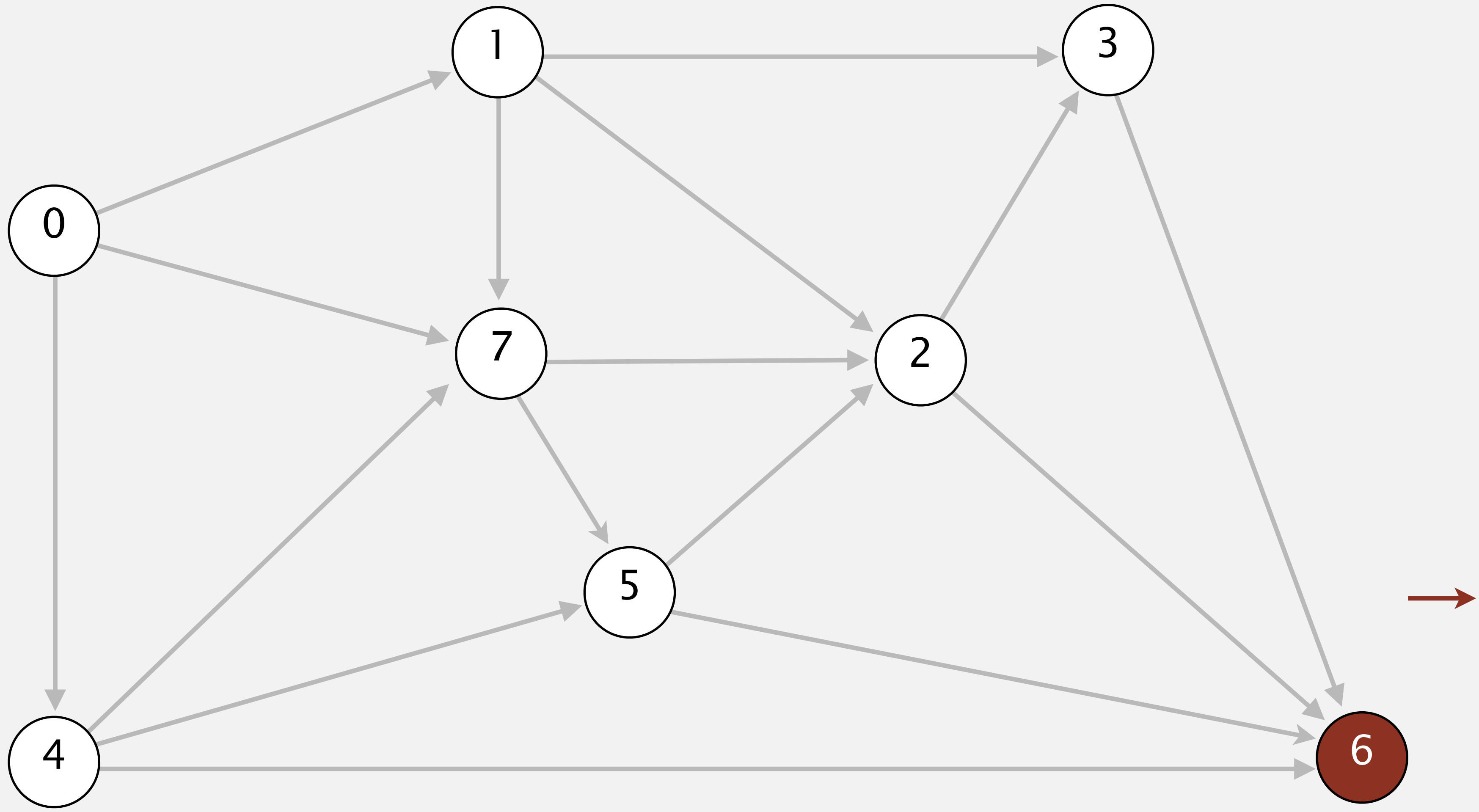


v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	14.0	5→2
3	17.0	2→3
4	9.0	0→4
5	13.0	4→5
→ 6	25.0	2→6
7	8.0	0→7

select vertex 6

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



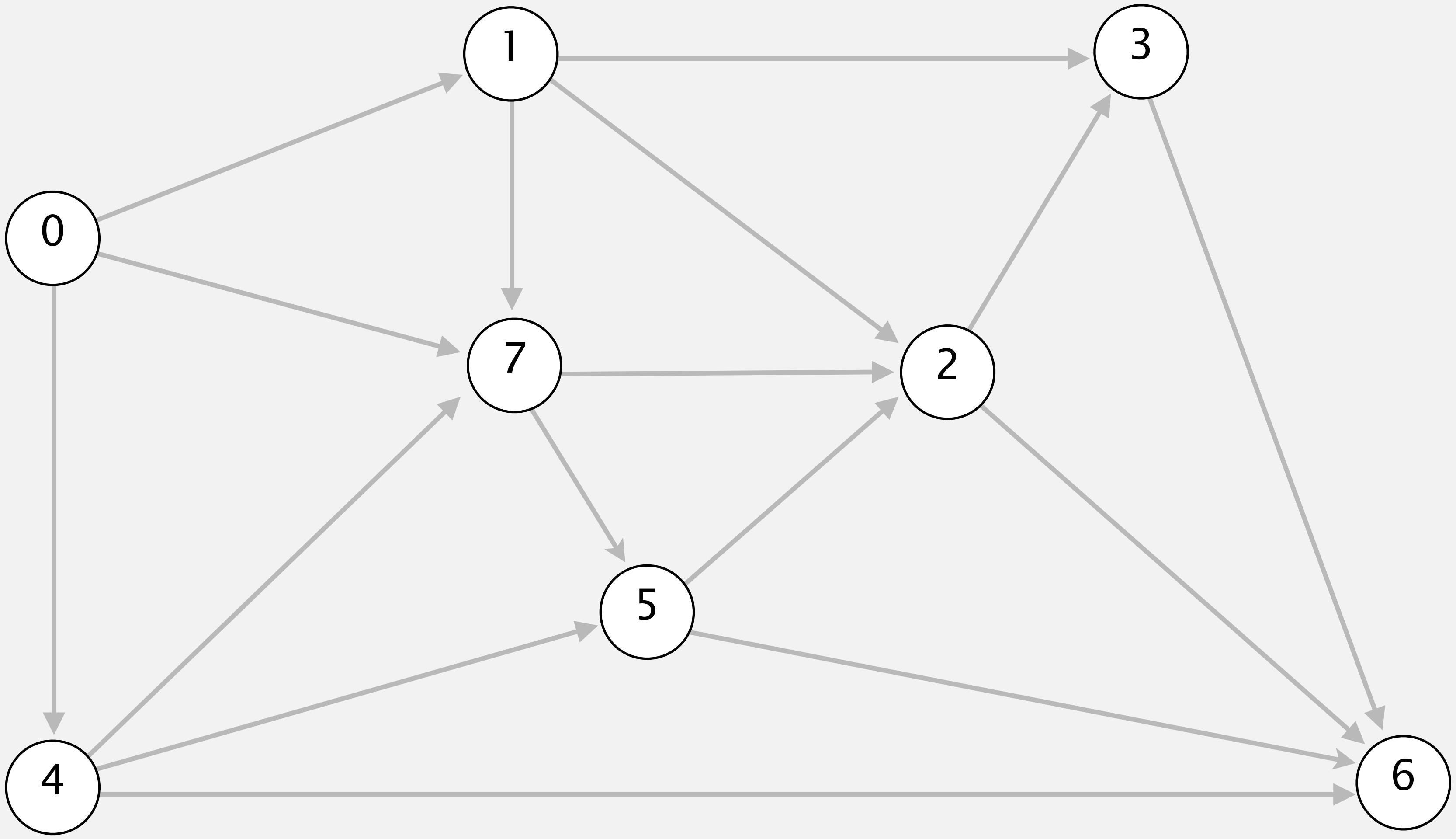
↓
0 1 4 7 5 2 3 **6**

	v	distTo[]	edgeTo[]
	0	0.0	-
	1	5.0	0→1
	2	14.0	5→2
	3	17.0	2→3
	4	9.0	0→4
	5	13.0	4→5
→	6	25.0	2→6
	7	8.0	0→7

relax all edges adjacent from 6

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.

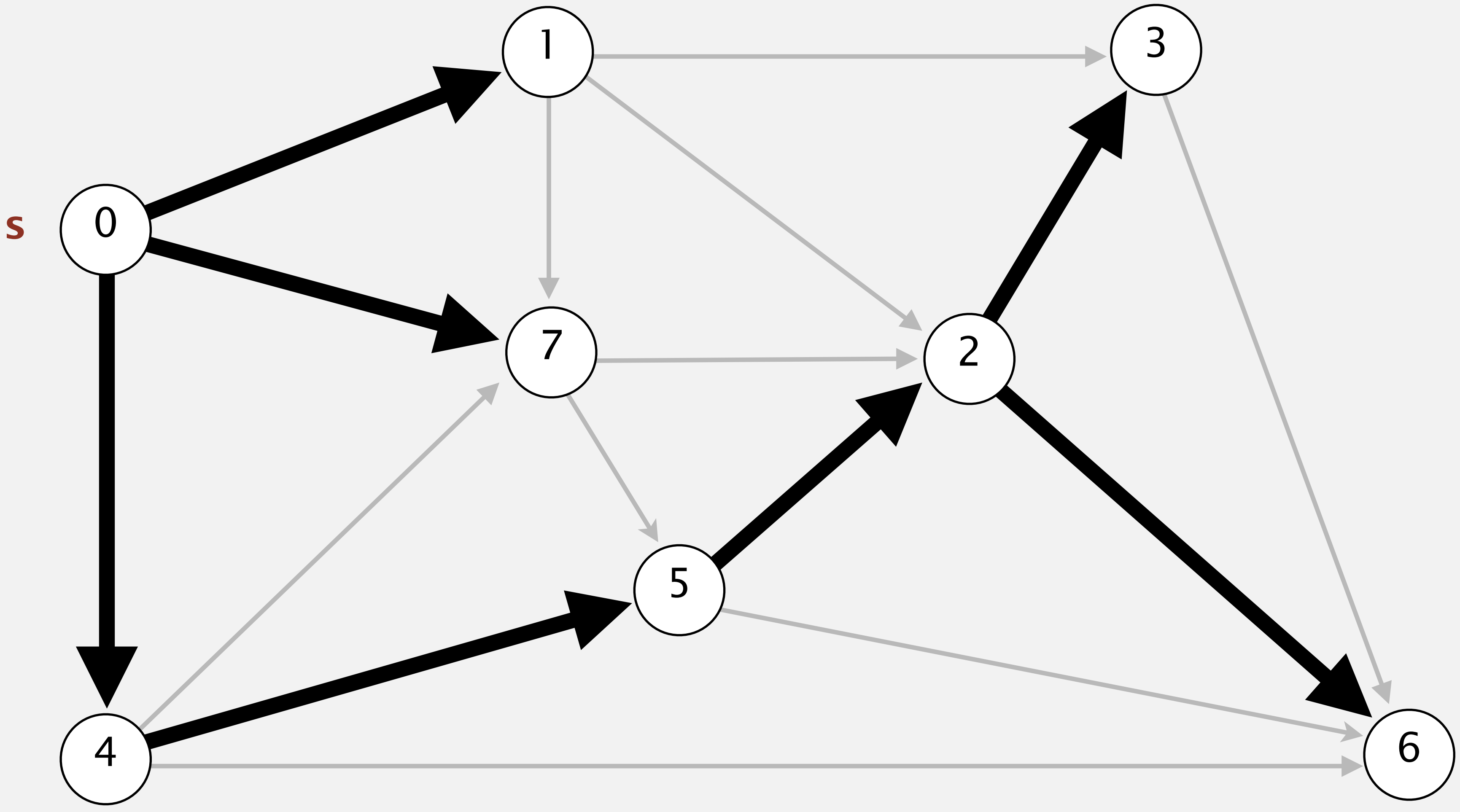


0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	14.0	5→2
3	17.0	2→3
4	9.0	0→4
5	13.0	4→5
6	25.0	2→6
7	8.0	0→7

Acyclic shortest paths demo

- Consider vertices in topological order.
- Relax all edges adjacent from that vertex.



0 1 4 7 5 2 3 6

v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	14.0	5→2
3	17.0	2→3
4	9.0	0→4
5	13.0	4→5
6	25.0	2→6
7	8.0	0→7

shortest-paths tree from vertex s