



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

PRIM'S ALGORITHM DEMO

- ▶ *Prim's algorithm*
- ▶ *lazy implementation*
- ▶ *eager implementation*



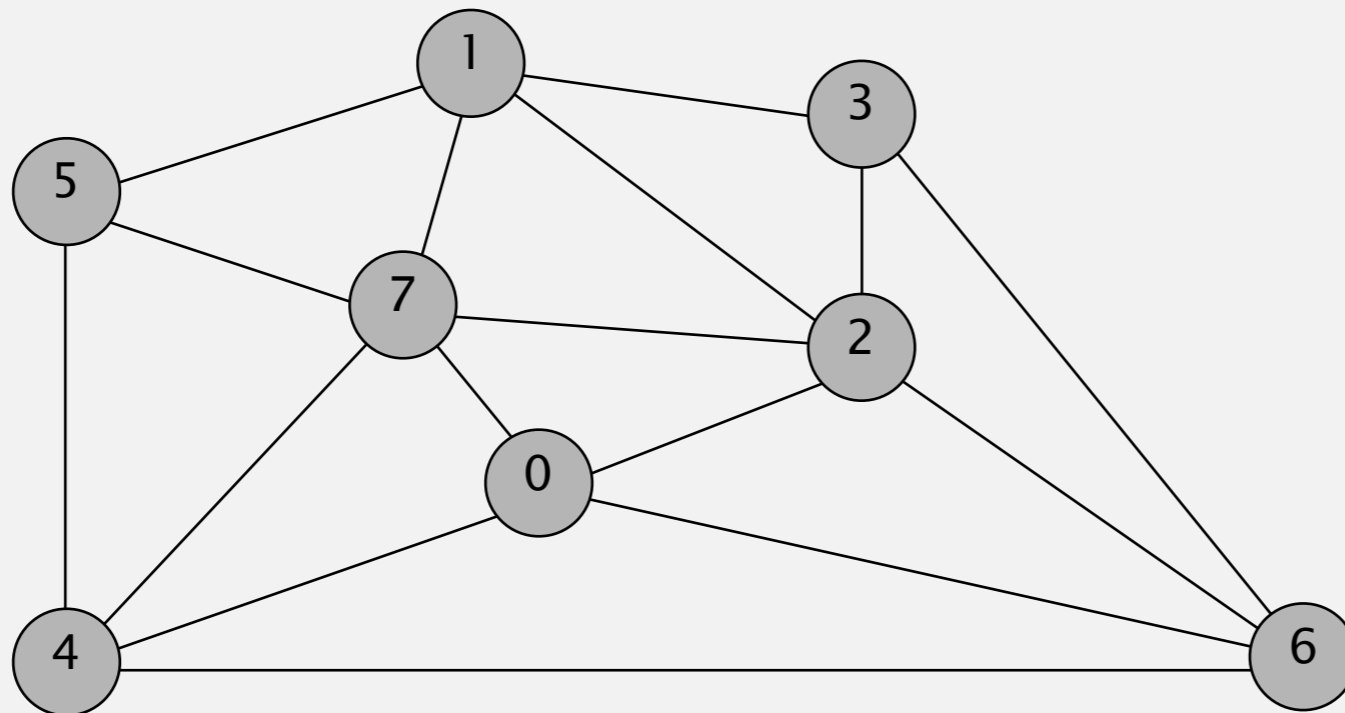
<https://algs4.cs.princeton.edu>

PRIM'S ALGORITHM DEMO

- ▶ *Prim's algorithm*
- ▶ *lazy implementation*
- ▶ *eager implementation*

Prim's algorithm demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

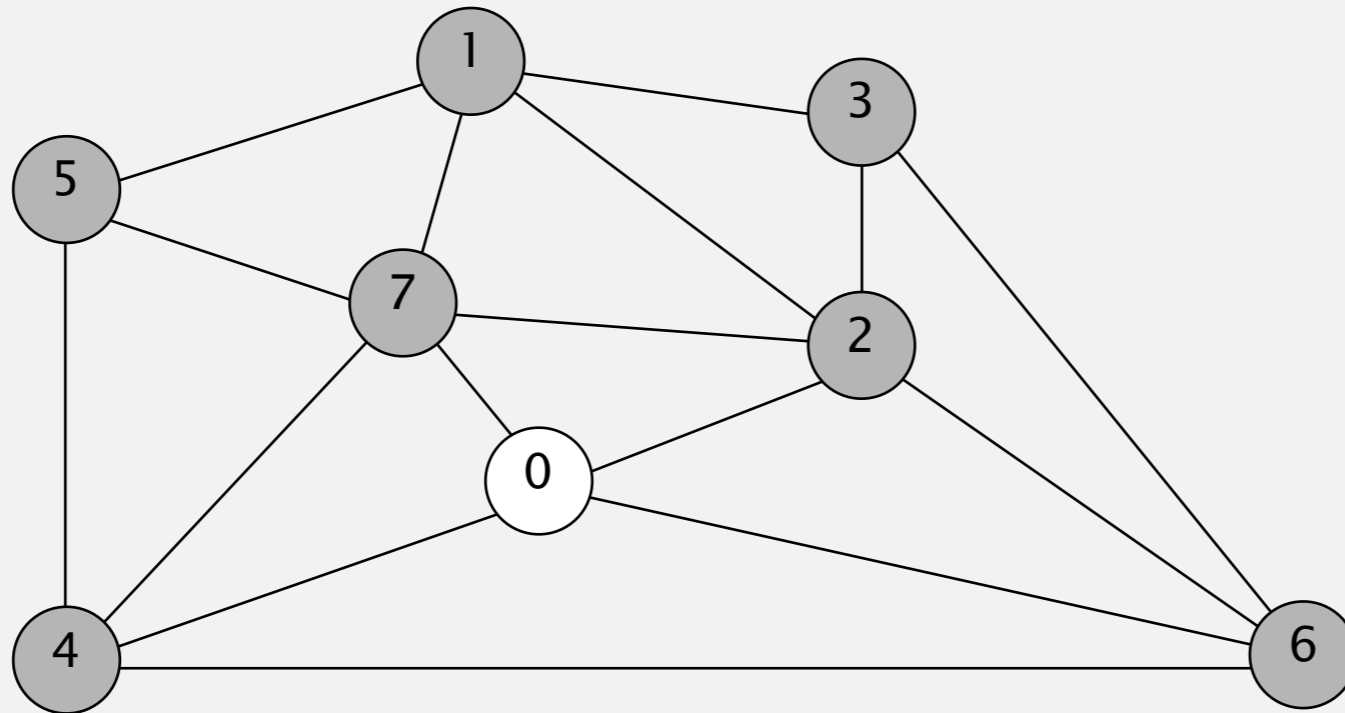


an edge-weighted graph

0-7	0.16
2-3	0.17
1-7	0.19
0-2	0.26
5-7	0.28
1-3	0.29
1-5	0.32
2-7	0.34
4-5	0.35
1-2	0.36
4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58
6-4	0.93

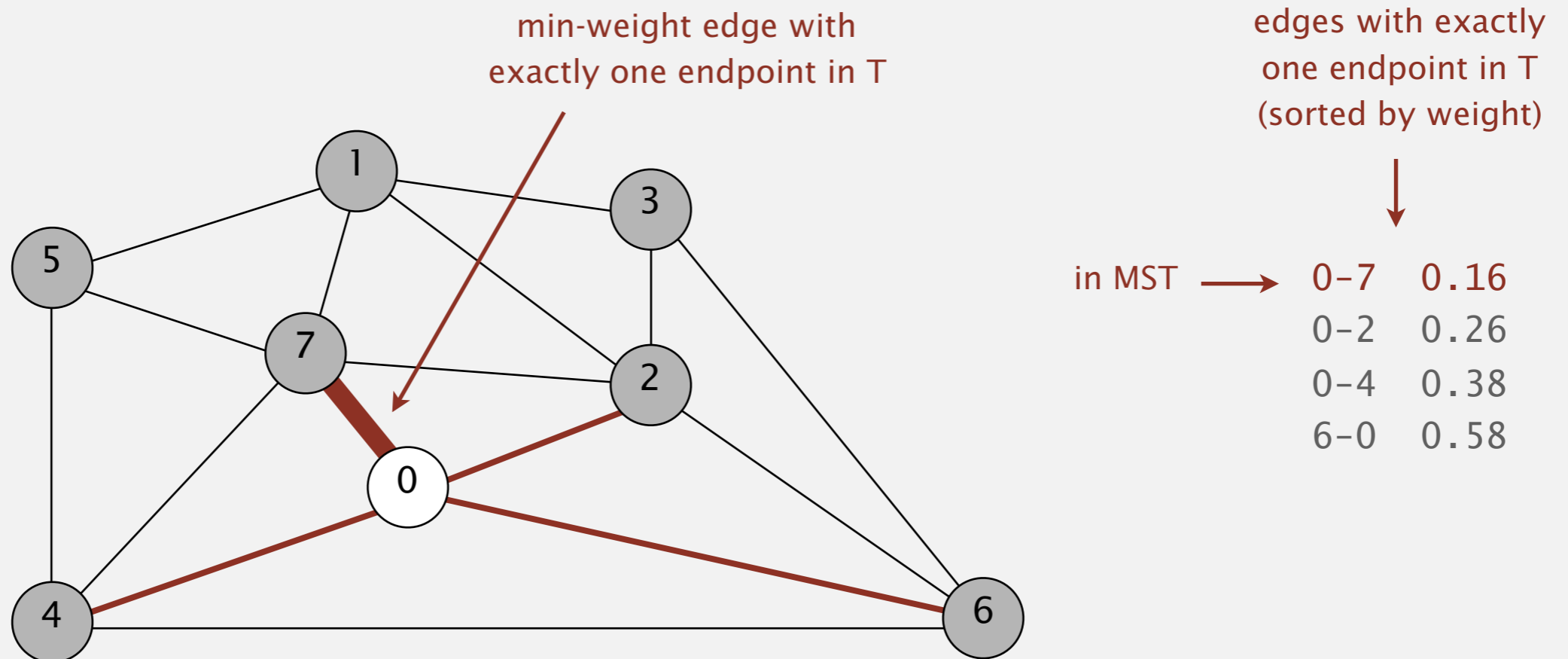
Prim's algorithm demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



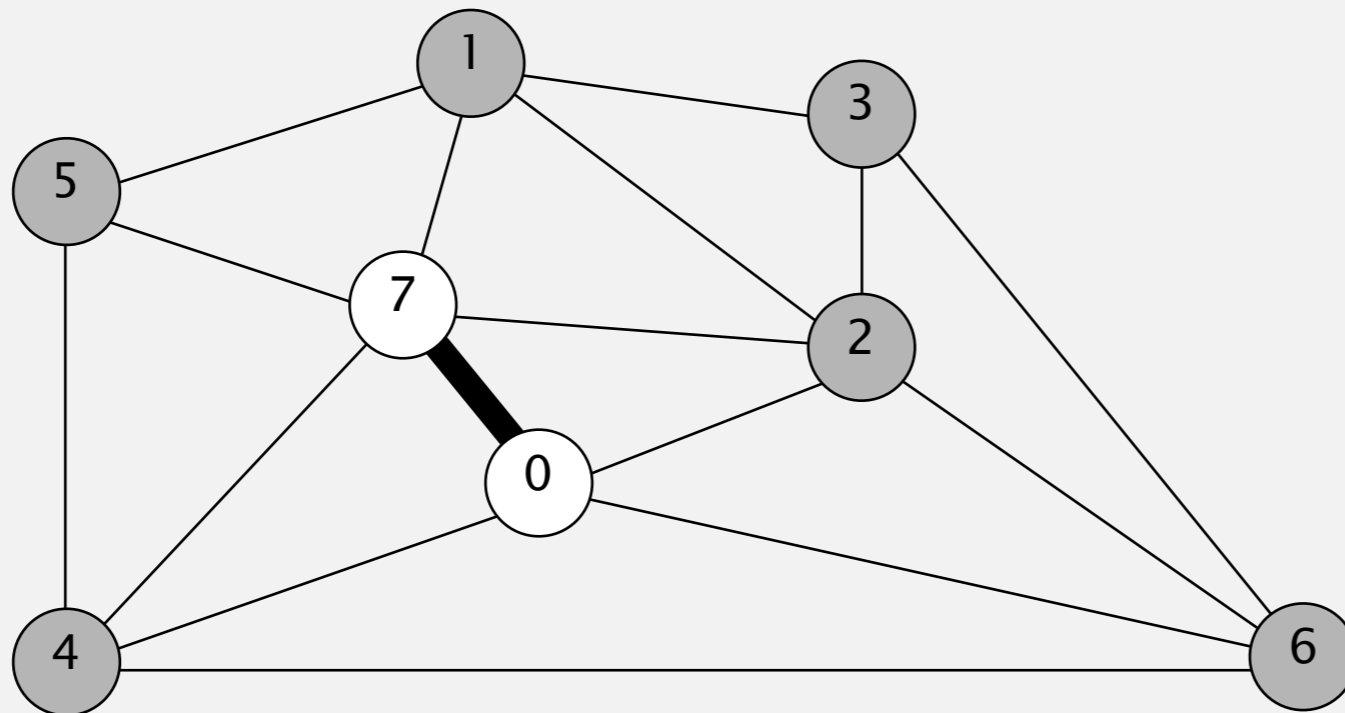
Prim's algorithm demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



Prim's algorithm demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

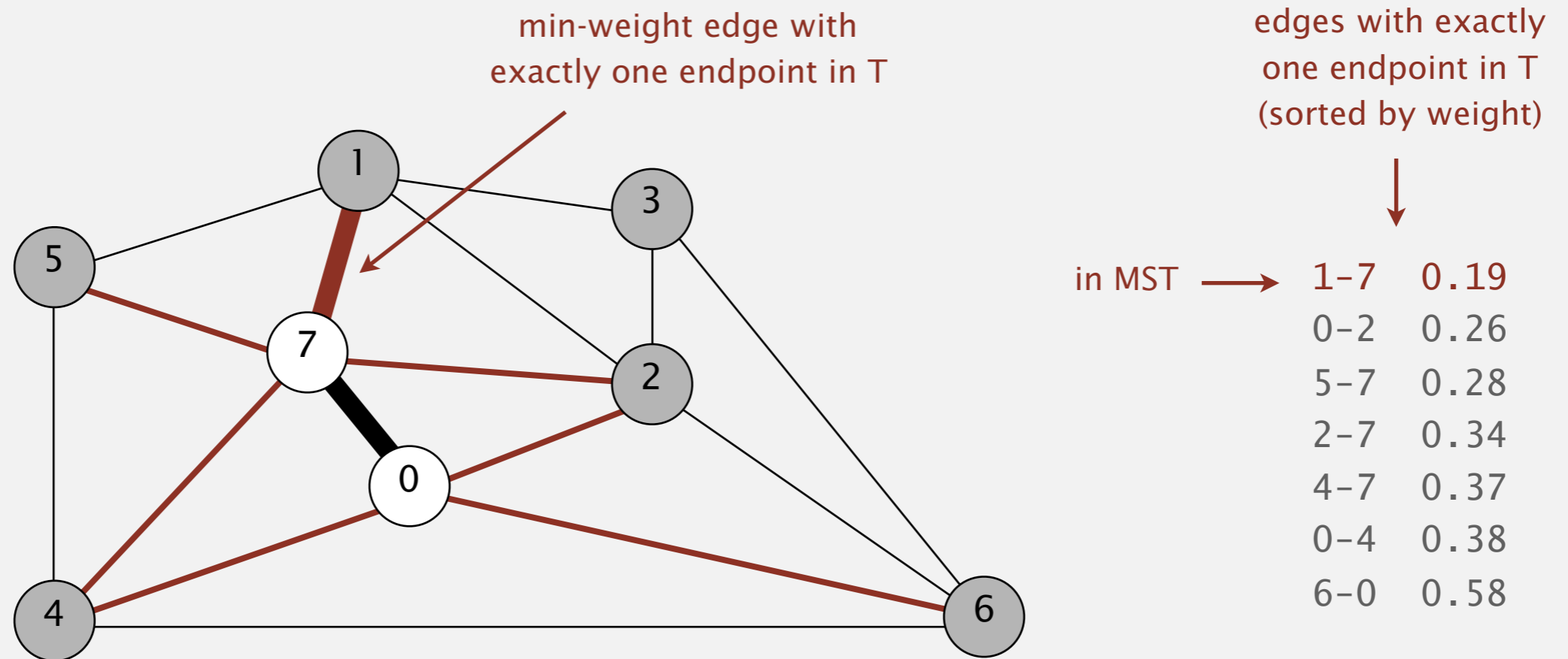


MST edges

0-7

Prim's algorithm demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

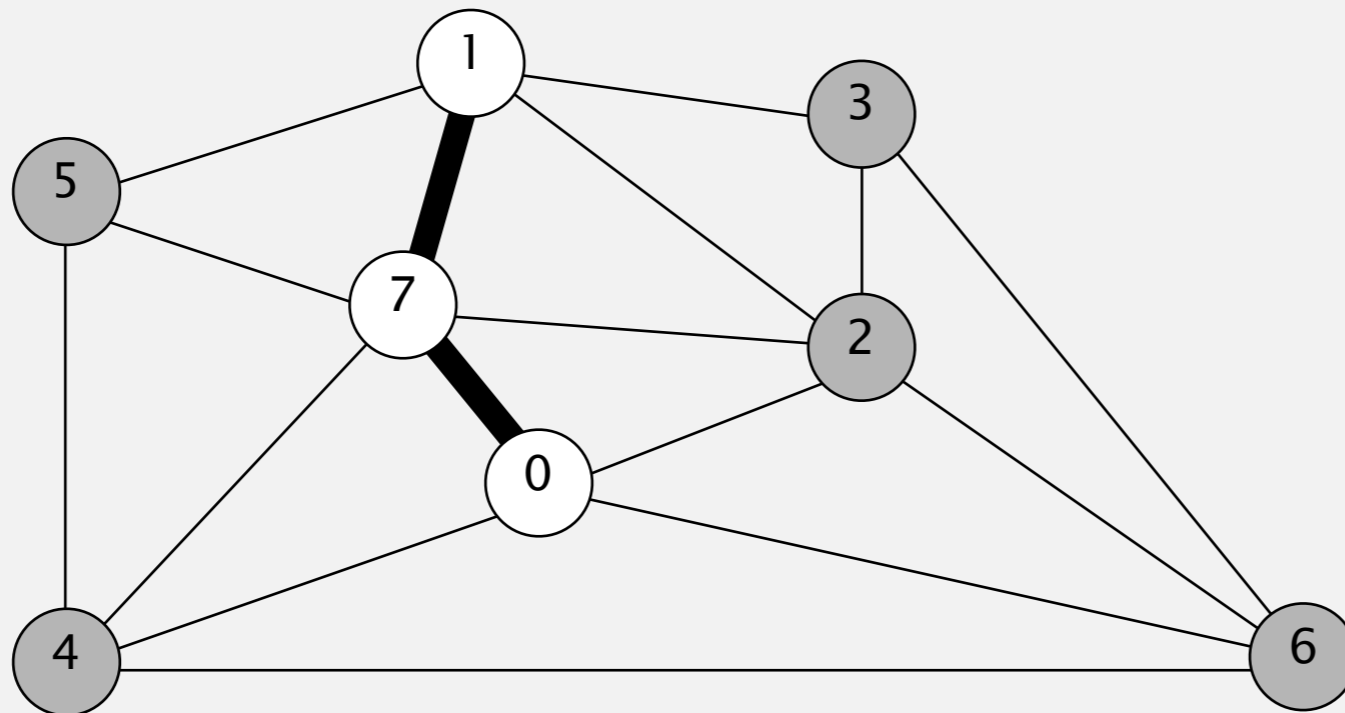


MST edges

0-7

Prim's algorithm demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

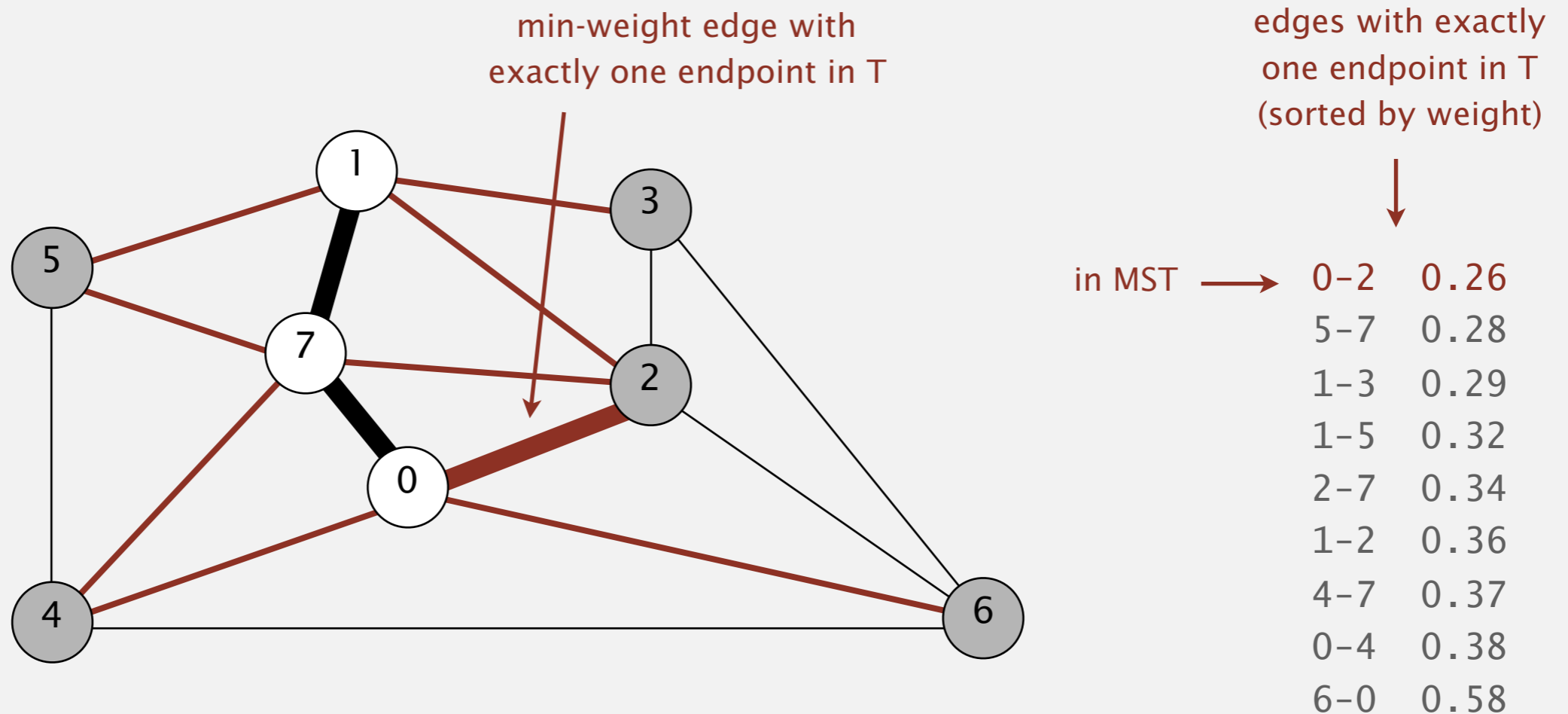


MST edges

0-7 1-7

Prim's algorithm demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

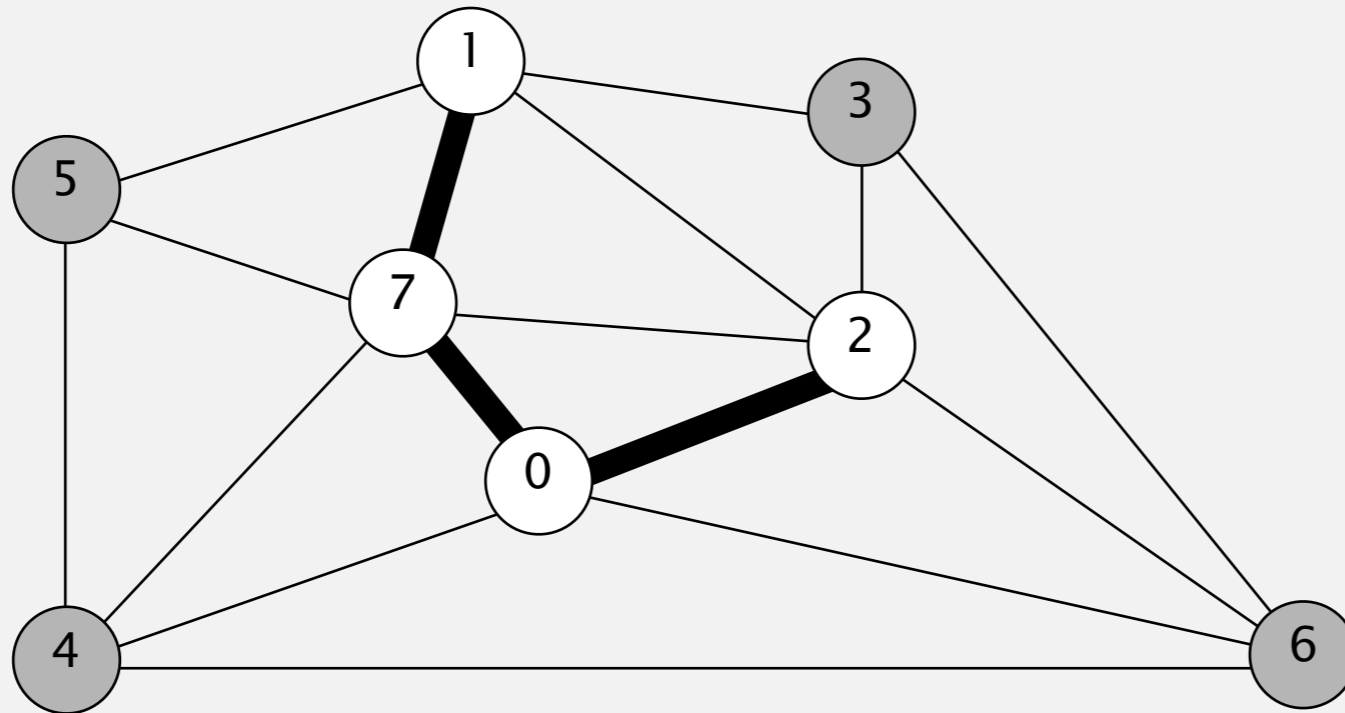


MST edges

0-7 1-7

Prim's algorithm demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

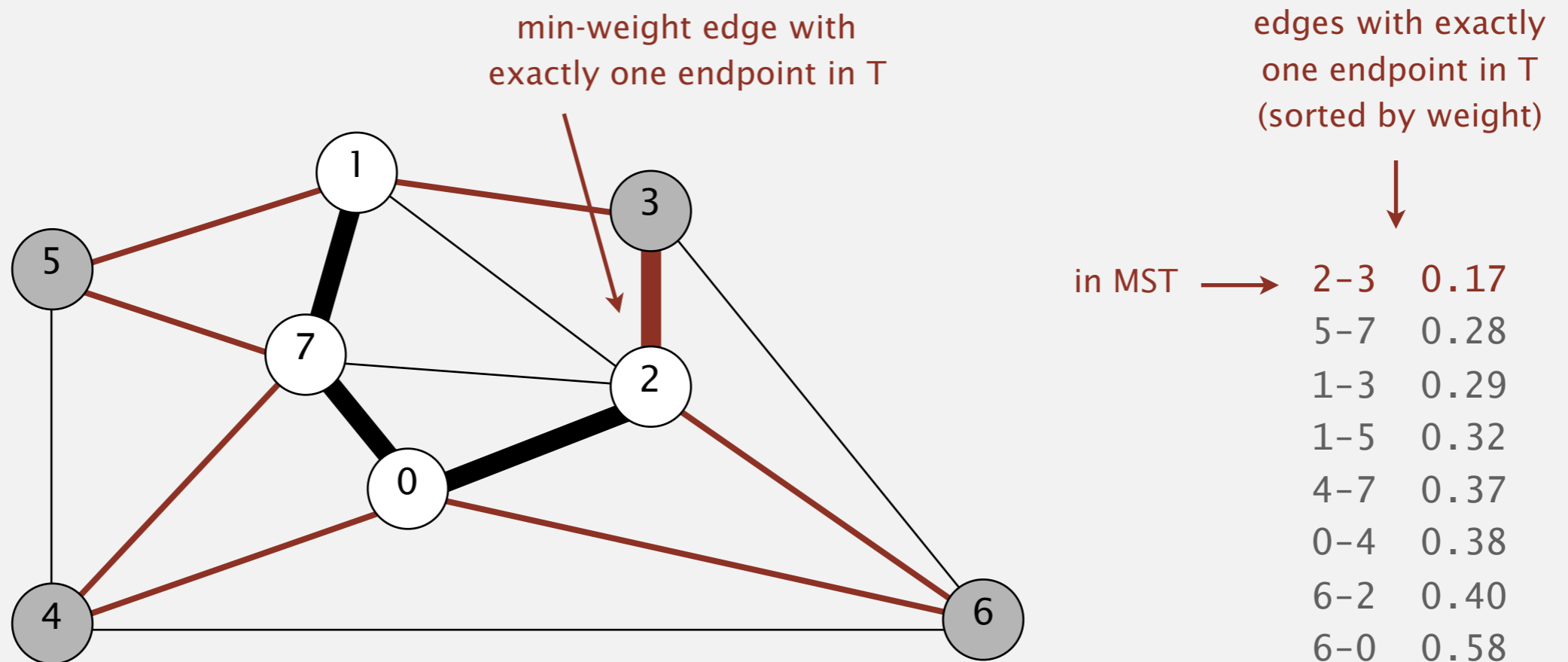


MST edges

0-7 1-7 0-2

Prim's algorithm demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

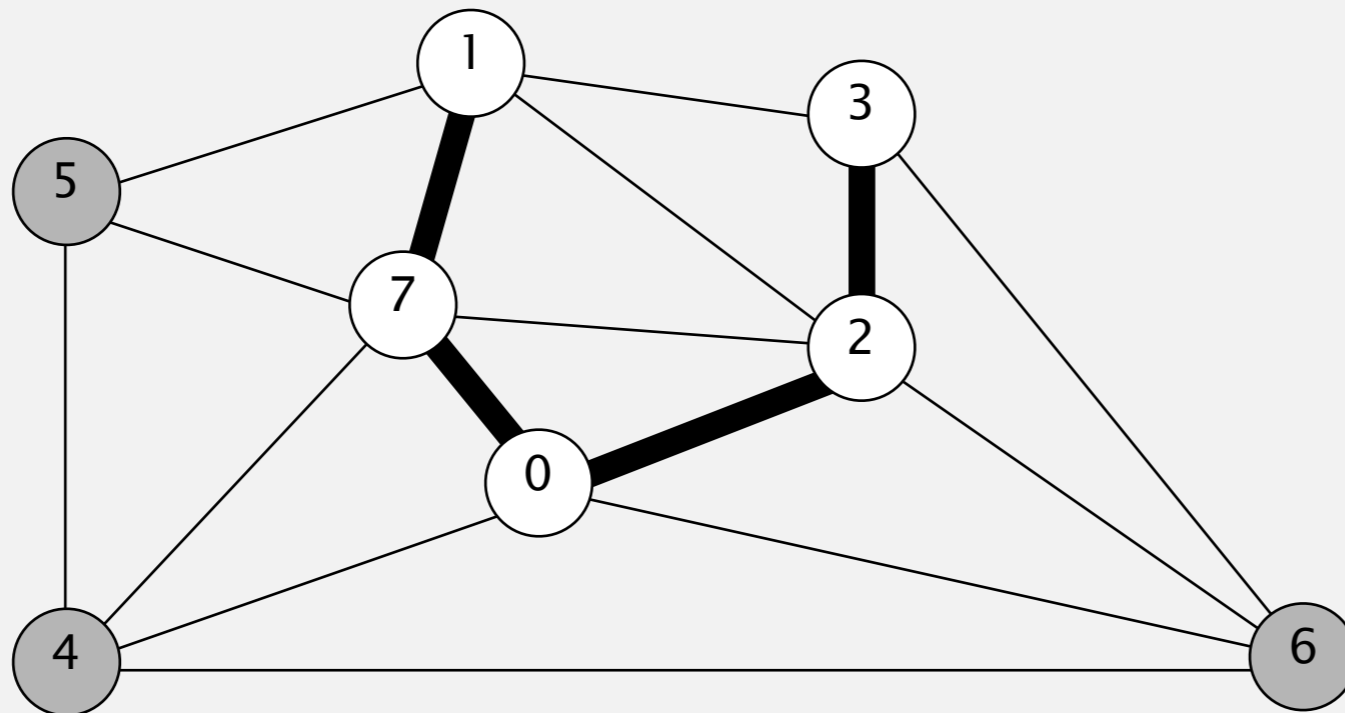


MST edges

0-7 1-7 0-2

Prim's algorithm demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



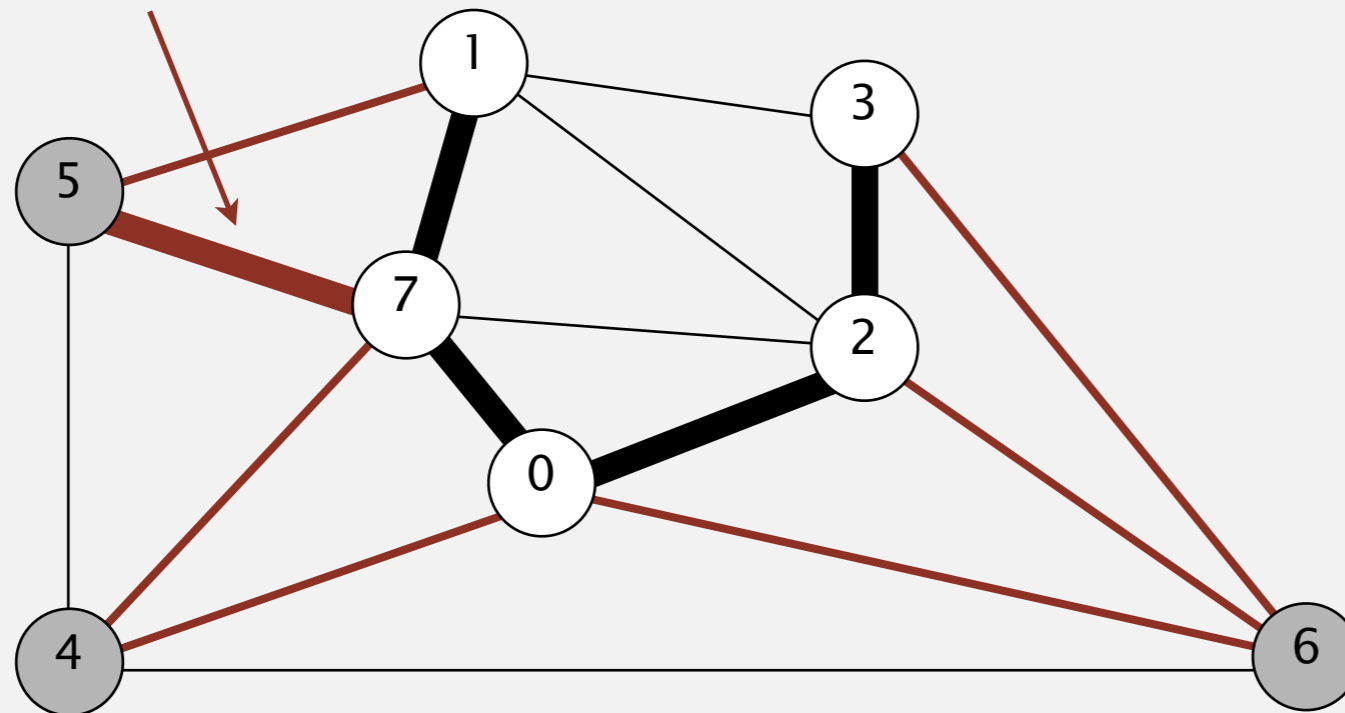
MST edges

0-7 1-7 0-2 2-3

Prim's algorithm demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

min-weight edge with exactly one endpoint in T



edges with exactly one endpoint in T (sorted by weight)

in MST →

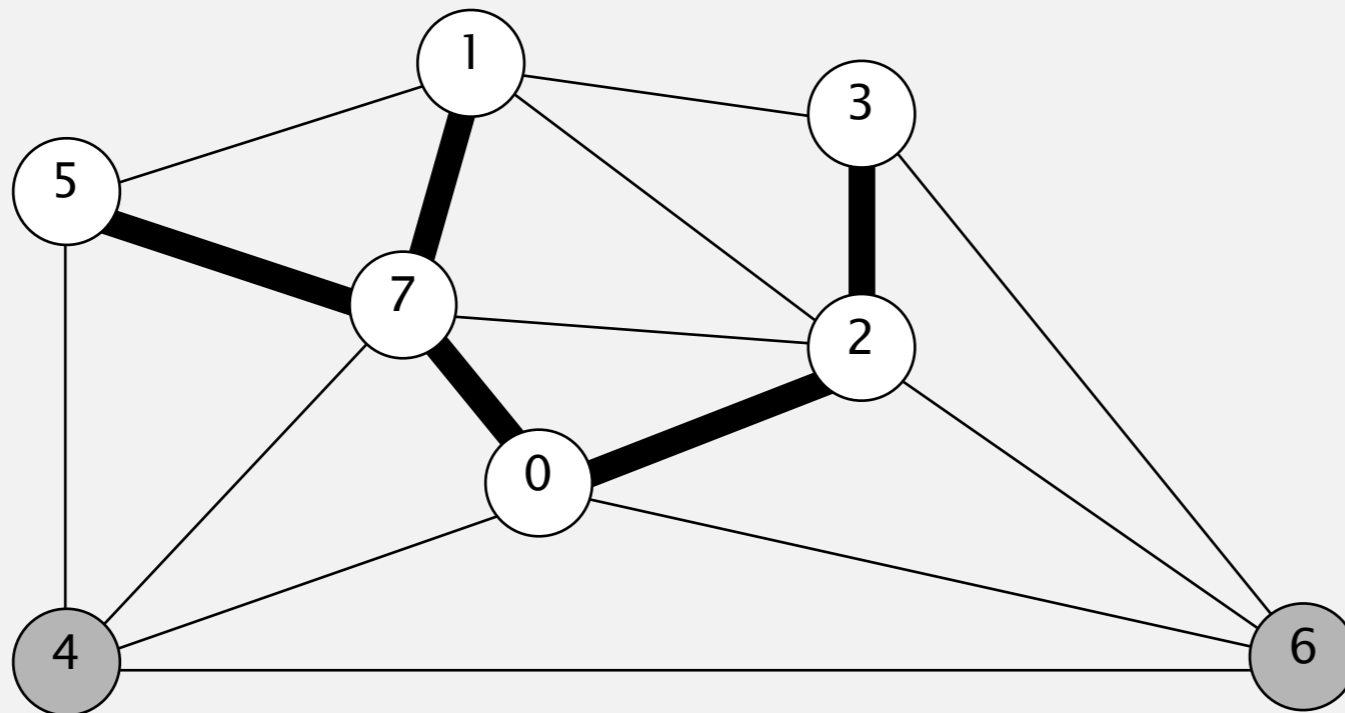
5-7	0.28
1-5	0.32
4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58

MST edges

0-7 1-7 0-2 2-3

Prim's algorithm demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



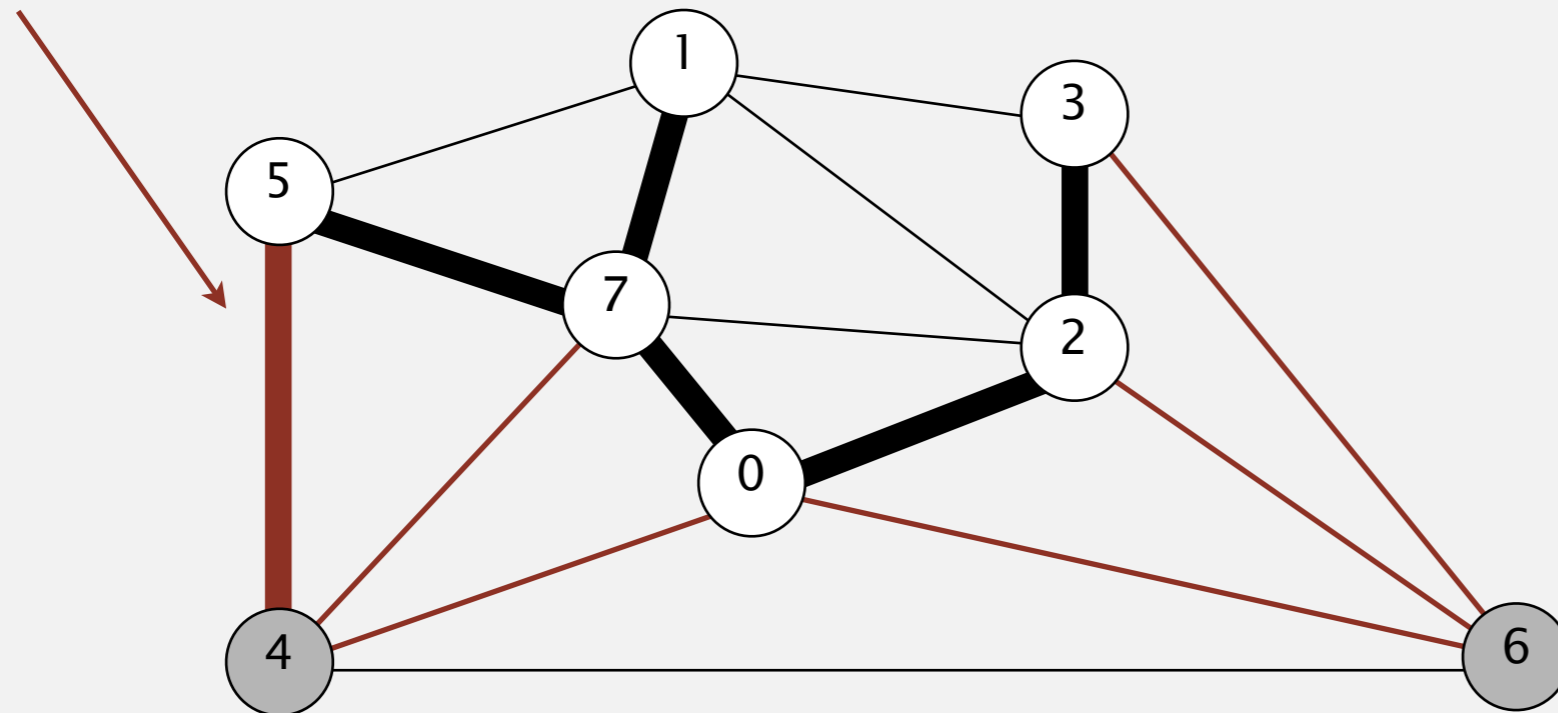
MST edges

0-7 1-7 0-2 2-3 5-7

Prim's algorithm demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

min-weight edge with exactly one endpoint in T



edges with exactly one endpoint in T (sorted by weight)

in MST →

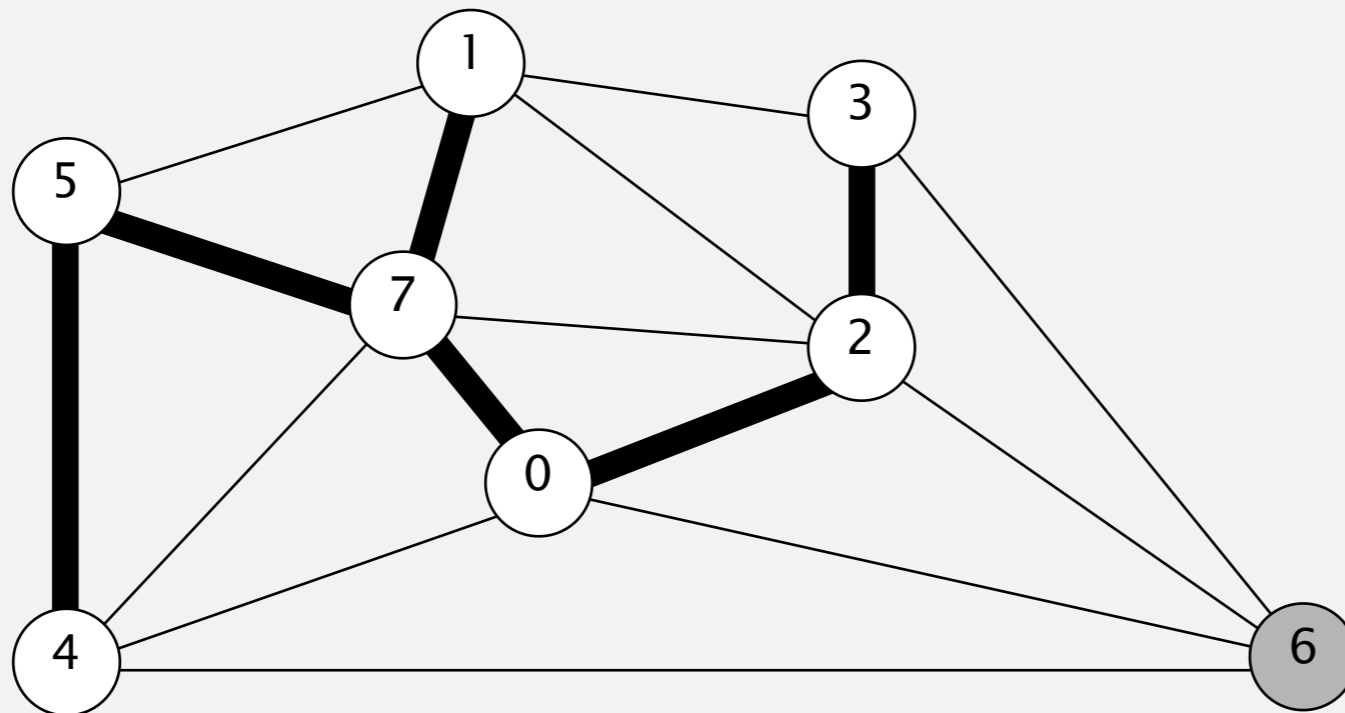
4-5	0.35
4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58

MST edges

0-7 1-7 0-2 2-3 5-7

Prim's algorithm demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

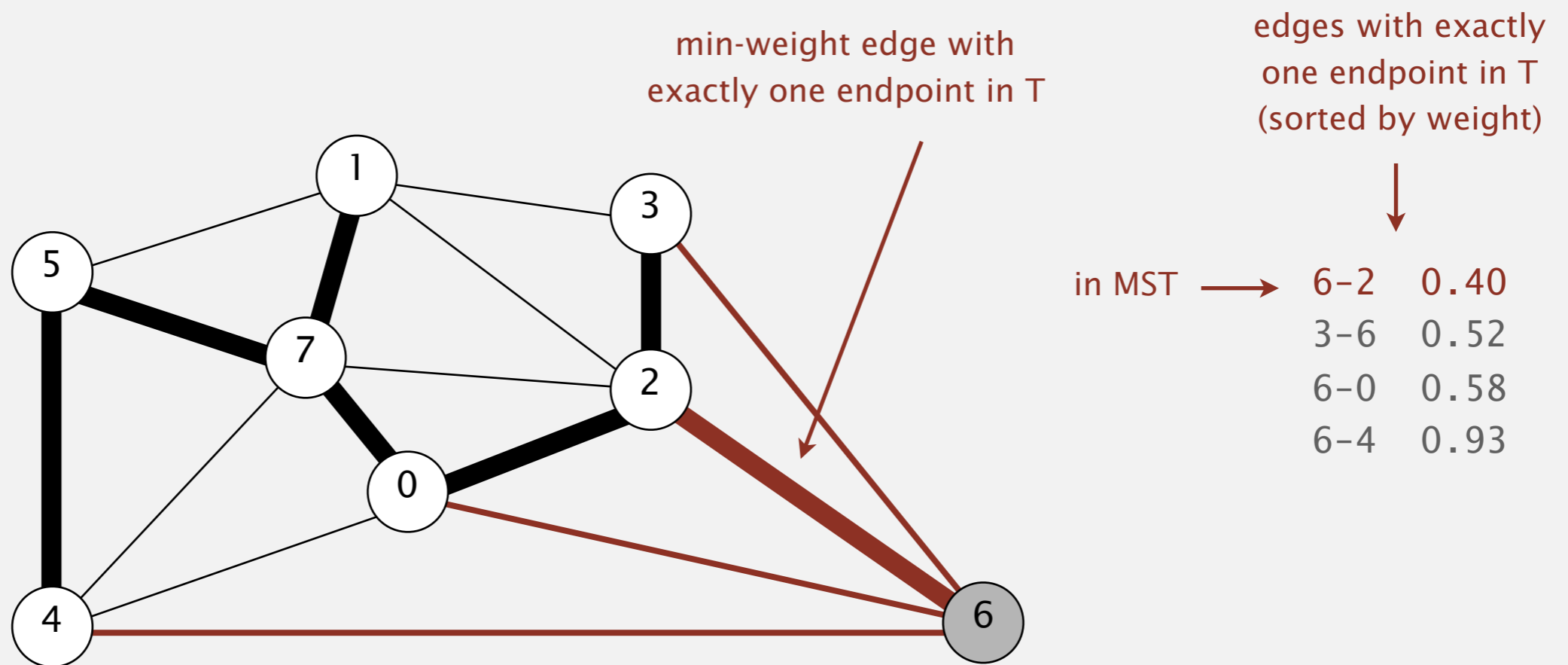


MST edges

0-7 1-7 0-2 2-3 5-7 4-5

Prim's algorithm demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

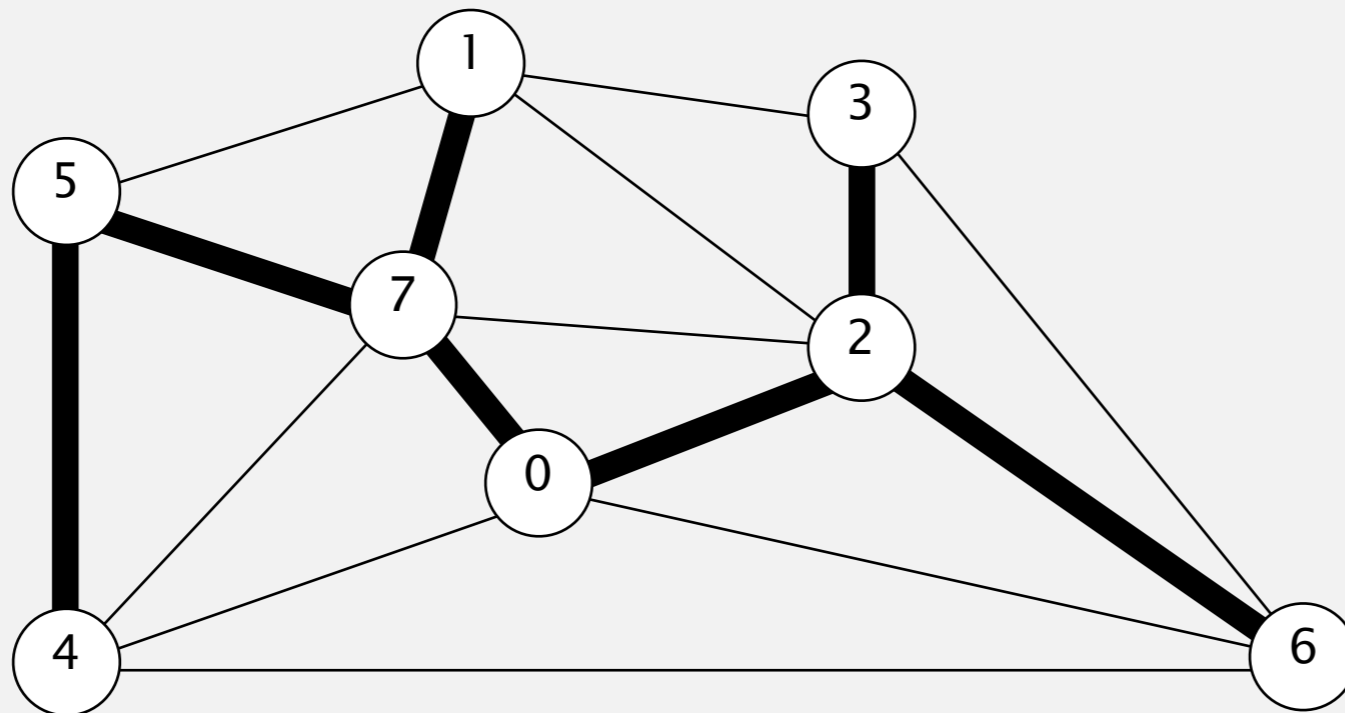


MST edges

0-7 1-7 0-2 2-3 5-7 4-5

Prim's algorithm demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



MST edges

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

PRIM'S ALGORITHM DEMO

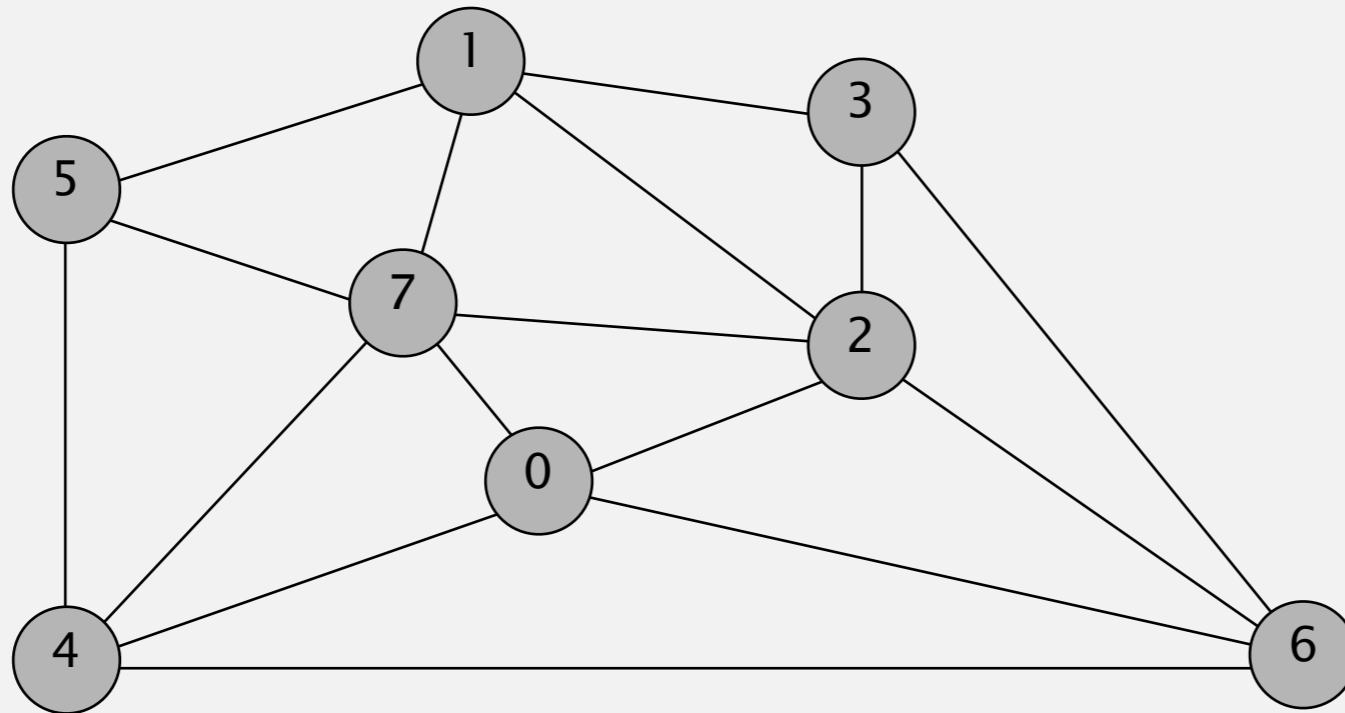
- ▶ *Prim's algorithm*
- ▶ *lazy implementation*
- ▶ *eager implementation*



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

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

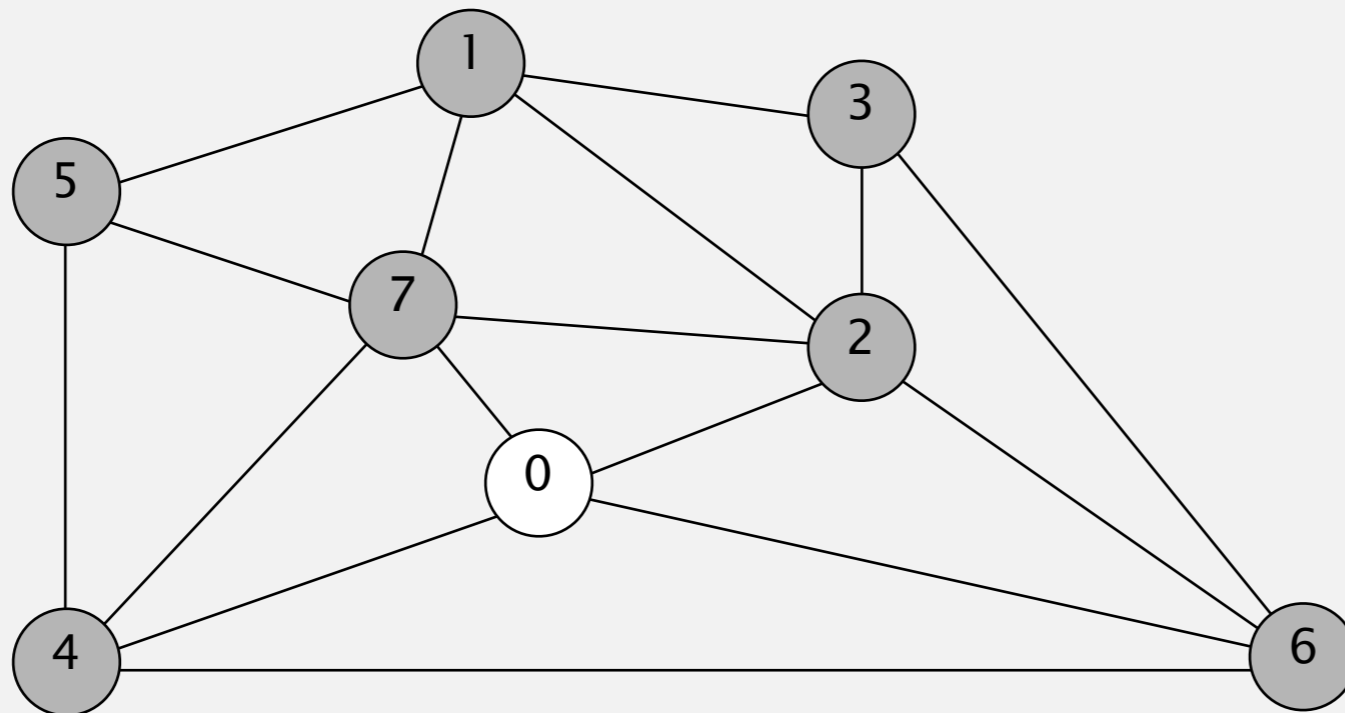


an edge-weighted graph

0-7	0.16
2-3	0.17
1-7	0.19
0-2	0.26
5-7	0.28
1-3	0.29
1-5	0.32
2-7	0.34
4-5	0.35
1-2	0.36
4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58
6-4	0.93

Prim's algorithm: lazy implementation demo

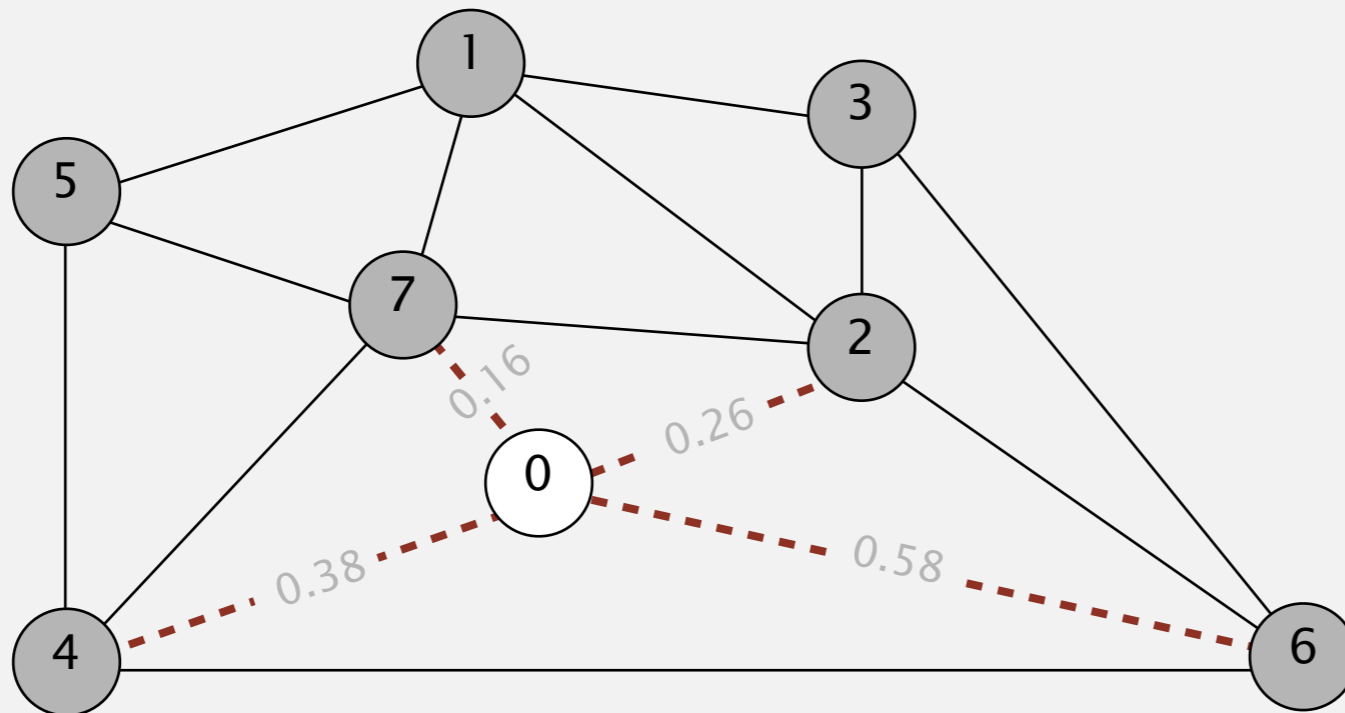
- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

add to PQ all edges incident to vertex 0



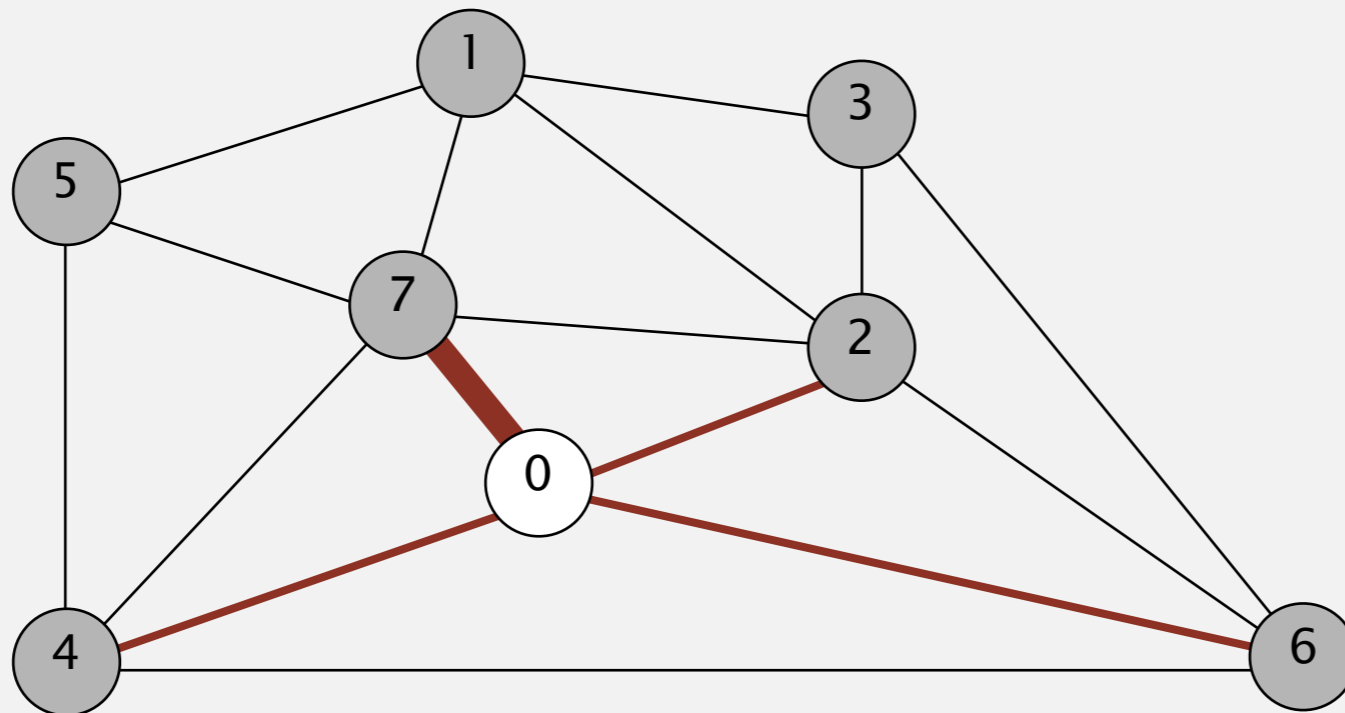
edges on PQ
(sorted by weight)

* 0-7	0.16
* 0-2	0.26
* 0-4	0.38
* 6-0	0.58

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

delete edge 0-7 from PQ and add to MST

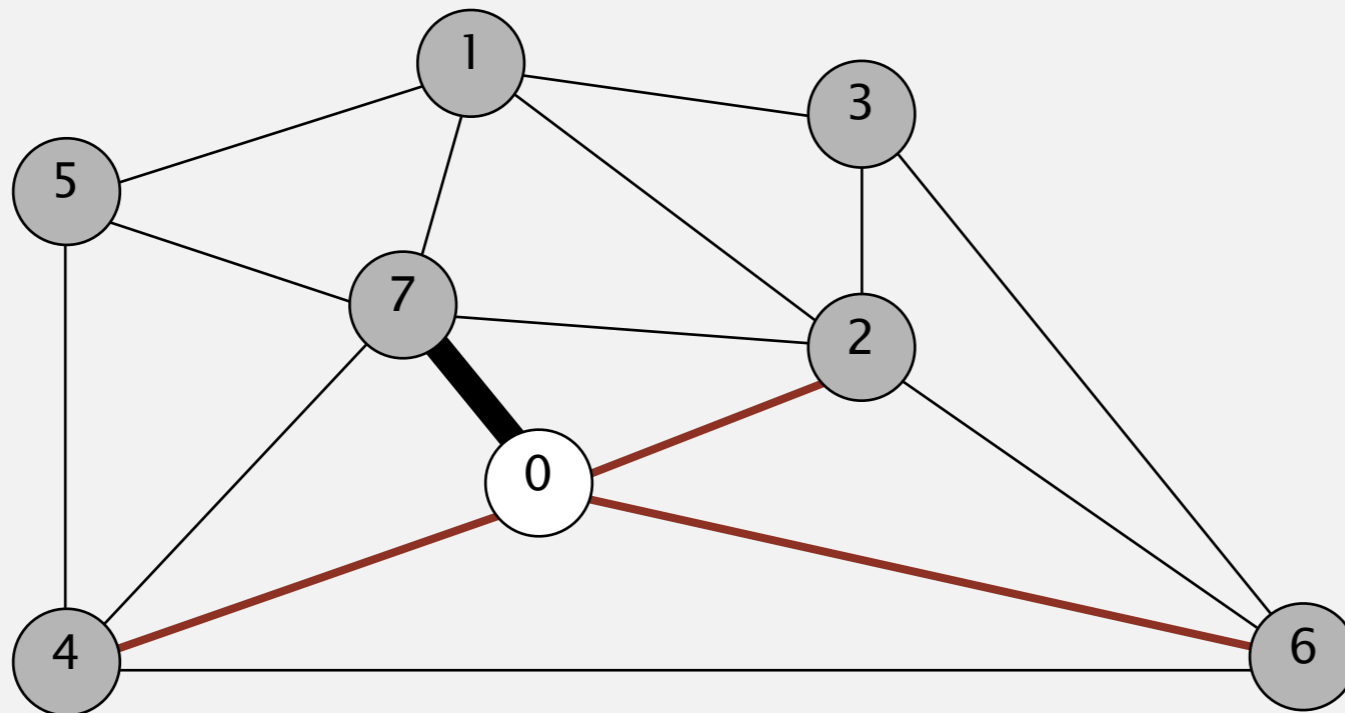


edges on PQ
(sorted by weight)

0-7	0.16
0-2	0.26
0-4	0.38
6-0	0.58

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



edges on PQ
(sorted by weight)

0-2	0.26
0-4	0.38
6-0	0.58

MST edges

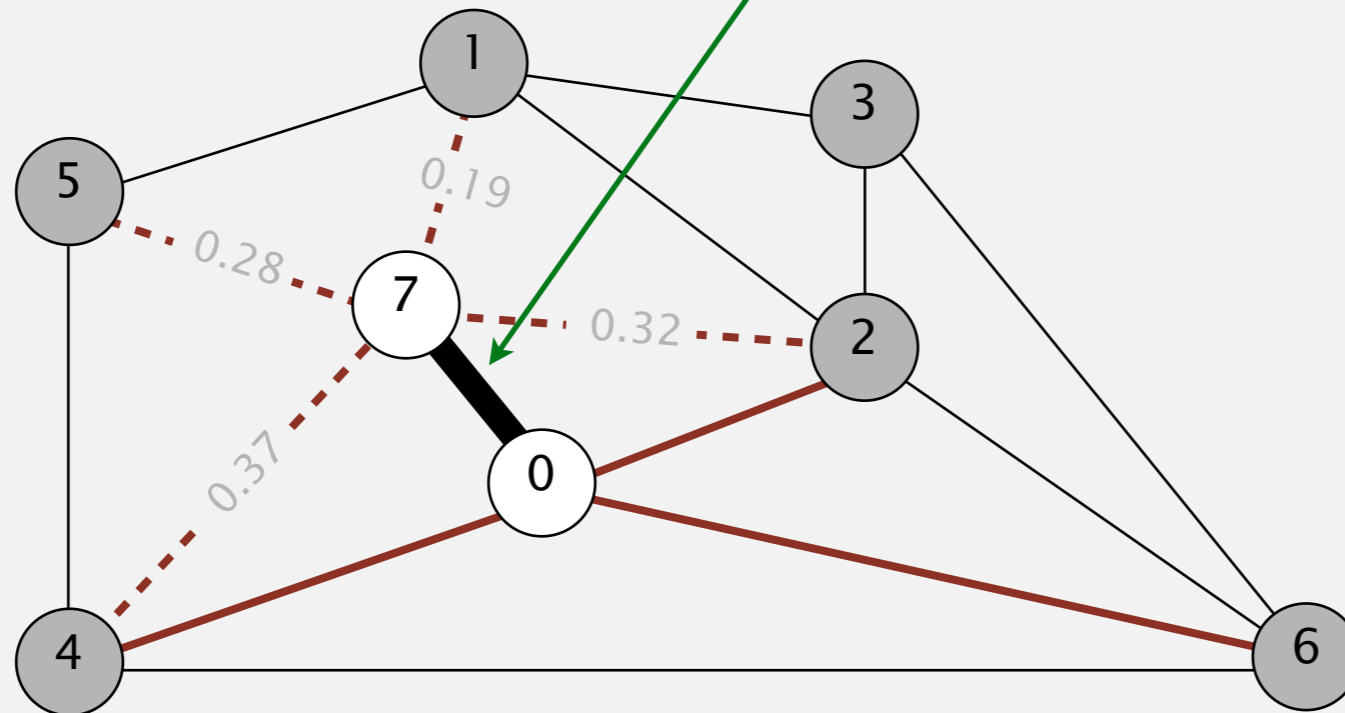
0-7

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

add to PQ all edges incident to vertex 7

no need to add 0-7
(because both endpoints are in T)



edges on PQ
(sorted by weight)

*	1-7	0.19
	0-2	0.26
*	5-7	0.28
*	2-7	0.34
*	4-7	0.37
	0-4	0.38
	6-0	0.58

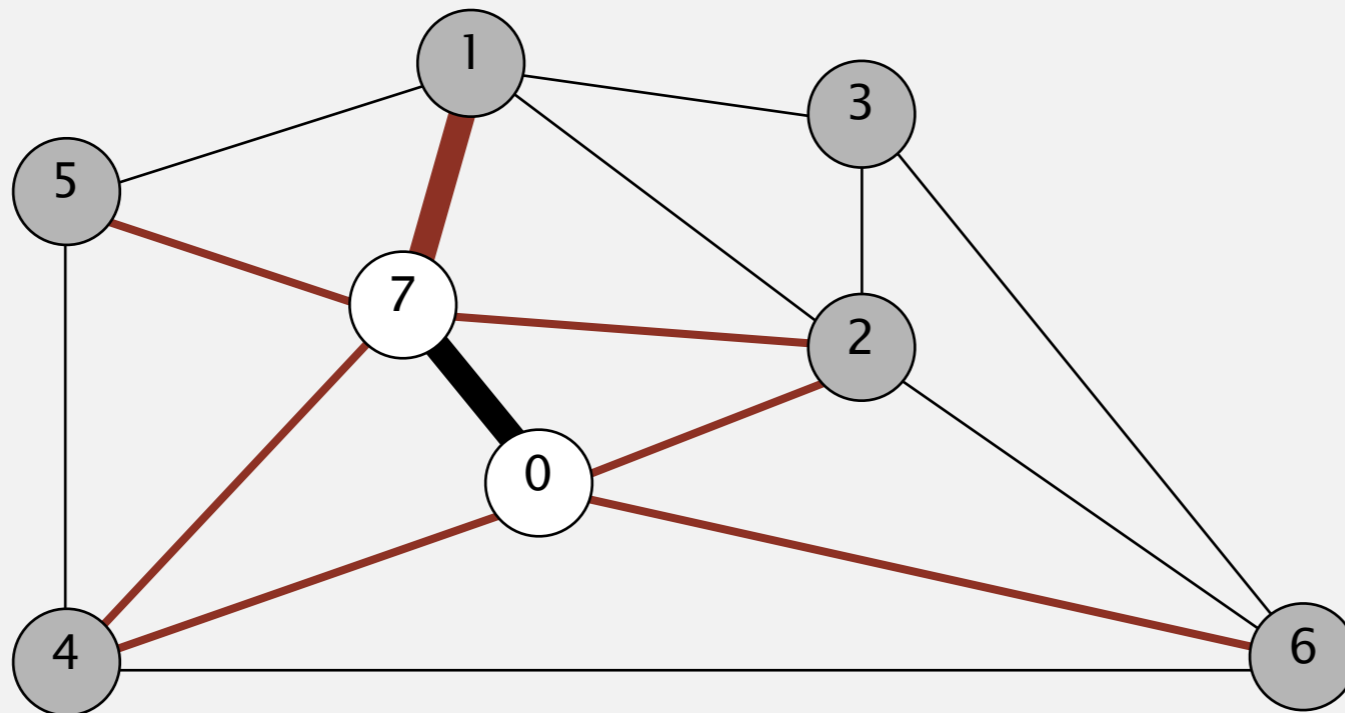
MST edges

0-7

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

delete edge 1-7 from PQ and add to MST



edges on PQ
(sorted by weight)

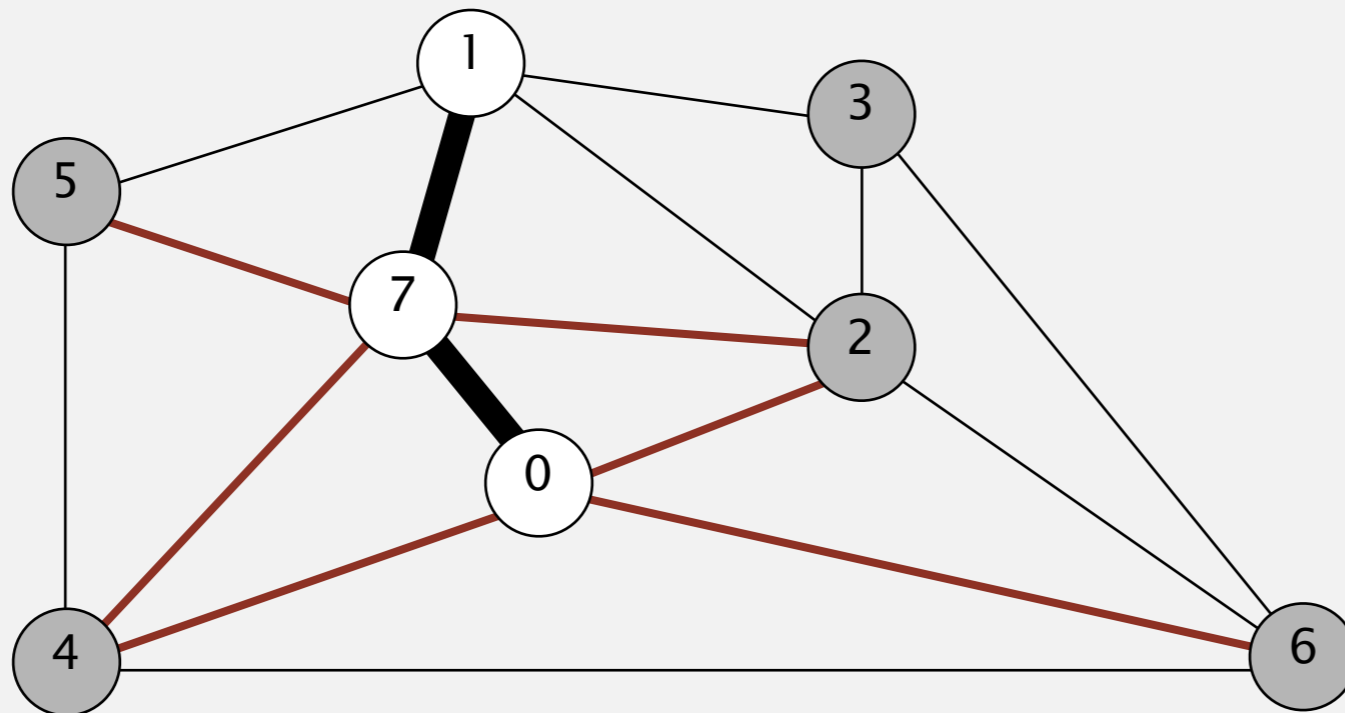
1-7	0.19
0-2	0.26
5-7	0.28
2-7	0.34
4-7	0.37
0-4	0.38
6-0	0.58

MST edges

0-7

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



edges on PQ
(sorted by weight)

0-2	0.26
5-7	0.28
2-7	0.34
4-7	0.37
0-4	0.38
6-0	0.58

MST edges

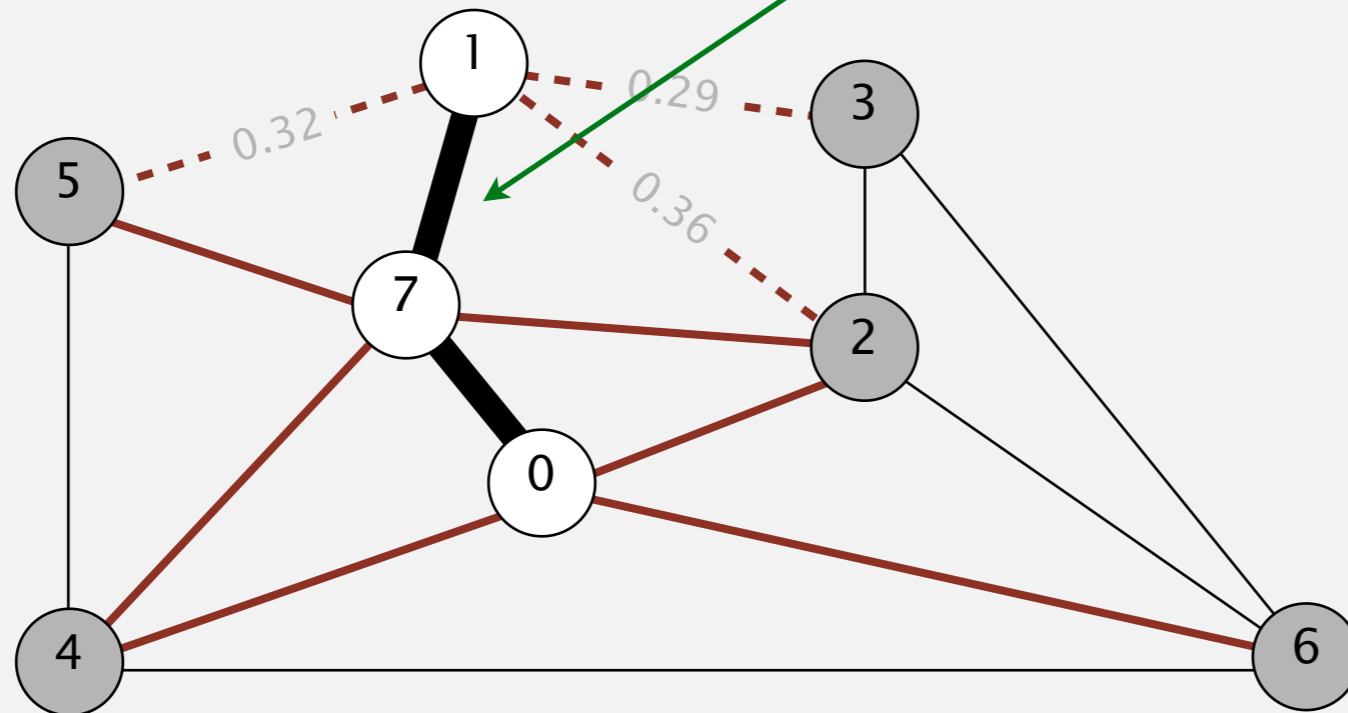
0-7 1-7

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

add to PQ all edges incident to vertex 1

no need to add 1-7
(because both endpoints are in T)



edges on PQ
(sorted by weight)

0-2	0.26
5-7	0.28
* 1-3	0.29
* 1-5	0.32
2-7	0.34
* 1-2	0.36
4-7	0.37
0-4	0.38
6-0	0.58

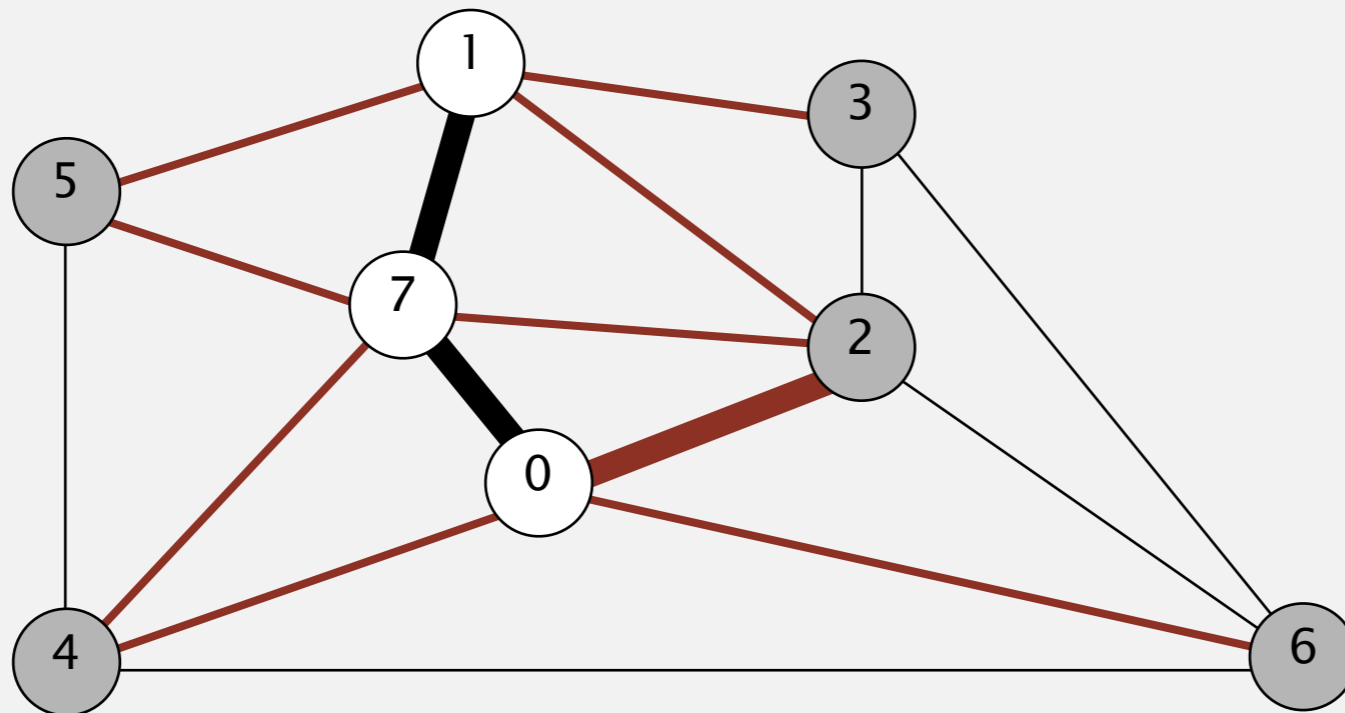
MST edges

0-7 1-7

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

delete edge 0-2 from PQ and add to MST



edges on PQ
(sorted by weight)

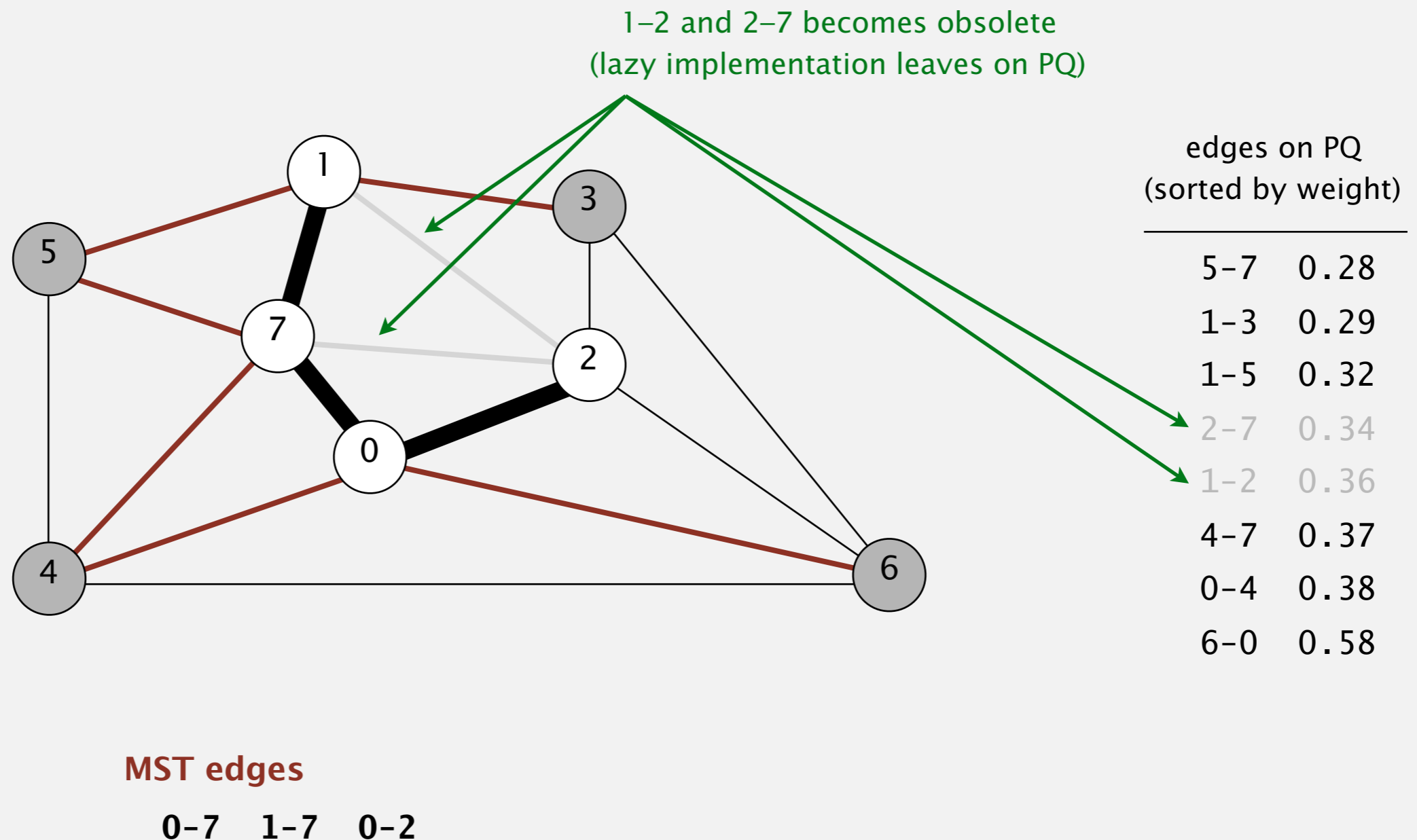
0-2	0.26
5-7	0.28
1-3	0.29
1-5	0.32
2-7	0.34
1-2	0.36
4-7	0.37
0-4	0.38
6-0	0.58

MST edges

0-7 1-7

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

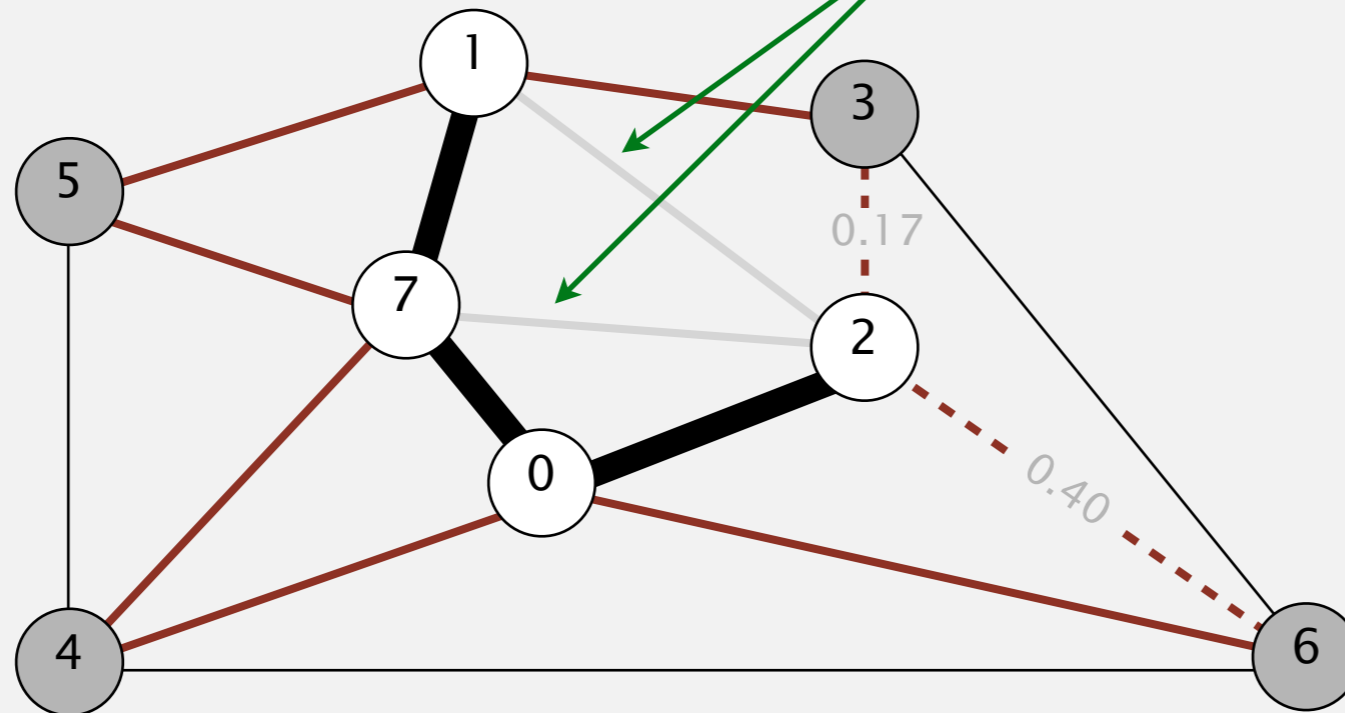


Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

add to PQ edges all incident to vertex 2

no need to add 0-2, 1-2, or 2-7
(because both endpoints are in T)



edges on PQ
(sorted by weight)

*	2-3	0.17
	5-7	0.28
	1-3	0.29
	1-5	0.32
	2-7	0.34
	1-2	0.36
	4-7	0.37
	0-4	0.38
*	6-2	0.40
	6-0	0.58

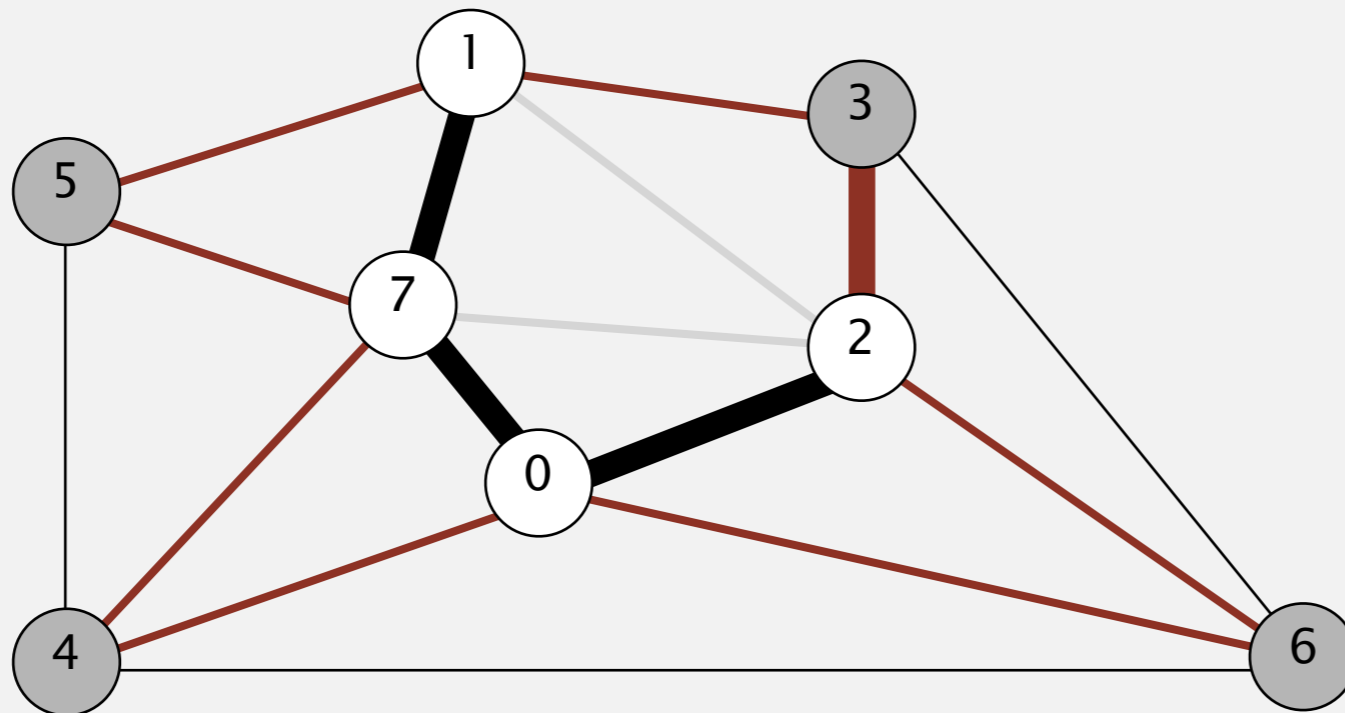
MST edges

0-7 1-7 0-2

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

delete edge 2-3 from PQ and add to MST



edges on PQ
(sorted by weight)

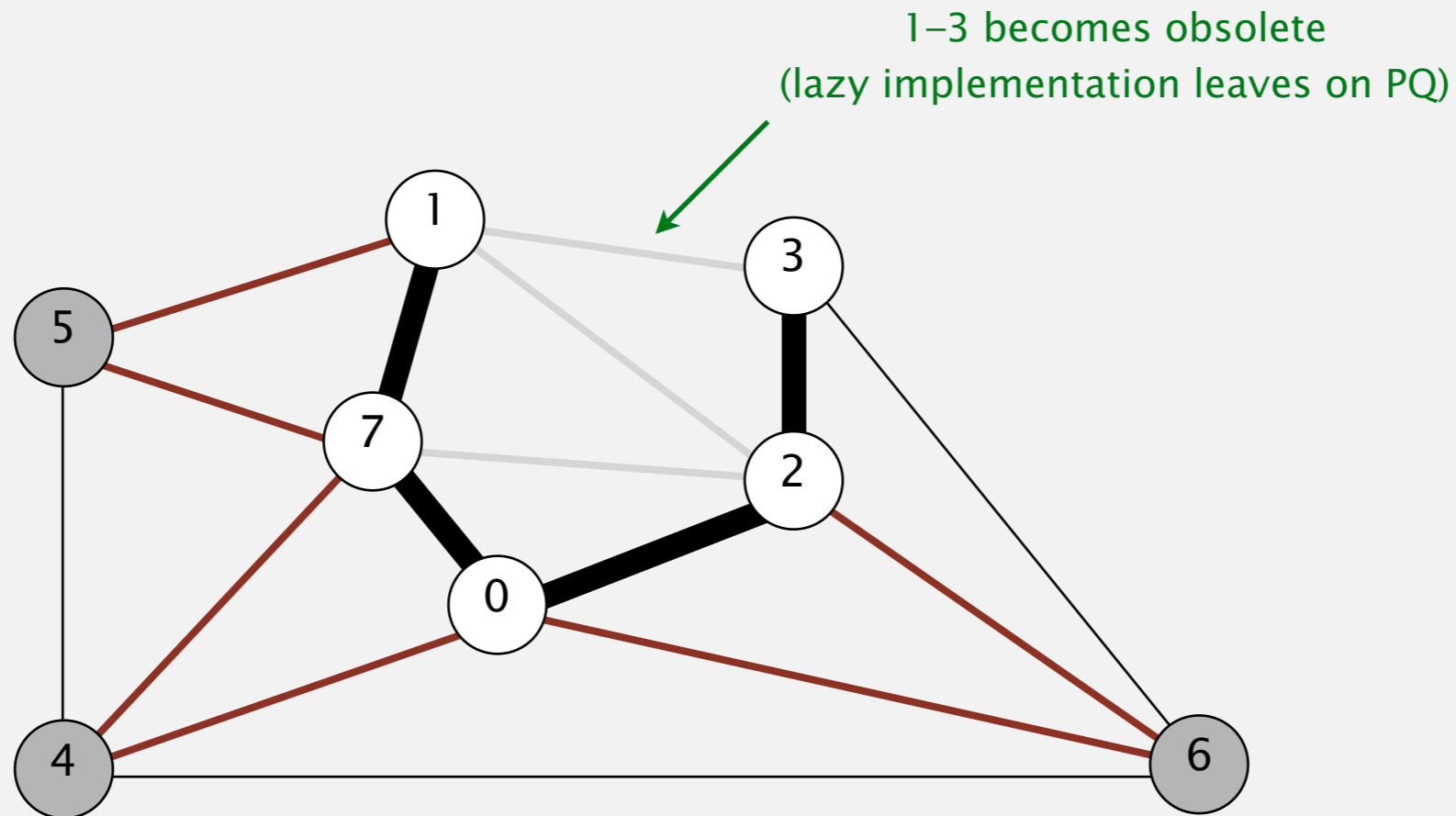
*	2-3	0.17
	5-7	0.28
	1-3	0.29
	1-5	0.32
	2-7	0.34
	1-2	0.36
	4-7	0.37
	0-4	0.38
*	6-2	0.40
	6-0	0.58

MST edges

0-7 1-7 0-2

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



edges on PQ
(sorted by weight)

5-7	0.28
1-3	0.29
1-5	0.32
2-7	0.34
1-2	0.36
4-7	0.37
0-4	0.38
6-2	0.40
6-0	0.58

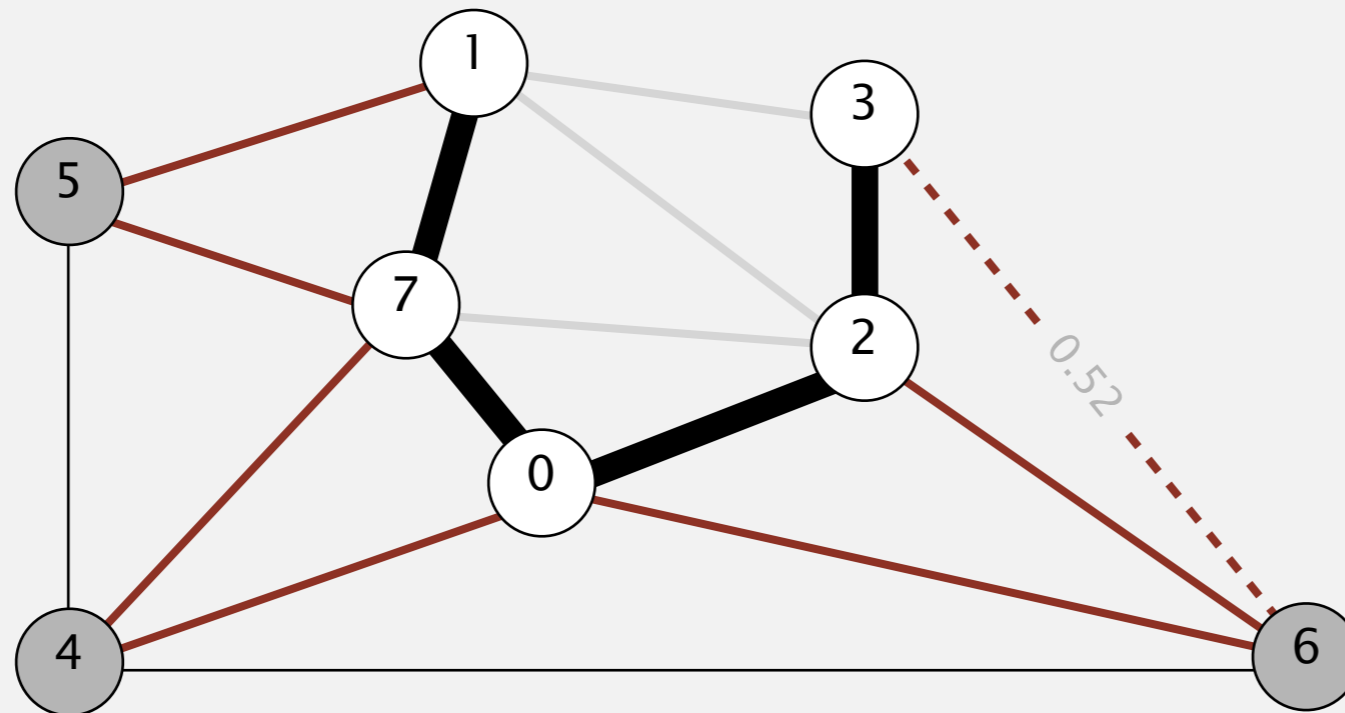
MST edges

0-7 1-7 0-2 2-3

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

add to PQ edges all incident to vertex 3



MST edges

0-7 1-7 0-2 2-3

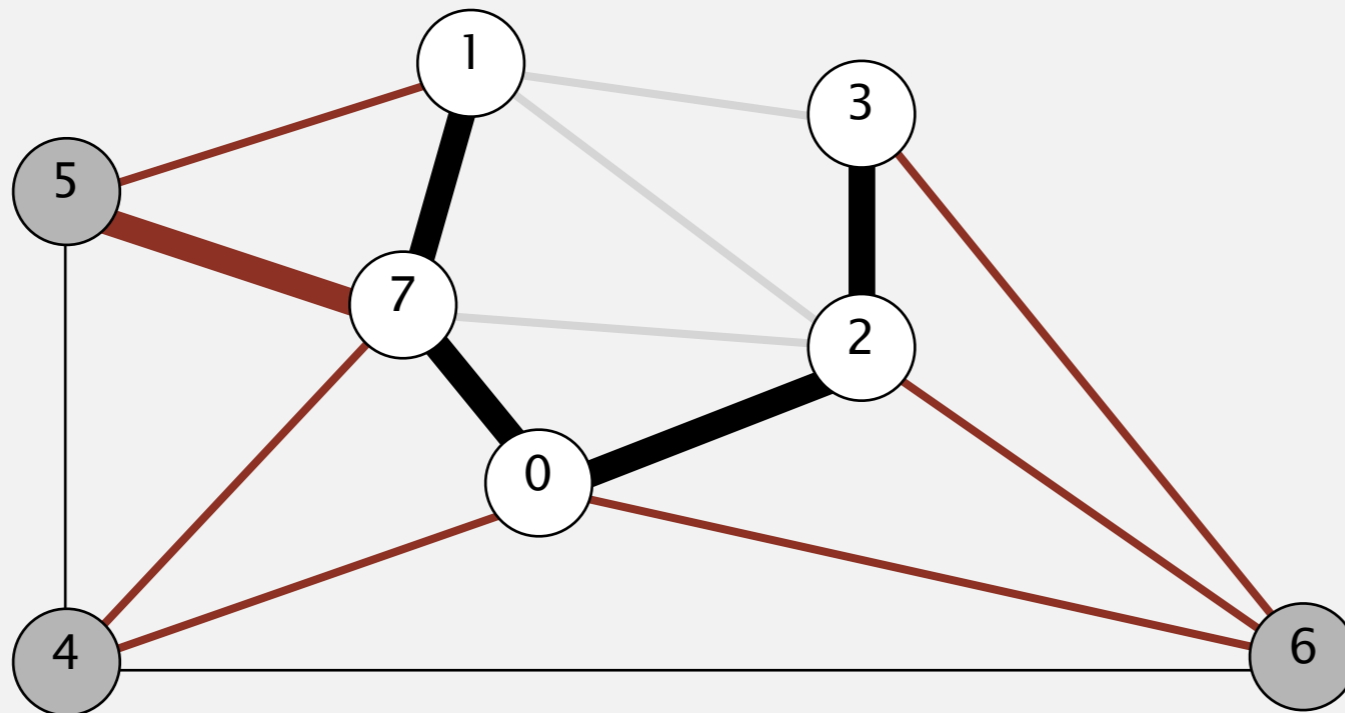
edges on PQ
(sorted by weight)

5-7	0.28
1-3	0.29
1-5	0.32
2-7	0.34
1-2	0.36
4-7	0.37
0-4	0.38
6-2	0.40
* 3-6	0.52
6-0	0.58

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

delete edge 5-7 from PQ and add to MST



edges on PQ
(sorted by weight)

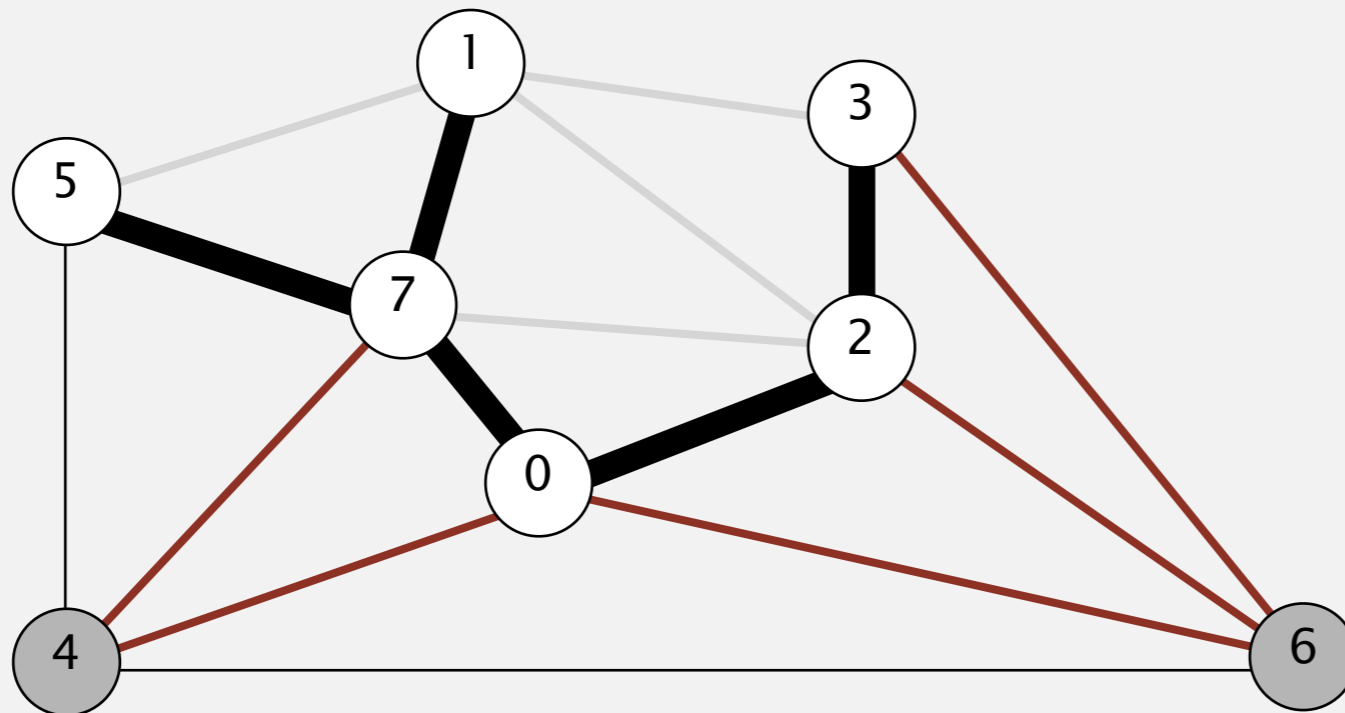
5-7	0.28
1-3	0.29
1-5	0.32
2-7	0.34
1-2	0.36
4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58

MST edges

0-7 1-7 0-2 2-3

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



edges on PQ
(sorted by weight)

1-3	0.29
1-5	0.32
2-7	0.34
1-2	0.36
4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58

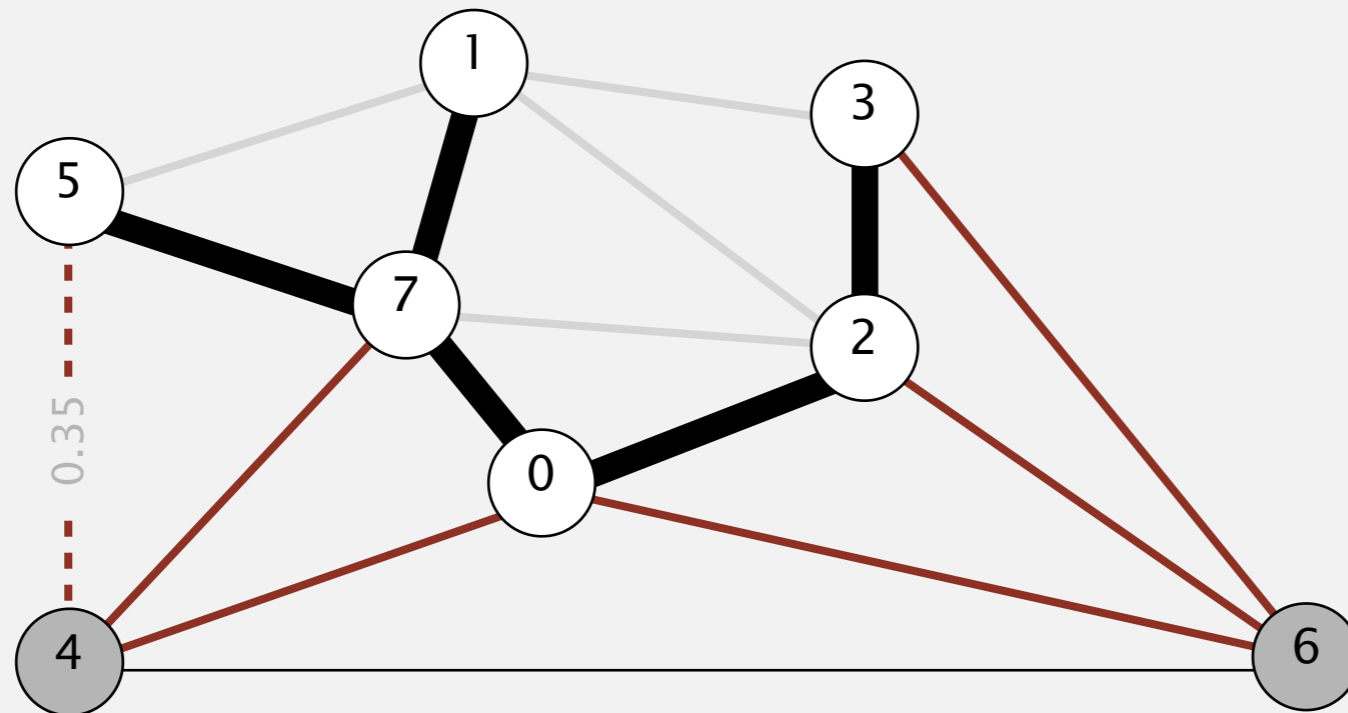
MST edges

0-7 1-7 0-2 2-3 5-7

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

add to PQ all edges incident to vertex 5



MST edges

0-7 1-7 0-2 2-3 5-7

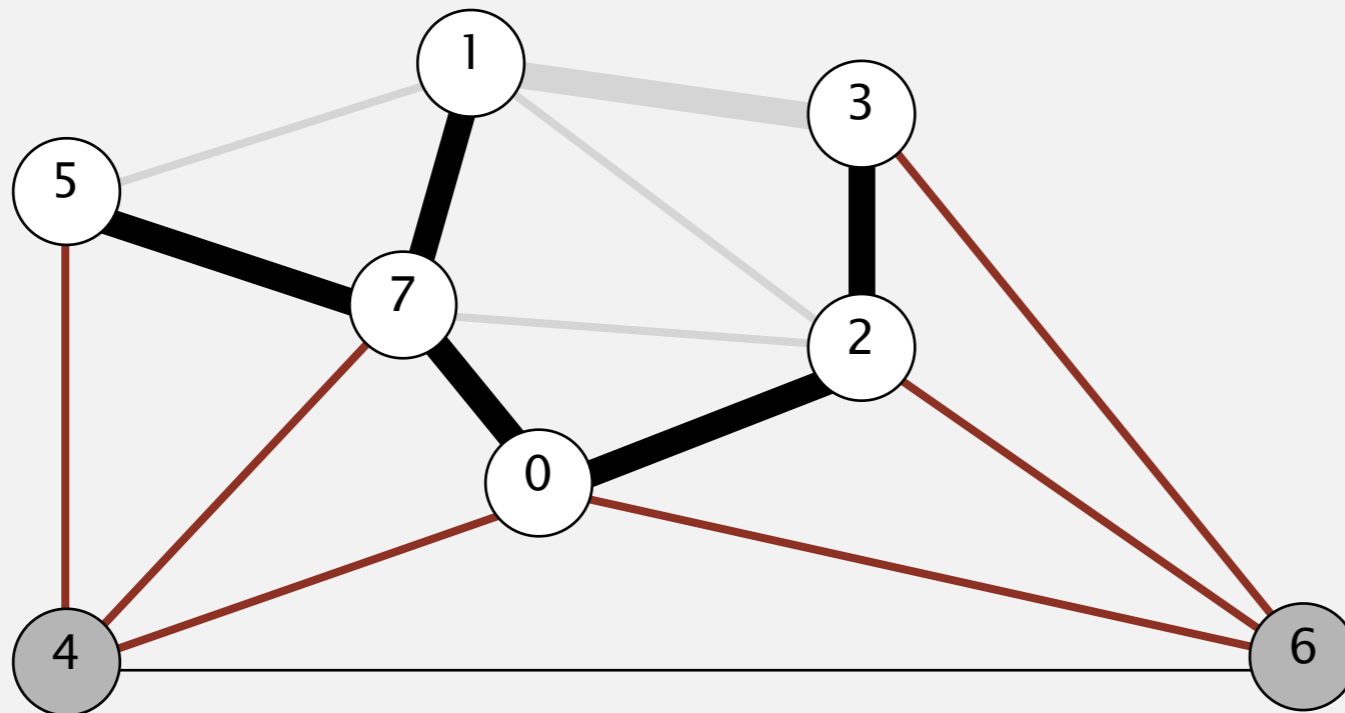
edges on PQ
(sorted by weight)

1-3	0.29
1-5	0.32
2-7	0.34
* 4-5	0.35
1-2	0.36
4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

delete edge 1-3 from PQ (obsolete edge)



edges on PQ
(sorted by weight)

1-3	0.29
1-5	0.32
2-7	0.34
4-5	0.35
1-2	0.36
4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58

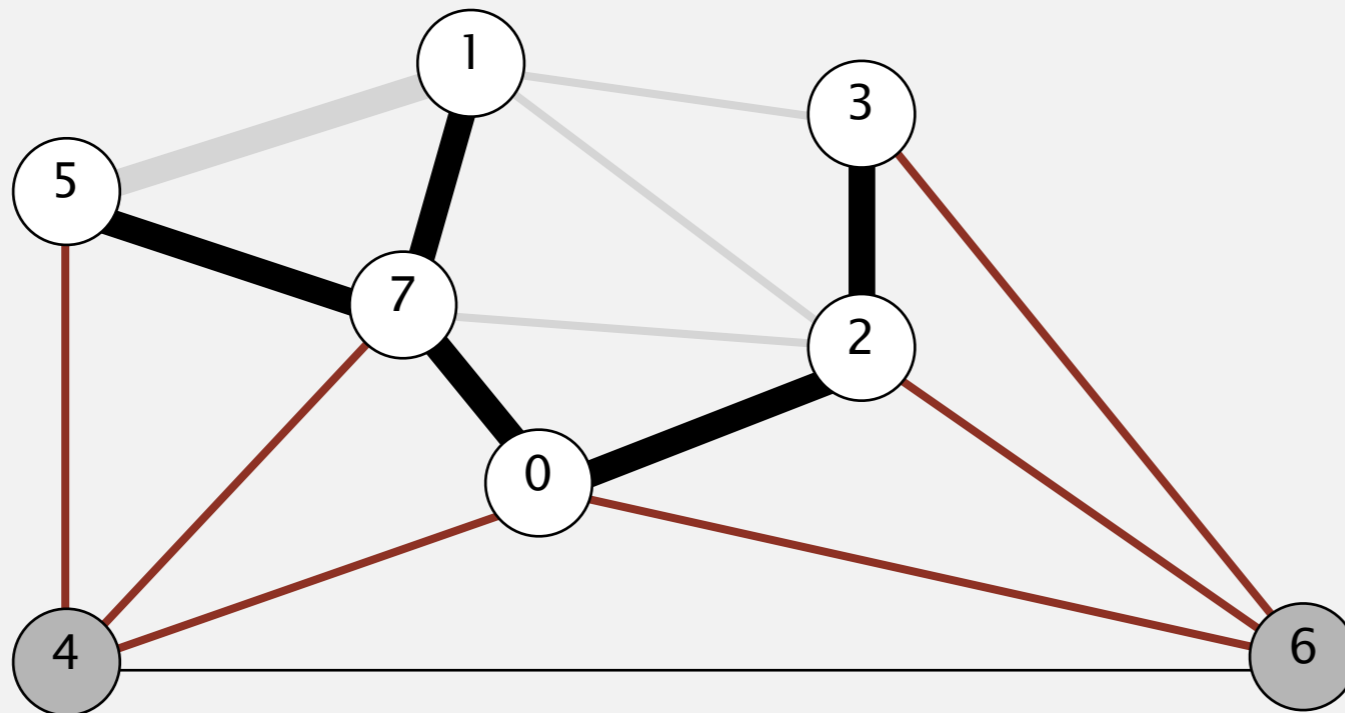
MST edges

0-7 1-7 0-2 2-3 5-7

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

delete edge 1-5 from PQ (obsolete edge)



edges on PQ
(sorted by weight)

1-5	0.32
2-7	0.34
4-5	0.35
1-2	0.36
4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58

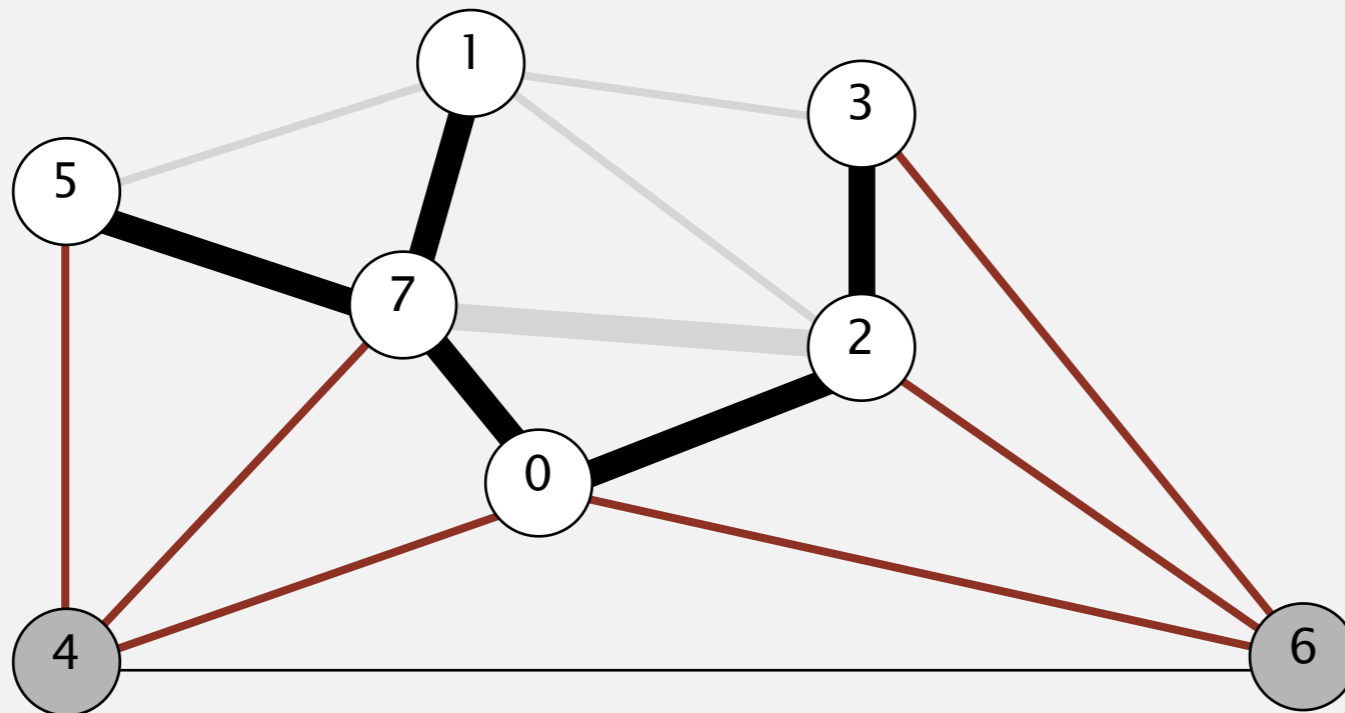
MST edges

0-7 1-7 0-2 2-3 5-7

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

delete edge 2-7 from PQ (obsolete edge)



edges on PQ
(sorted by weight)

2-7	0.34
4-5	0.35
1-2	0.36
4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58

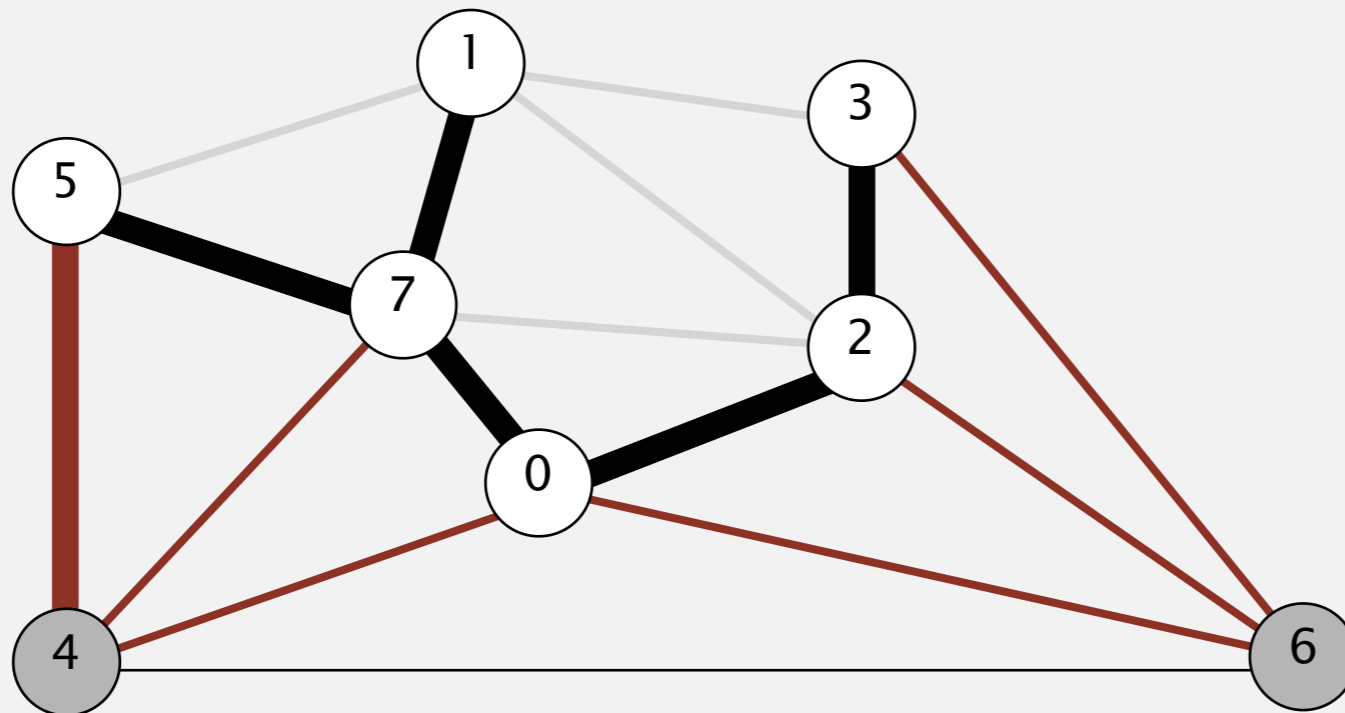
MST edges

0-7 1-7 0-2 2-3 5-7

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

delete edge 4-5 from PQ and add to MST



edges on PQ
(sorted by weight)

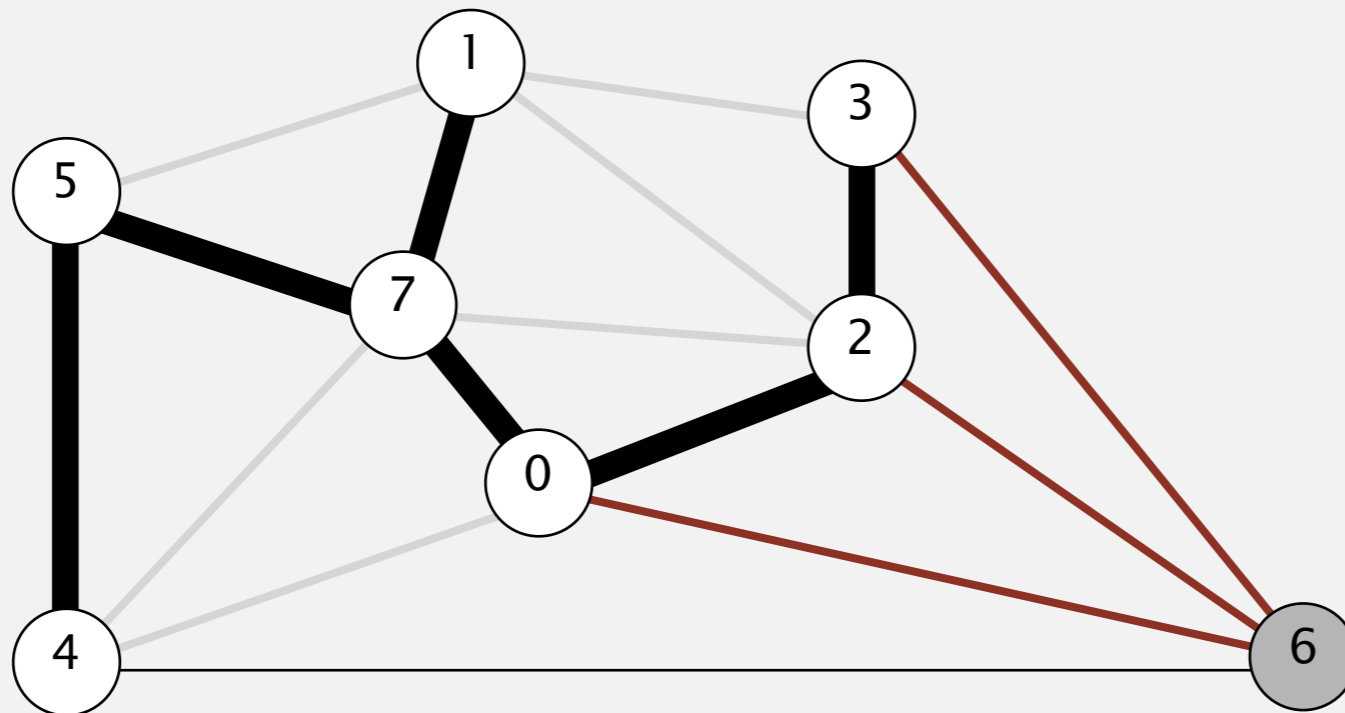
4-5	0.35
1-2	0.36
4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58

MST edges

0-7 1-7 0-2 2-3 5-7

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



edges on PQ
(sorted by weight)

1-2	0.36
4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58

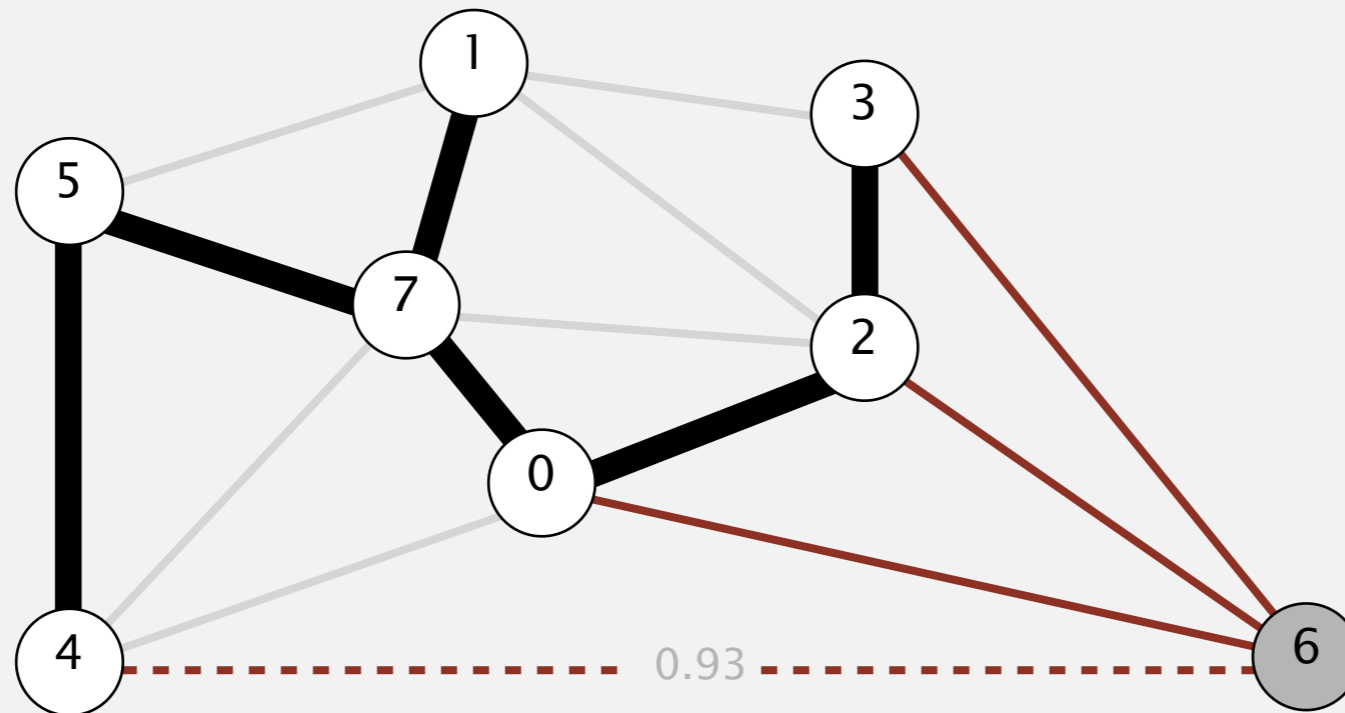
MST edges

0-7 1-7 0-2 2-3 5-7 4-5

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

add to PQ all edges incident to vertex 4



edges on PQ
(sorted by weight)

1-2	0.36
4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58
* 6-4	0.93

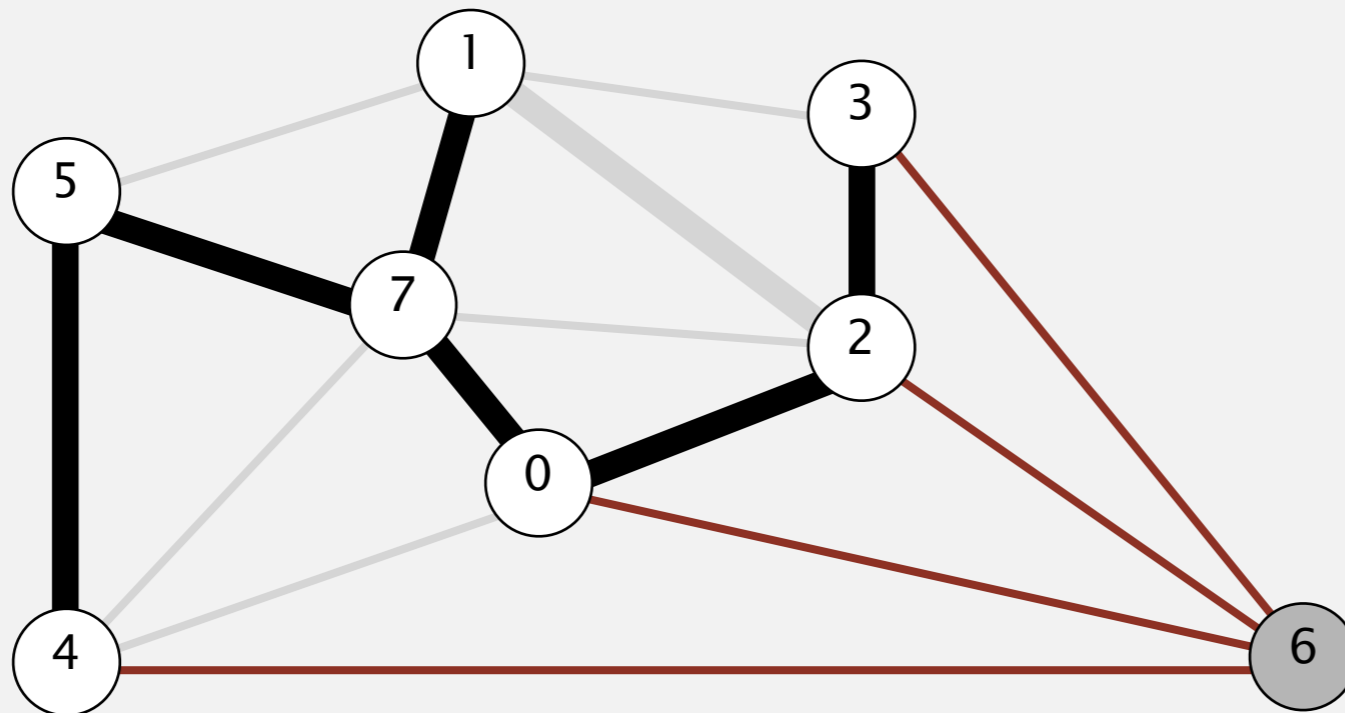
MST edges

0-7 1-7 0-2 2-3 5-7 4-5

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

delete edge 1-2 from PQ (obsolete edge)



edges on PQ
(sorted by weight)

1-2	0.36
4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58
6-4	0.93

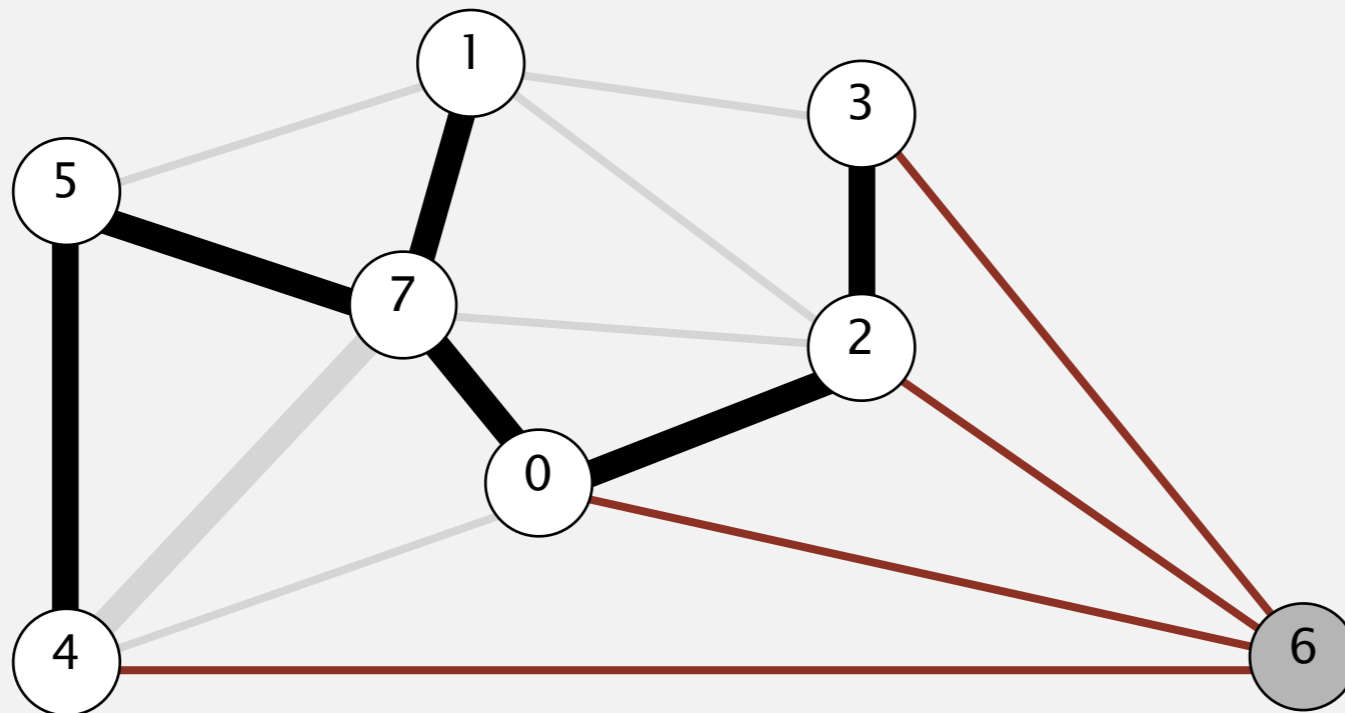
MST edges

0-7 1-7 0-2 2-3 5-7 4-5

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

delete edge 4-7 from PQ (obsolete edge)



edges on PQ
(sorted by weight)

4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58
6-4	0.93

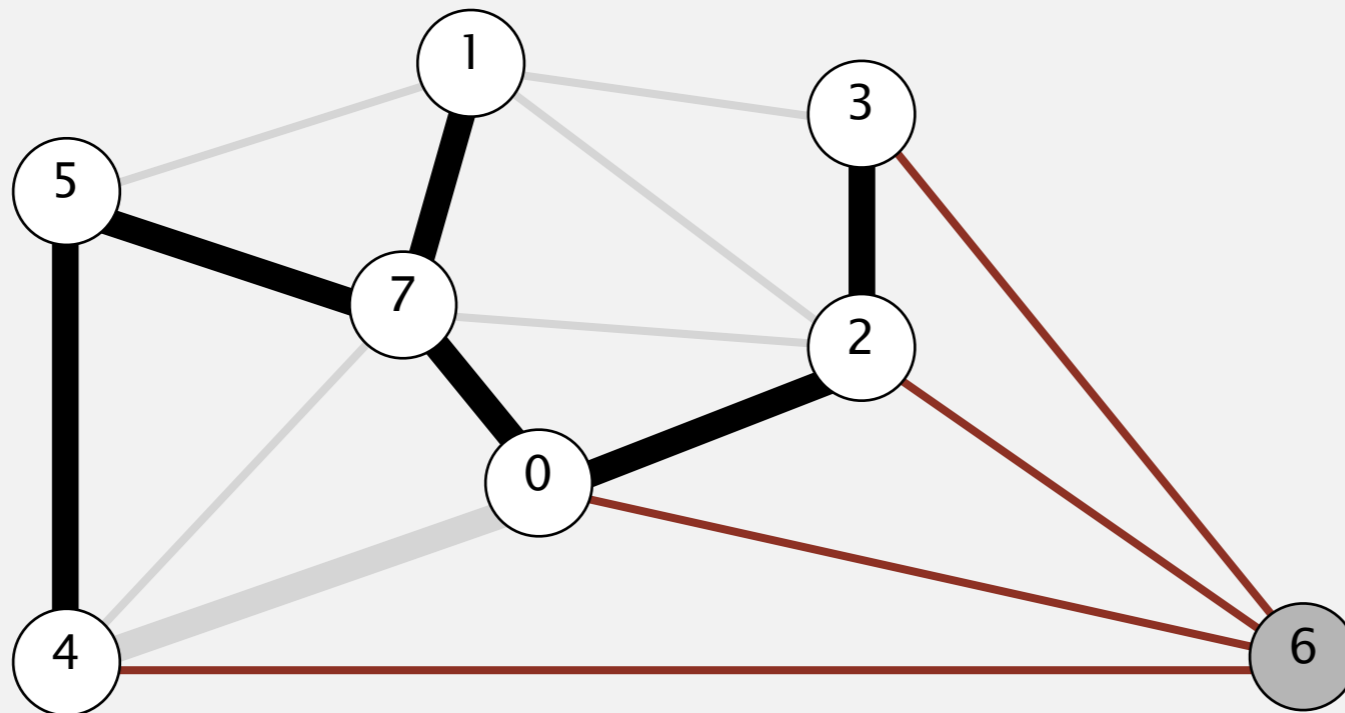
MST edges

0-7 1-7 0-2 2-3 5-7 4-5

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

delete edge 0-4 from PQ (obsolete edge)



edges on PQ
(sorted by weight)

0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58
6-4	0.93

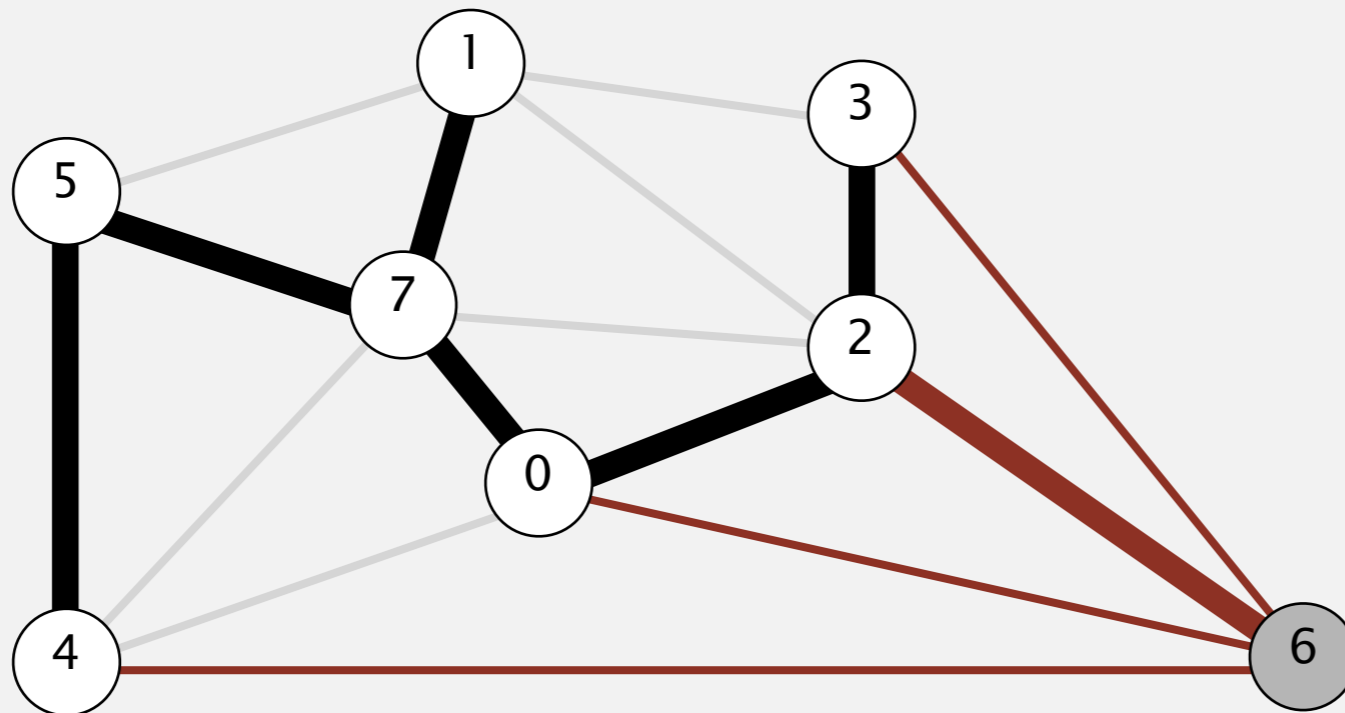
MST edges

0-7 1-7 0-2 2-3 5-7 4-5

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

delete edge 6-2 from PQ and add to MST



edges on PQ
(sorted by weight)

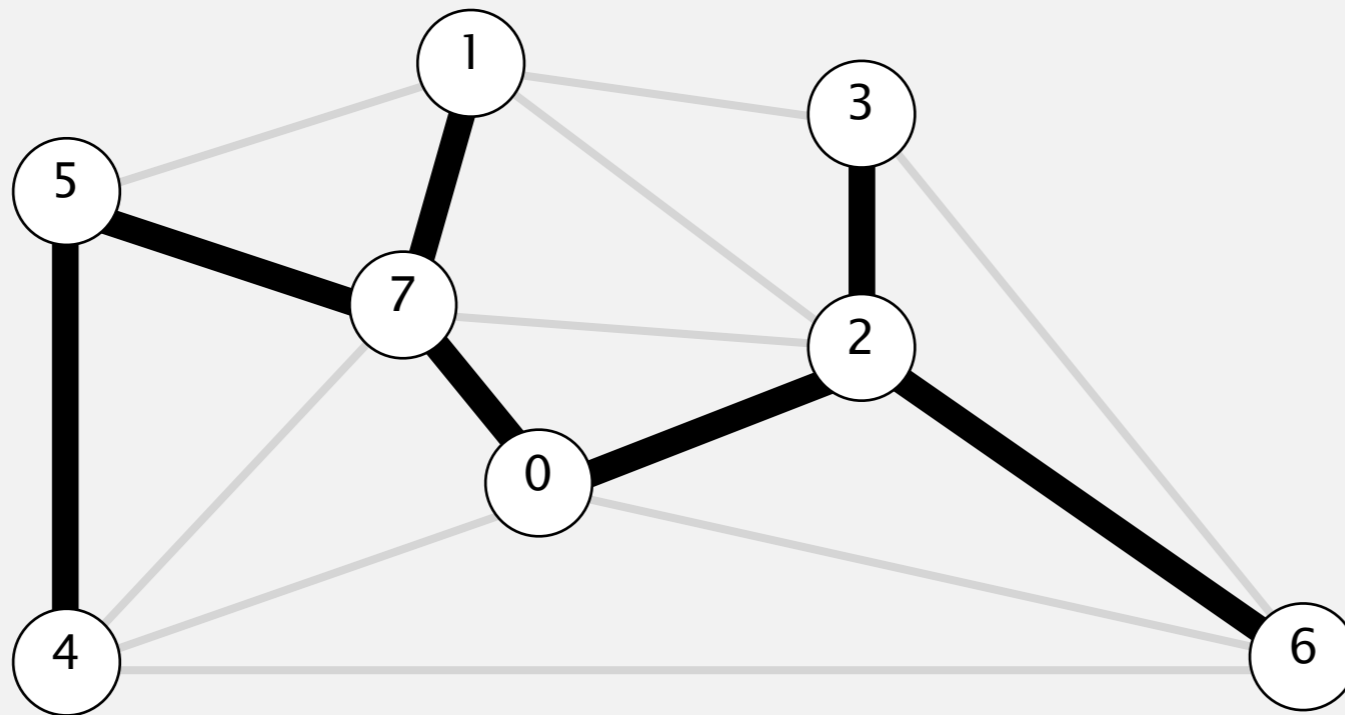
6-2	0.40
3-6	0.52
6-0	0.58
6-4	0.93

MST edges

0-7 1-7 0-2 2-3 5-7 4-5

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



edges on PQ
(sorted by weight)

3-6	0.52
6-0	0.58
6-4	0.93

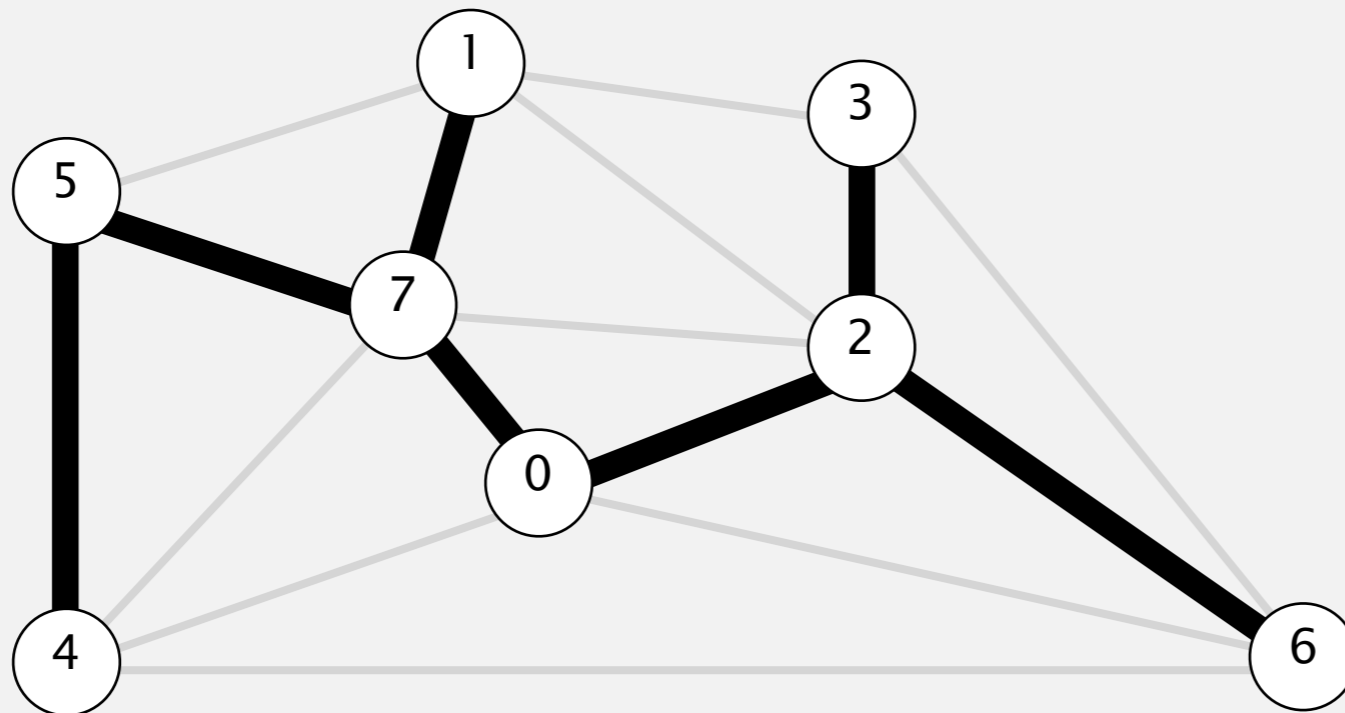
MST edges

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

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

stop since $V-1$ edges



edges on PQ
(sorted by weight)

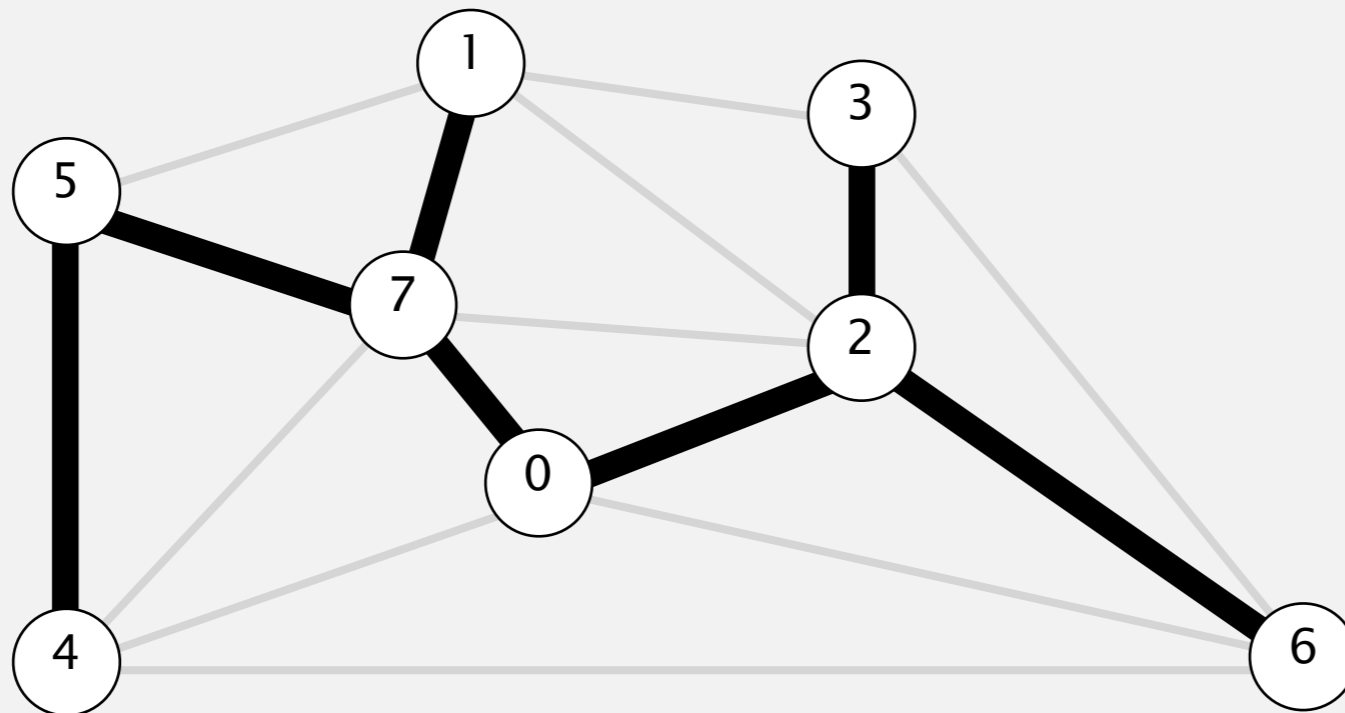
3-6	0.52
6-0	0.58
6-4	0.93

MST edges

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

Prim's algorithm: lazy implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



MST edges

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



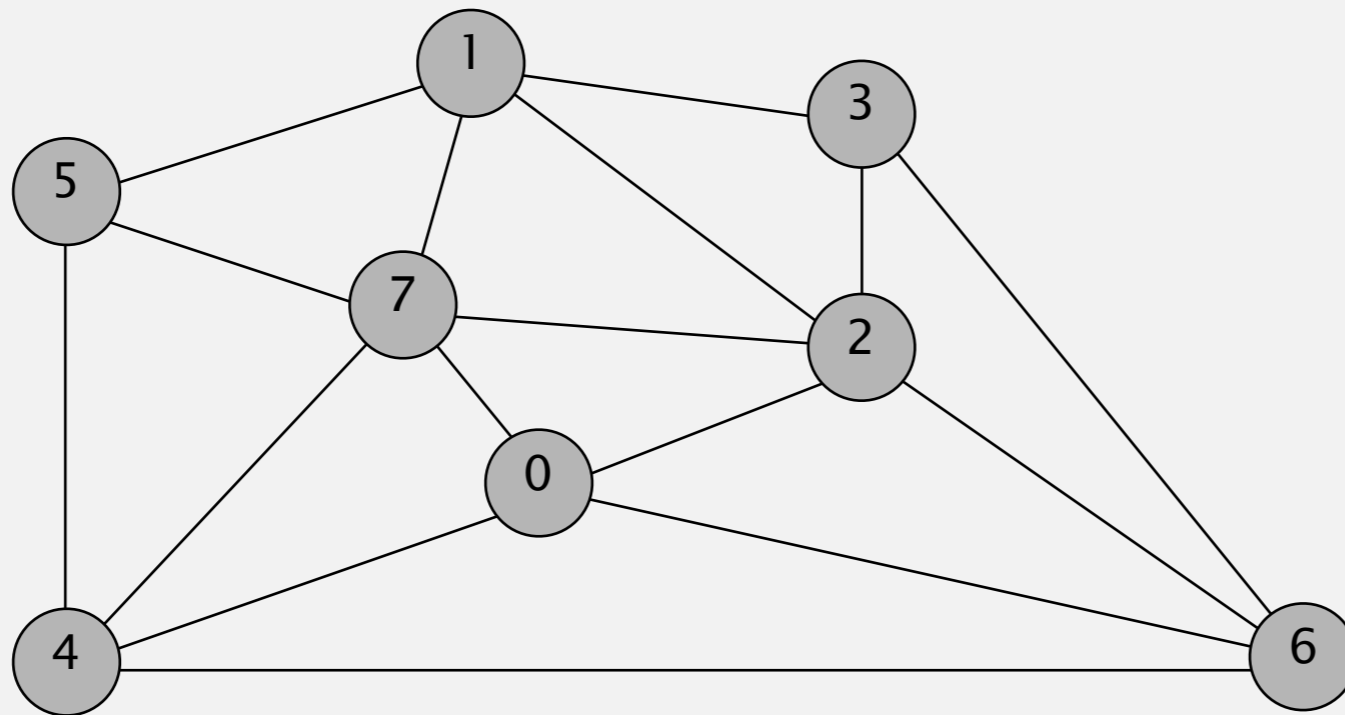
<https://algs4.cs.princeton.edu>

PRIM'S ALGORITHM DEMO

- ▶ *Prim's algorithm*
- ▶ *lazy implementation*
- ▶ *eager implementation*

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

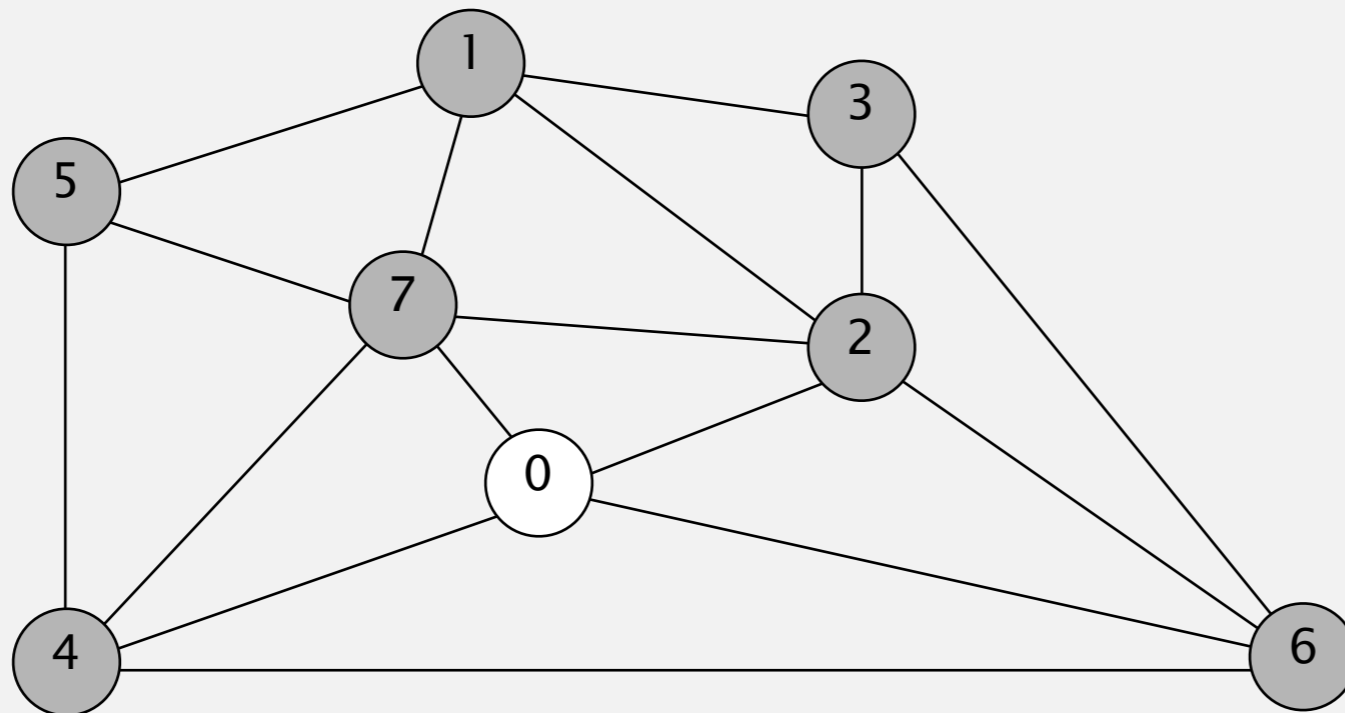


an edge-weighted graph

0-7	0.16
2-3	0.17
1-7	0.19
0-2	0.26
5-7	0.28
1-3	0.29
1-5	0.32
2-7	0.34
4-5	0.35
1-2	0.36
4-7	0.37
0-4	0.38
6-2	0.40
3-6	0.52
6-0	0.58
6-4	0.93

Prim's algorithm: eager implementation demo

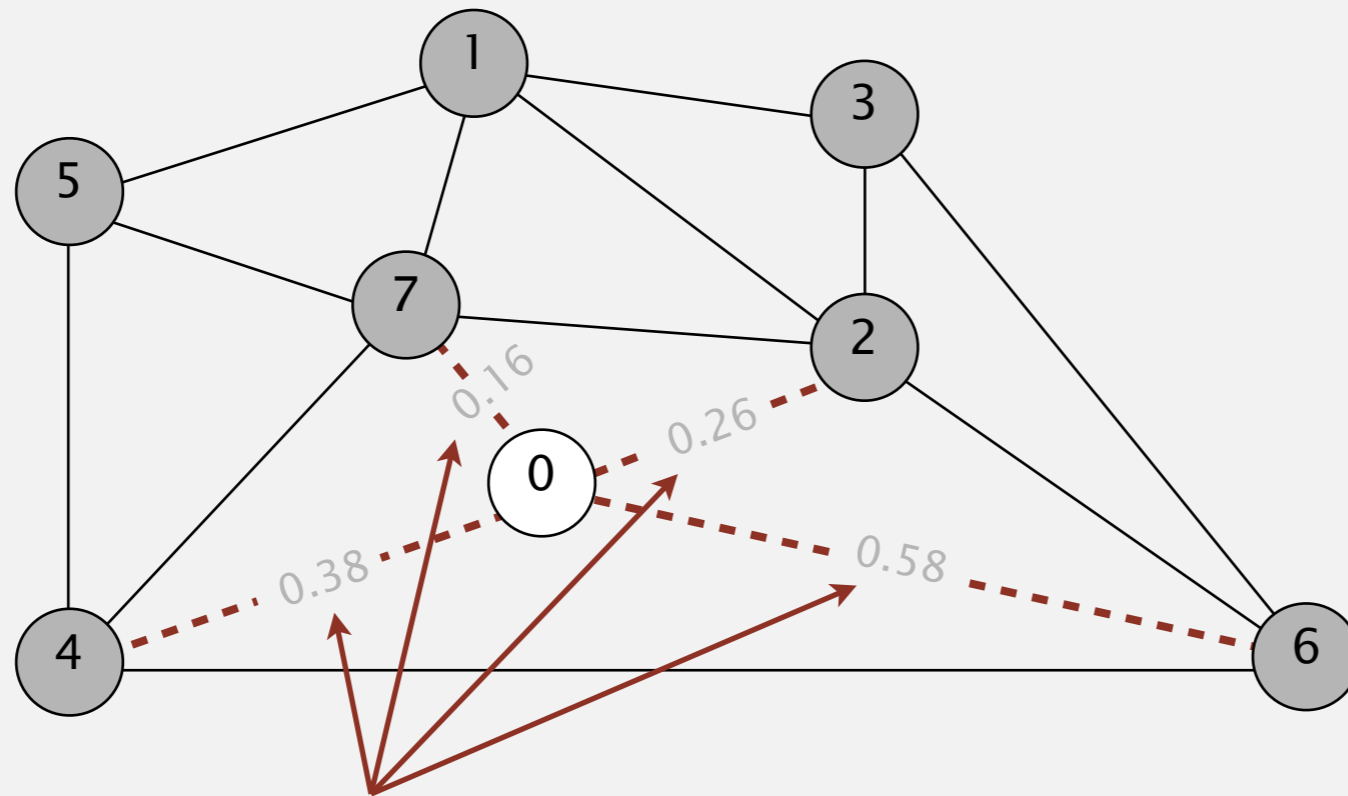
- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



	v	edgeTo[]	distTo[]
→	0	-	-

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



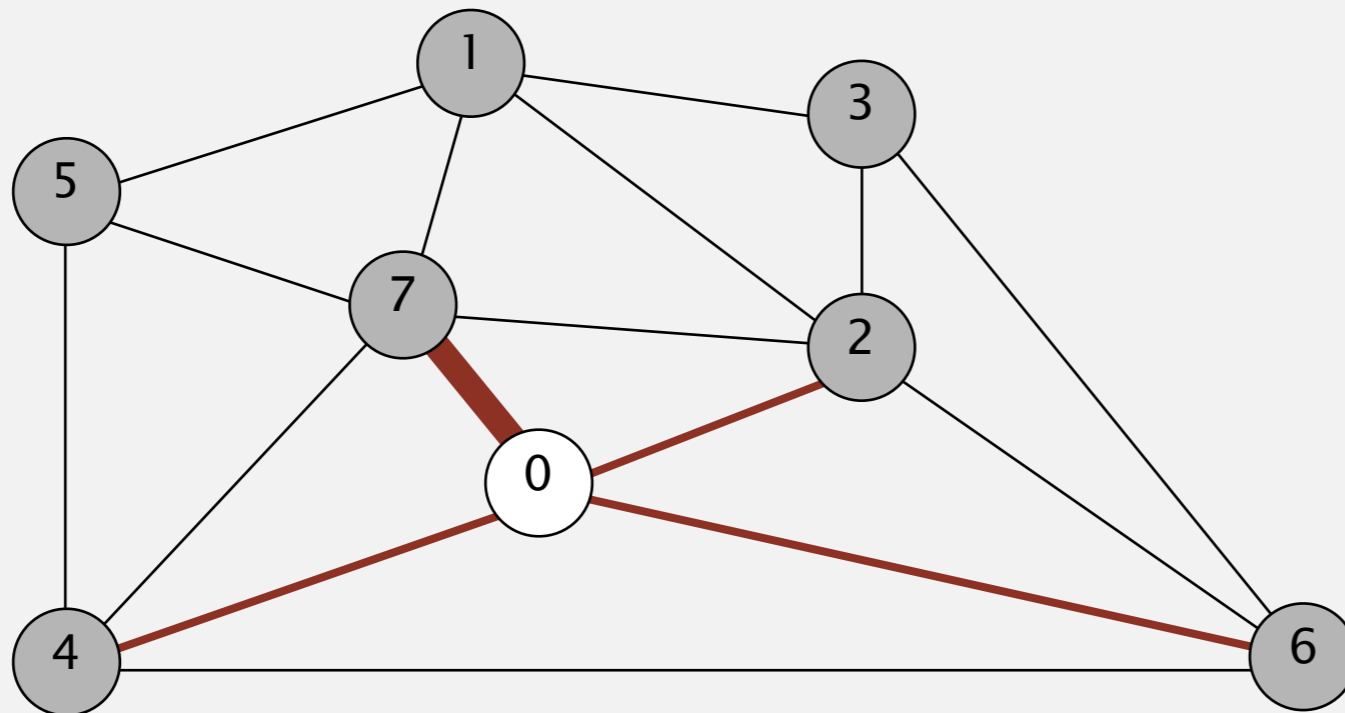
found connections to 7, 2, 4, and 6
(add to PQ)

v	edgeTo[]	distTo[]
→ 0	-	-
7	0-7	0.16
2	0-2	0.26
4	0-4	0.38
6	6-0	0.58

vertices on PQ
(sorted by weight)

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

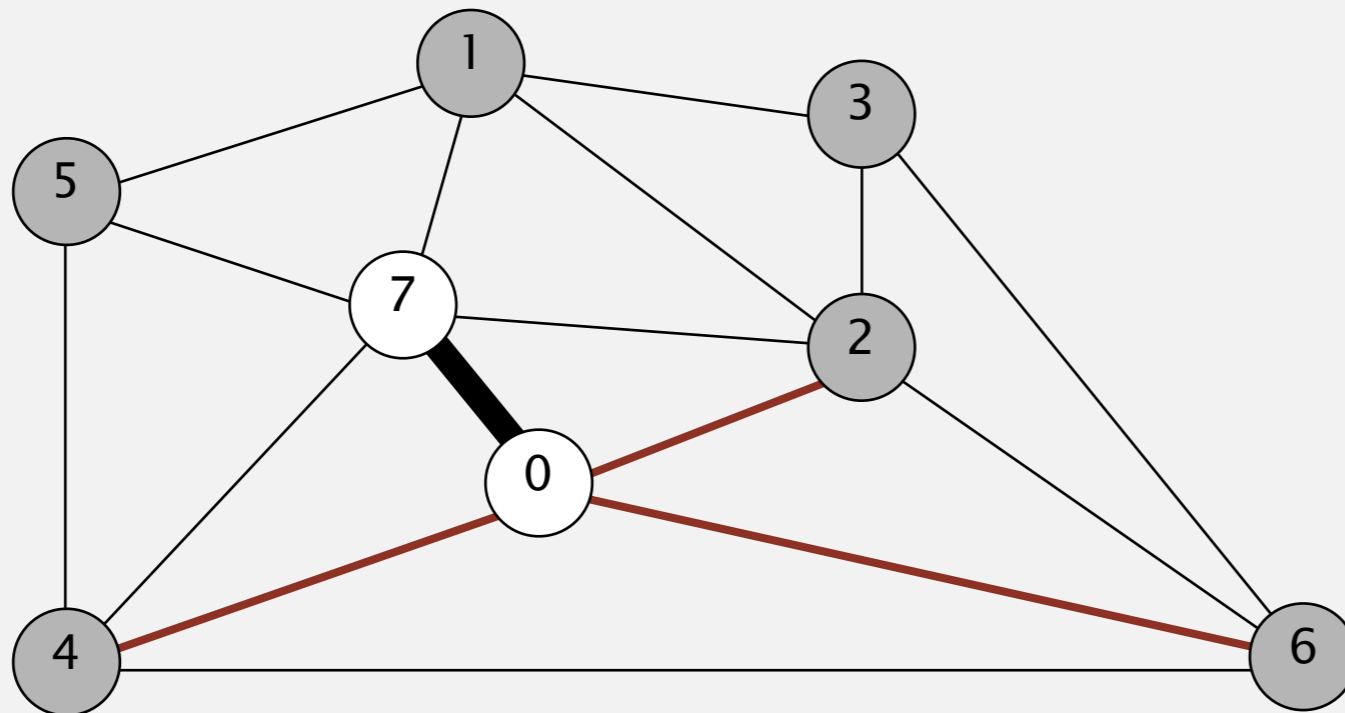


v	edgeTo[]	distTo[]
0	-	-
→ 7	0-7	0.16
2	0-2	0.26
4	0-4	0.38
6	6-0	0.58

vertices on PQ
(sorted by weight)

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



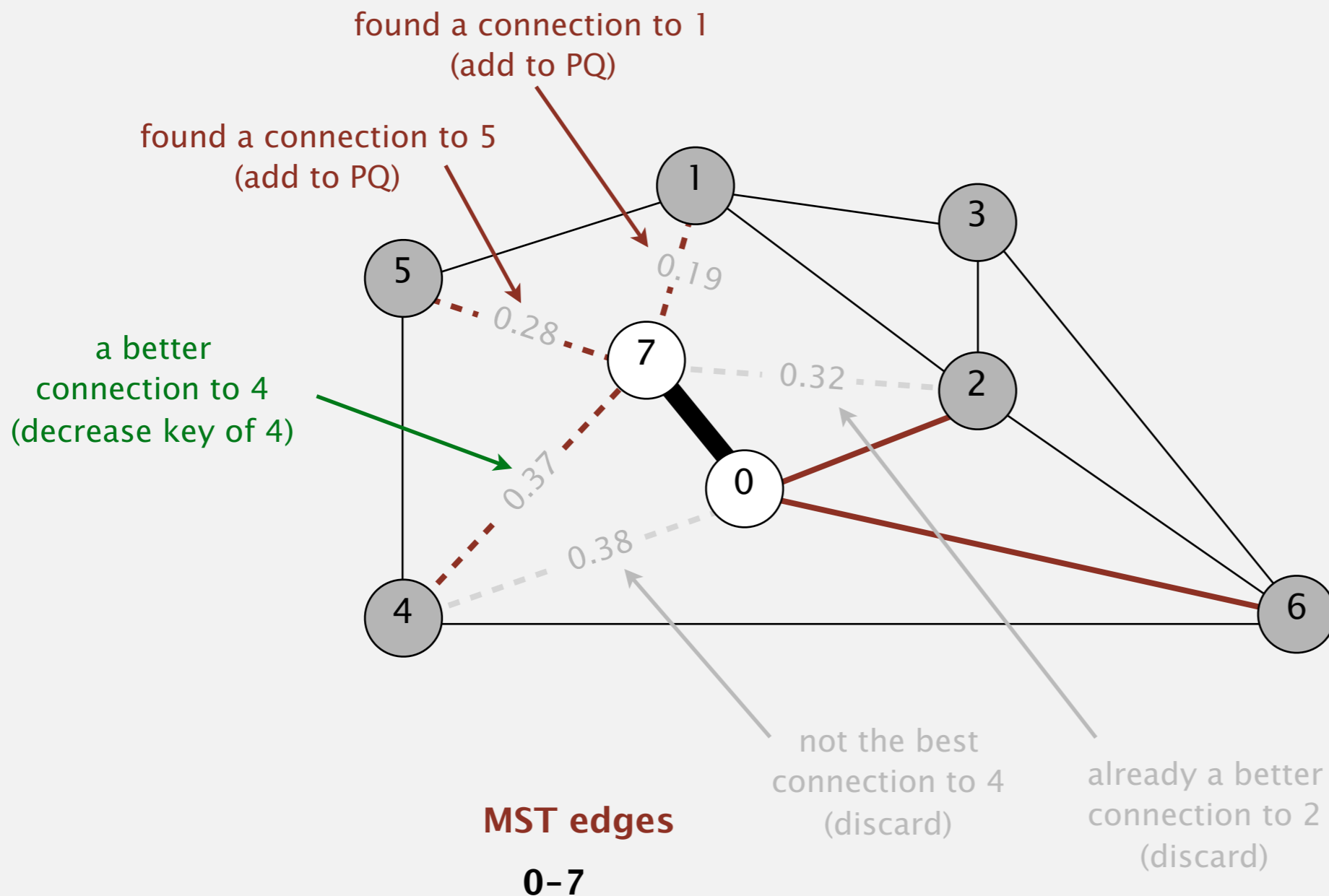
<u>v</u>	<u>edgeTo[]</u>	<u>distTo[]</u>
0	-	-
→ 7	0-7	0.16
2	0-2	0.26
4	0-4	0.38
6	6-0	0.58

MST edges

0-7

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

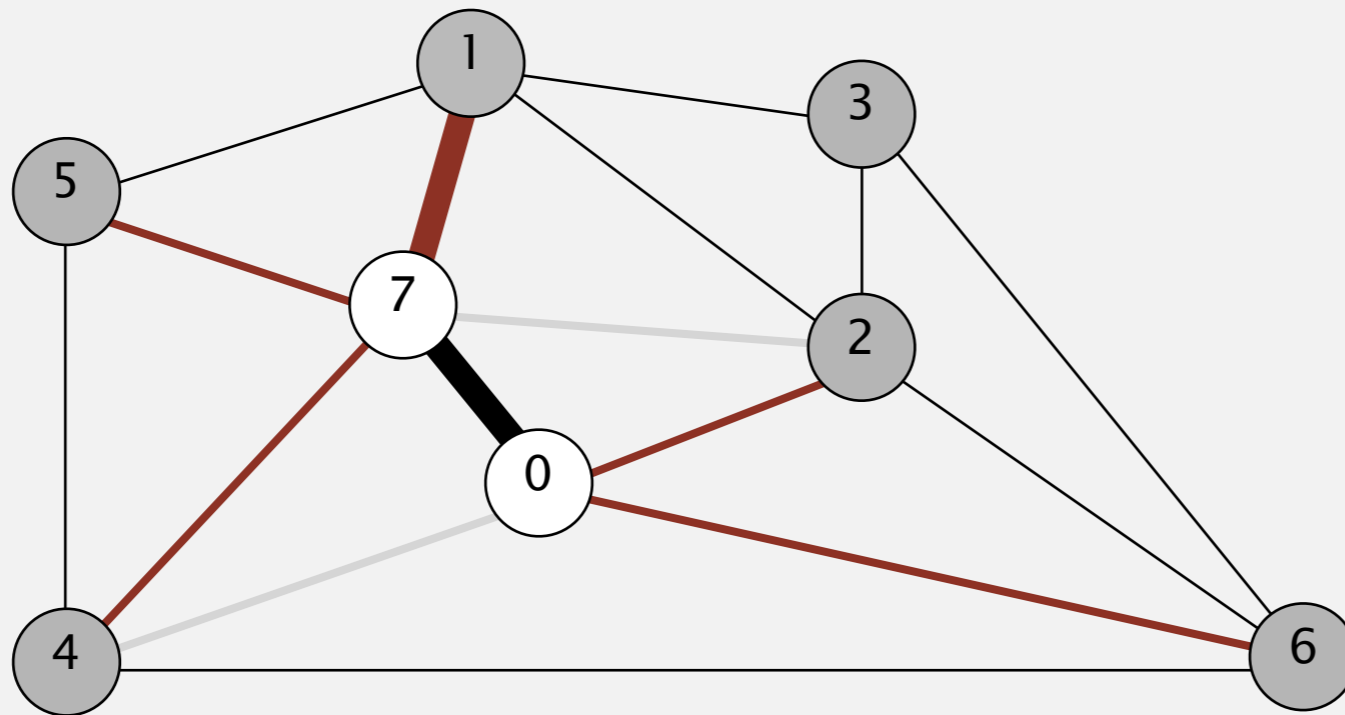


v	edgeTo[]	distTo[]
0	-	-
→ 7	0-7	0.16
①	1-7	0.19
2	0-2	0.26
⑤	5-7	0.28
④	0-4 4-7	0.38 0.37
6	6-0	0.58

vertices on PQ
(sorted by weight)

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



v	edgeTo[]	distTo[]
0	-	-
7	0-7	0.16
→ 1	1-7	0.19
2	0-2	0.26
5	5-7	0.28
4	4-7	0.37
6	6-0	0.58

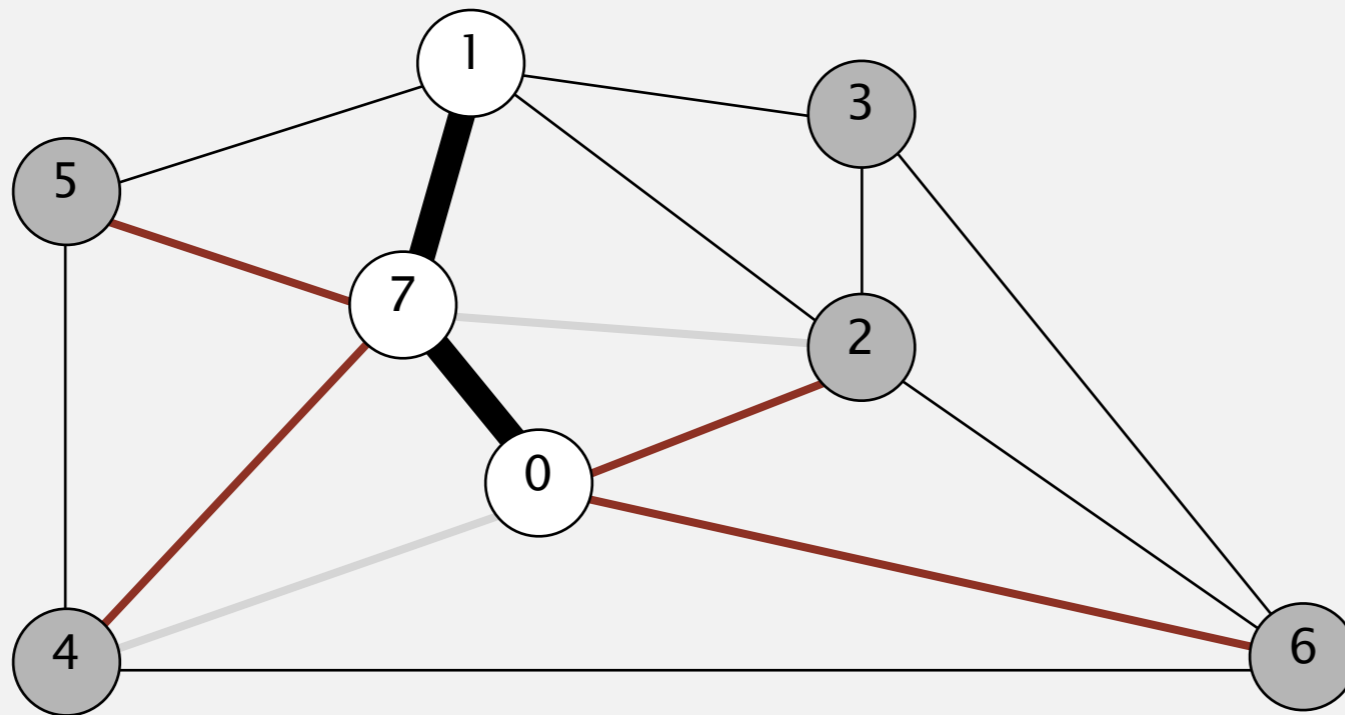
vertices on PQ
(sorted by weight)

MST edges

0-7 1-7

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



<u>v</u>	<u>edgeTo[]</u>	<u>distTo[]</u>
0	-	-
7	0-7	0.16
→ 1	1-7	0.19
2	0-2	0.26
5	5-7	0.28
4	4-7	0.37
6	6-0	0.58

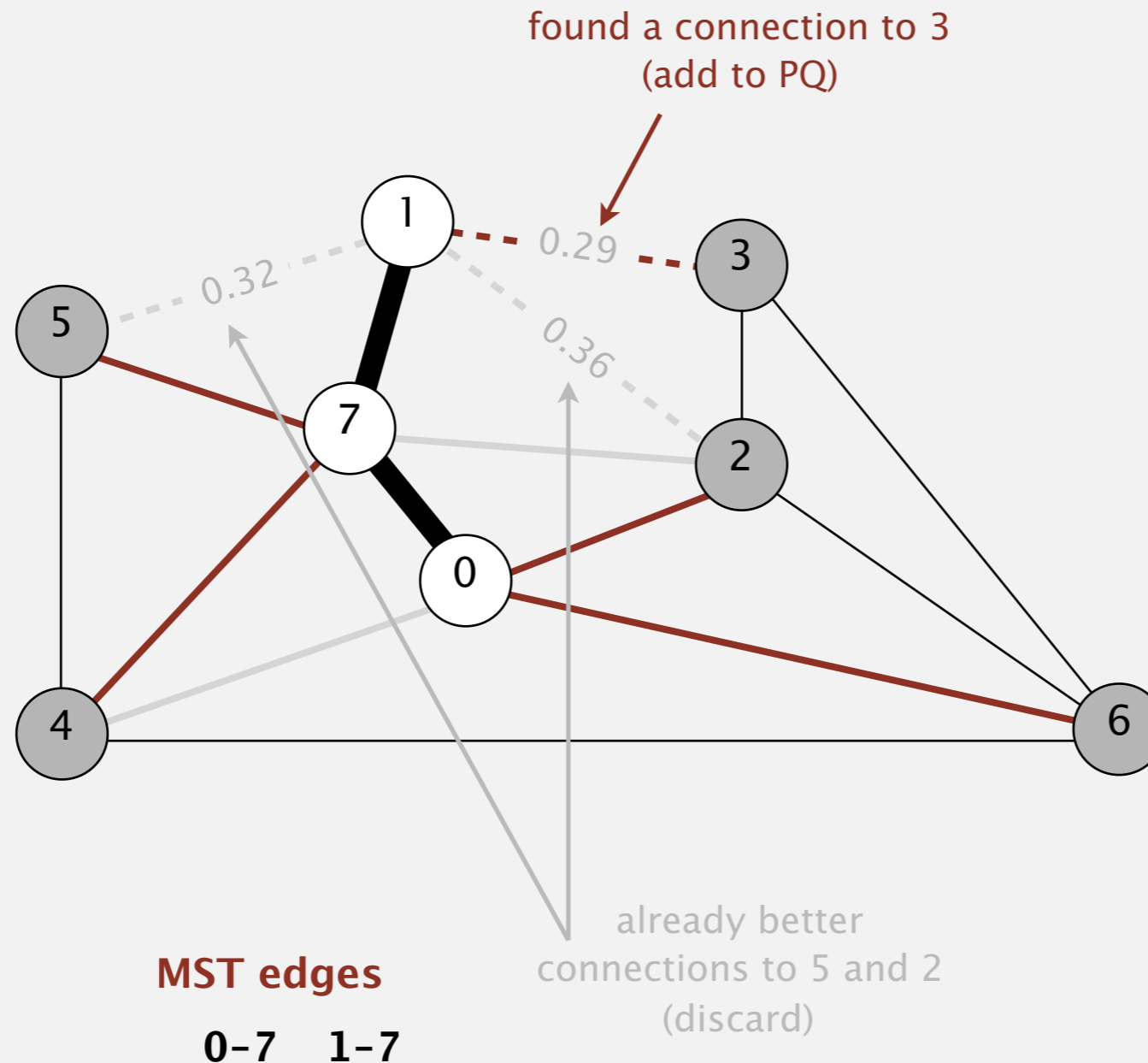
vertices on PQ
(sorted by weight)

MST edges

0-7 1-7

Prim's algorithm: eager implementation demo

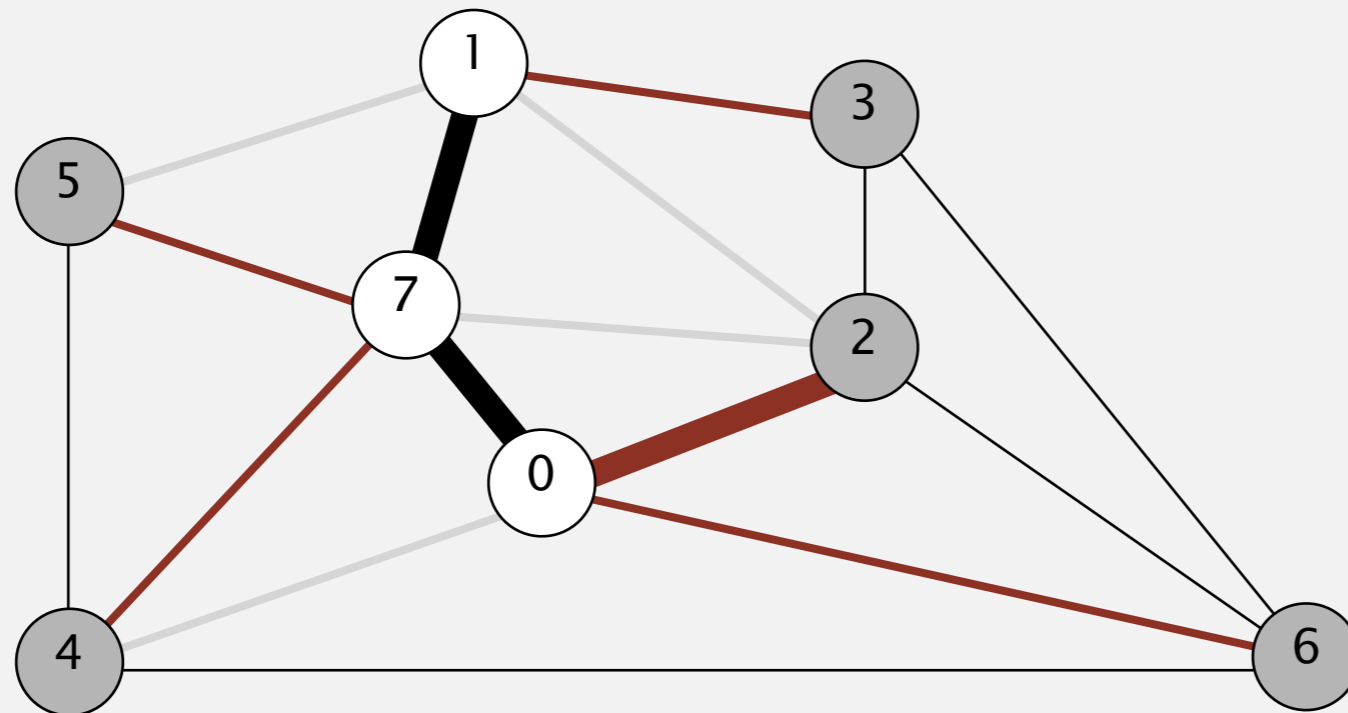
- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



v	edgeTo[]	distTo[]
0	-	-
7	0-7	0.16
→ 1	1-7	0.19
2	0-2	0.26
5	5-7	0.28
3	1-3	0.29
4	4-7	0.37
6	6-0	0.58

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



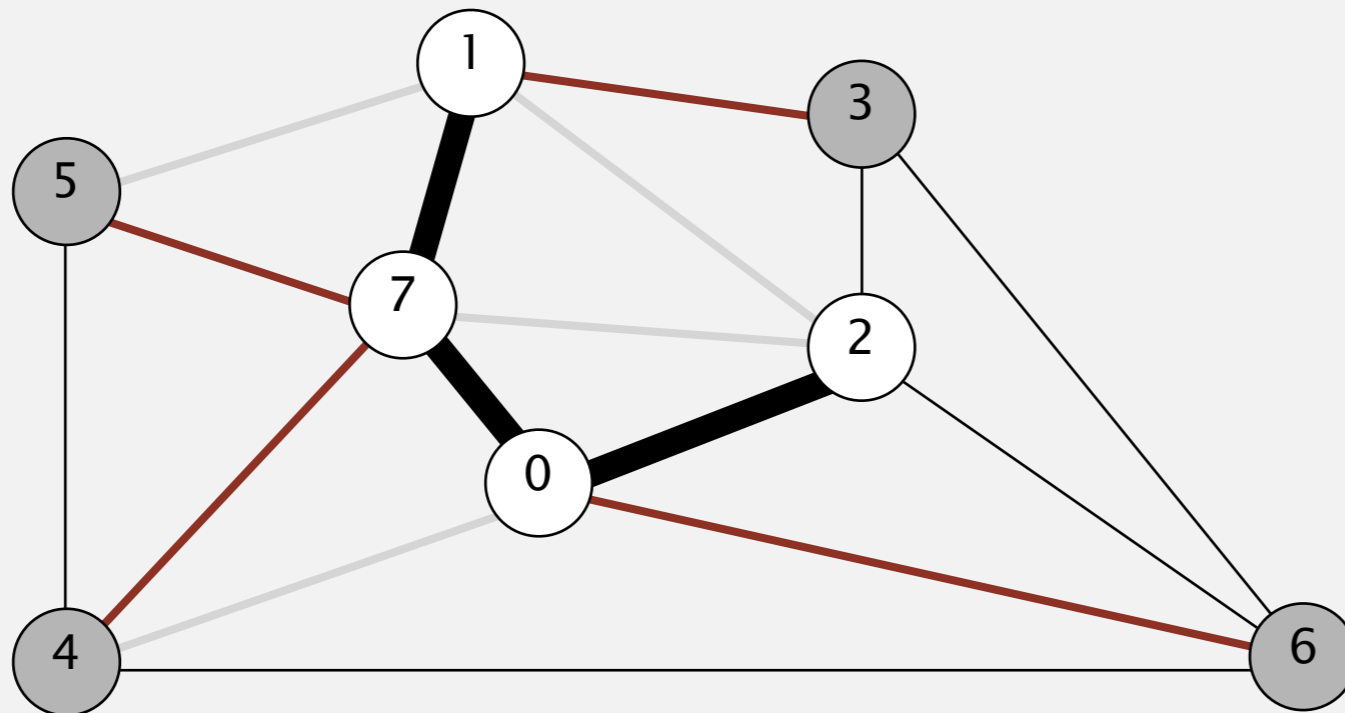
<u>v</u>	<u>edgeTo[]</u>	<u>distTo[]</u>
0	-	-
7	0-7	0.16
1	1-7	0.19
→ 2	0-2	0.26
5	5-7	0.28
3	1-3	0.29
4	4-7	0.37
6	6-0	0.58

MST edges

0-7 1-7

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



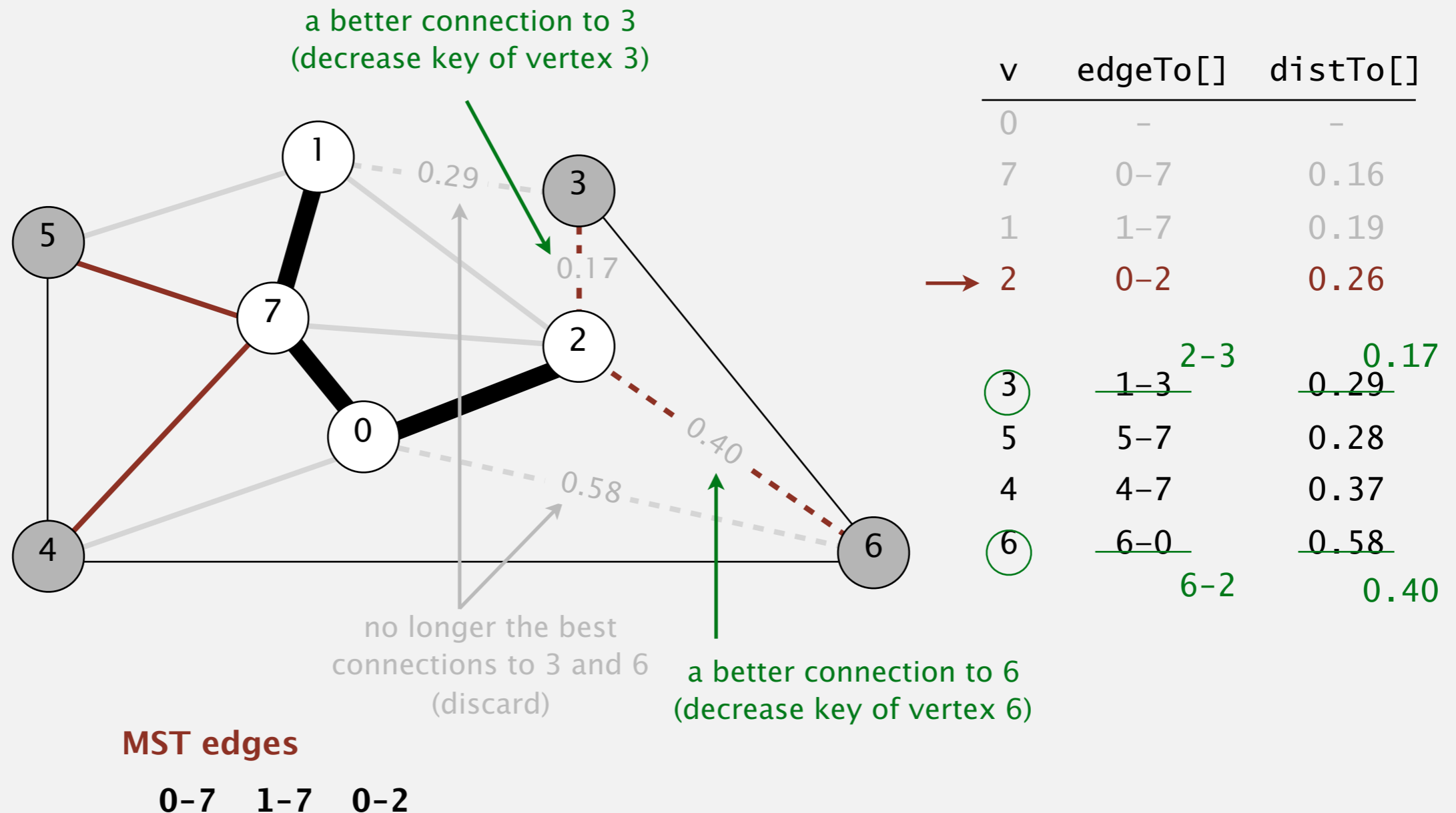
v	edgeTo[]	distTo[]
0	-	-
7	0-7	0.16
1	1-7	0.19
→ 2	0-2	0.26
5	5-7	0.28
3	1-3	0.29
4	4-7	0.37
6	6-0	0.58

MST edges

0-7 1-7 0-2

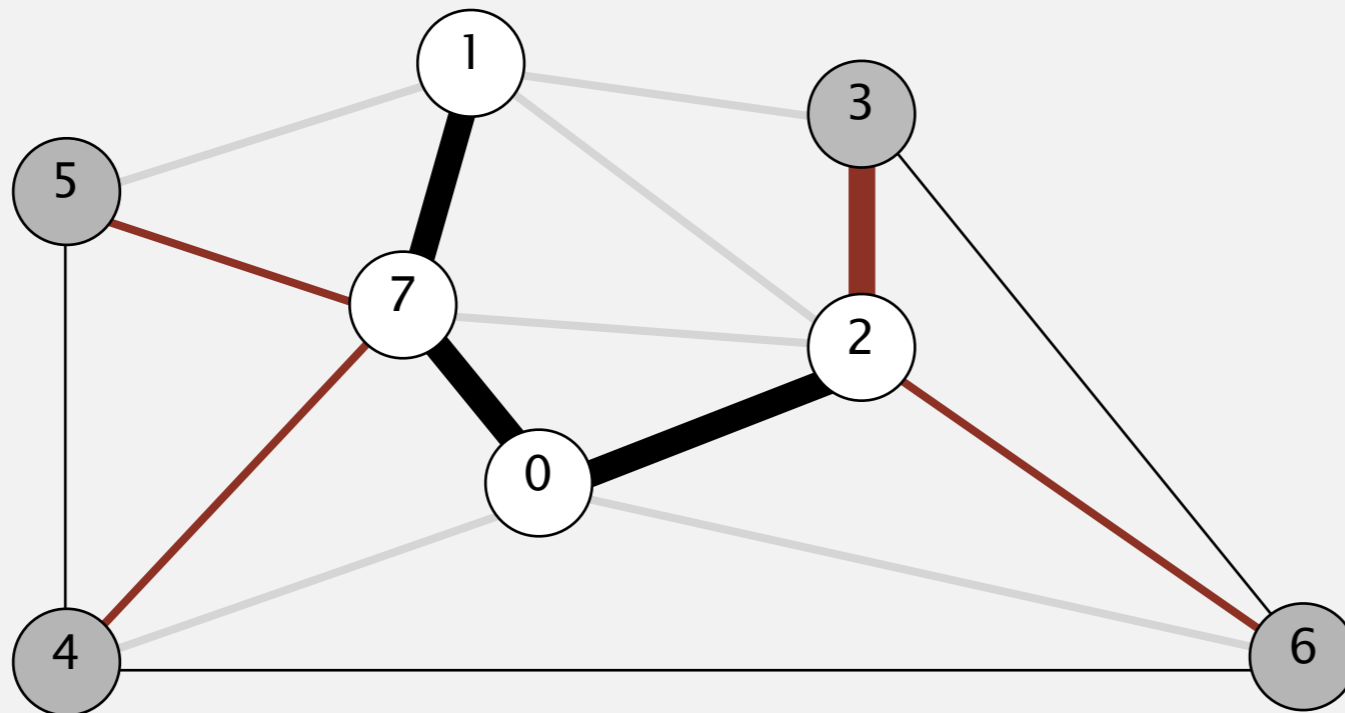
Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



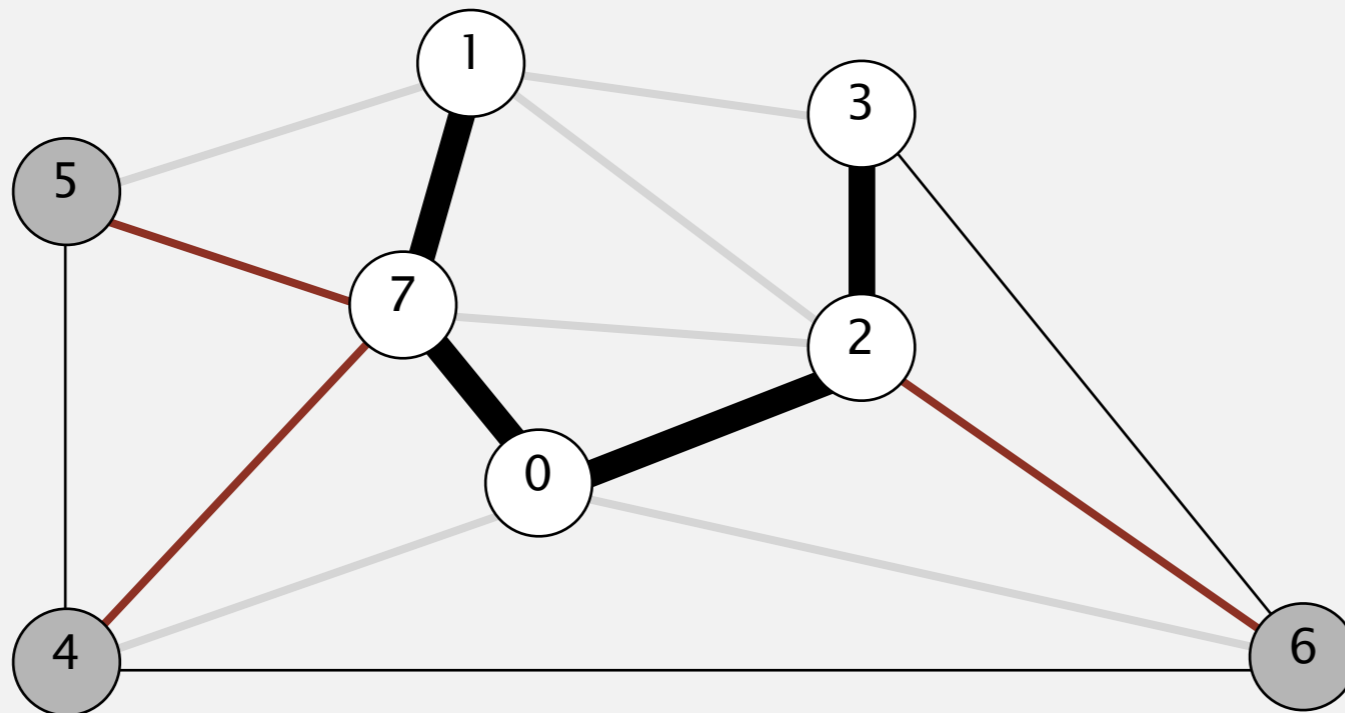
v	edgeTo[]	distTo[]
0	-	-
7	0-7	0.16
1	1-7	0.19
2	0-2	0.26
→ 3	2-3	0.17
5	5-7	0.28
4	4-7	0.37
6	6-2	0.40

MST edges

0-7 1-7 0-2 2-3

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



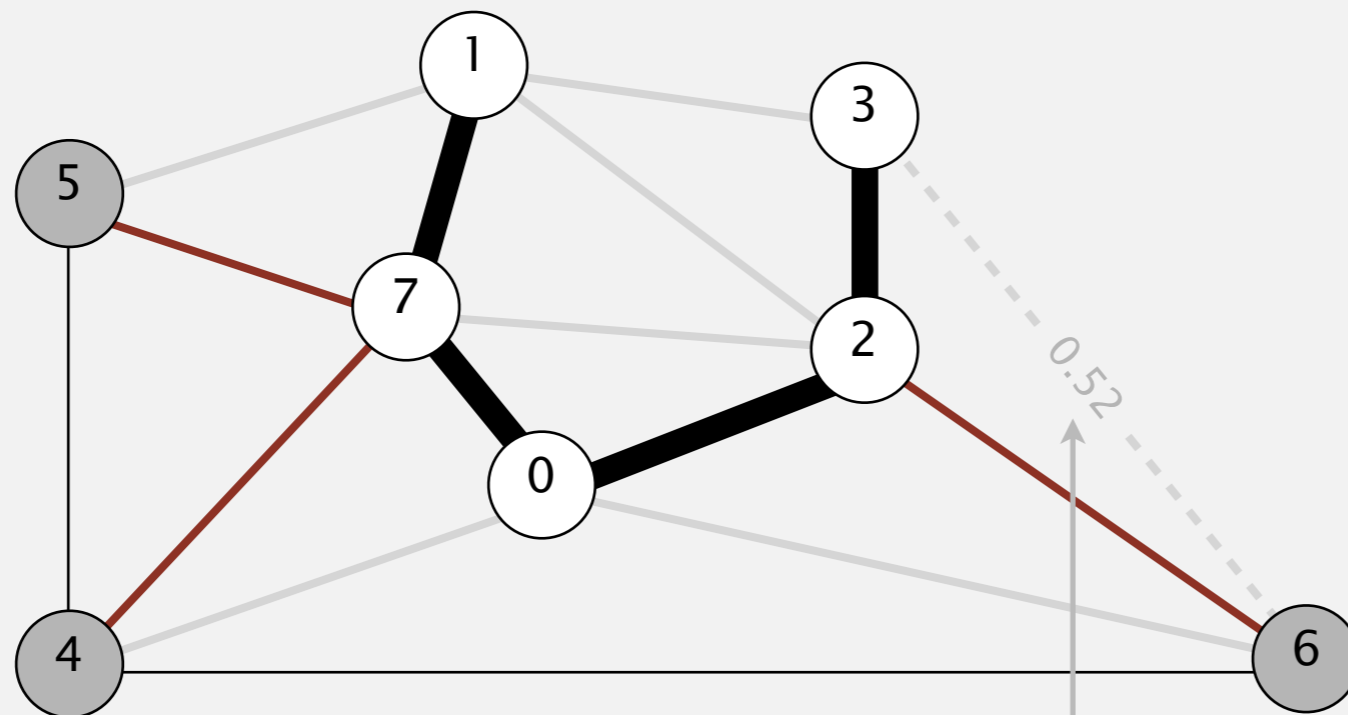
<u>v</u>	<u>edgeTo[]</u>	<u>distTo[]</u>
0	-	-
7	0-7	0.16
1	1-7	0.19
2	0-2	0.26
→ 3	2-3	0.17
5	5-7	0.28
4	4-7	0.37
6	6-2	0.40

MST edges

0-7 1-7 0-2 2-3

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



MST edges

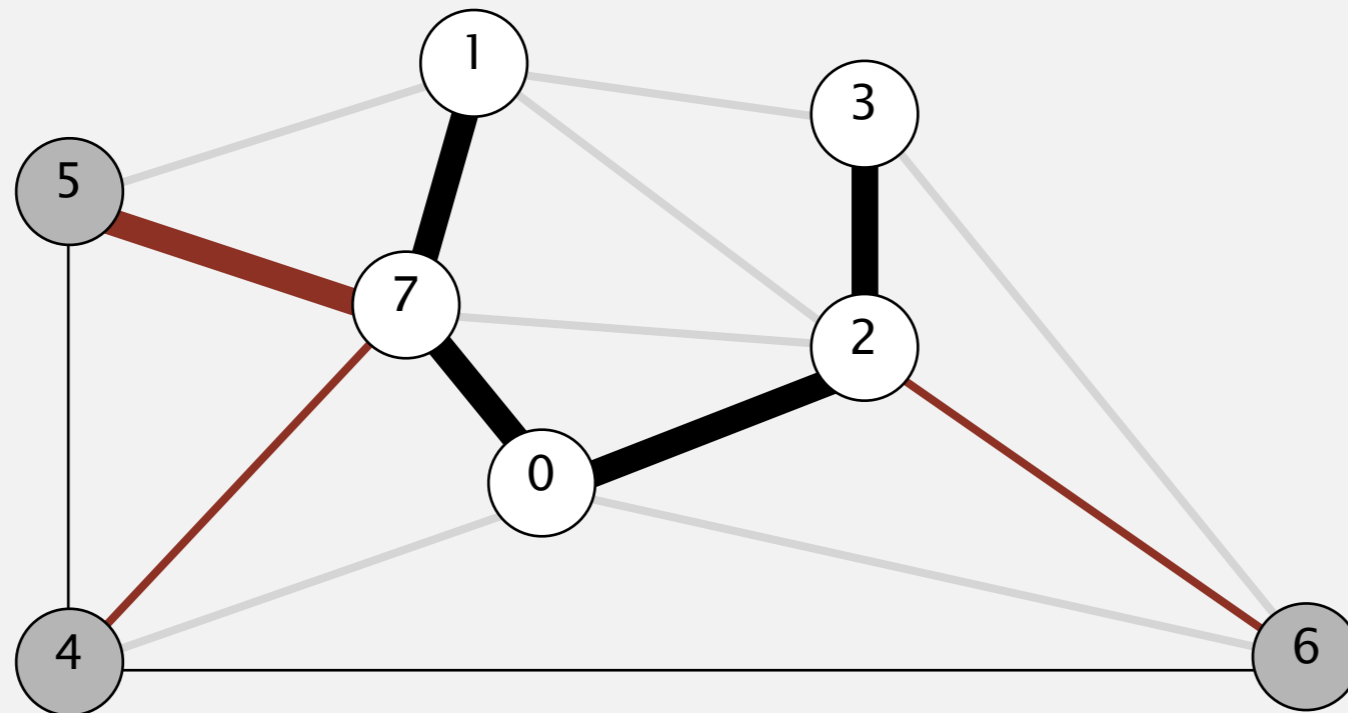
0-7 1-7 0-2 2-3

already a better
connection to 6
(discard)

v	edgeTo[]	distTo[]
0	-	-
7	0-7	0.16
1	1-7	0.19
2	0-2	0.26
→ 3	2-3	0.17
5	5-7	0.28
4	4-7	0.37
6	6-2	0.40

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



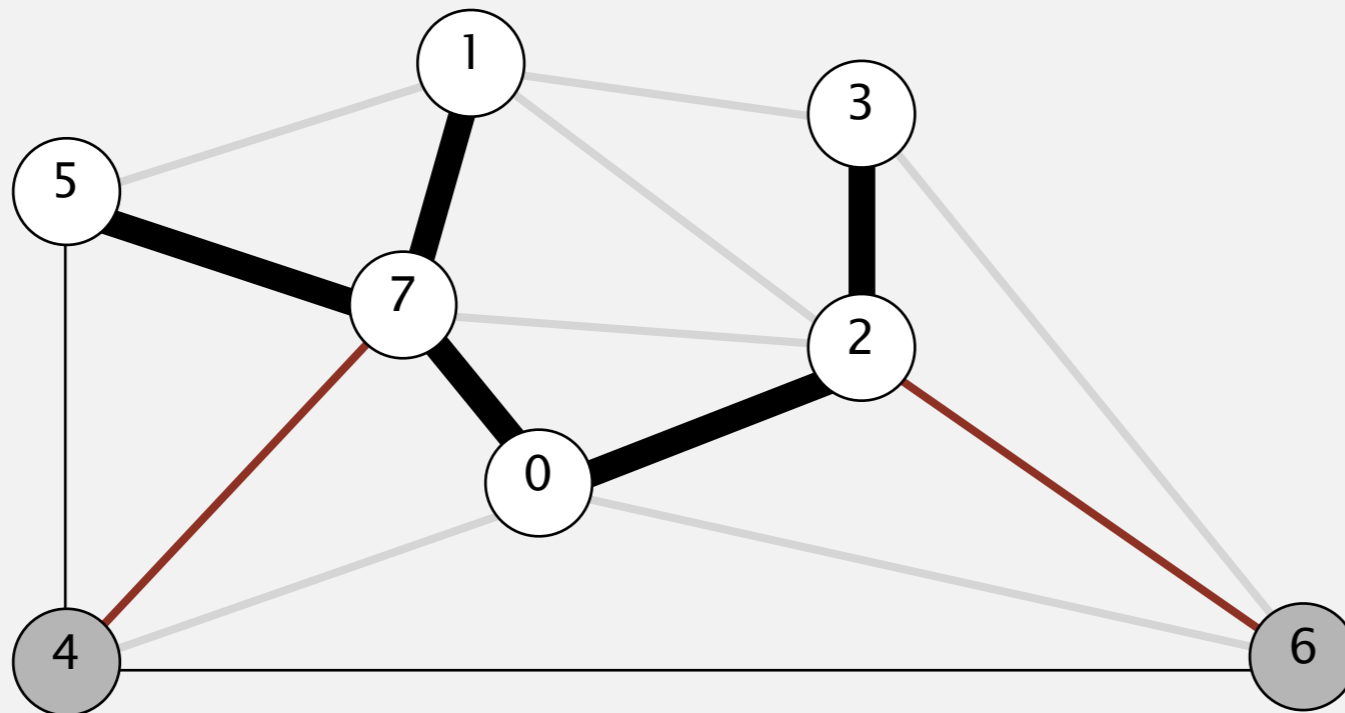
v	edgeTo[]	distTo[]
0	-	-
7	0-7	0.16
1	1-7	0.19
2	0-2	0.26
3	2-3	0.17
→ 5	5-7	0.28
4	4-7	0.37
6	6-2	0.40

MST edges

0-7 1-7 0-2 2-3

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



v	edgeTo[]	distTo[]
0	-	-
7	0-7	0.16
1	1-7	0.19
2	0-2	0.26
3	2-3	0.17
→ 5	5-7	0.28
4	4-7	0.37
6	6-2	0.40

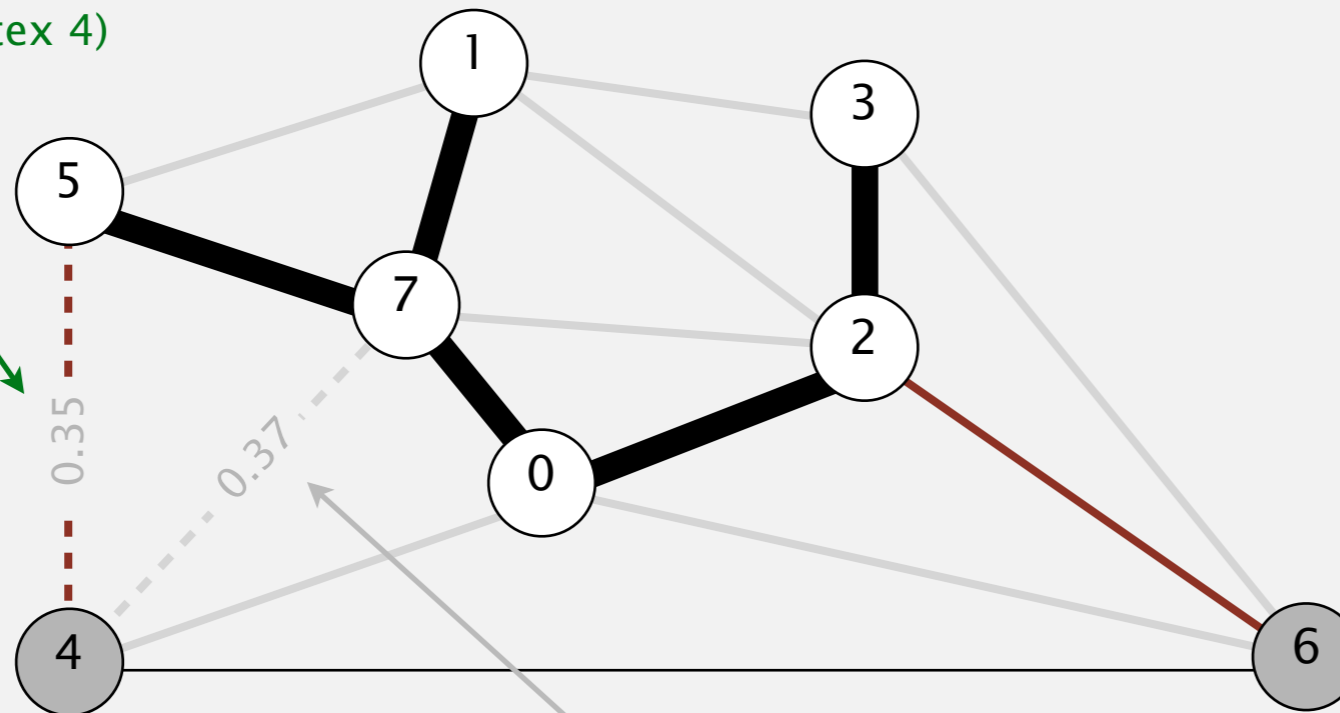
MST edges

0-7 1-7 0-2 2-3 5-7

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T

a better connection to 4
(decrease key of vertex 4)



no longer the best
connection to 4
(discard)

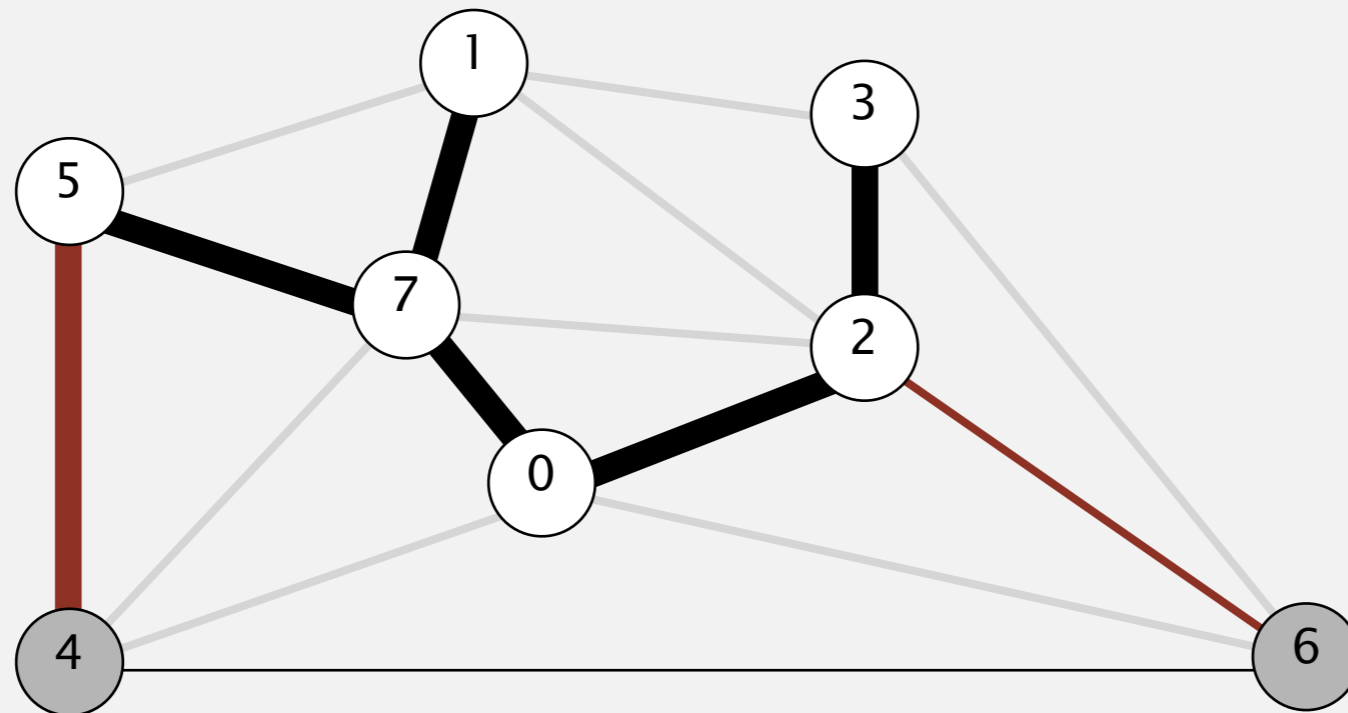
MST edges

0-7 1-7 0-2 2-3 5-7

v	edgeTo[]	distTo[]
0	-	-
7	0-7	0.16
1	1-7	0.19
2	0-2	0.26
3	2-3	0.17
5	5-7	0.28
4	4-7 ⁴⁻⁵	0.37 ^{0.35}
6	6-2	0.40

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



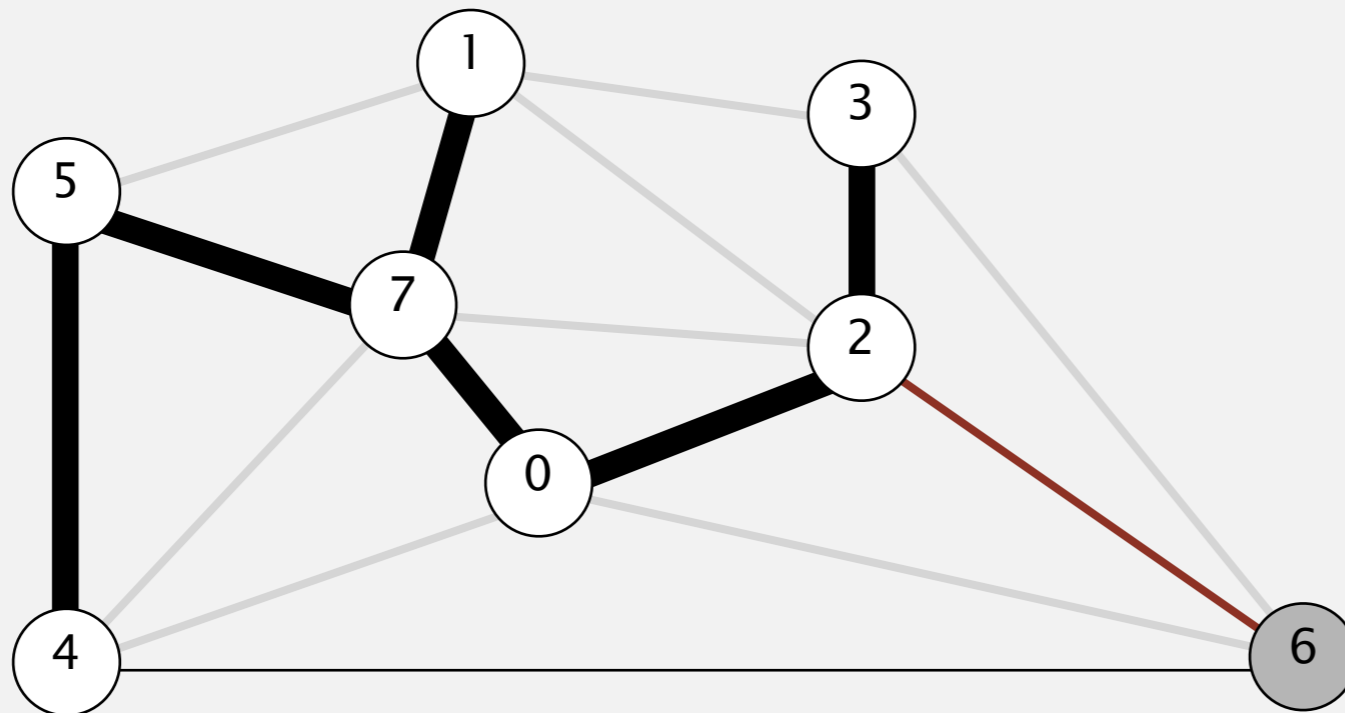
v	edgeTo[]	distTo[]
0	-	-
7	0-7	0.16
1	1-7	0.19
2	0-2	0.26
3	2-3	0.17
5	5-7	0.28
4	4-5	0.35
6	6-2	0.40

MST edges

0-7 1-7 0-2 2-3 5-7

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



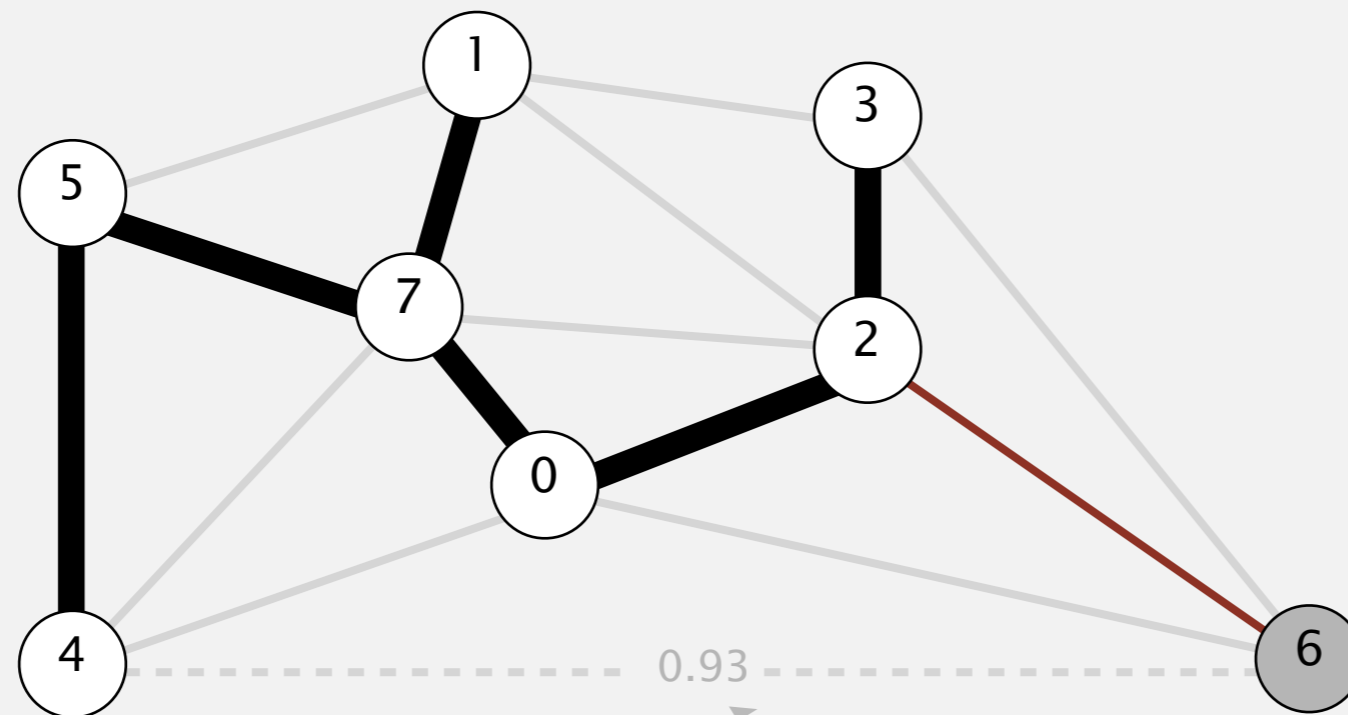
<u>v</u>	<u>edgeTo[]</u>	<u>distTo[]</u>
0	-	-
7	0-7	0.16
1	1-7	0.19
2	0-2	0.26
3	2-3	0.17
5	5-7	0.28
→ 4	4-5	0.35
6	6-2	0.40

MST edges

0-7 1-7 0-2 2-3 5-7 4-5

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



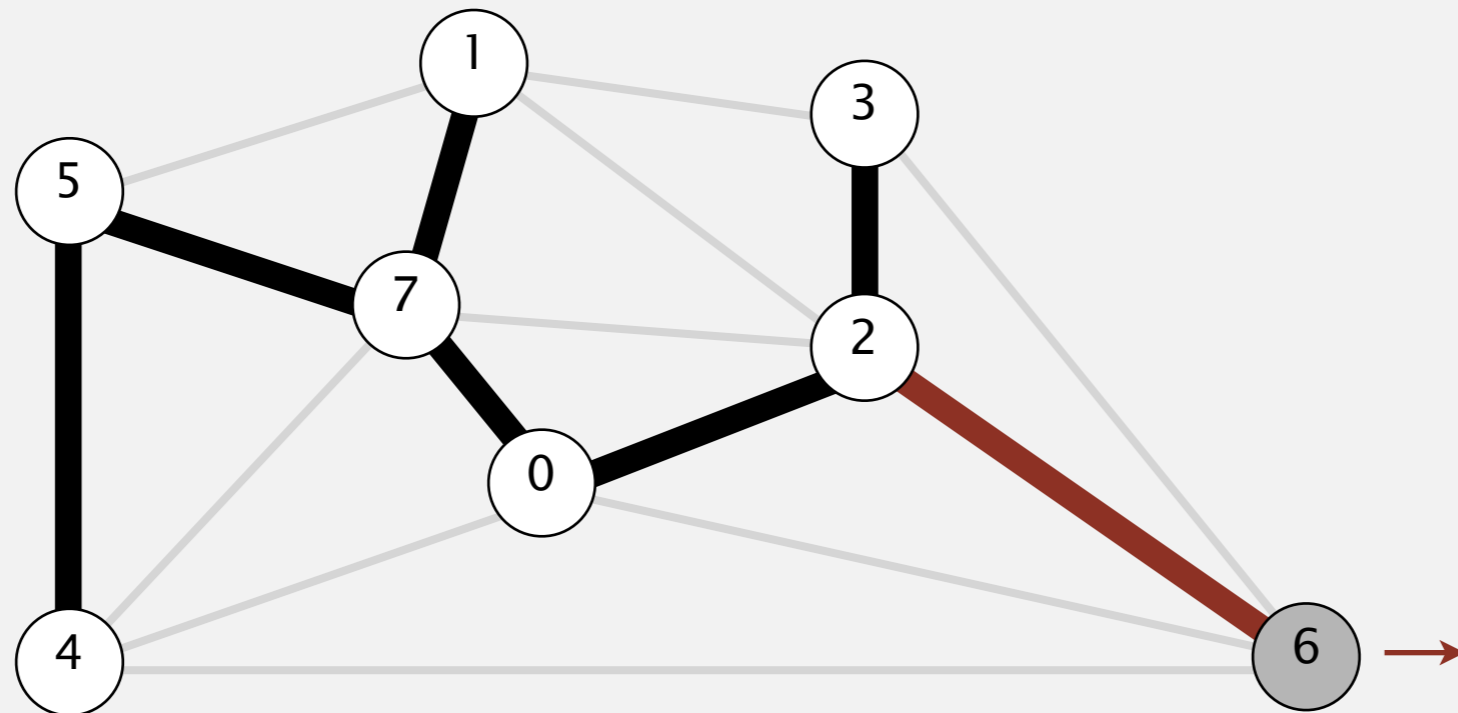
v	edgeTo[]	distTo[]
0	-	-
7	0-7	0.16
1	1-7	0.19
2	0-2	0.26
3	2-3	0.17
5	5-7	0.28
4	4-5	0.35
6	6-2	0.40

MST edges

0-7 1-7 0-2 2-3 5-7 4-5

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



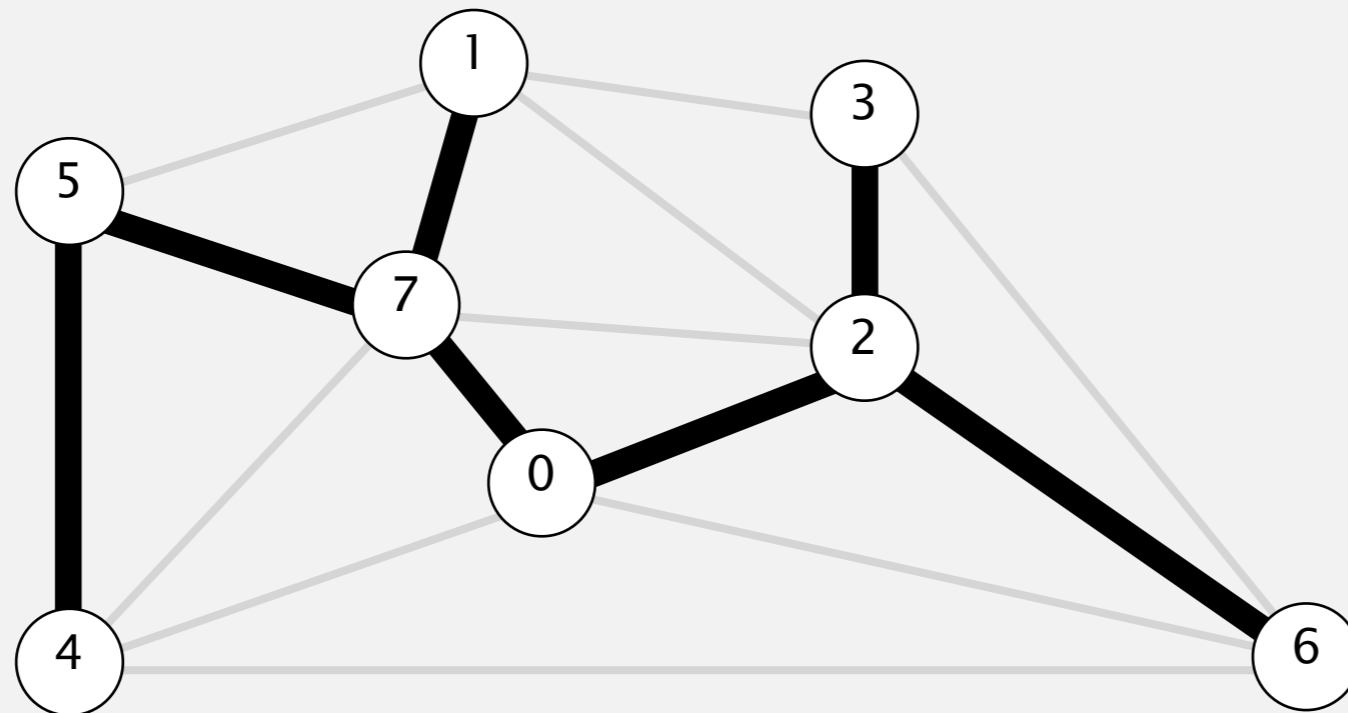
v	edgeTo[]	distTo[]
0	-	-
7	0-7	0.16
1	1-7	0.19
2	0-2	0.26
3	2-3	0.17
5	5-7	0.28
4	4-5	0.35
6	6-2	0.40

MST edges

0-7 1-7 0-2 2-3 5-7 4-5

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



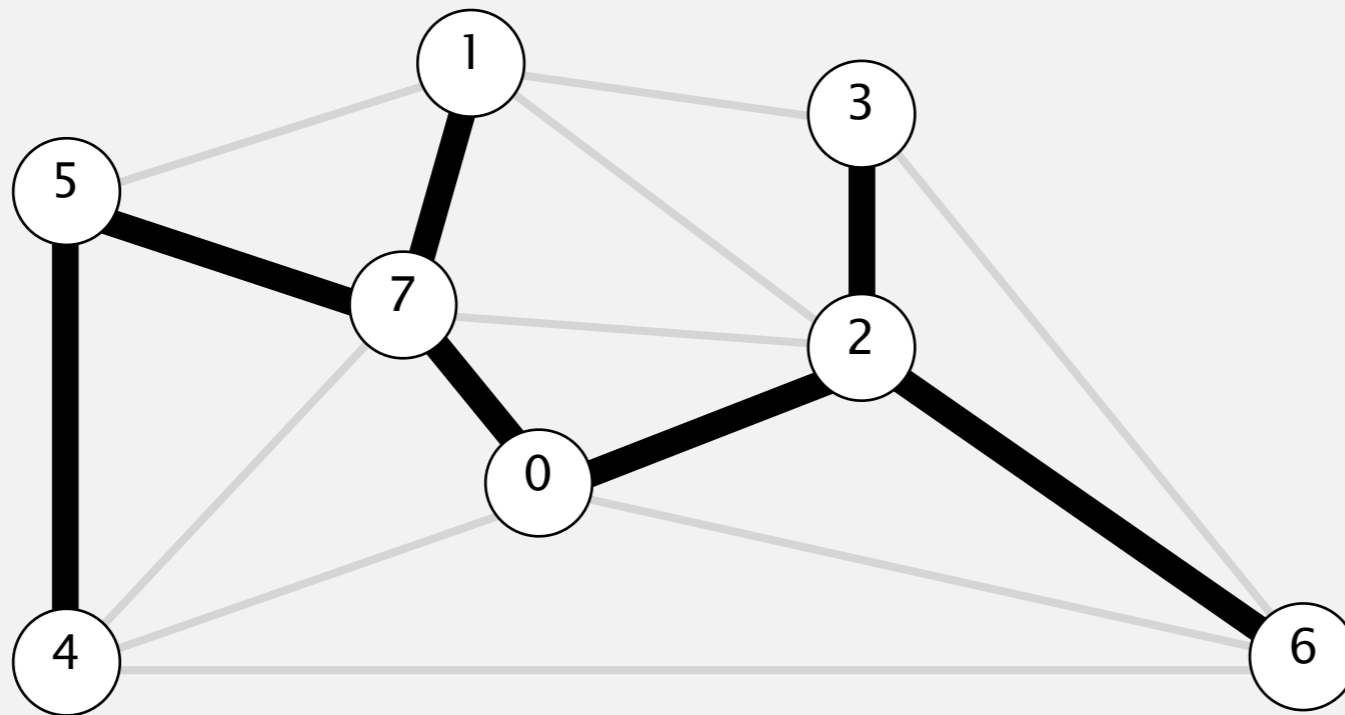
<u>v</u>	<u>edgeTo[]</u>	<u>distTo[]</u>
0	-	-
7	0-7	0.16
1	1-7	0.19
2	0-2	0.26
3	2-3	0.17
5	5-7	0.28
4	4-5	0.35
→ 6	6-2	0.40

MST edges

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

Prim's algorithm: eager implementation demo

- Start with vertex 0 and grow tree T .
- Repeat until $V - 1$ edges:
 - add to T the min-weight edge with exactly one endpoint in T



v	edgeTo[]	distTo[]
0	-	-
7	0-7	0.16
1	1-7	0.19
2	0-2	0.26
3	2-3	0.17
5	5-7	0.28
4	4-5	0.35
6	6-2	0.40

MST edges

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