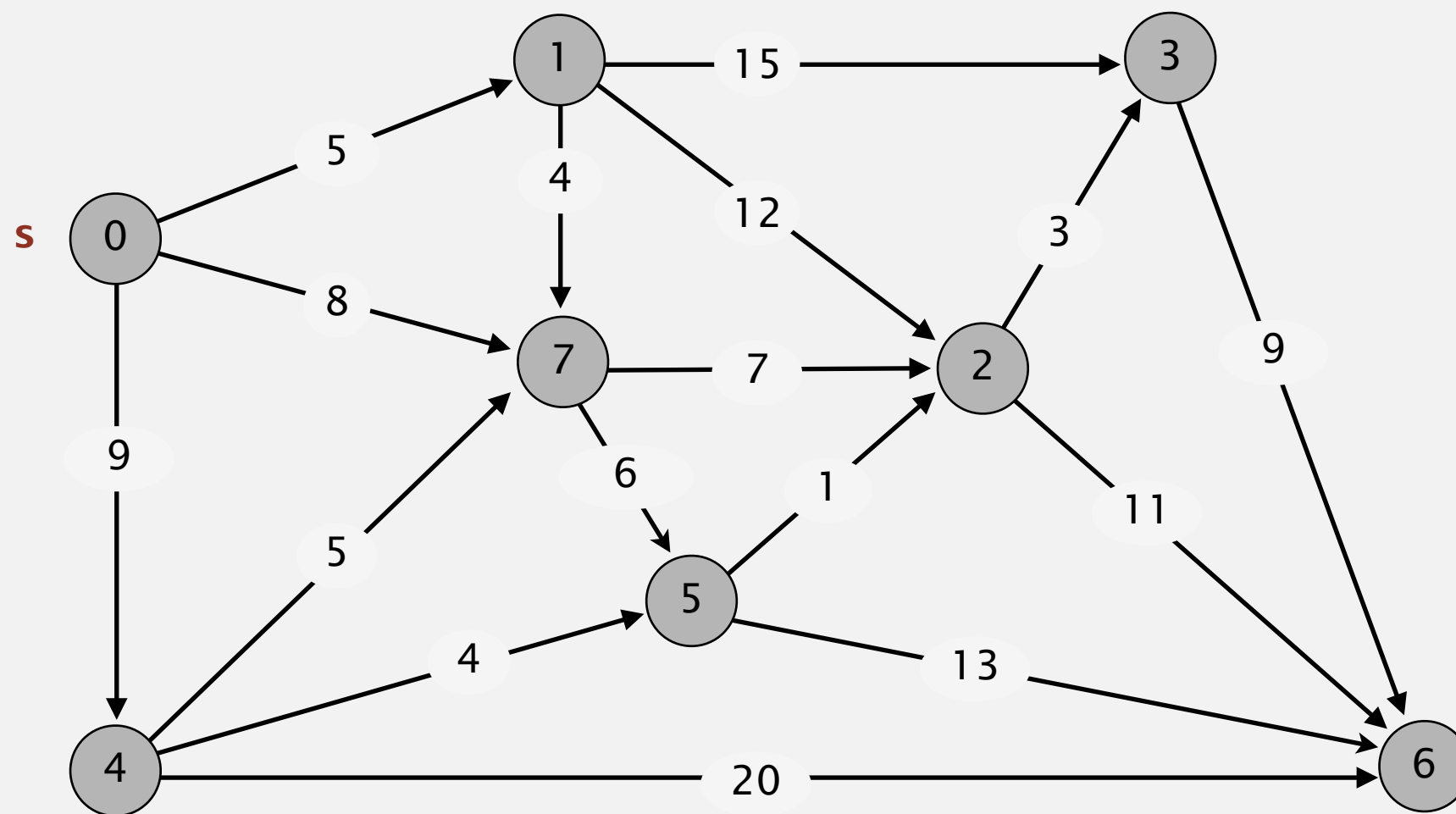


# ACYCLIC SHORTEST PATHS DEMO

# Topological sort algorithm

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

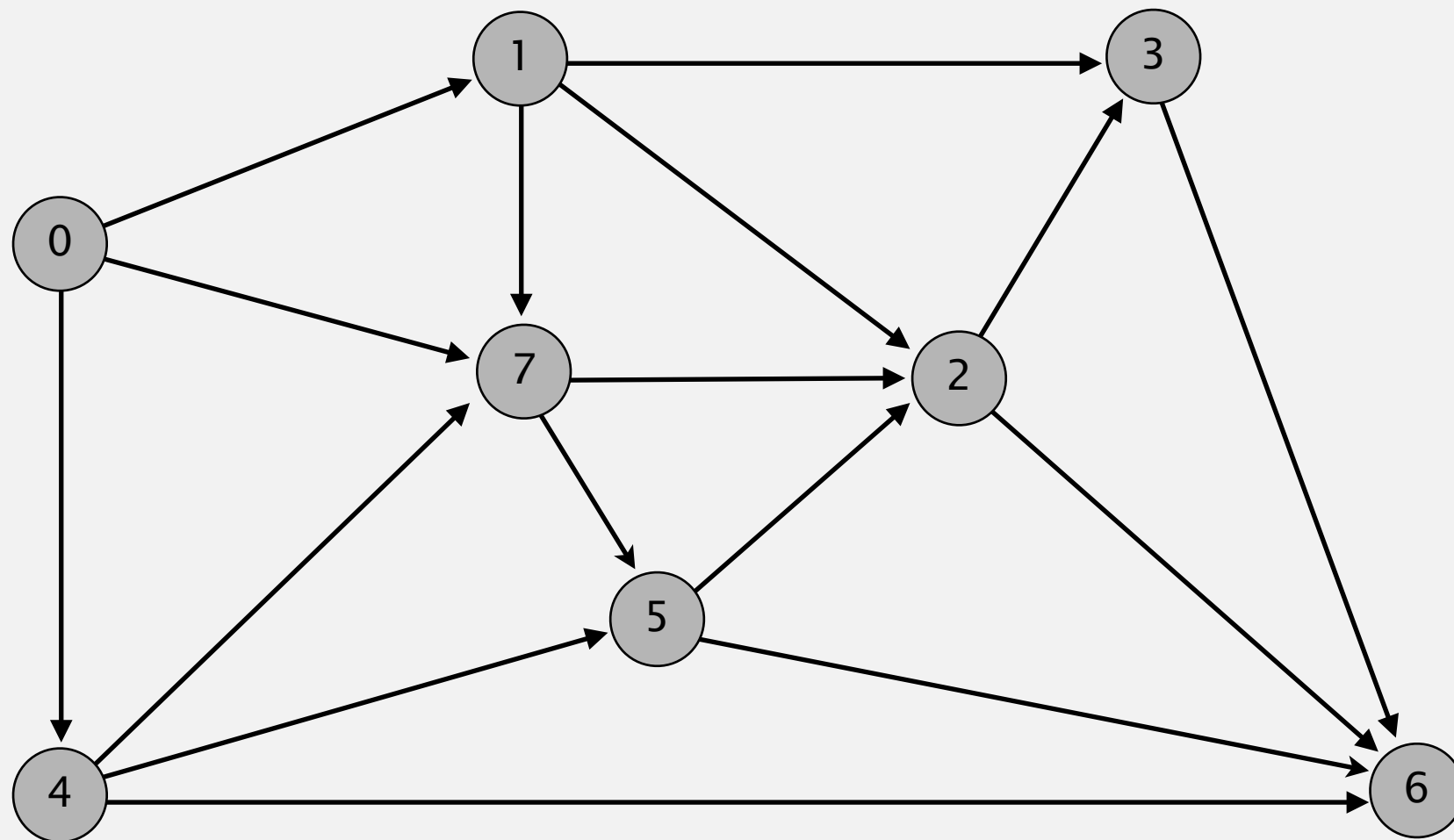


0→1	5.0
0→4	9.0
0→7	8.0
1→2	12.0
1→3	15.0
1→7	4.0
2→3	3.0
2→6	11.0
3→6	9.0
4→5	4.0
4→6	20.0
4→7	5.0
5→2	1.0
5→6	13.0
7→5	6.0
7→2	7.0

**an edge-weighted DAG**

# Topological sort algorithm

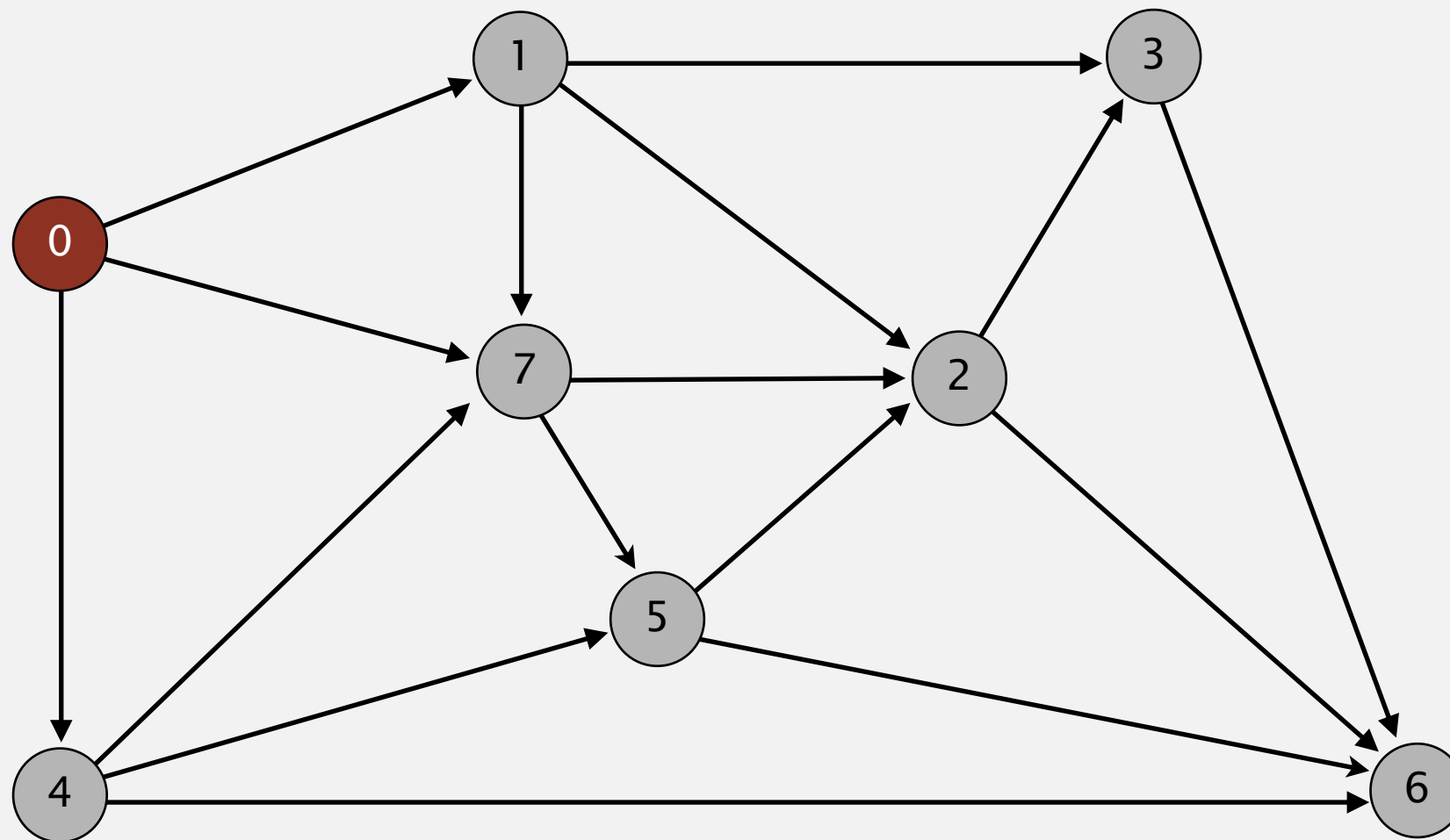
- Consider vertices in topological order.
- Relax all edges incident from that vertex.



**topological order: 0 1 4 7 5 2 3 6**

# Topological sort algorithm

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

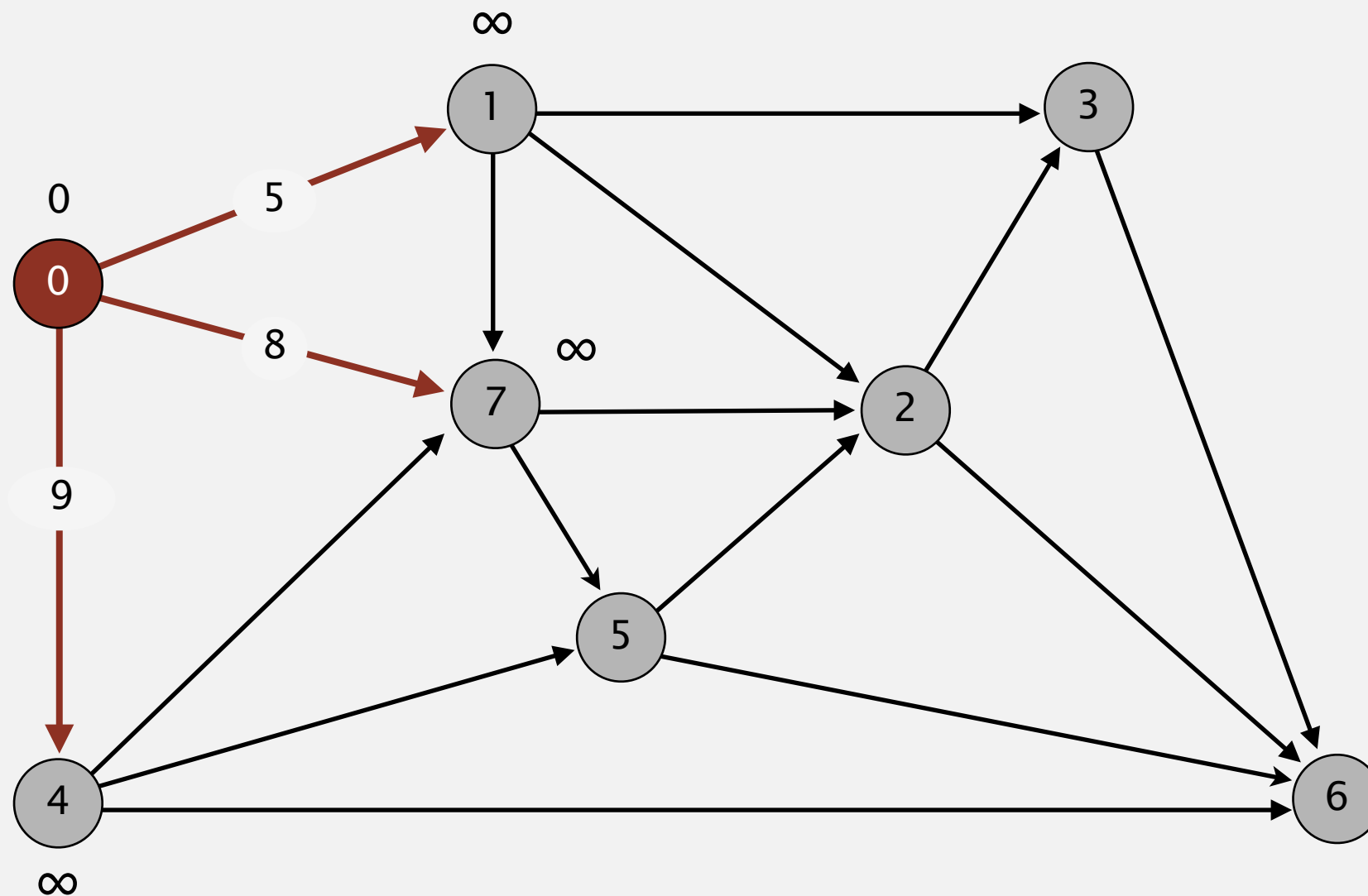


choose vertex 0

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

# Topological sort algorithm

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

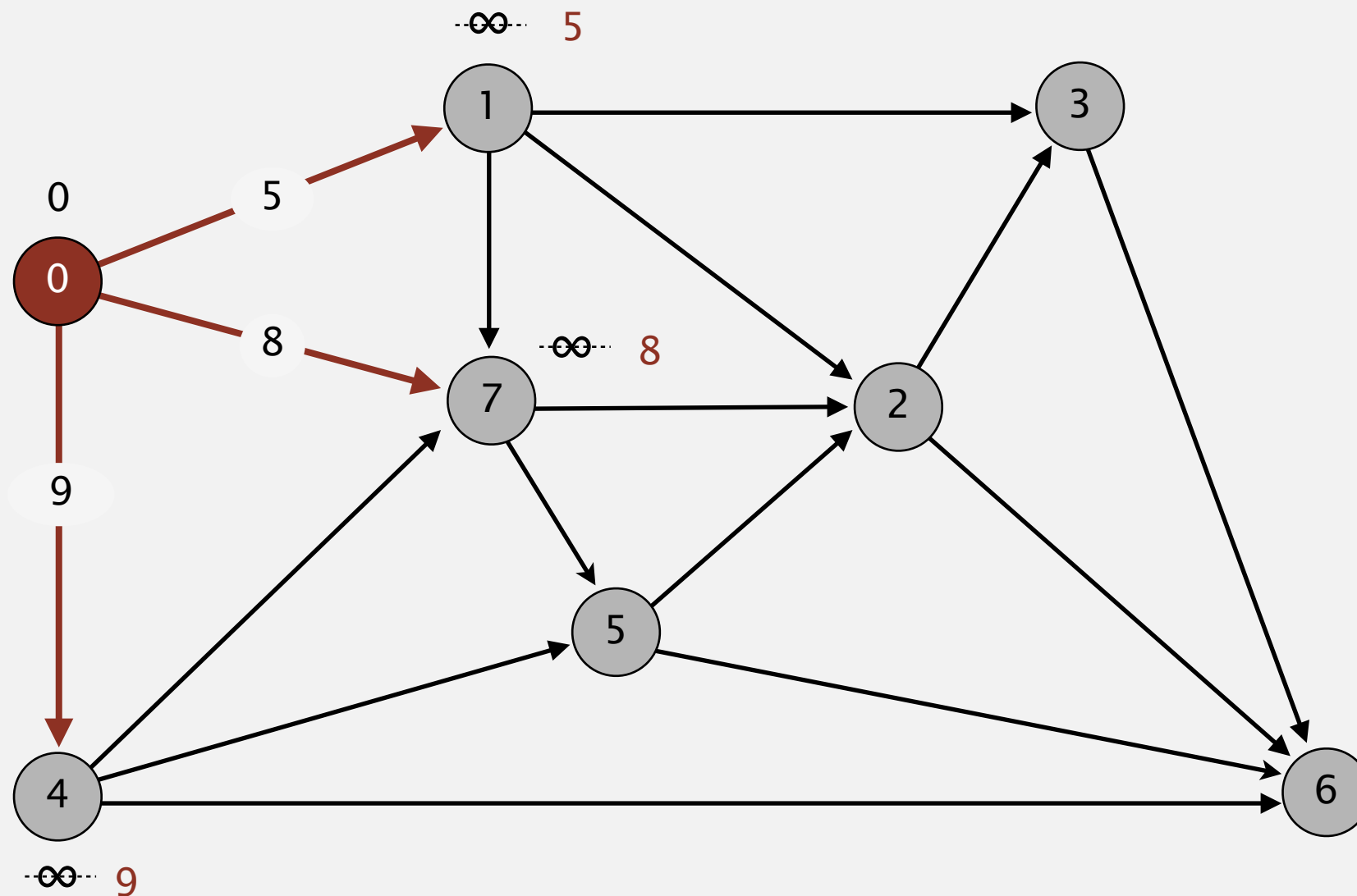


v	distTo[]	edgeTo[]
→ 0	0.0	-
1		
2		
3		
4		
5		
6		
7		

relax all edges incident from 0

# Topological sort algorithm

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

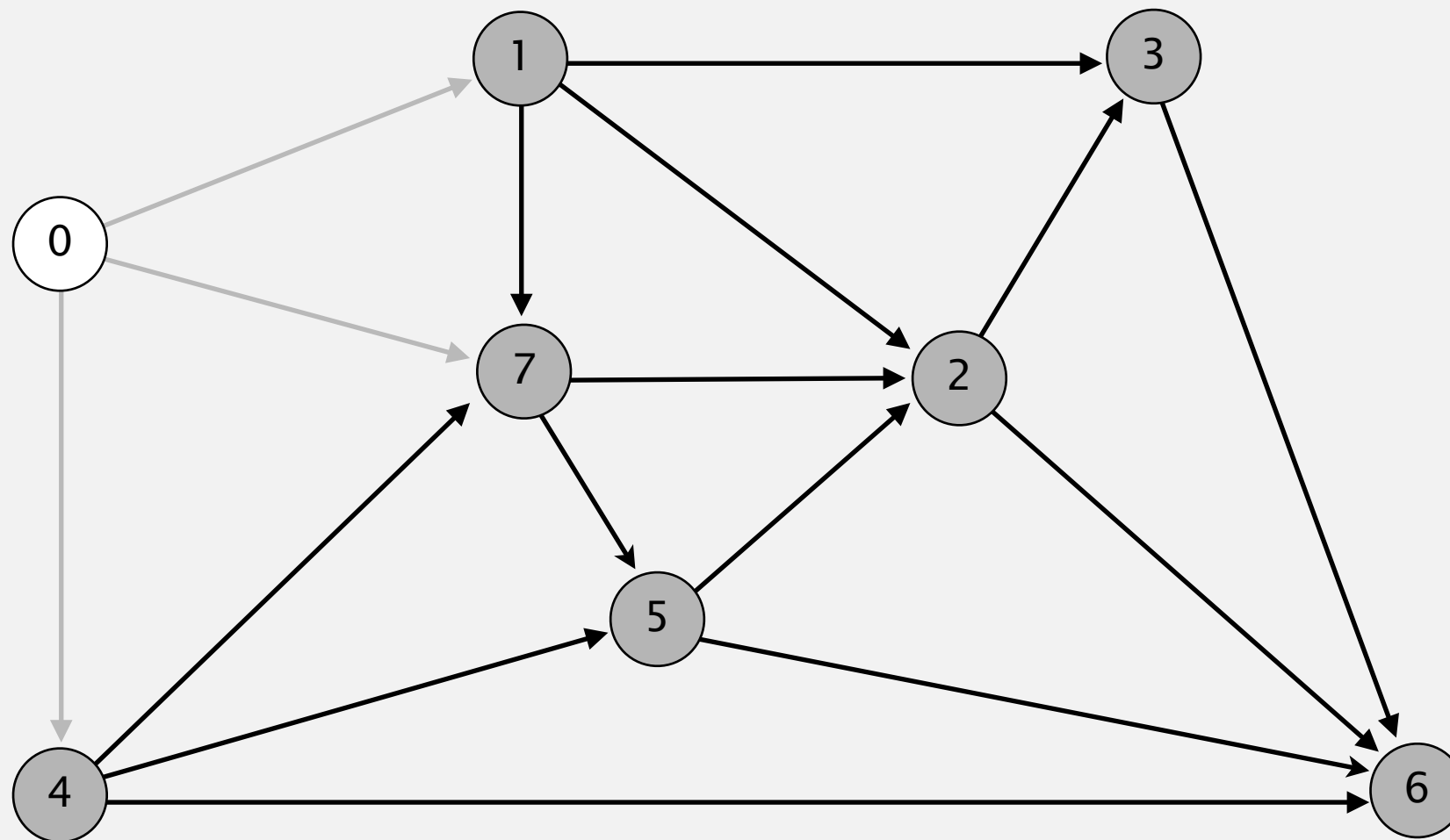


relax all edges incident from 0

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

# Topological sort algorithm

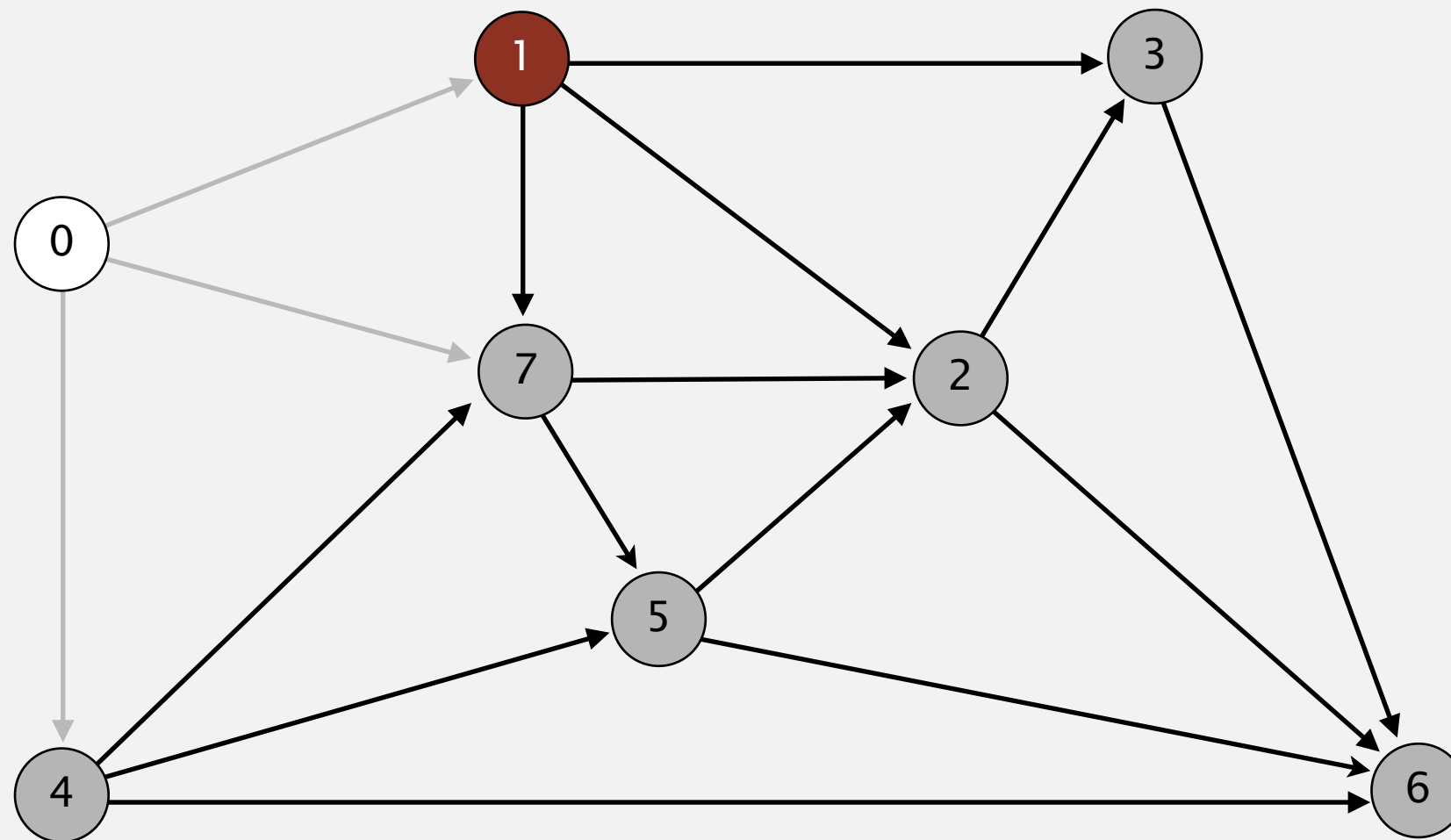
- Consider vertices in topological order.
- Relax all edges incident from that vertex.



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

# Topological sort algorithm

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



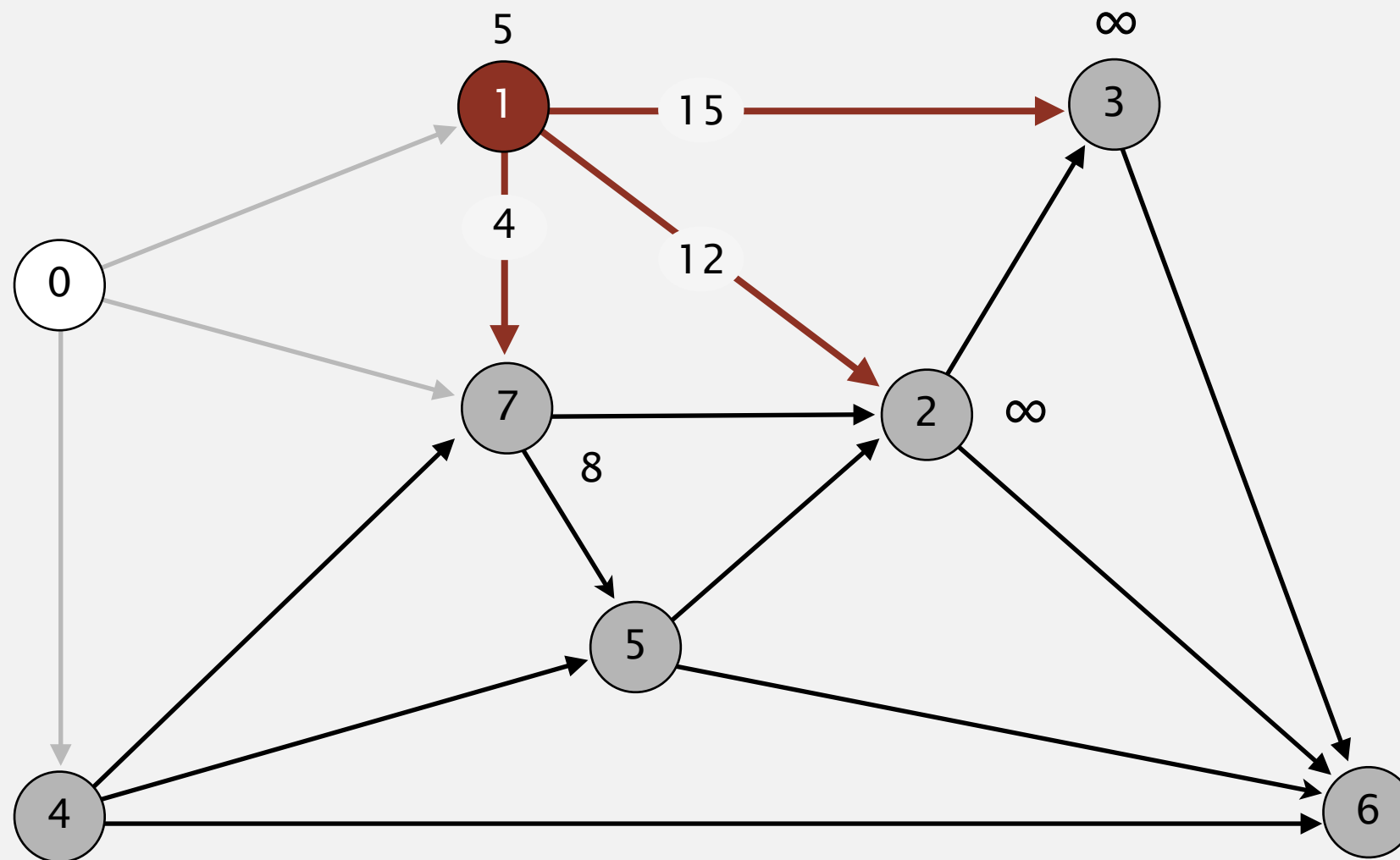
**choose vertex 1**

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



# Topological sort algorithm

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

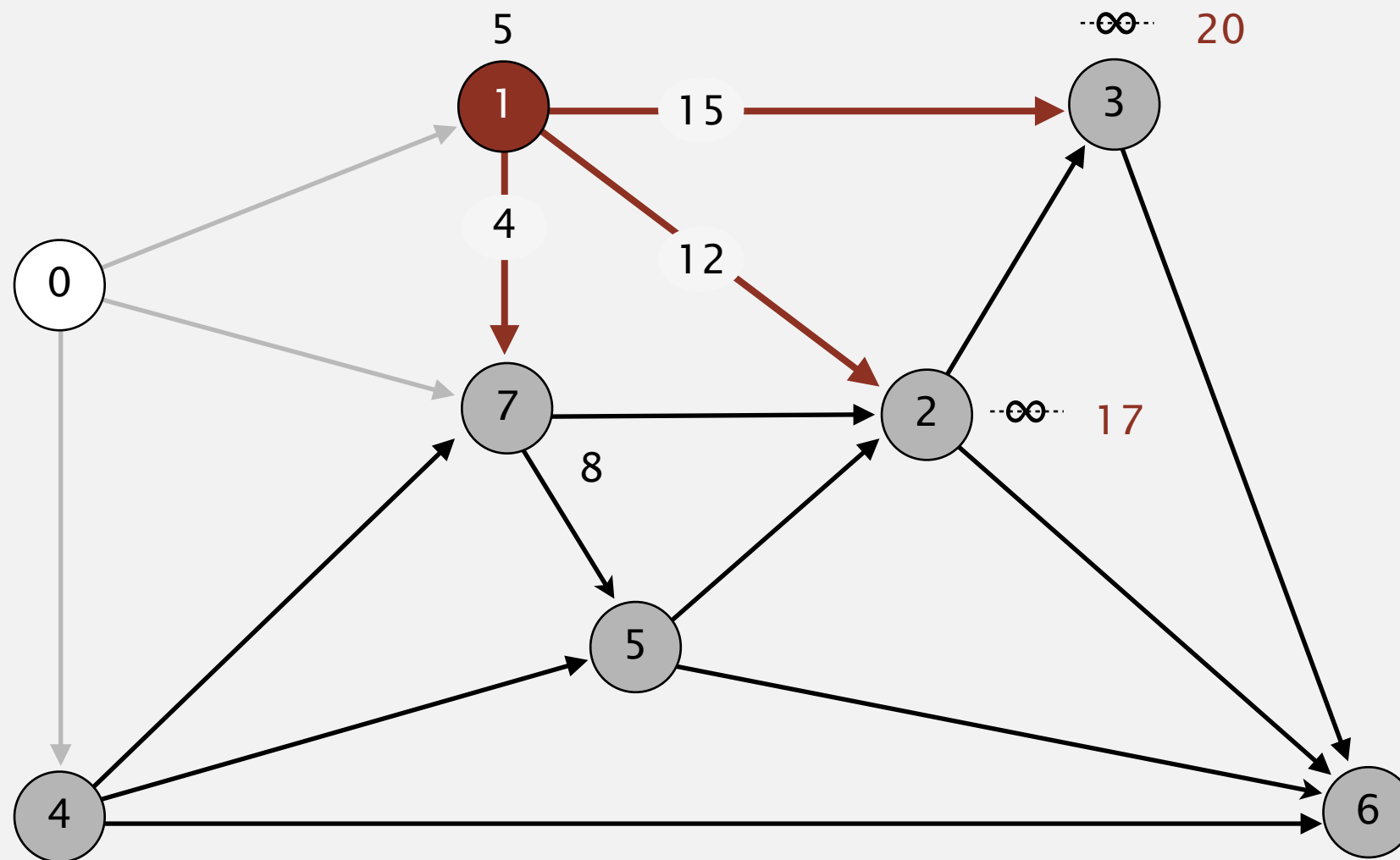


v	distTo[]	edgeTo[]
0	0.0	-
→ 1	5.0	0→1
2		
3		
4	9.0	0→4
5		
6		
7	8.0	0→7

relax all edges incident from 1

# Topological sort algorithm

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

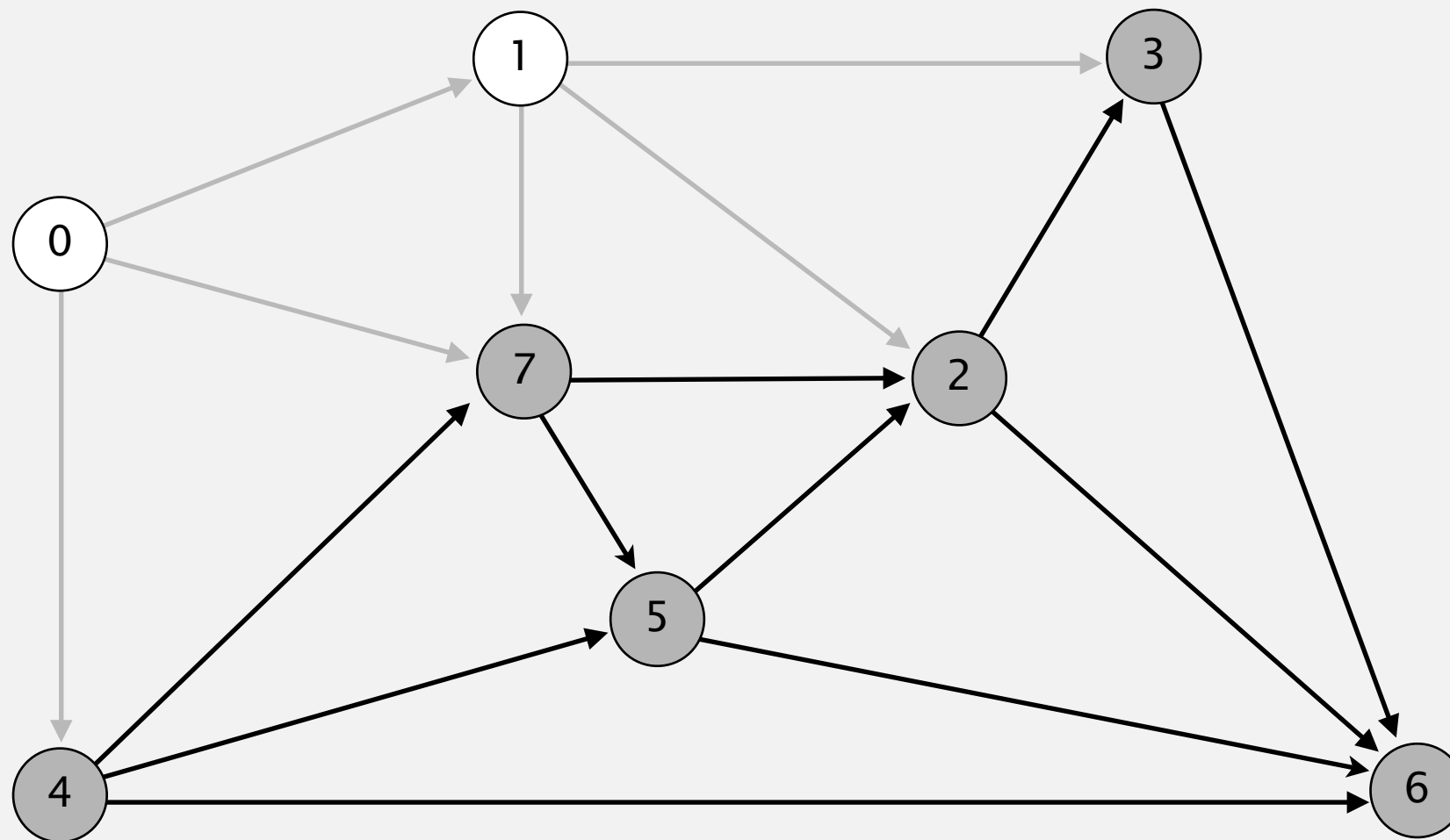


relax all edges incident from 1

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

# Topological sort algorithm

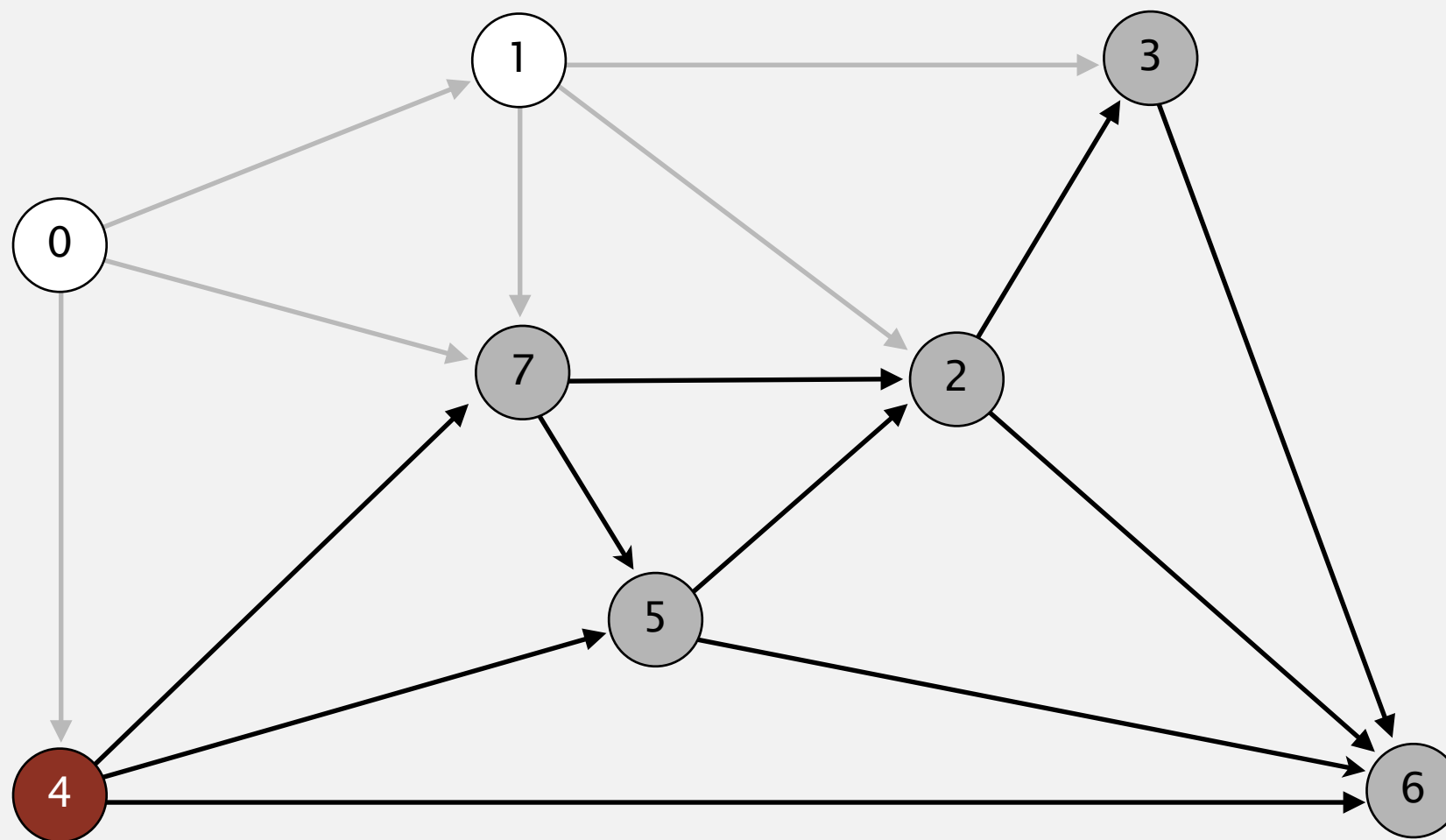
- Consider vertices in topological order.
- Relax all edges incident from that vertex.



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

# Topological sort algorithm

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

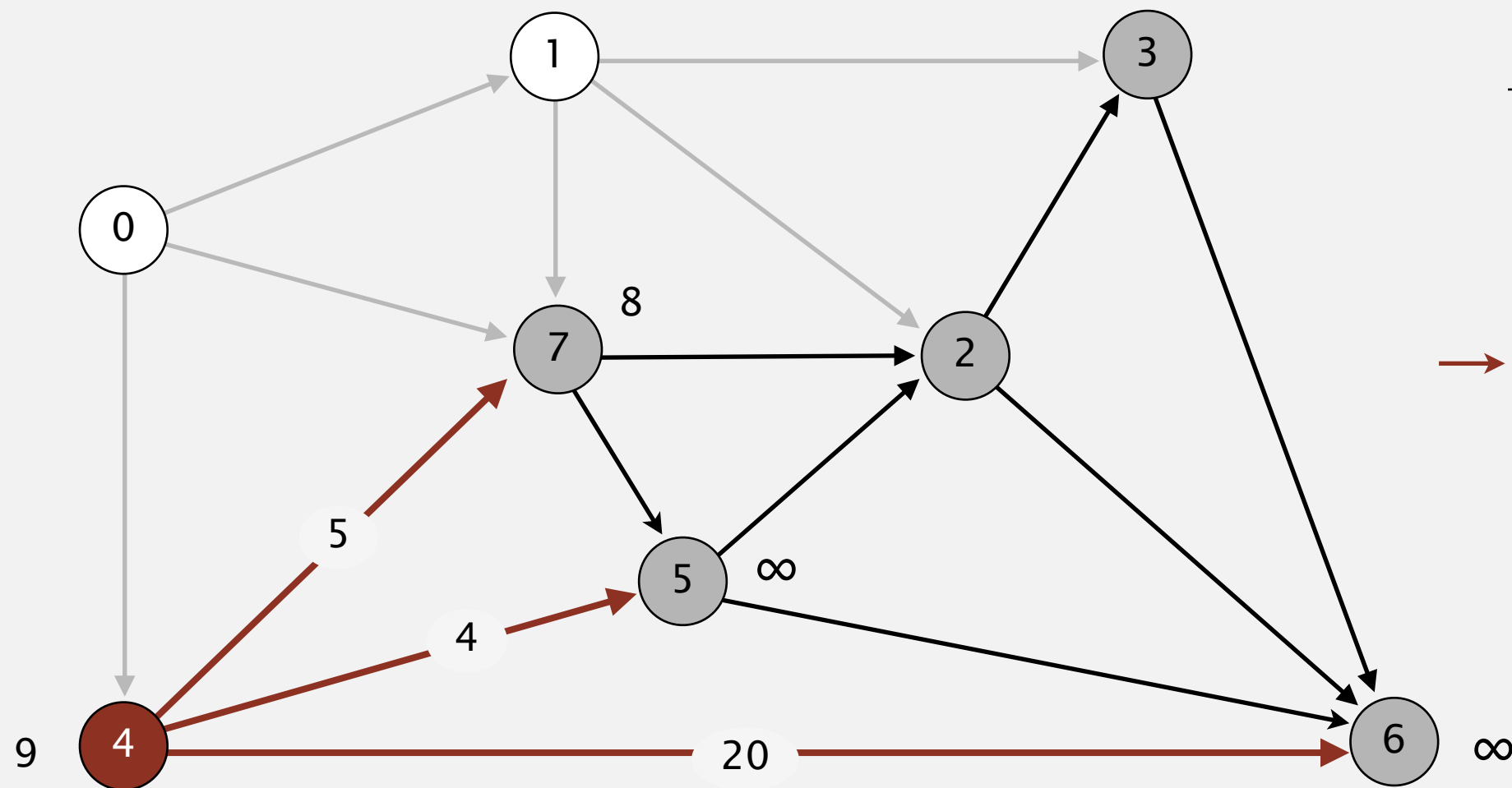


v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	17.0	1→2
3	20.0	1→3
→ 4	9.0	0→4
5		
6		
7	8.0	0→7

**select vertex 4**  
**(Dijkstra would have selected vertex 7)**

# Topological sort algorithm

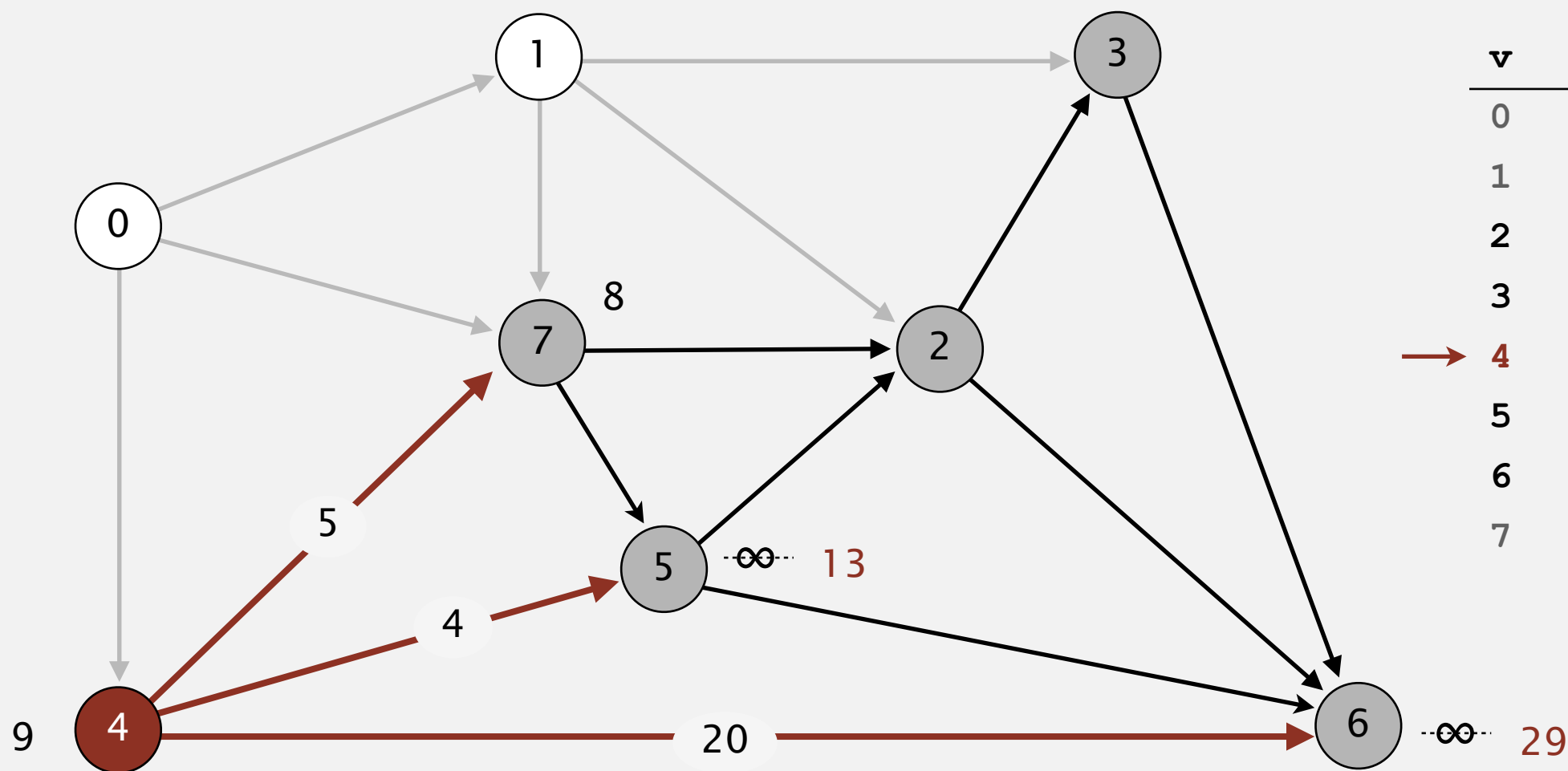
- Consider vertices in topological order.
- Relax all edges incident from that vertex.



relax all edges incident from 4

# Topological sort algorithm

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

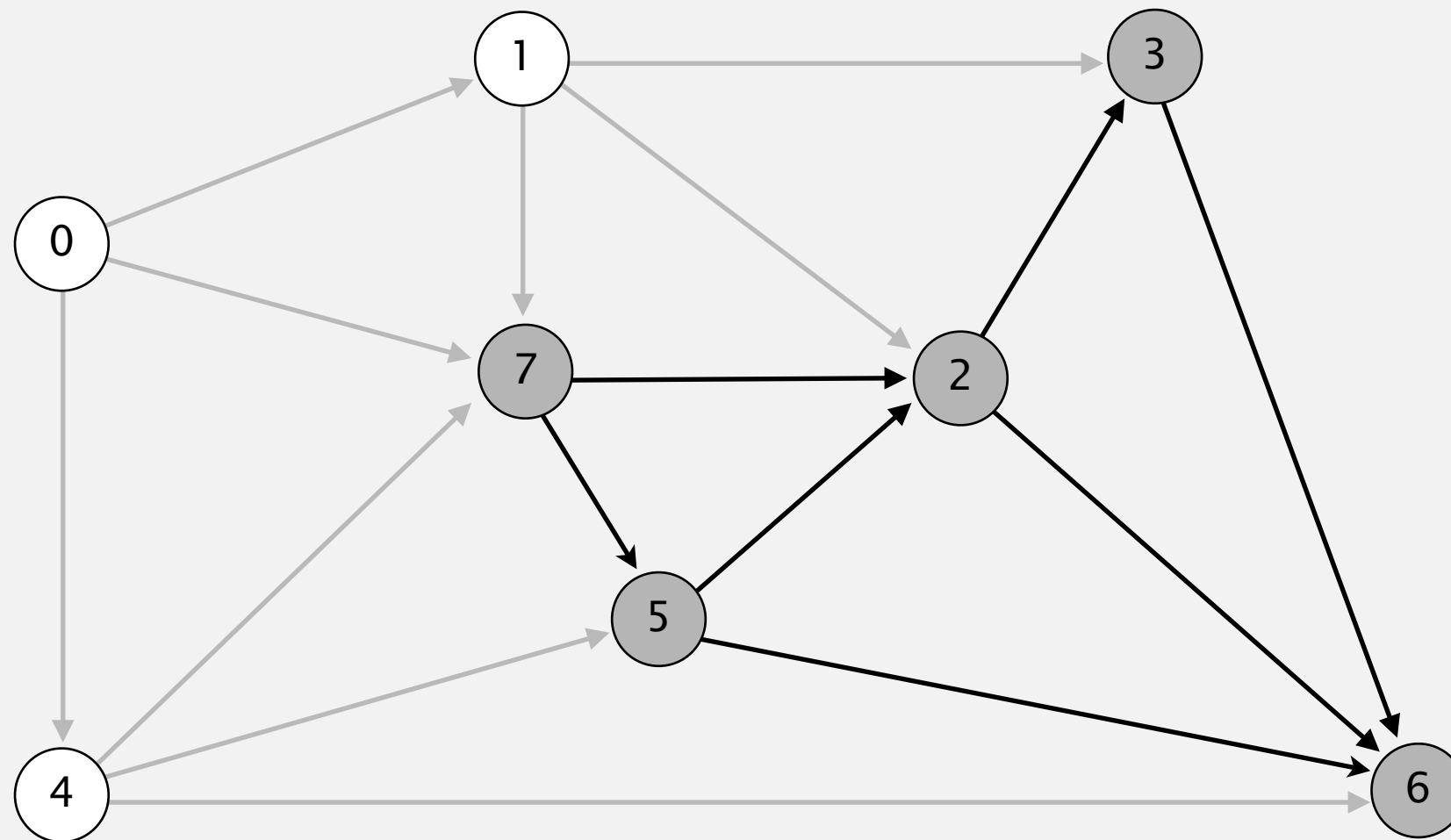


v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	17.0	1→2
3	20.0	1→3
4	9.0	0→4
5	13.0	4→5
6	29.0	4→6
7	8.0 ✓	0→7

relax all edges incident from 4

# Topological sort algorithm

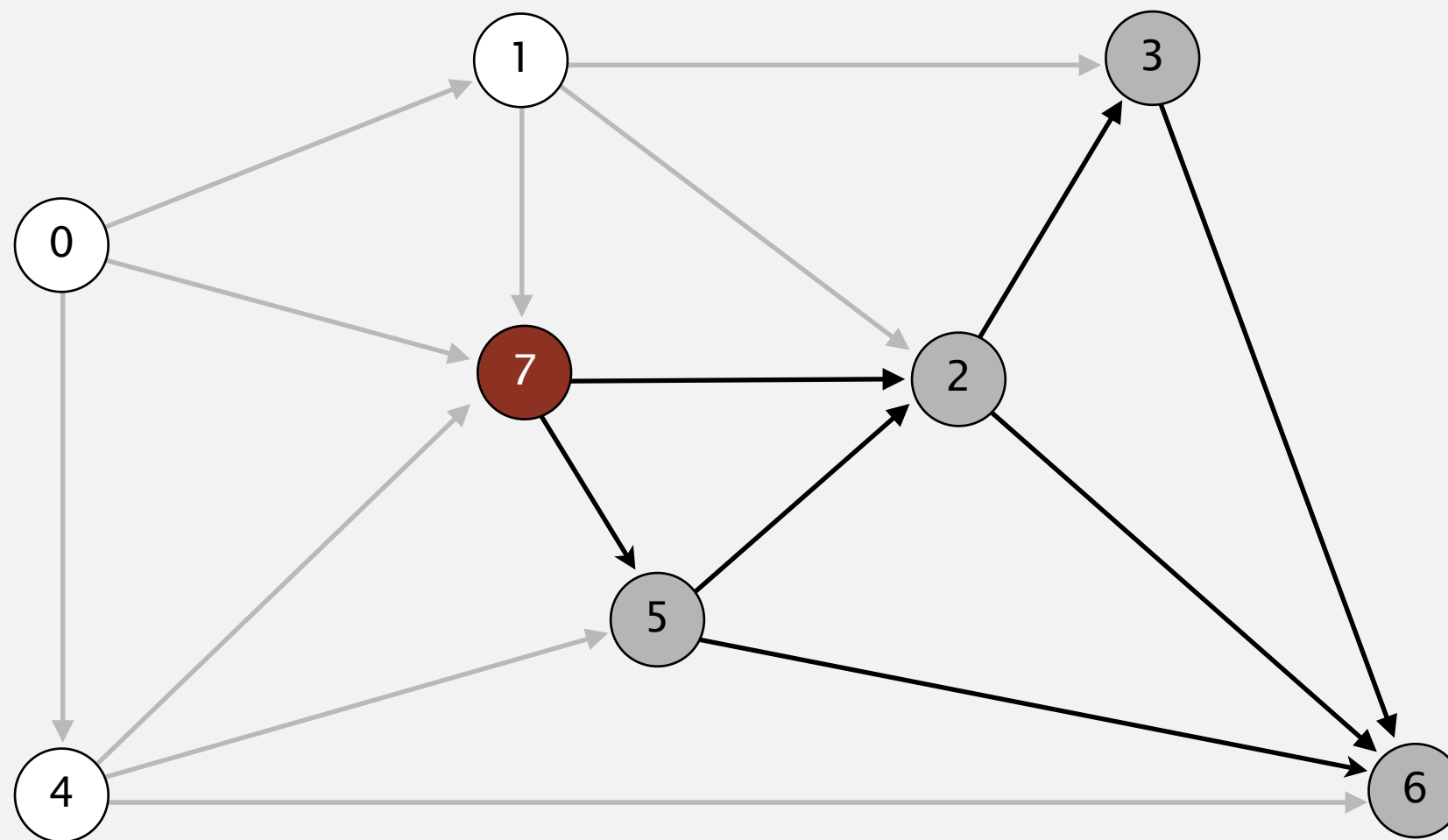
- Consider vertices in topological order.
- Relax all edges incident from that vertex.



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

# Topological sort algorithm

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



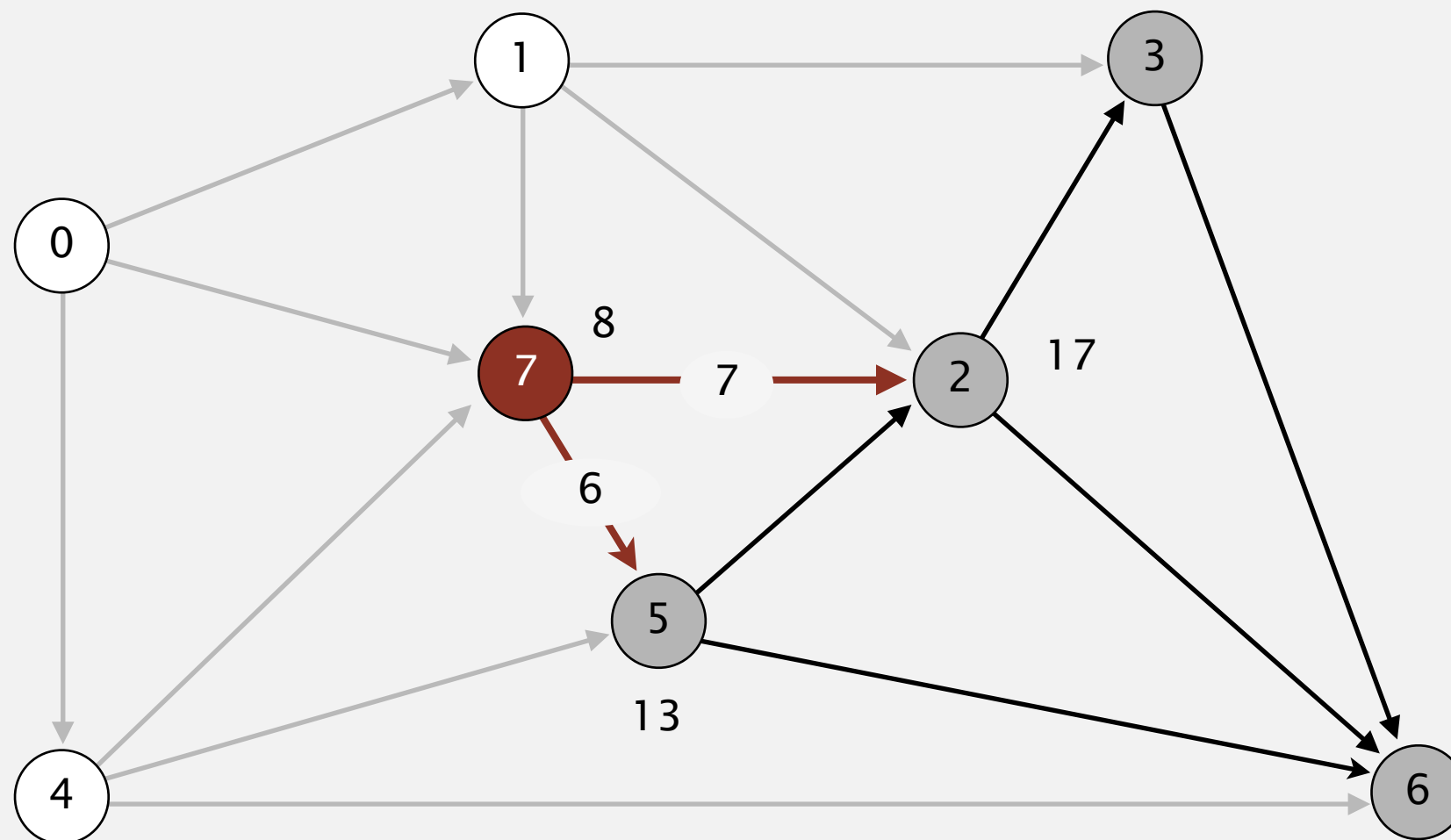
choose vertex 7

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



# Topological sort algorithm

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

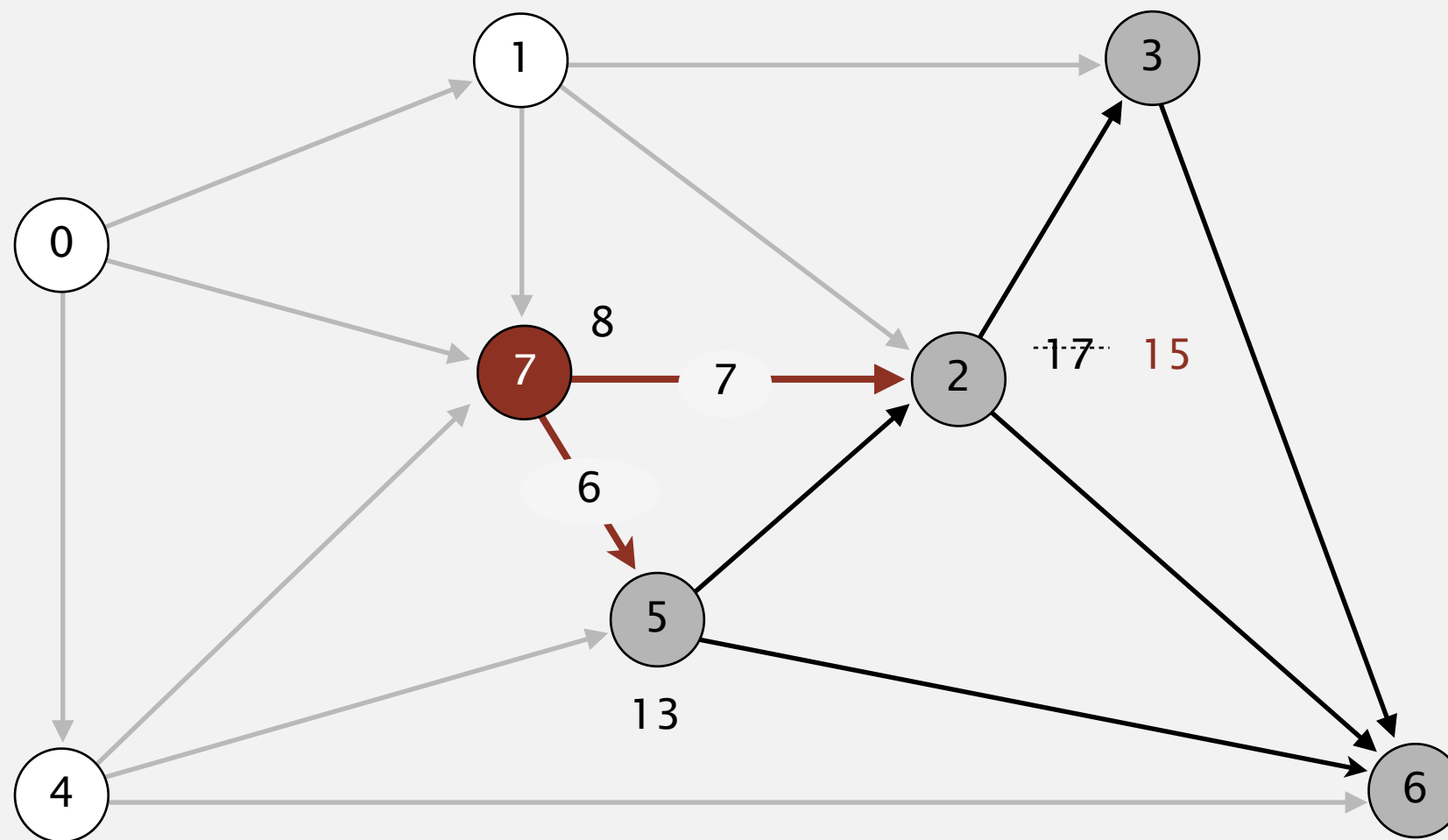


v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	17.0	1→2
3	20.0	1→3
4	9.0	0→4
5	13.0	4→5
6	29.0	4→6
→ 7	8.0	0→7

**relax all edges incident from 7**

# Topological sort algorithm

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

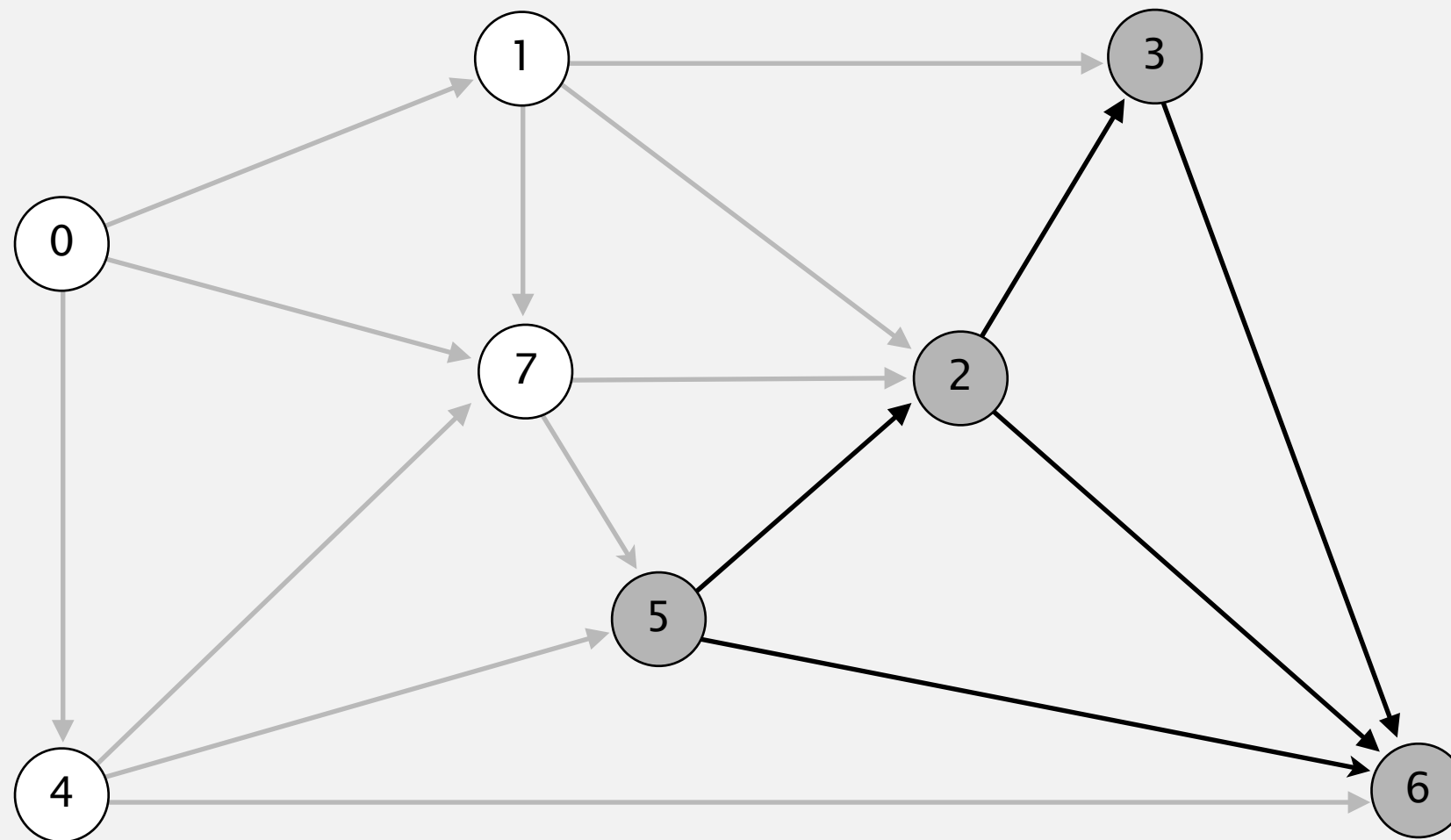


v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	15.0	7→2
3	20.0	1→3
4	9.0	0→4
5	13.0	4→5
6	29.0	4→6
→ 7	8.0	0→7

relax all edges incident from 7

# Topological sort algorithm

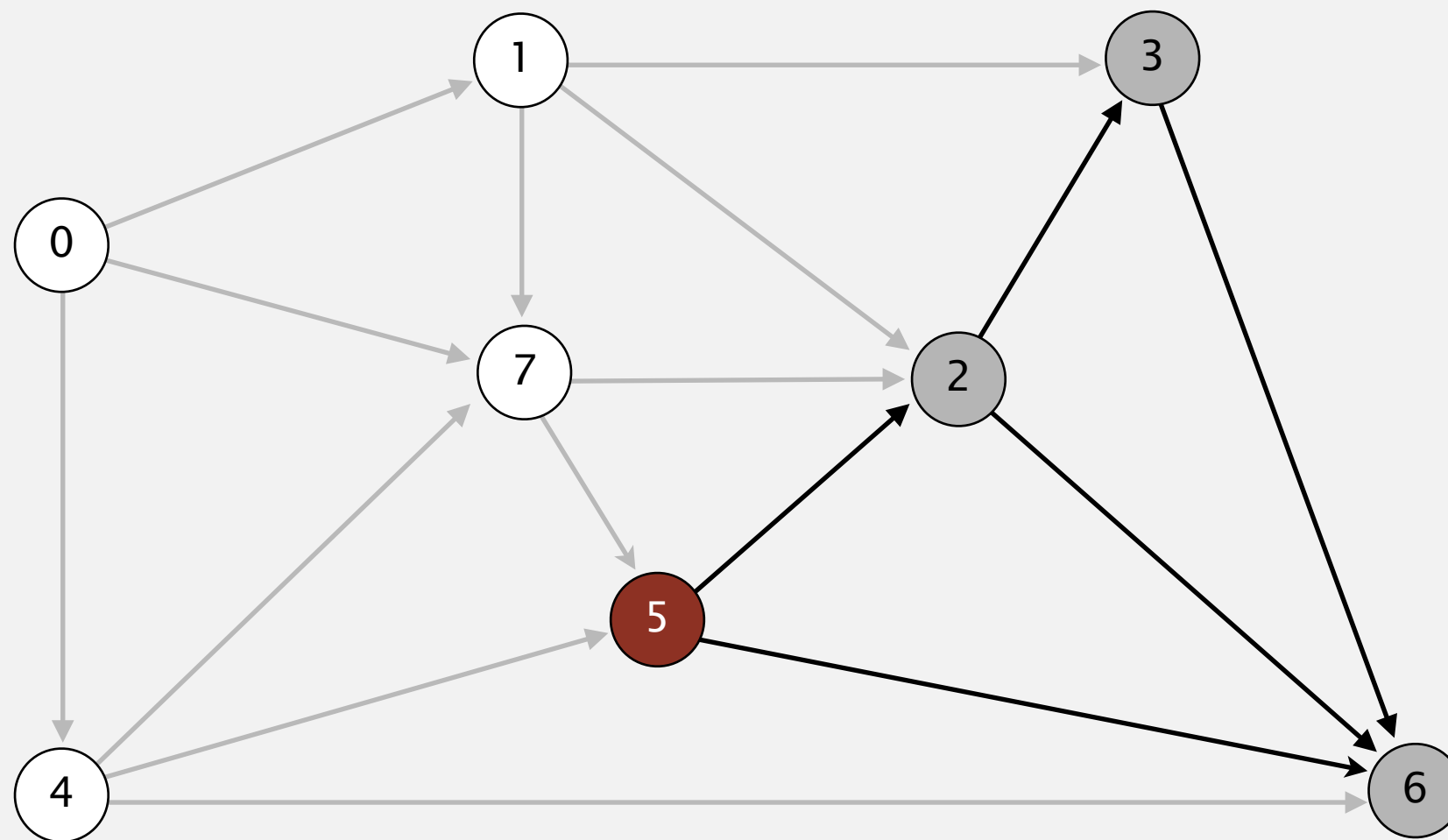
- Consider vertices in topological order.
- Relax all edges incident from that vertex.



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

# Topological sort algorithm

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

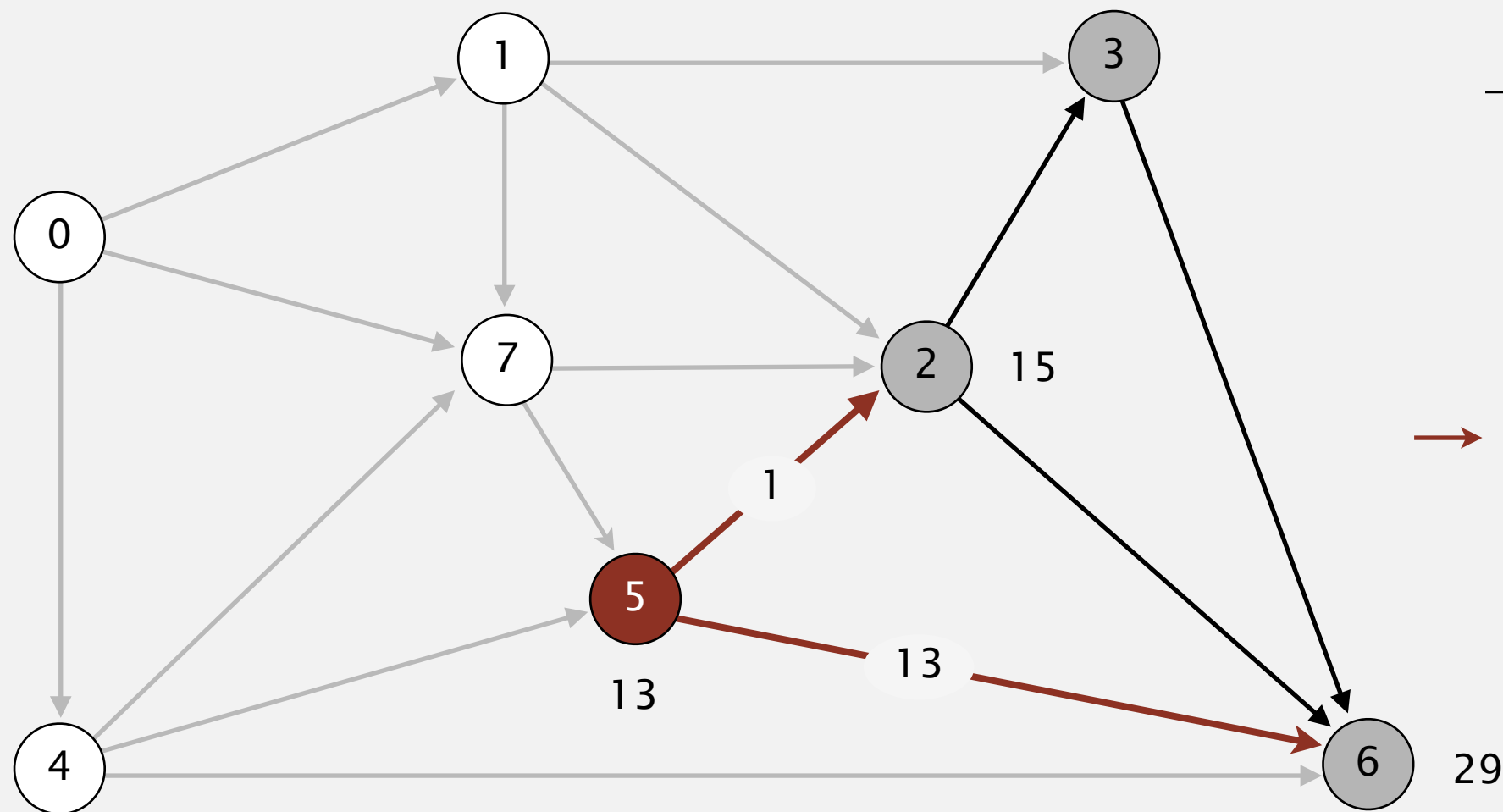


v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	15.0	7→2
3	20.0	1→3
4	9.0	0→4
→ 5	13.0	4→5
6	29.0	4→6
7	8.0	0→7

**select vertex 5**

# Topological sort algorithm

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

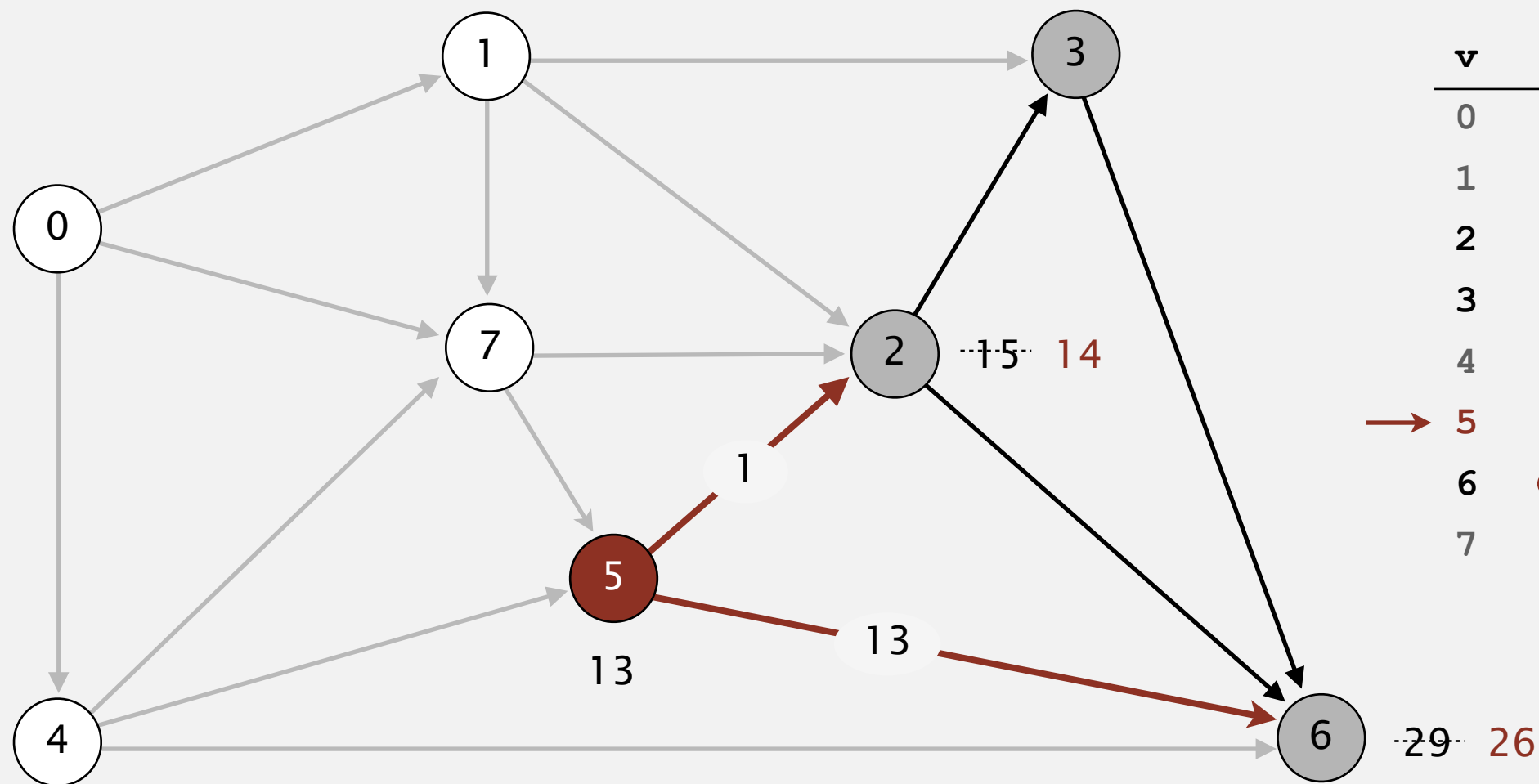


v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	15.0	7→2
3	20.0	1→3
4	9.0	0→4
→ 5	13.0	4→5
6	29.0	4→6
7	8.0	0→7

**relax all edges incident from 5**

# Topological sort algorithm

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

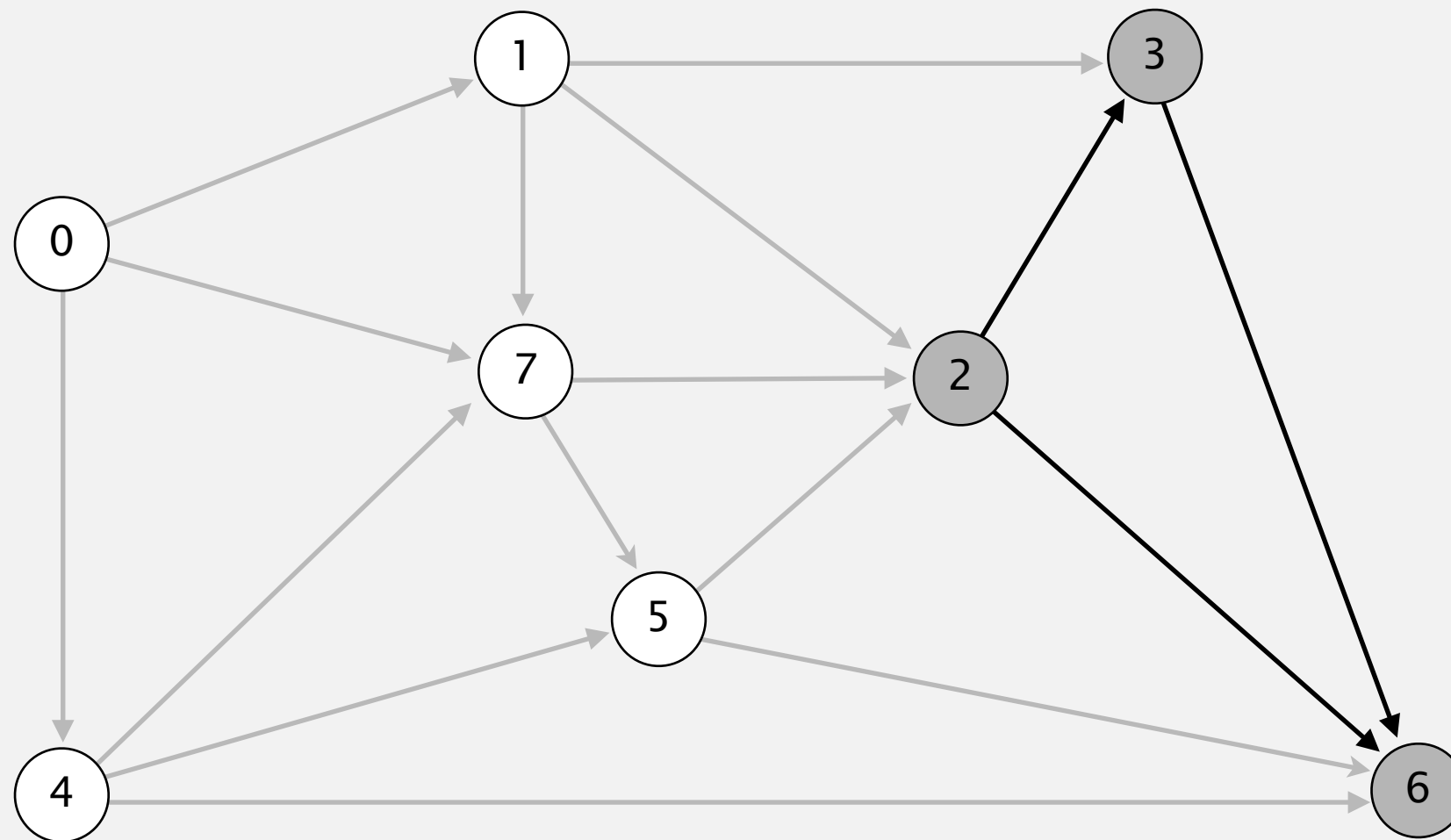


v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	14.0	5→2
3	20.0	1→3
4	9.0	0→4
→ 5	13.0	4→5
6	26.0	5→6
7	8.0	0→7

relax all edges incident from 5

# Topological sort algorithm

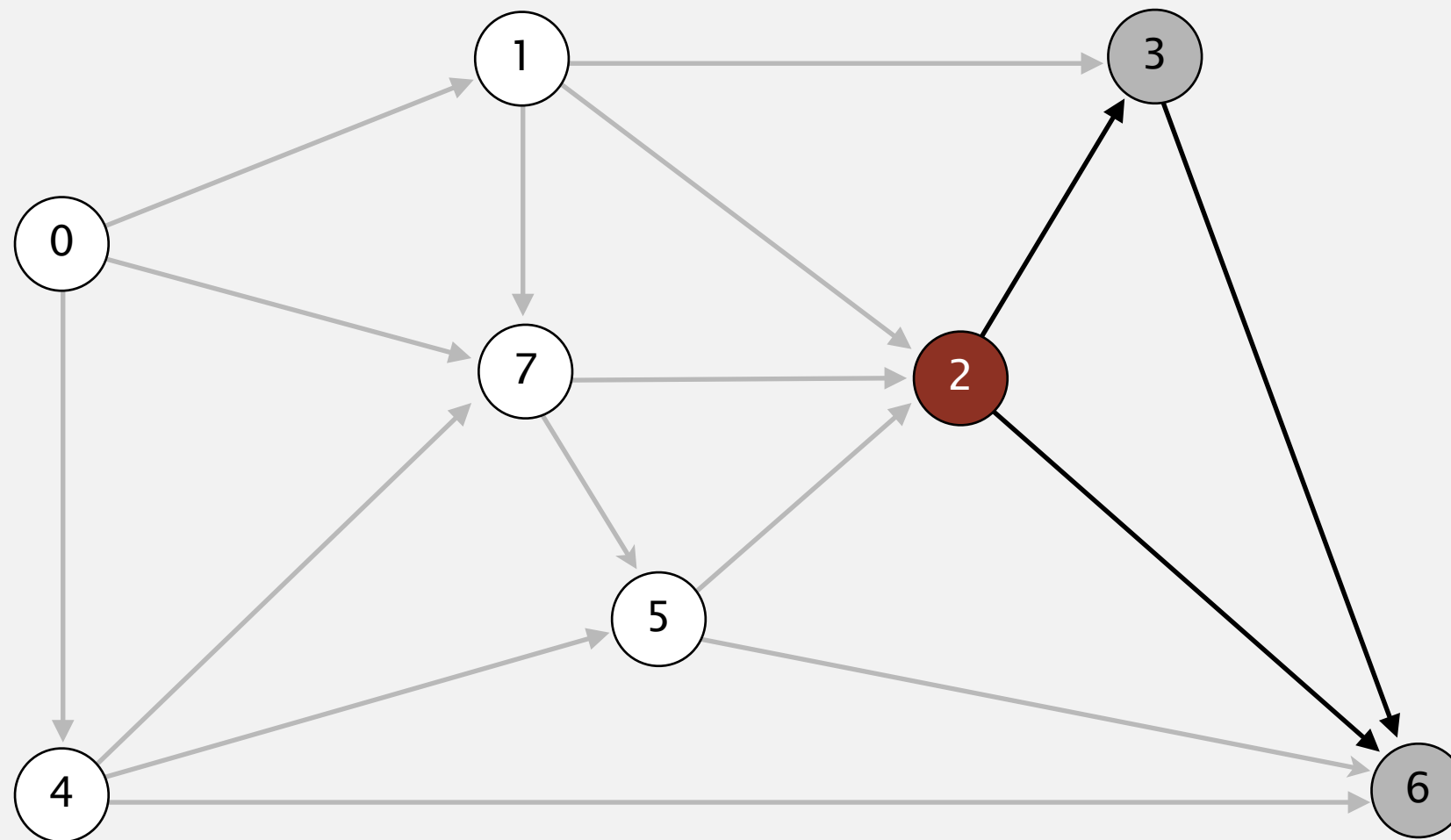
- Consider vertices in topological order.
- Relax all edges incident from that vertex.



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

# Topological sort algorithm

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



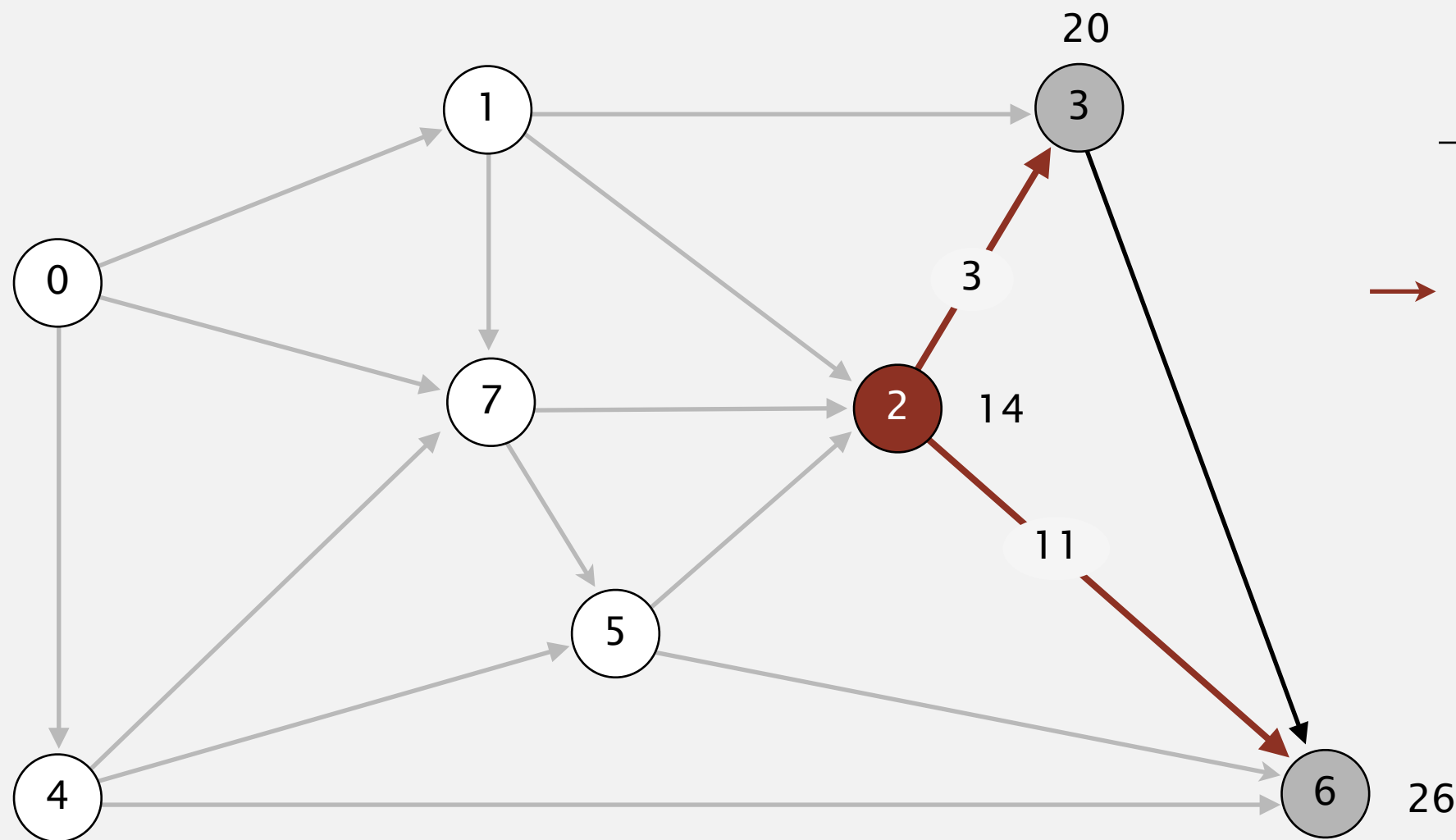
**select vertex 2**

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



# Topological sort algorithm

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

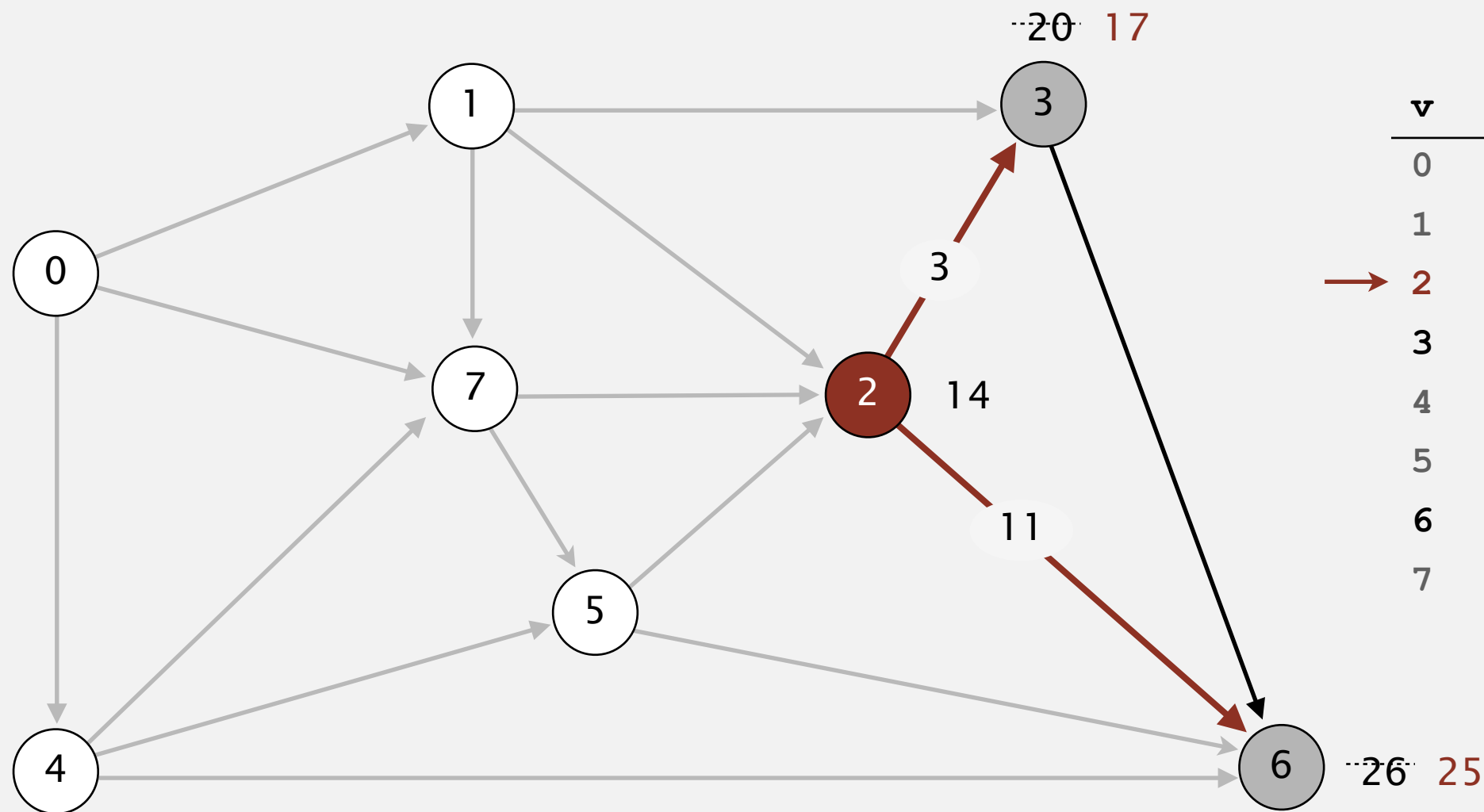


v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
→ 2	14.0	5→2
3	20.0	1→3
4	9.0	0→4
5	13.0	4→5
6	26.0	5→6
7	8.0	0→7

relax all edges incident from 2

# Topological sort algorithm

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

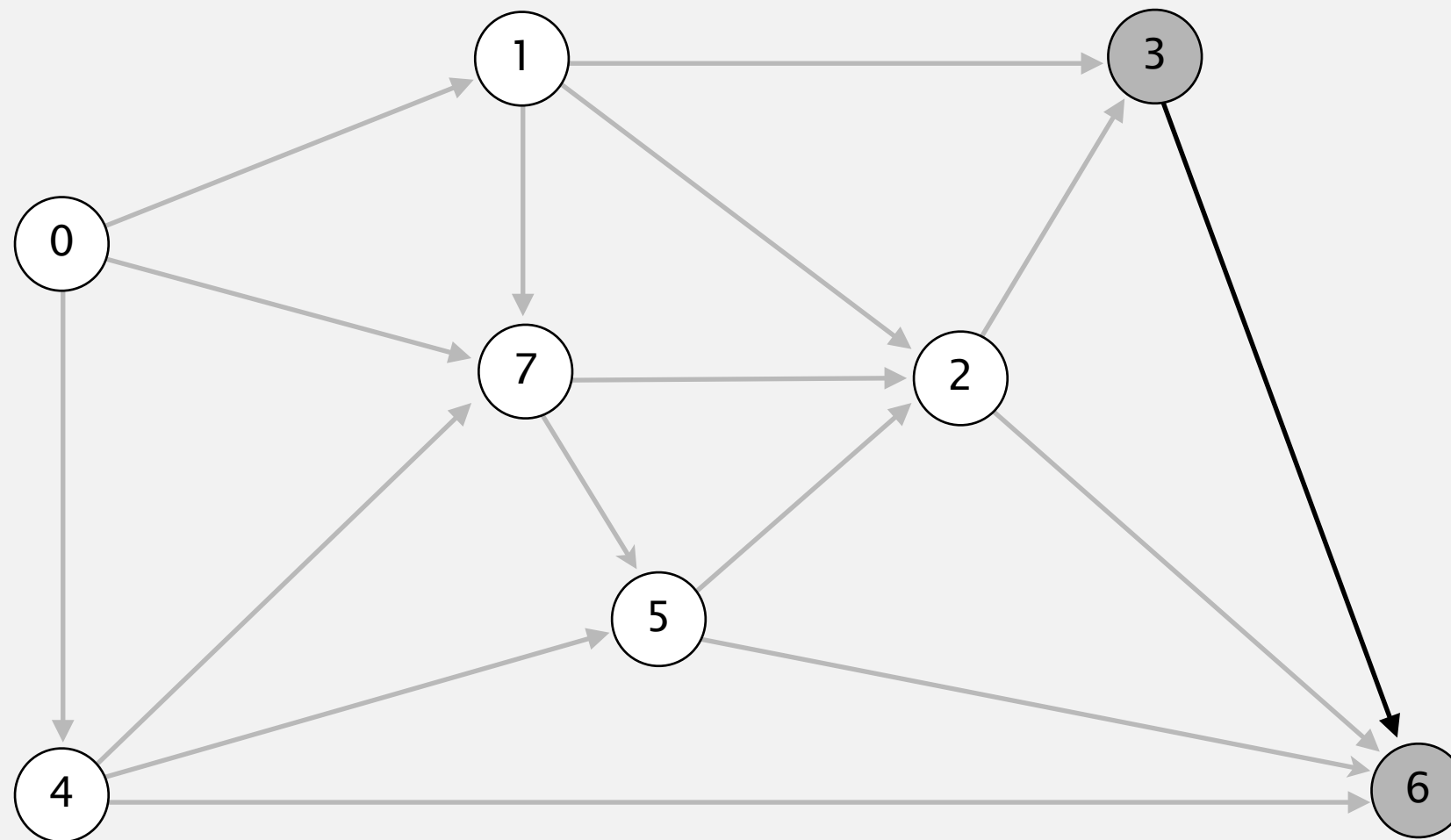


v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
→ 2	14.0	5→2
3	17.0	2→3
4	9.0	0→4
5	13.0	4→5
6	25.0	2→6
7	8.0	0→7

relax all edges incident from 2

# Topological sort algorithm

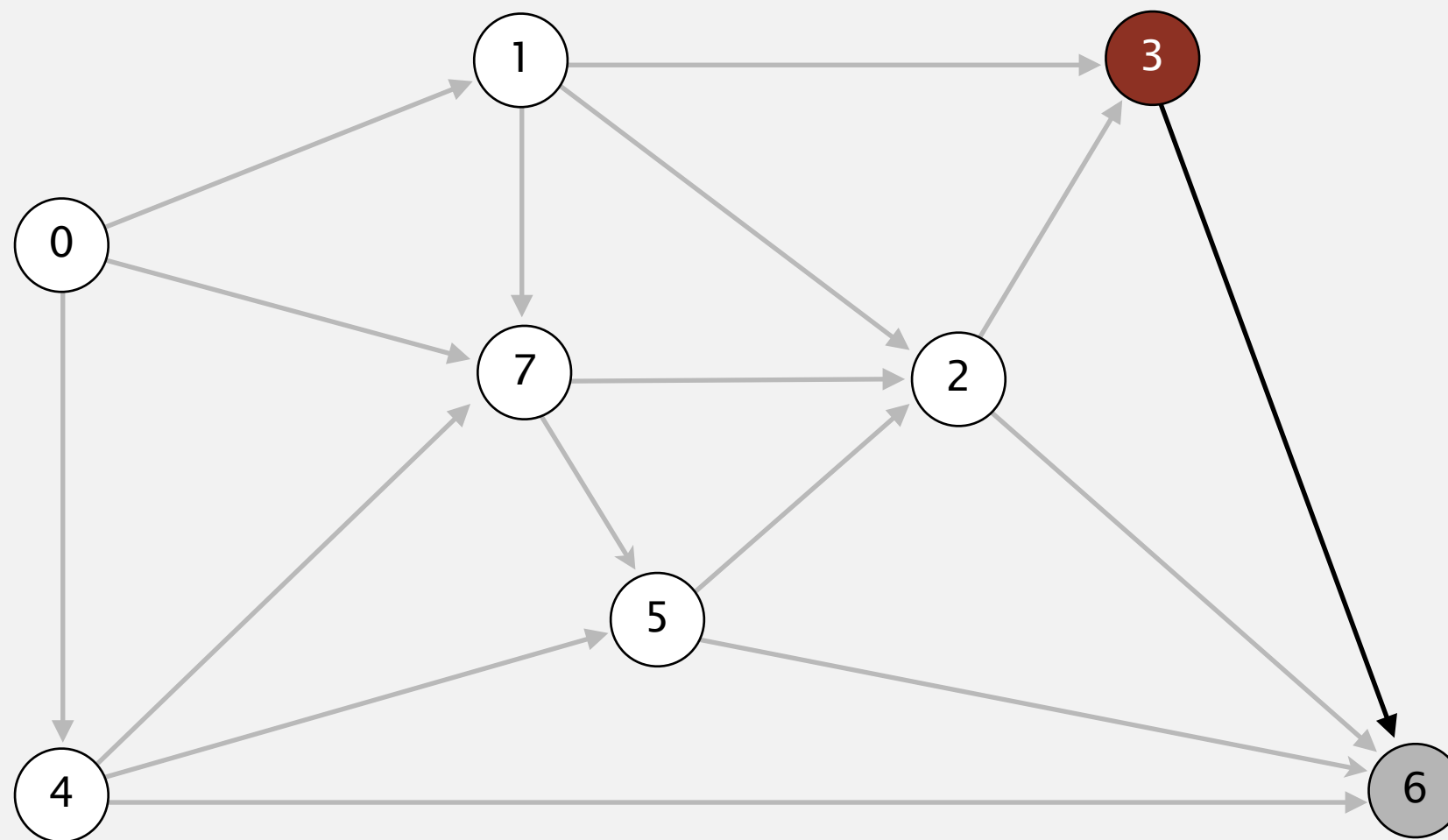
- Consider vertices in topological order.
- Relax all edges incident from that vertex.



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

# Topological sort algorithm

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

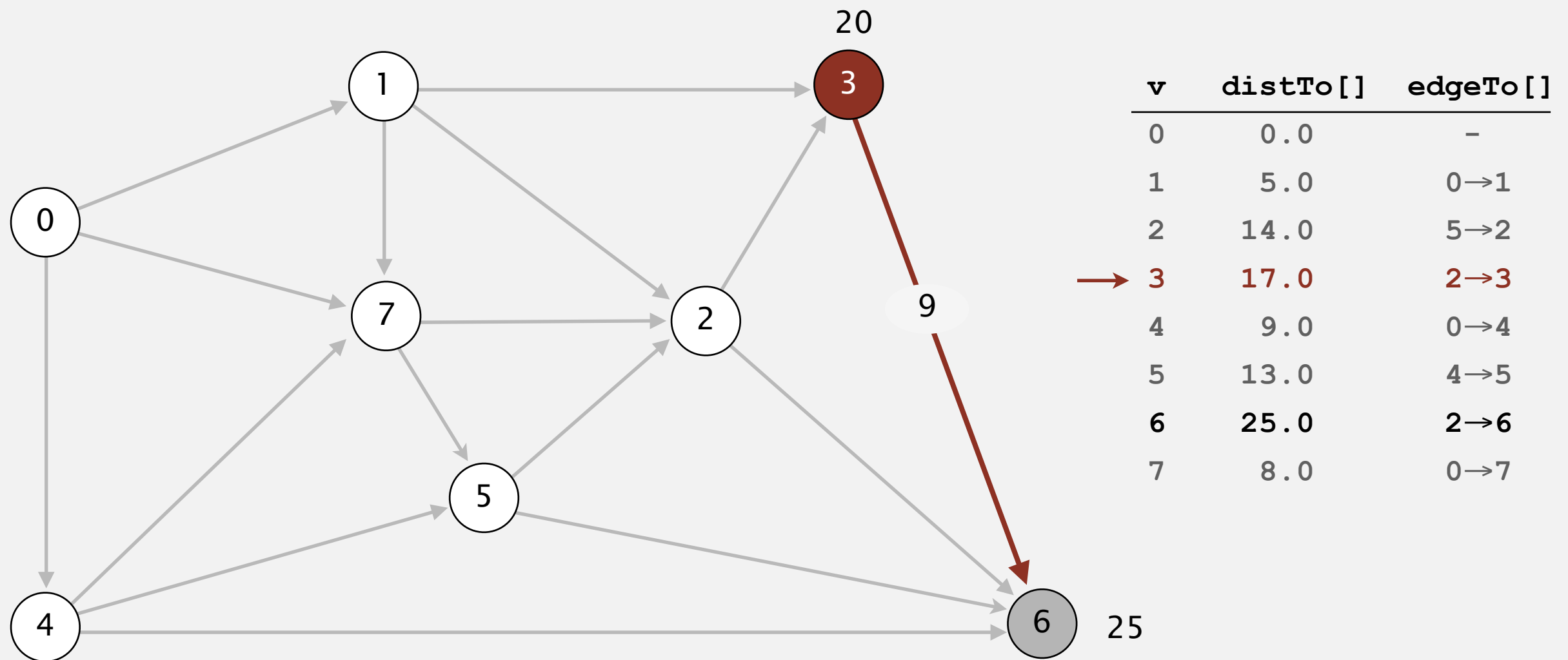


**select vertex 3**

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

# Topological sort algorithm

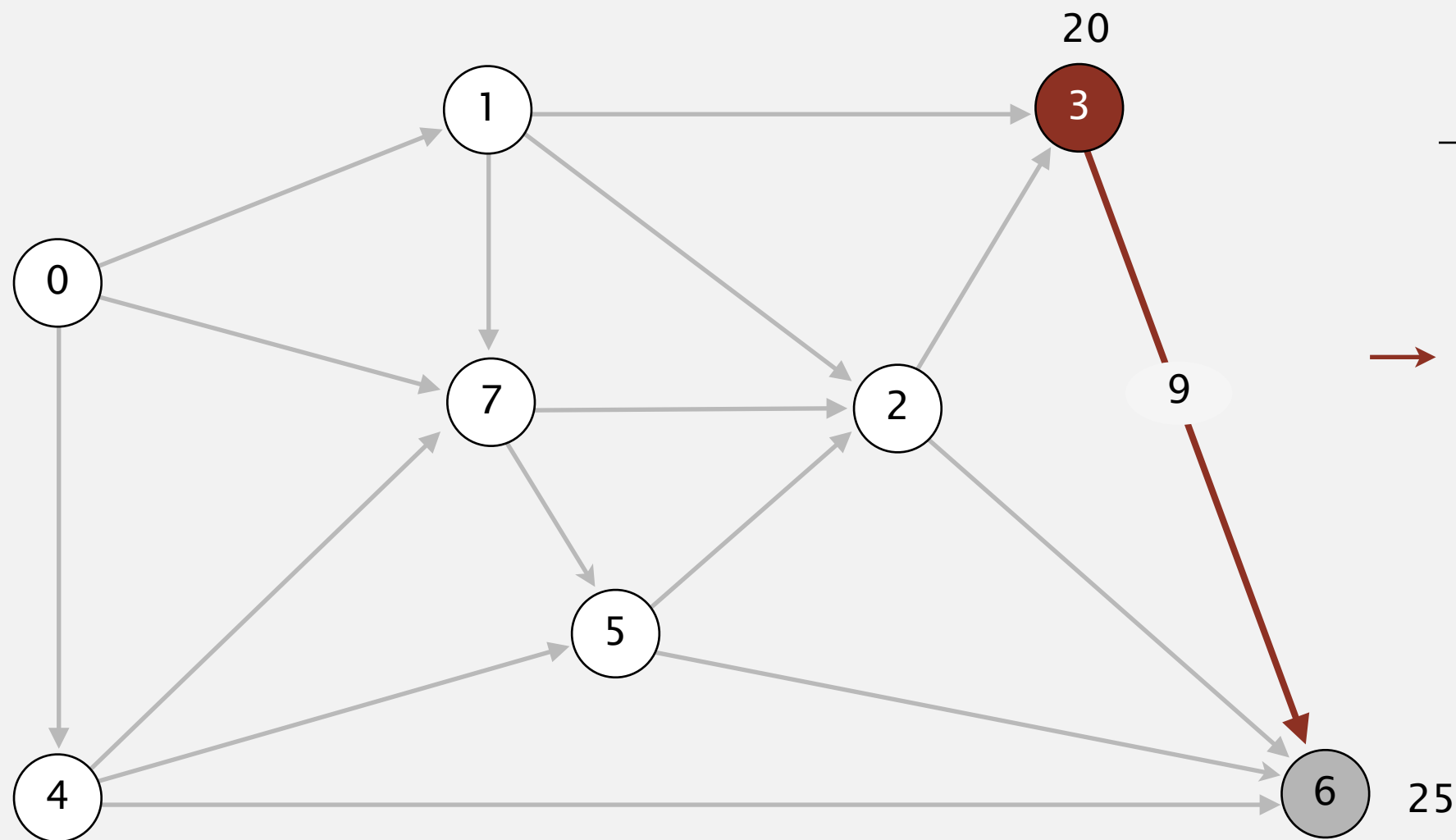
- Consider vertices in topological order.
- Relax all edges incident from that vertex.



relax all edges incident from 3

# Topological sort algorithm

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

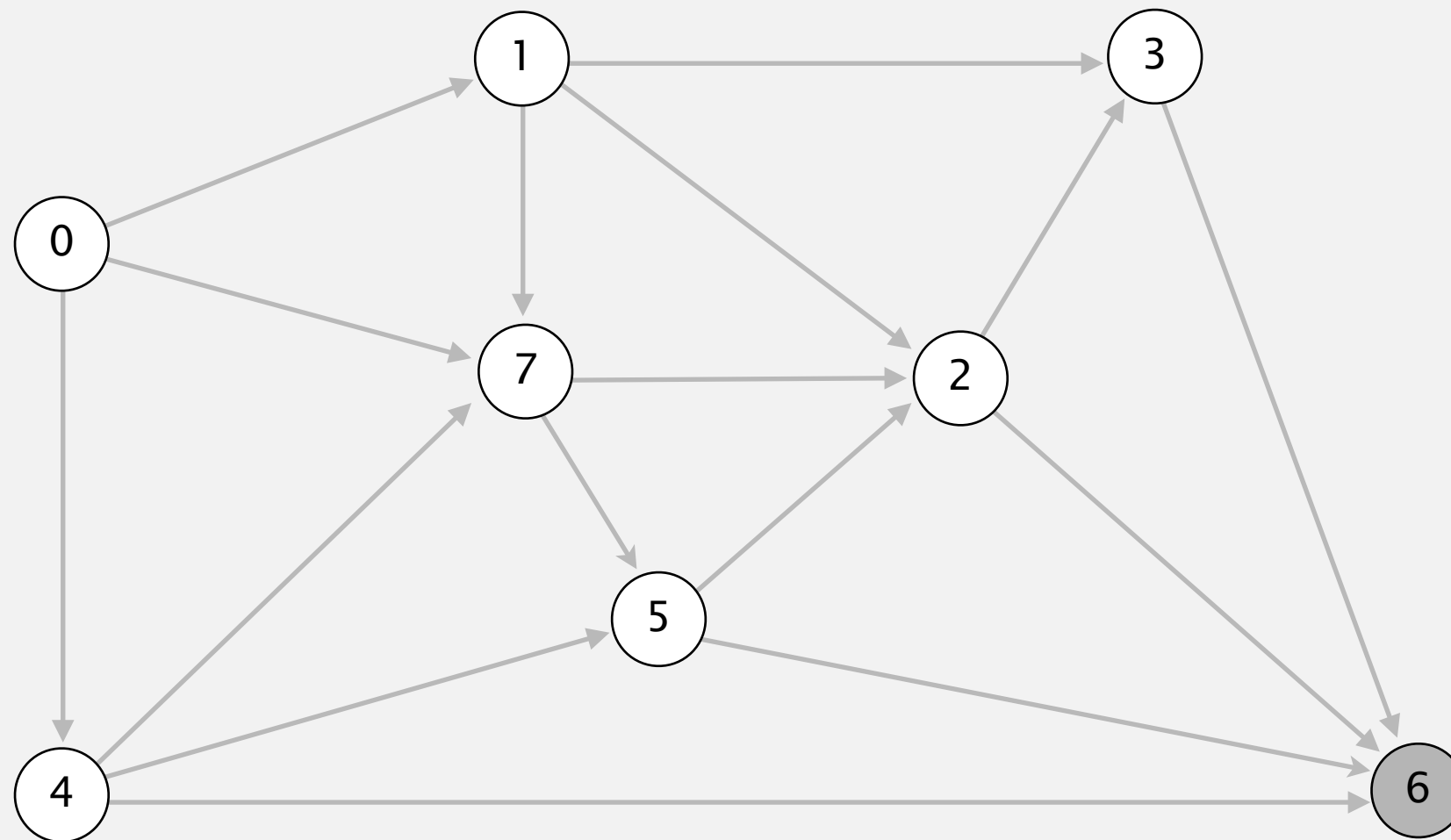


v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	14.0	5→2
3	17.0	2→3
4	9.0	0→4
5	13.0	4→5
6	25.0 ✓	2→6
7	8.0	0→7

relax all edges incident from 3

# Topological sort algorithm

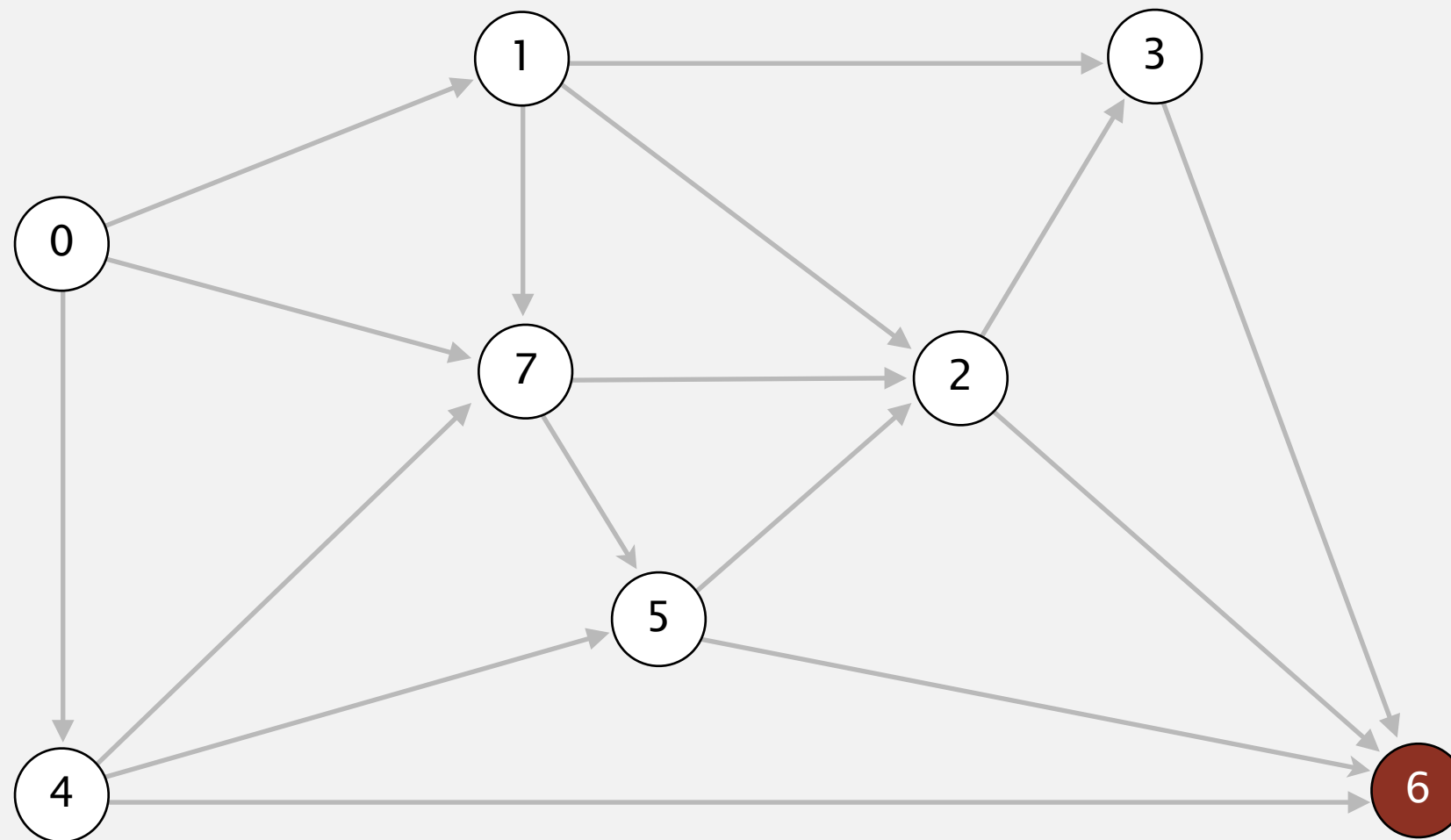
- Consider vertices in topological order.
- Relax all edges incident from that vertex.



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

# Topological sort algorithm

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



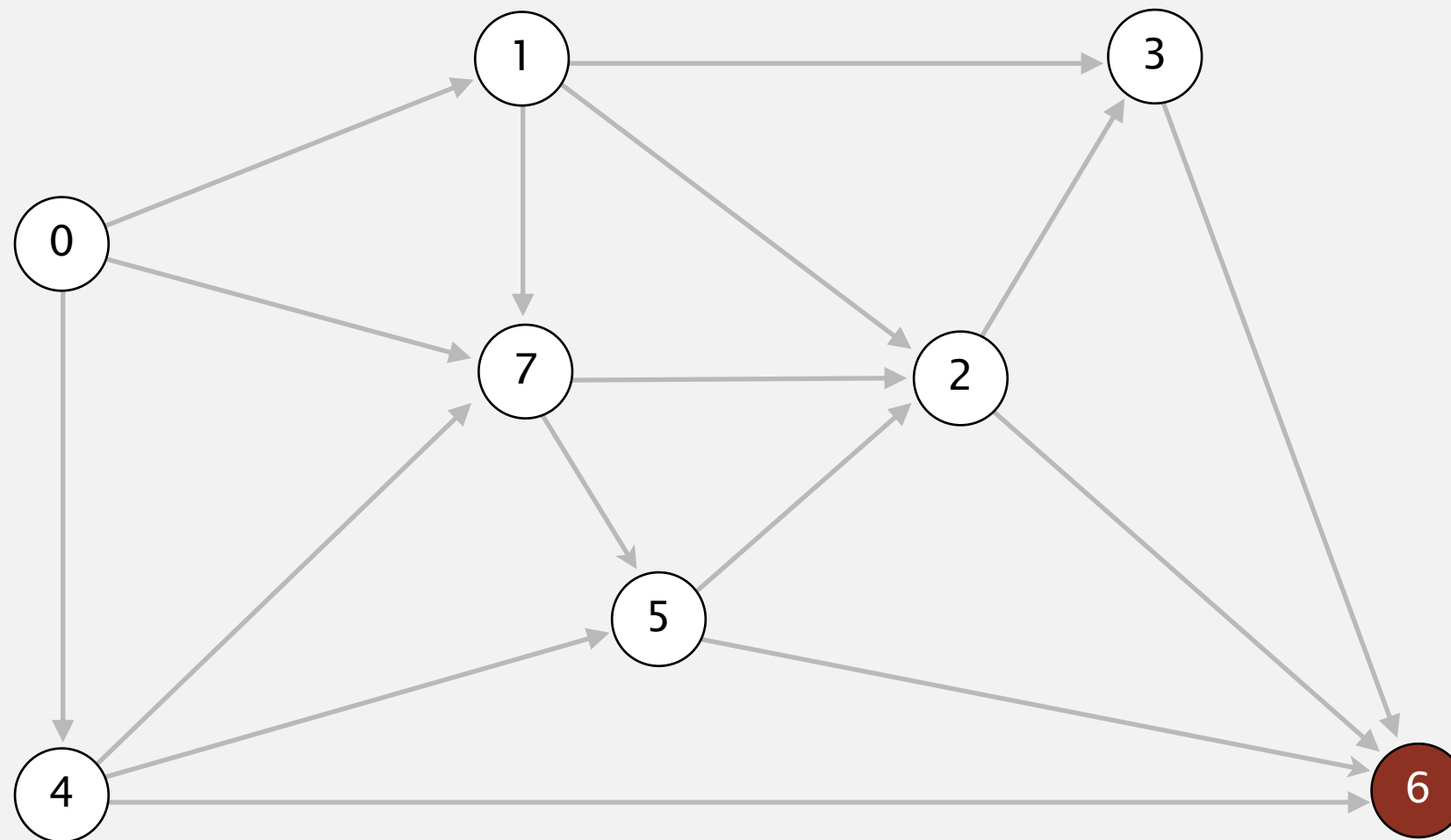
**select vertex 6**

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	<b>25.0</b>	<b>2→6</b>
7	8.0	0→7



# Topological sort algorithm

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

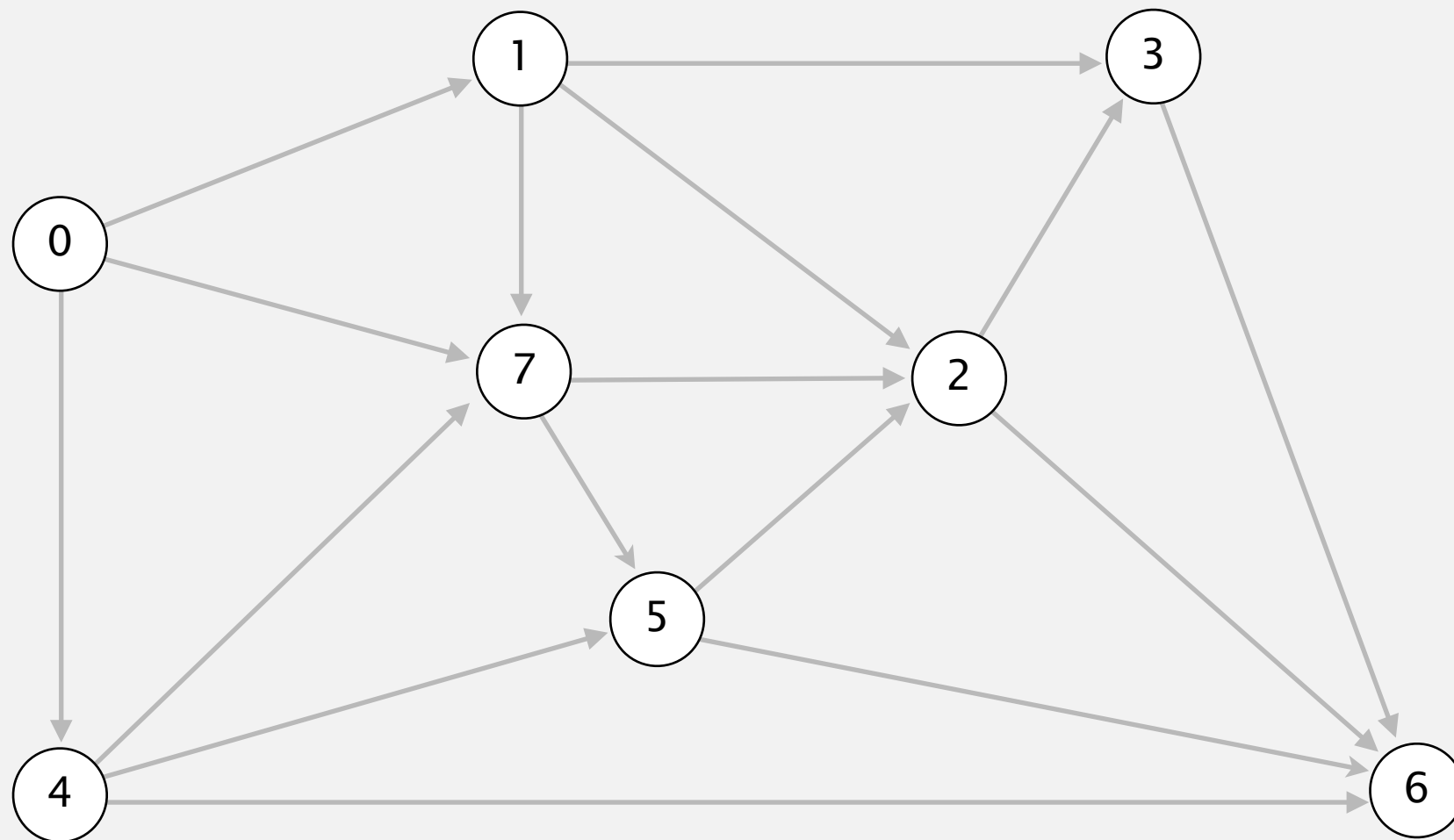


v	distTo[]	edgeTo[]
0	0.0	-
1	5.0	0→1
2	14.0	5→2
3	17.0	2→3
4	9.0	0→4
5	13.0	4→5
→ 6	25.0	2→6
7	8.0	0→7

**relax all edges incident from 6**

# Topological sort algorithm

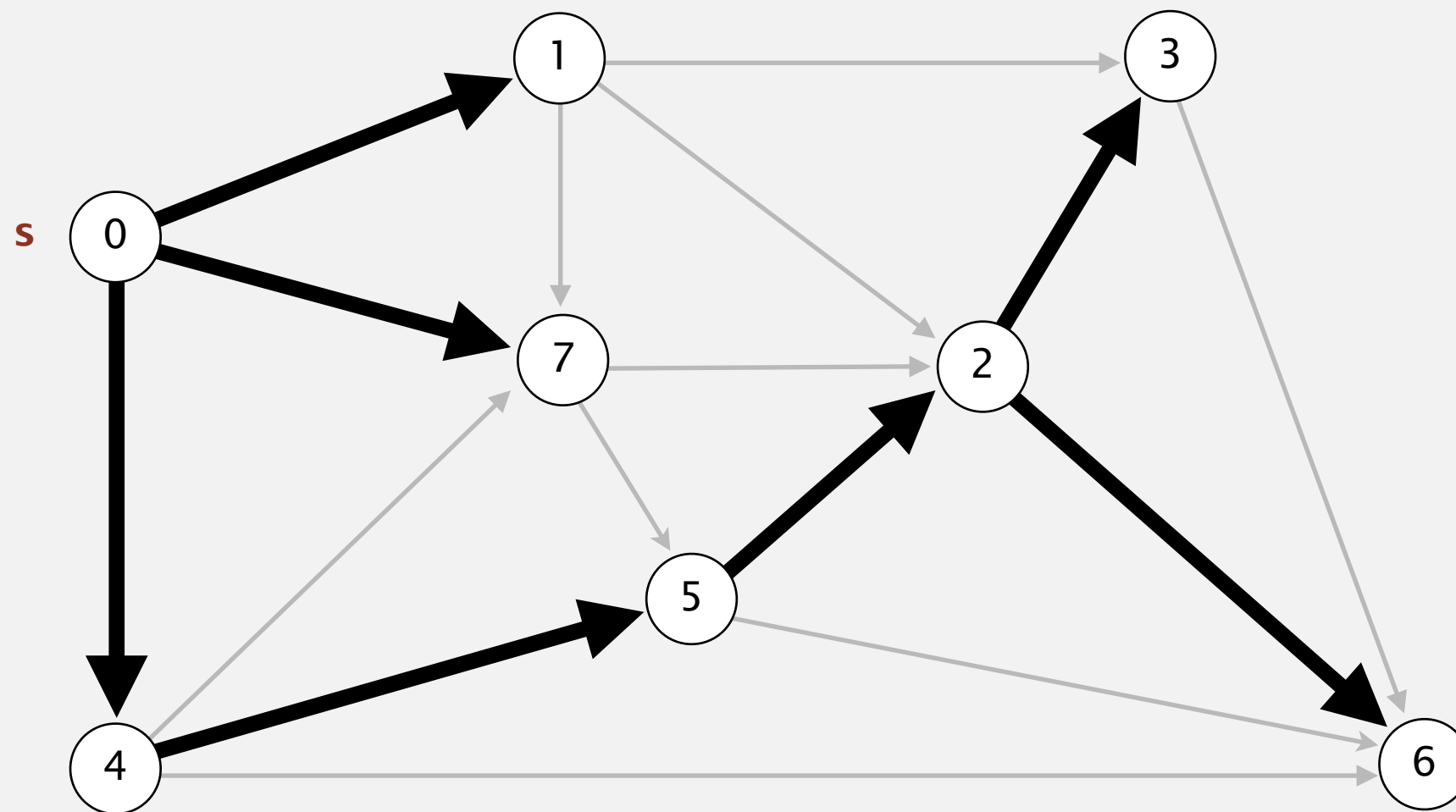
- Consider vertices in topological order.
- Relax all edges incident from that vertex.



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

# Topological sort algorithm

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



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**