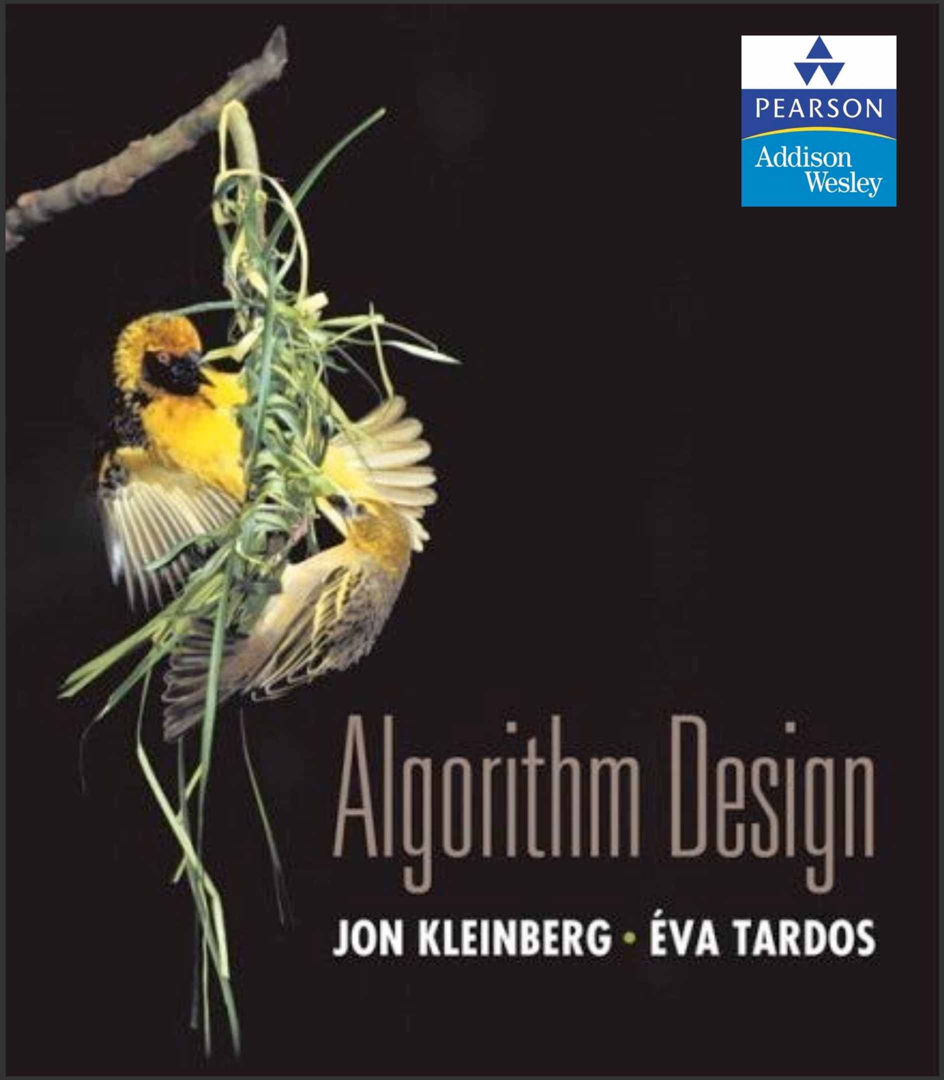


4. Greedy Algorithms II

- *Prim's algorithm demo*

Lecture slides by Kevin Wayne

http://www.cs.princeton.edu/~wayne/kleinberg-tardos

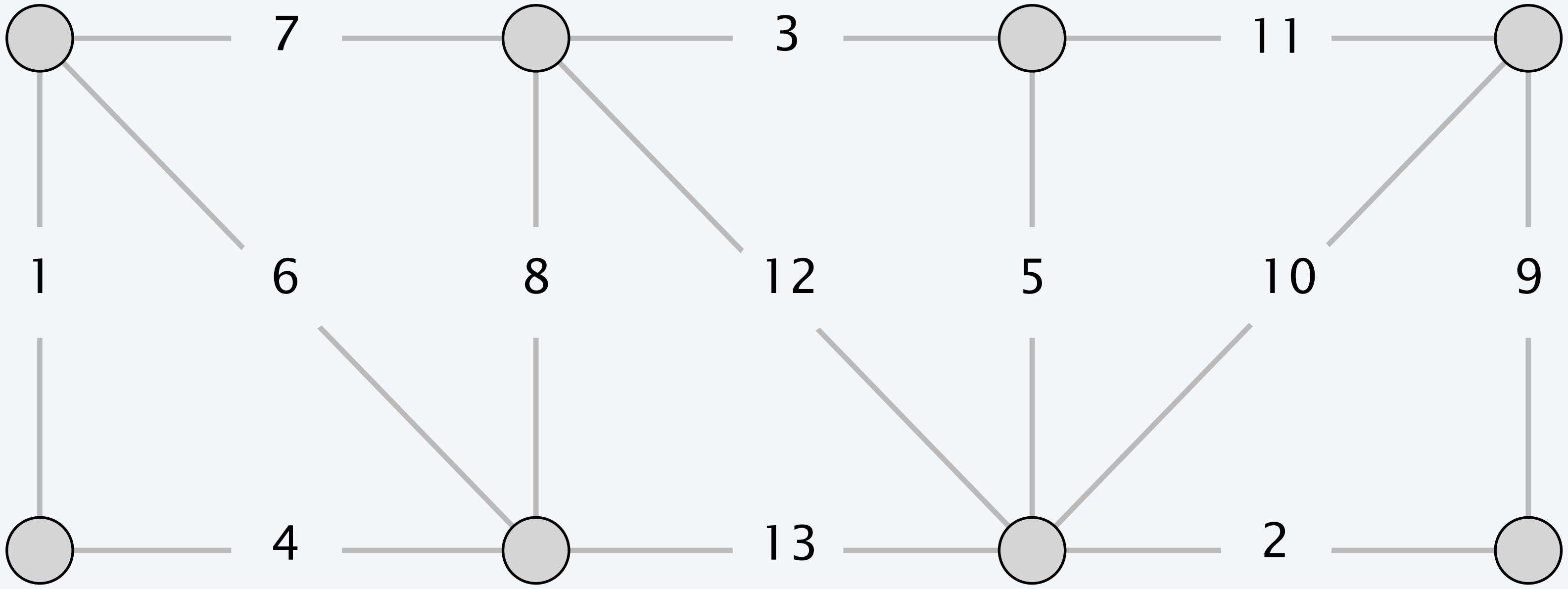
Last updated on 11/7/17 4:07 PM

Prim's algorithm demo

Initialize S = any node, $T = \varnothing$.

Repeat $n - 1$ times:

- Add to T a min-weight edge with one endpoint in S.
- Add new node to S.

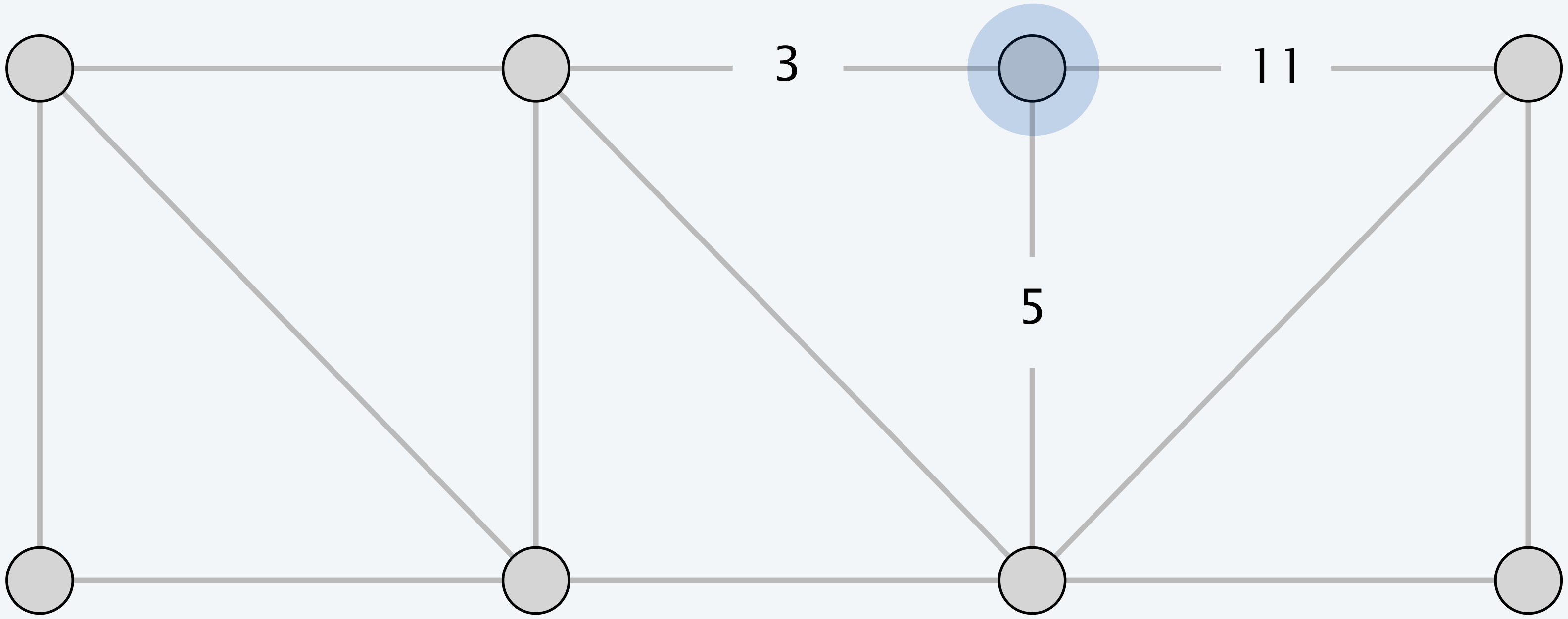


Prim's algorithm demo

Initialize S = any node, $T = \varnothing$.

Repeat $n - 1$ times:

- Add to T a min-weight edge with one endpoint in S.
- Add new node to S.

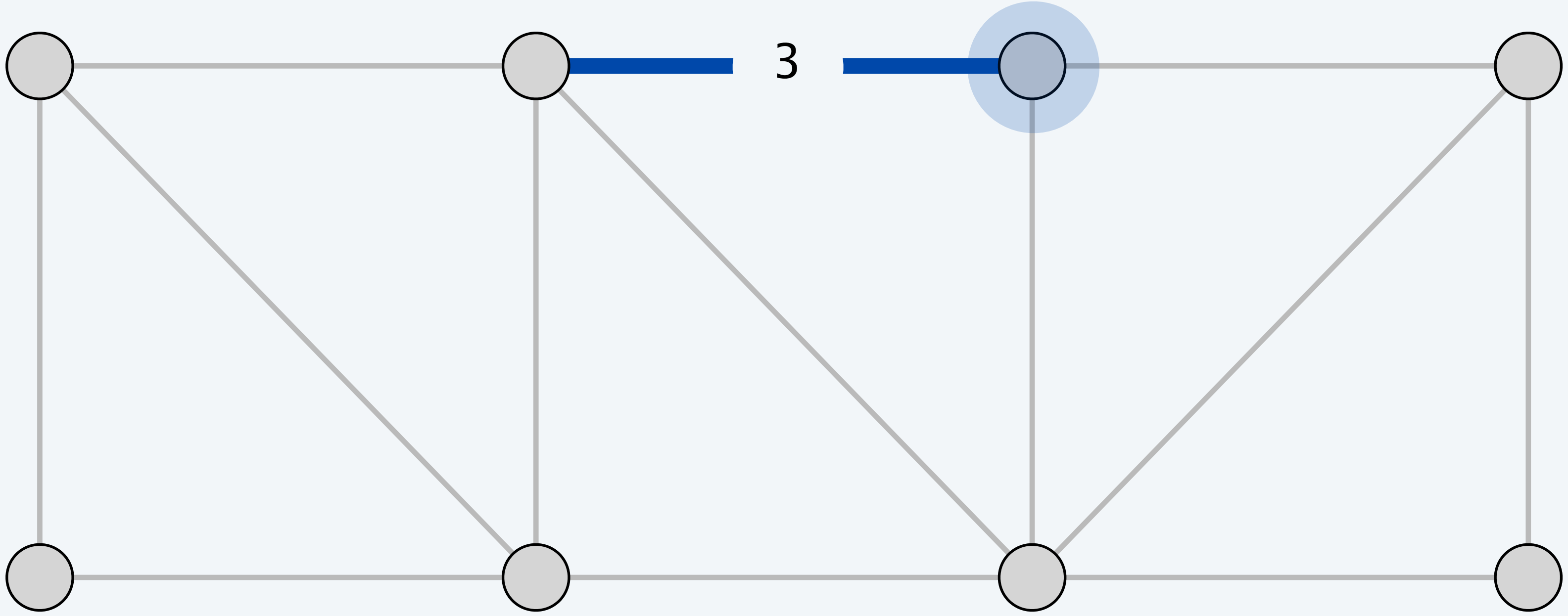


Prim's algorithm demo

Initialize S = any node, $T = \varnothing$.

Repeat $n - 1$ times:

- Add to T a min-weight edge with one endpoint in S.
- Add new node to S.

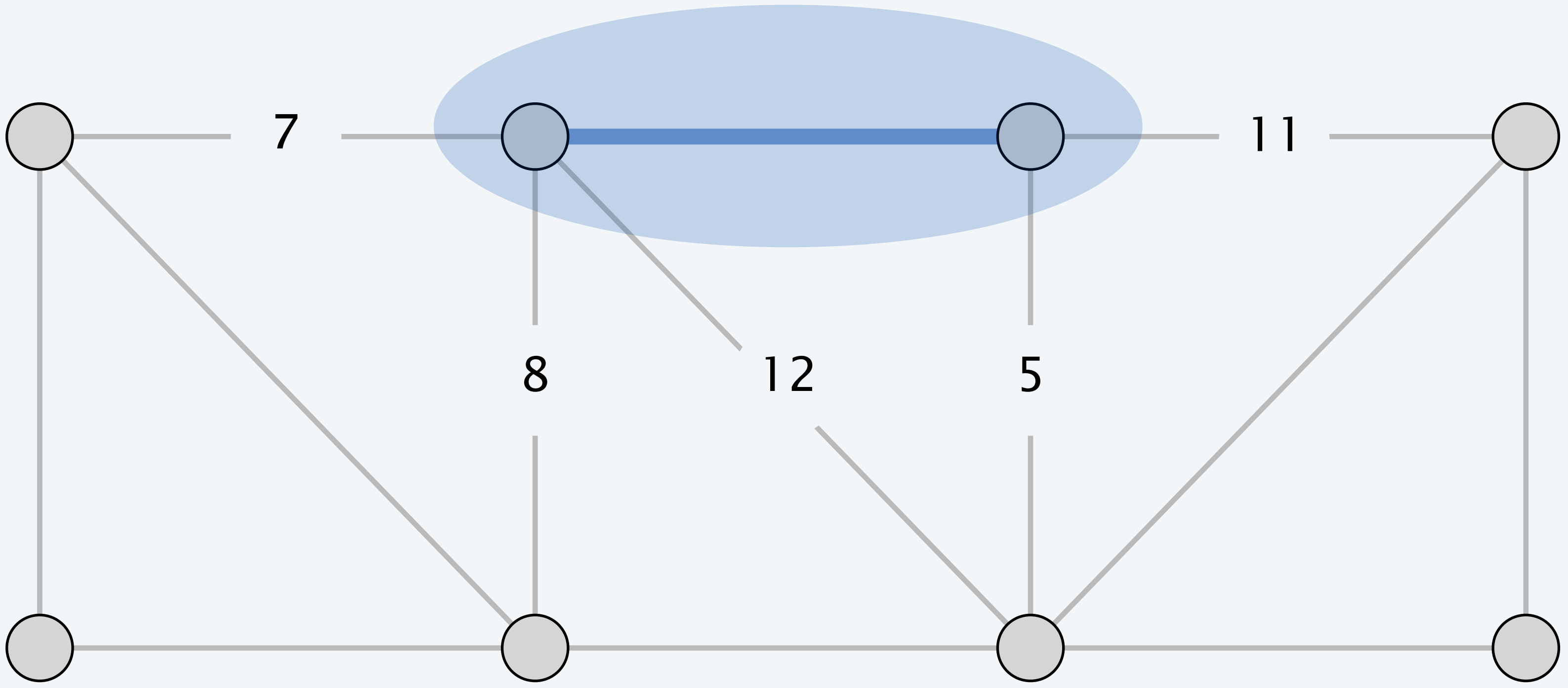


Prim's algorithm demo

Initialize S = any node, $T = \varnothing$.

Repeat $n - 1$ times:

- Add to T a min-weight edge with one endpoint in S.
- Add new node to S.

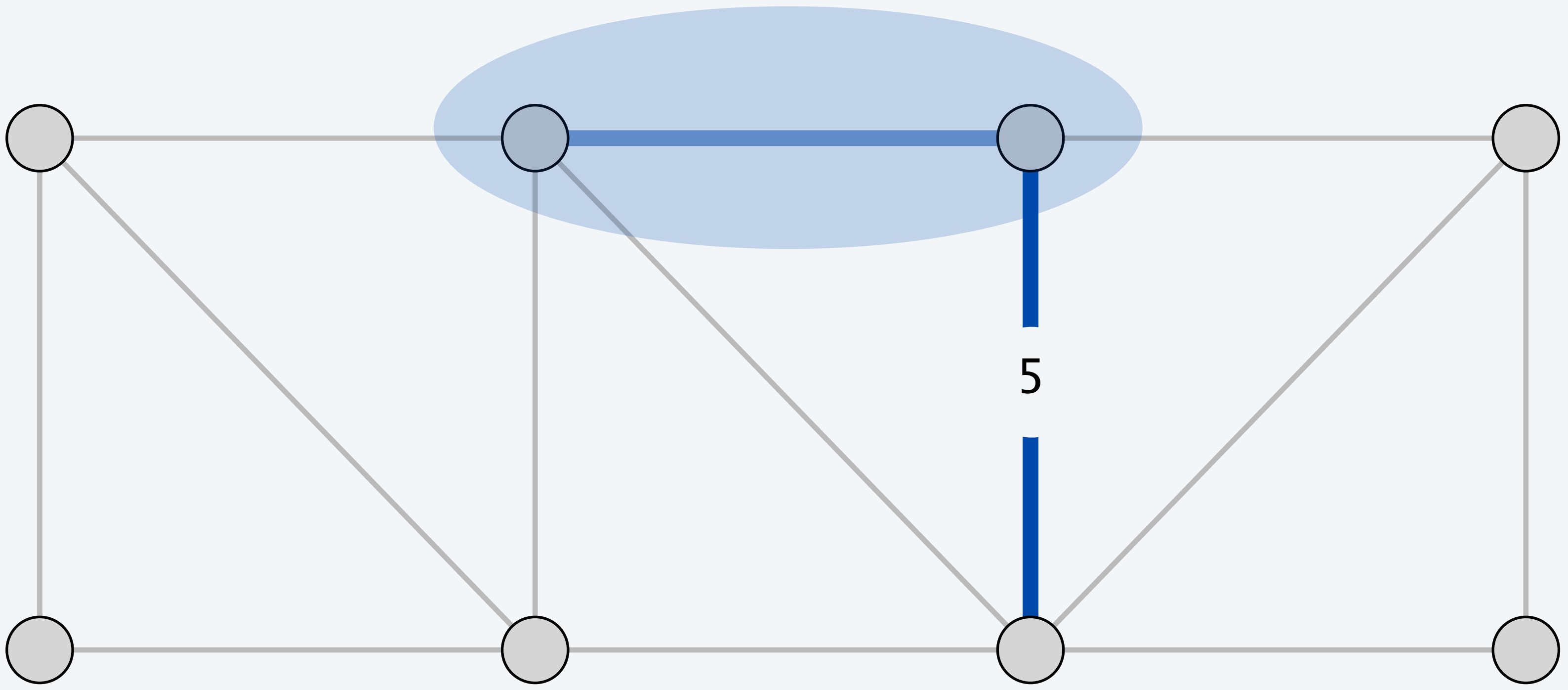


Prim's algorithm demo

Initialize S = any node, $T = \varnothing$.

Repeat $n - 1$ times:

- Add to T a min-weight edge with one endpoint in S.
- Add new node to S.

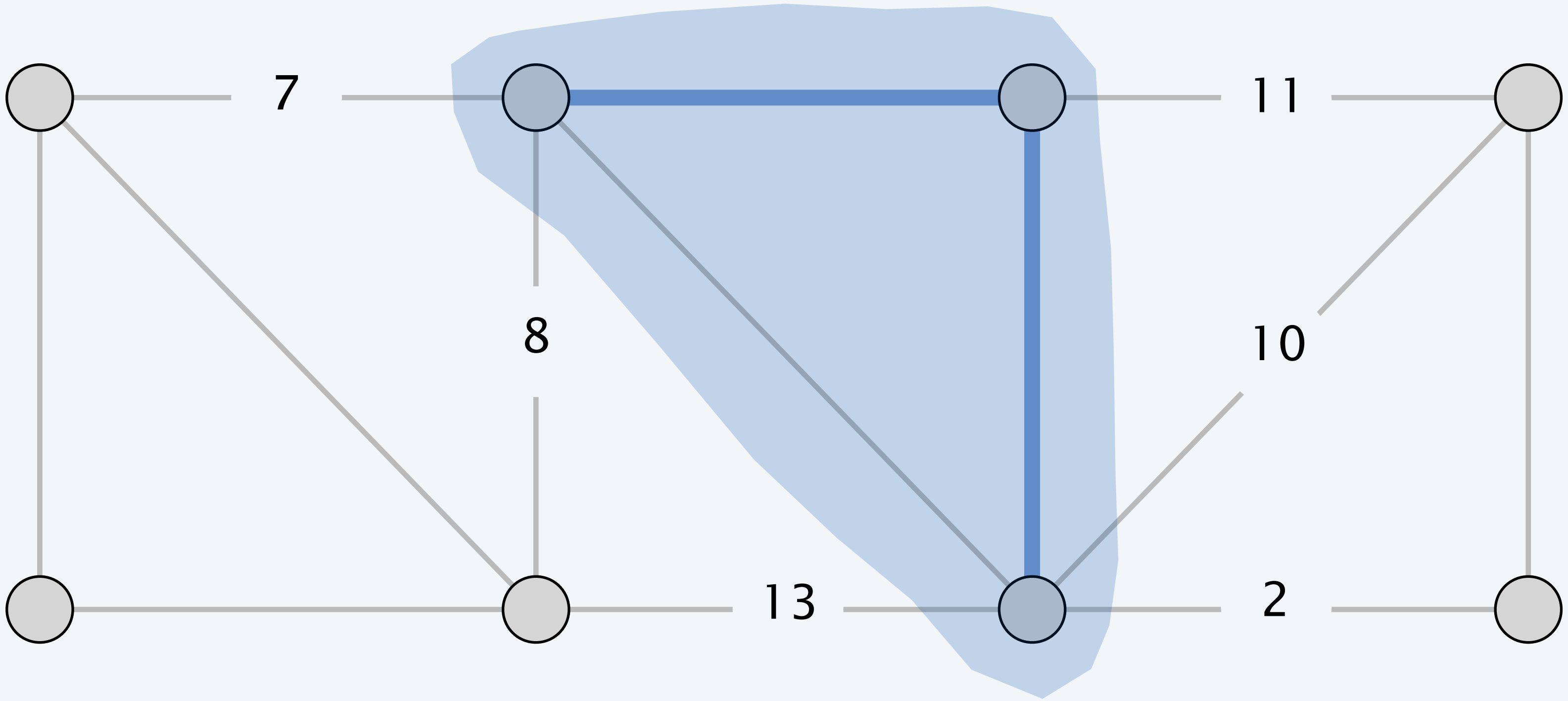


Prim's algorithm demo

Initialize S = any node, $T = \varnothing$.

Repeat $n - 1$ times:

- Add to T a min-weight edge with one endpoint in S.
- Add new node to S.

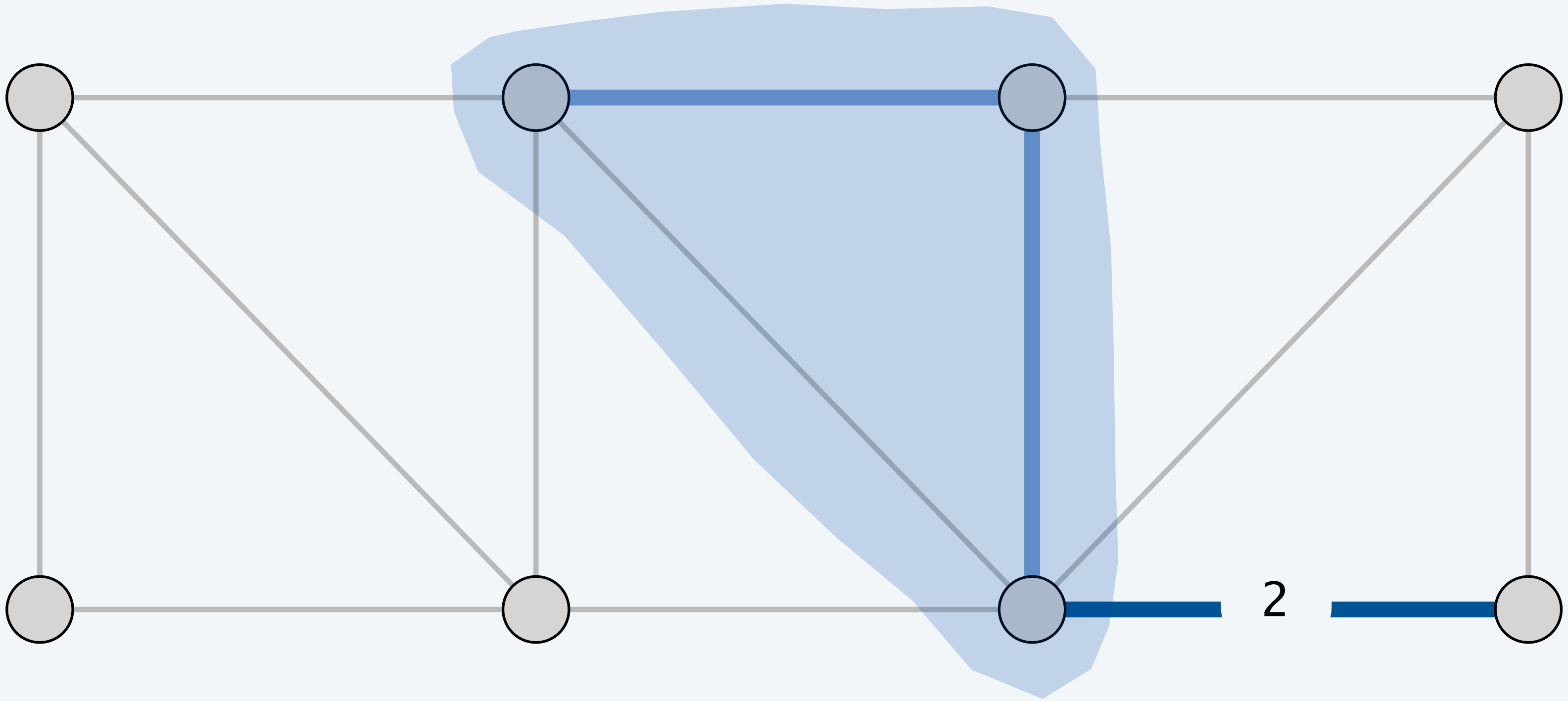


Prim's algorithm demo

Initialize S = any node, $T = \varnothing$.

Repeat $n - 1$ times:

- Add to T a min-weight edge with one endpoint in S.
- Add new node to S.

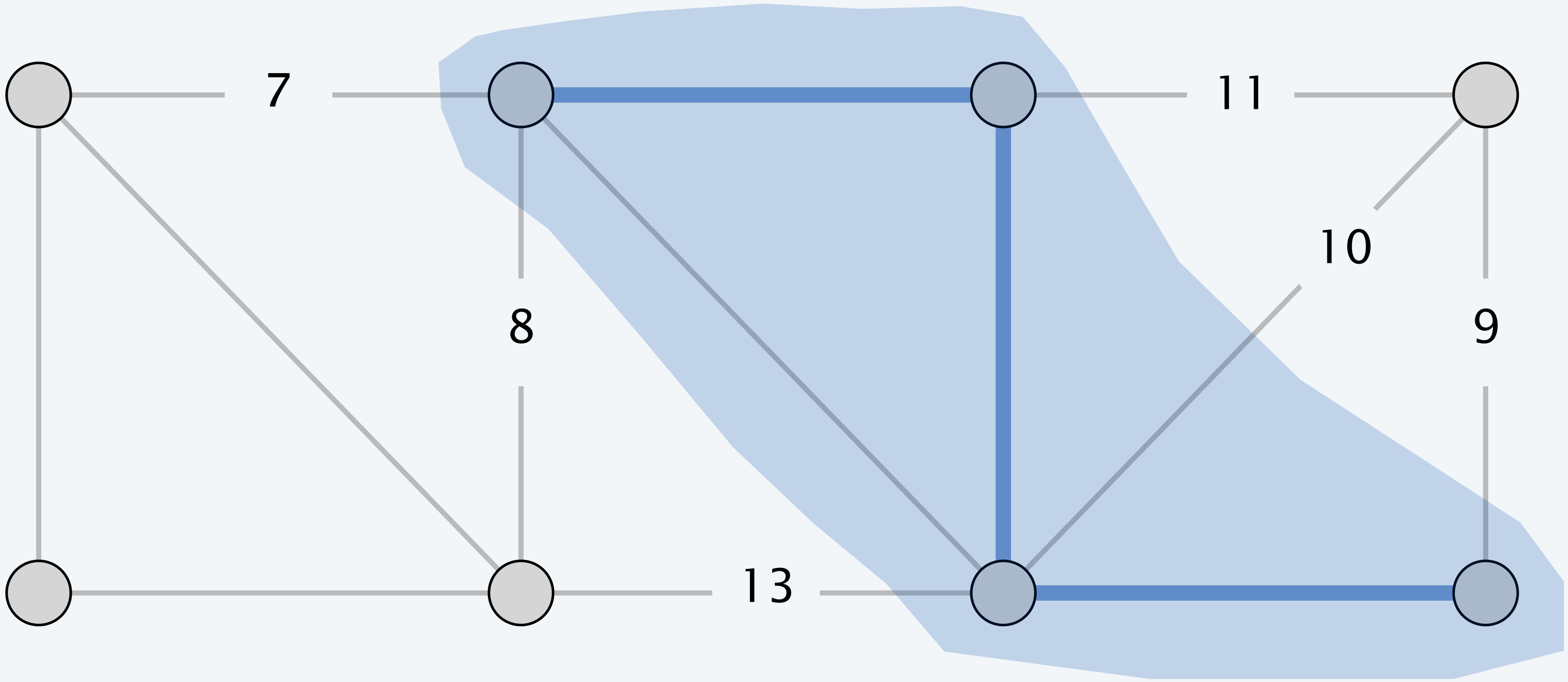


Prim's algorithm demo

Initialize S = any node, $T = \varnothing$.

Repeat $n - 1$ times:

- Add to T a min-weight edge with one endpoint in S.
- Add new node to S.

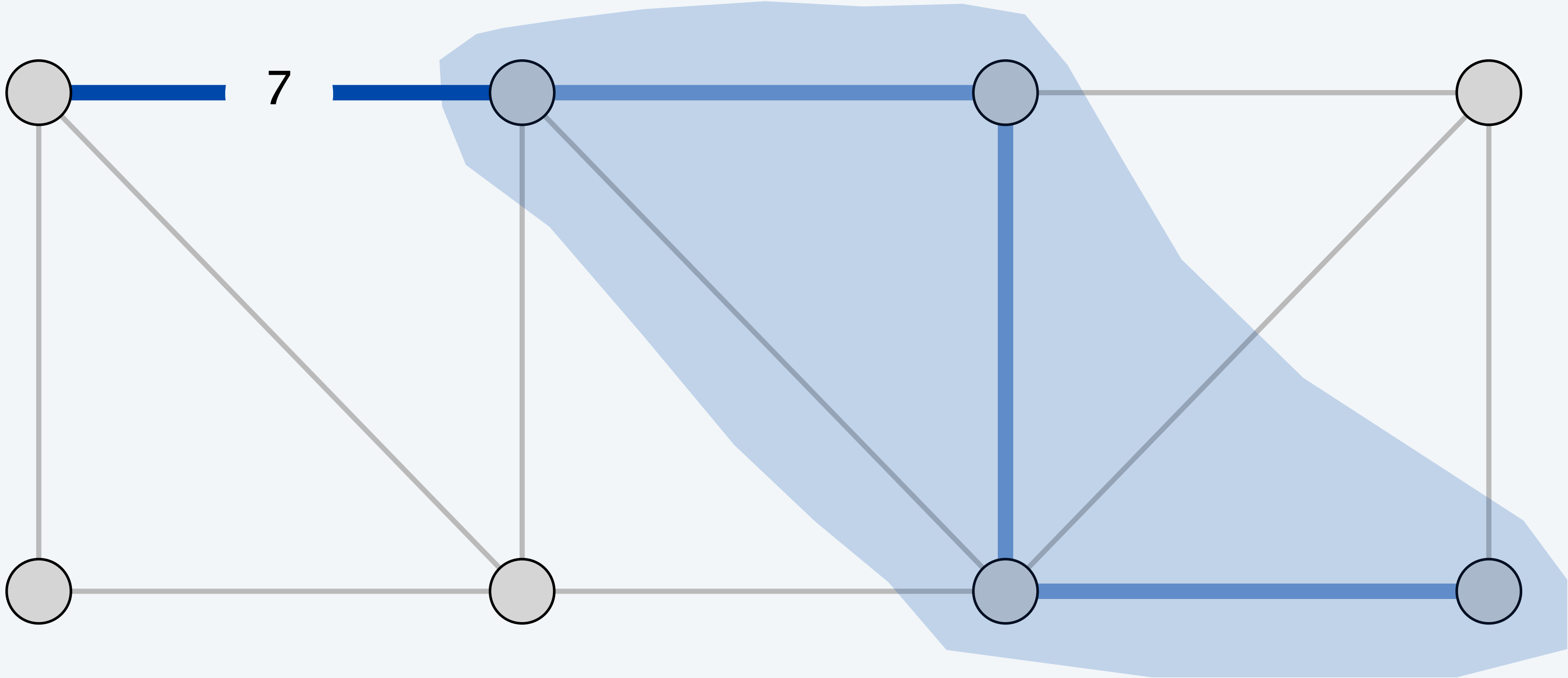


Prim's algorithm demo

Initialize S = any node, $T = \varnothing$.

Repeat $n - 1$ times:

- Add to T a min-weight edge with one endpoint in S.
- Add new node to S.

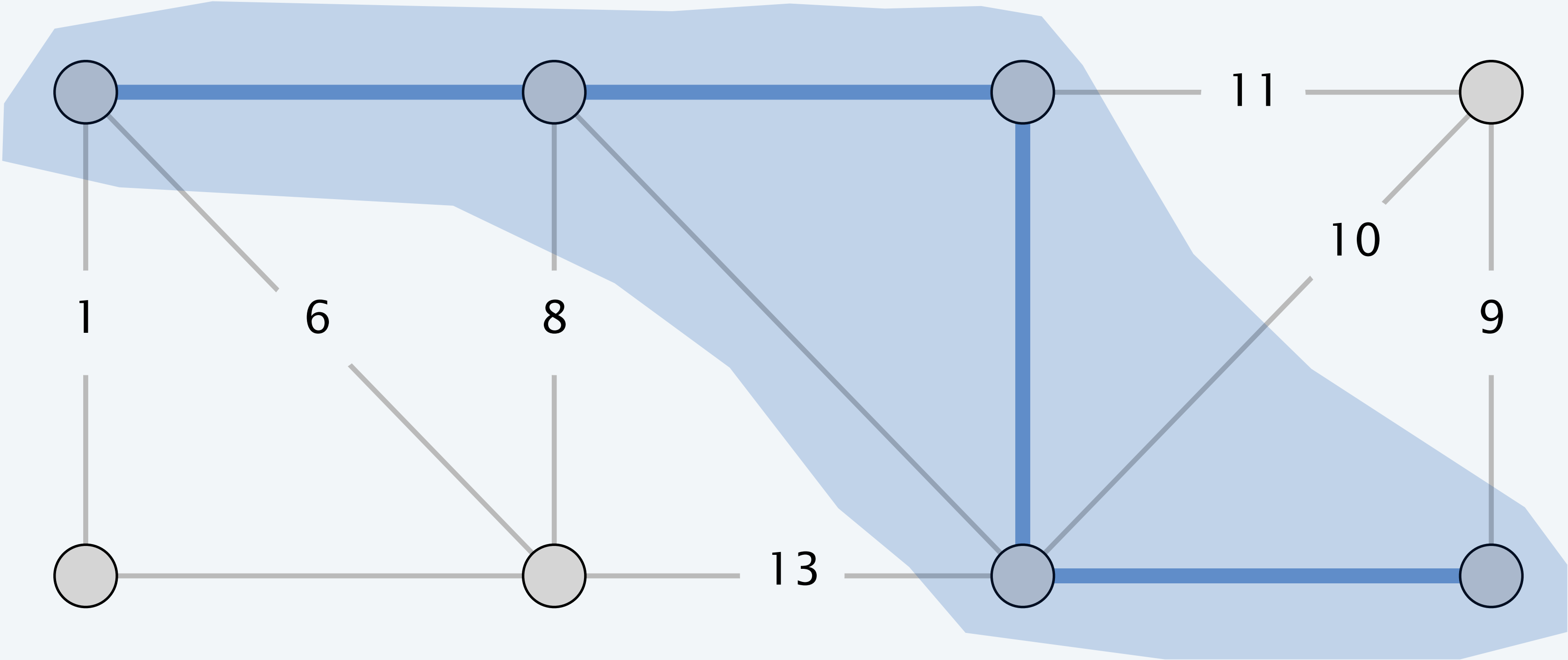


Prim's algorithm demo

Initialize S = any node, $T = \varnothing$.

Repeat $n - 1$ times:

- Add to T a min-weight edge with one endpoint in S.
- Add new node to S.

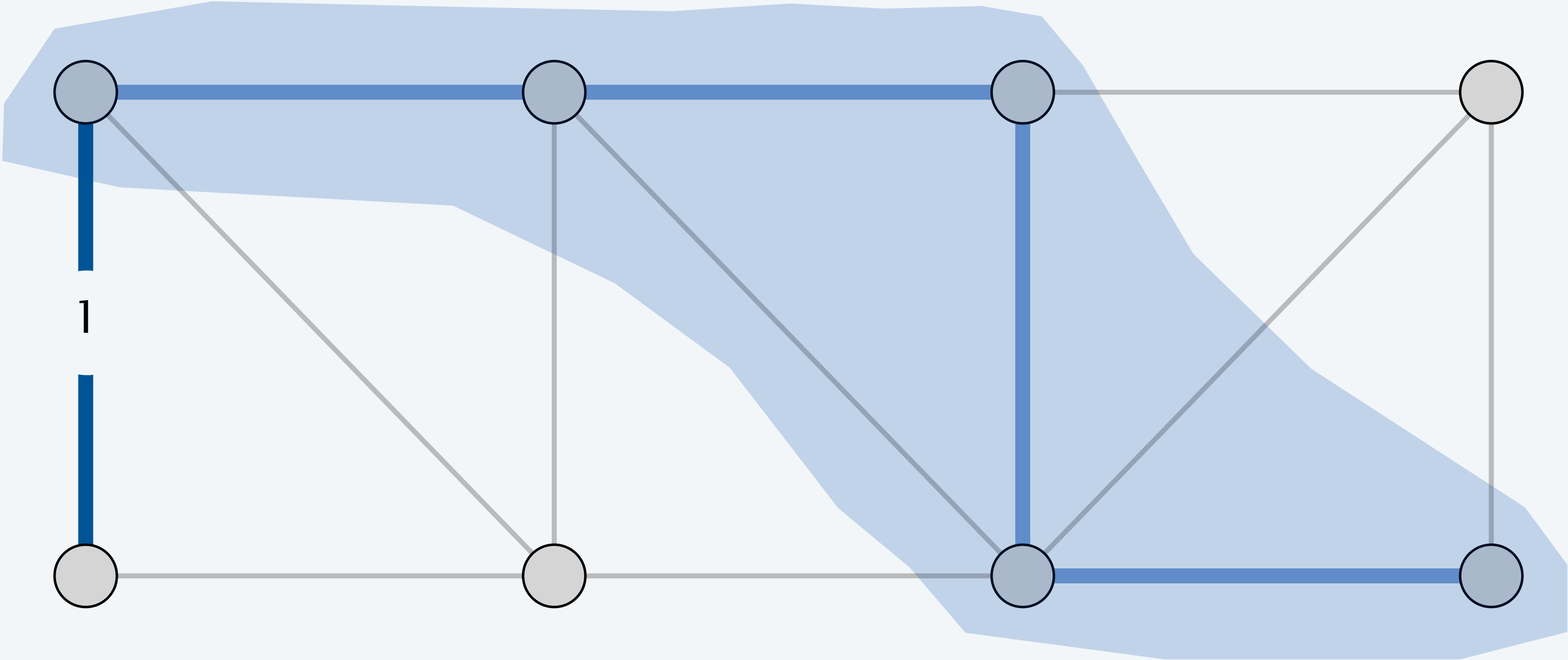


Prim's algorithm demo

Initialize S = any node, $T = \varnothing$.

Repeat $n - 1$ times:

- Add to T a min-weight edge with one endpoint in S.
- Add new node to S.

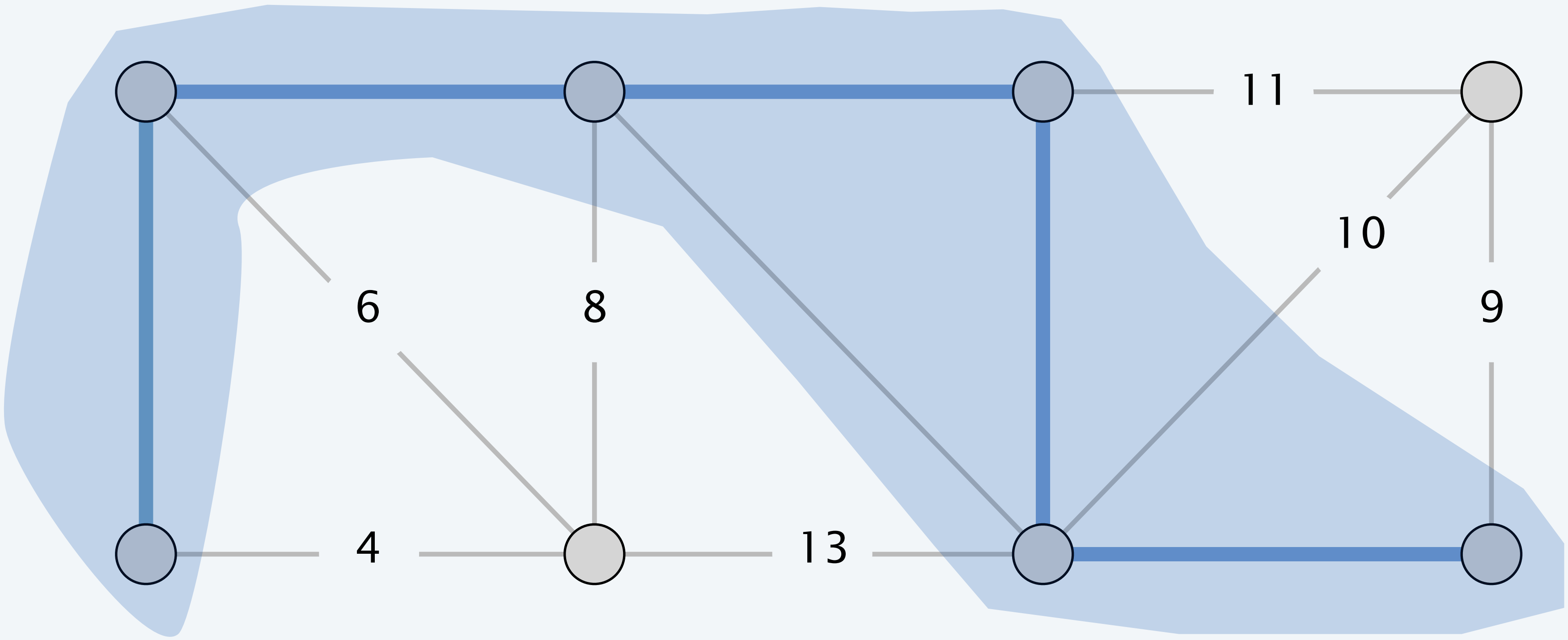


Prim's algorithm demo

Initialize S = any node, $T = \varnothing$.

Repeat $n - 1$ times:

- Add to T a min-weight edge with one endpoint in S.
- Add new node to S.

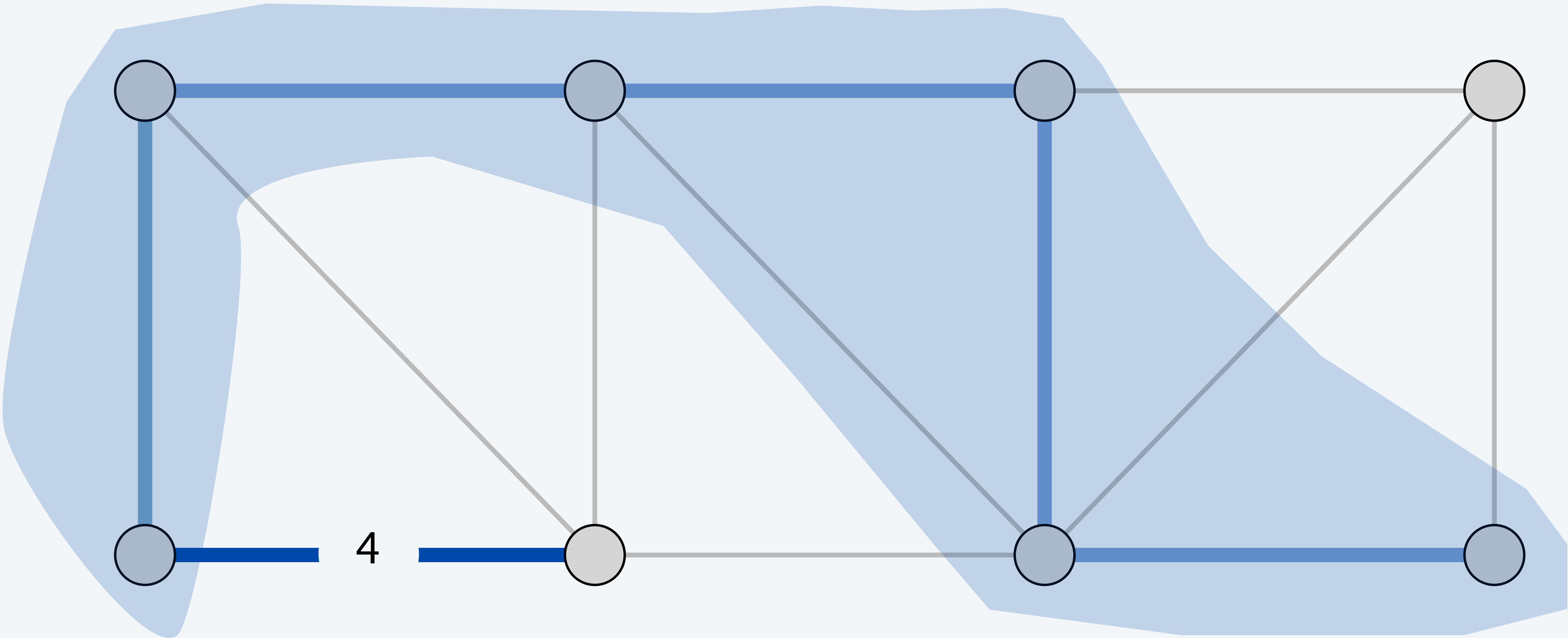


Prim's algorithm demo

Initialize S = any node, $T = \varnothing$.

Repeat $n - 1$ times:

- Add to T a min-weight edge with one endpoint in S.
- Add new node to S.

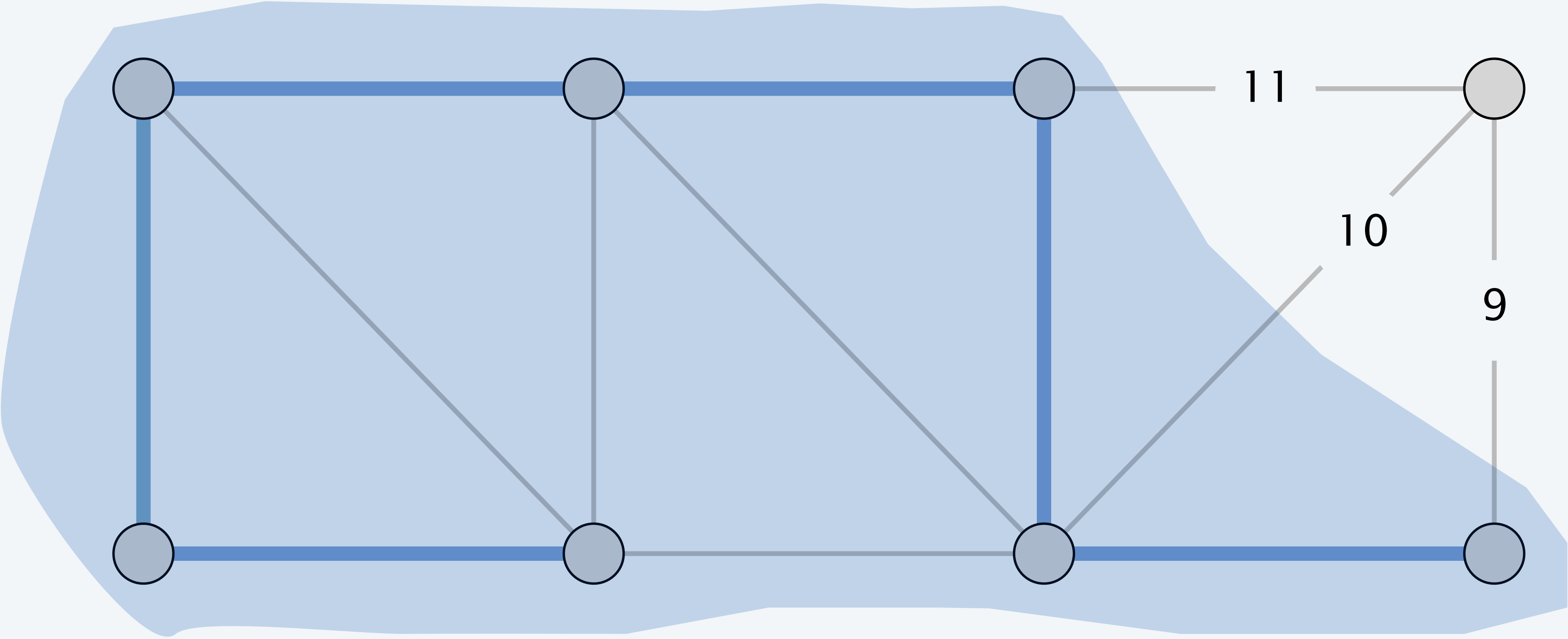


Prim's algorithm demo

Initialize S = any node, $T = \varnothing$.

Repeat $n - 1$ times:

- Add to T a min-weight edge with one endpoint in S.
- Add new node to S.

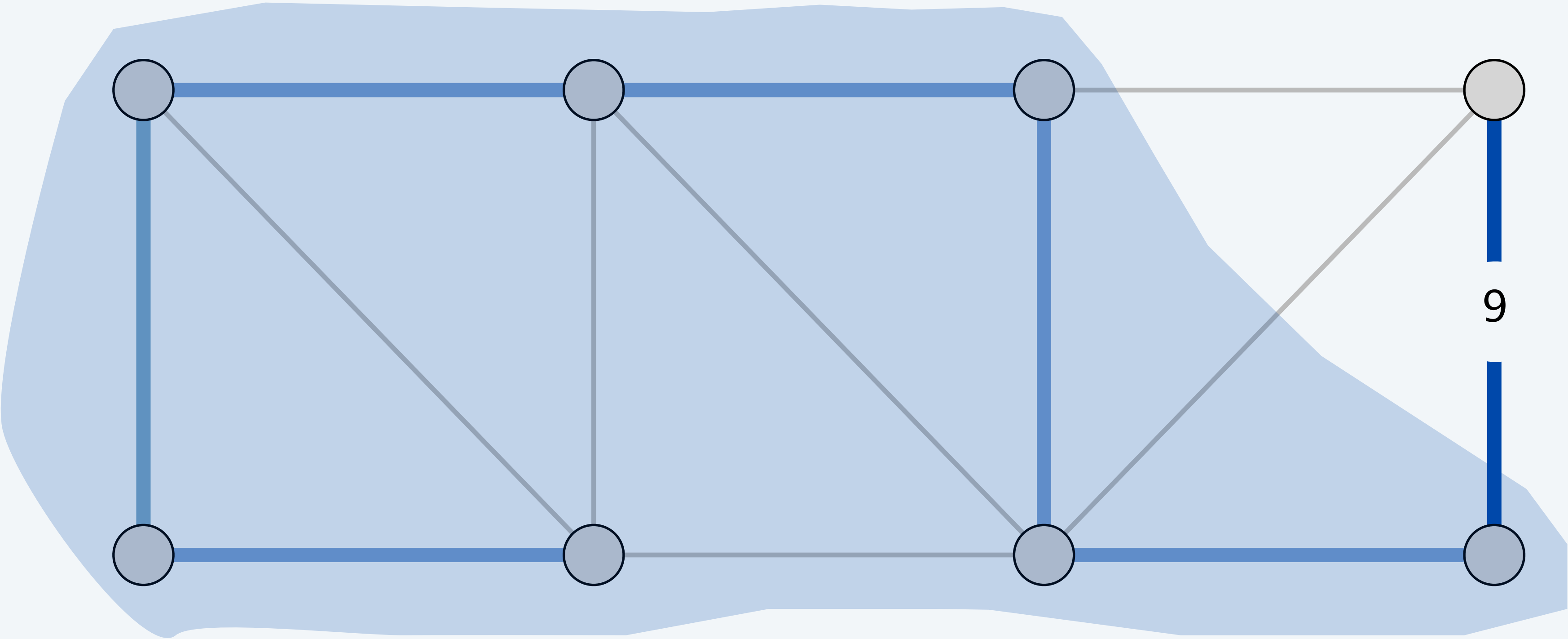


Prim's algorithm demo

Initialize S = any node, $T = \varnothing$.

Repeat $n - 1$ times:

- Add to T a min-weight edge with one endpoint in S.
- Add new node to S.

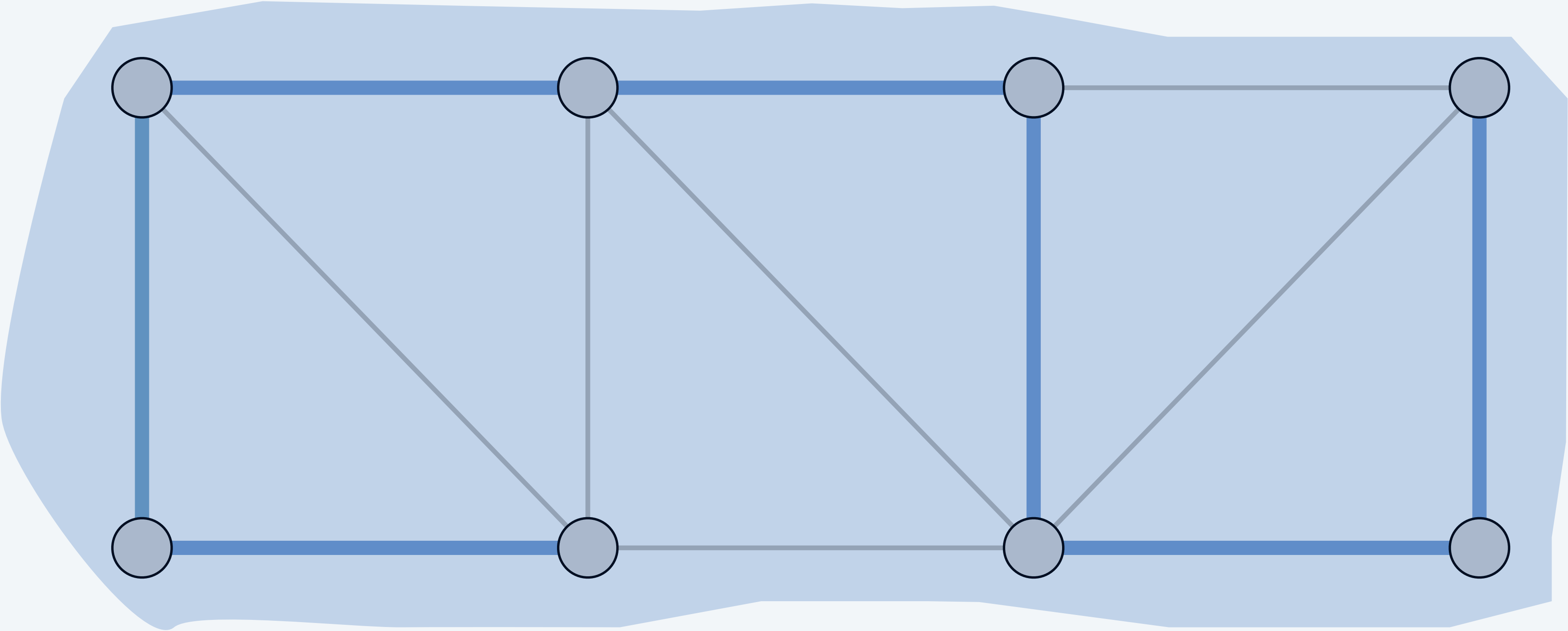


Prim's algorithm demo

Initialize S = any node, $T = \varnothing$.

Repeat $n - 1$ times:

- Add to T a min-weight edge with one endpoint in S.
- Add new node to S.



Prim's algorithm demo

Initialize S = any node, $T = \varnothing$.

Repeat $n - 1$ times:

- Add to T a min-weight edge with one endpoint in S.
- Add new node to S.