Prim's Algorithm

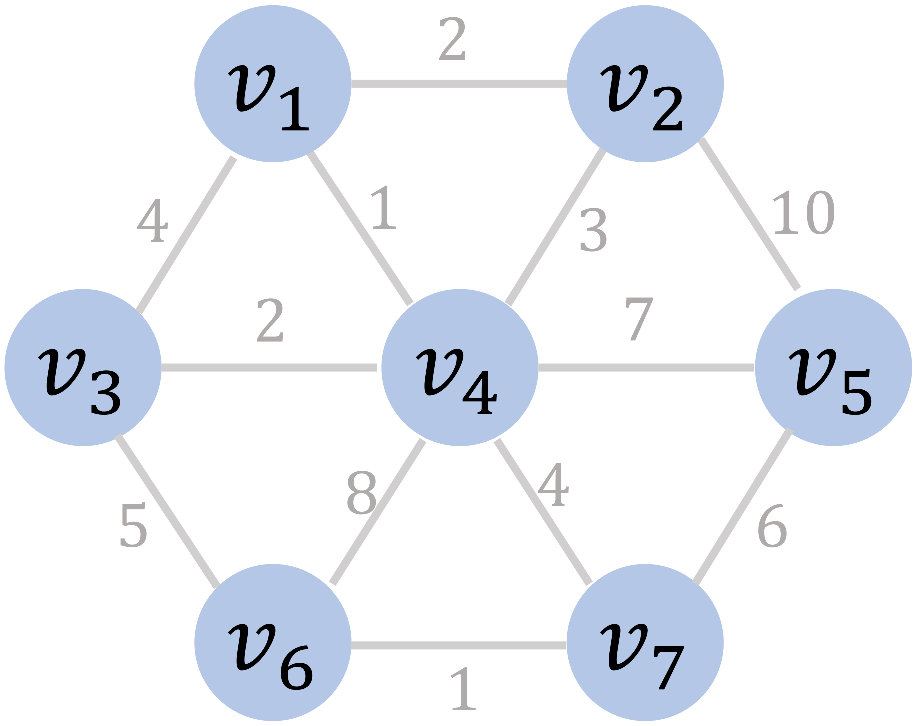
Shusen Wang

Prim's Algorithm

Basic idea: Grow the tree in successive stages.

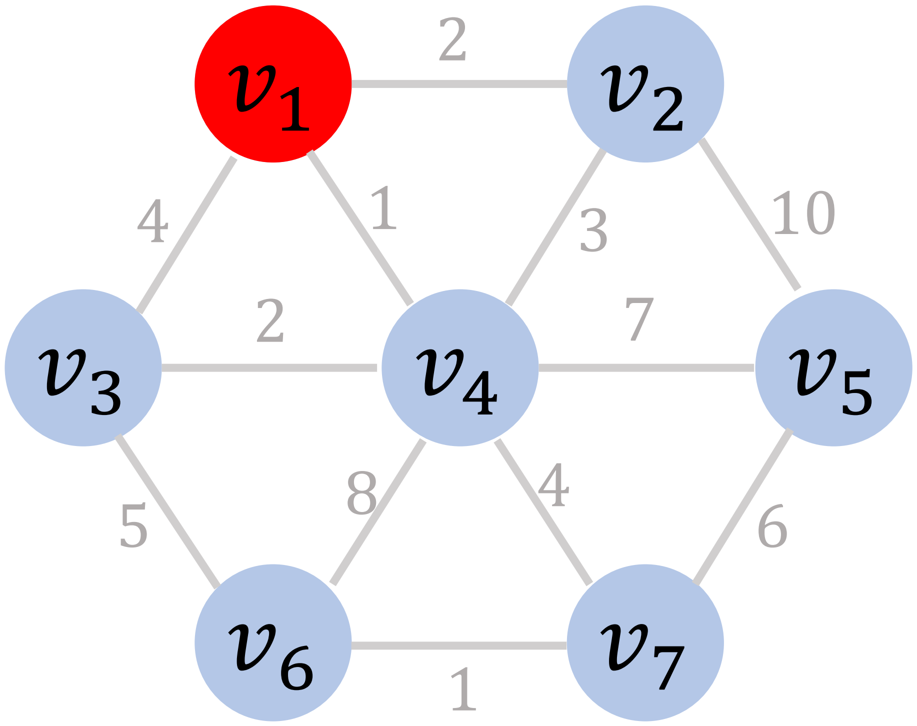
- Initially, the tree has one vertex and no edge.
- In each iteration, add one vertex and one edge to the tree.
- Throughout, maintain the properties of trees:
  - Connectivity.
  - No cycle.
- The algorithm runs in $|\mathcal{V}|$ iterations.

Initial State



$\mathcal{U}$: vertices of spanning tree

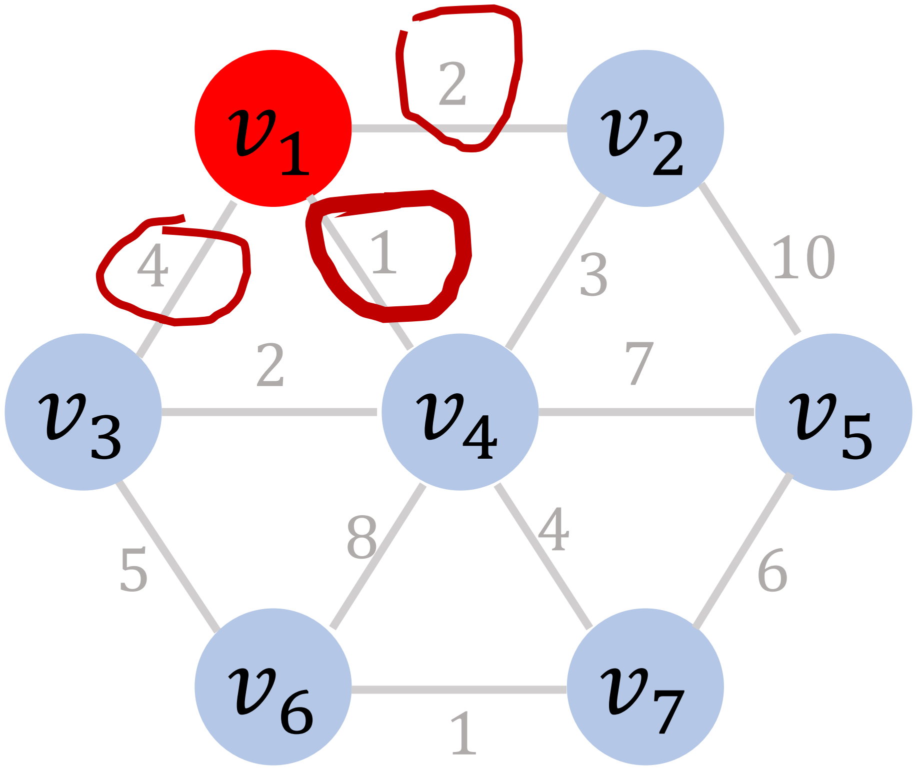
Iteration 1



- Pick any vertex in the graph.
- Maybe pick v_1.
- Add v_1 to $\mathcal{U}$.

$\mathcal{U} = \{v_1\}$

Iteration 2



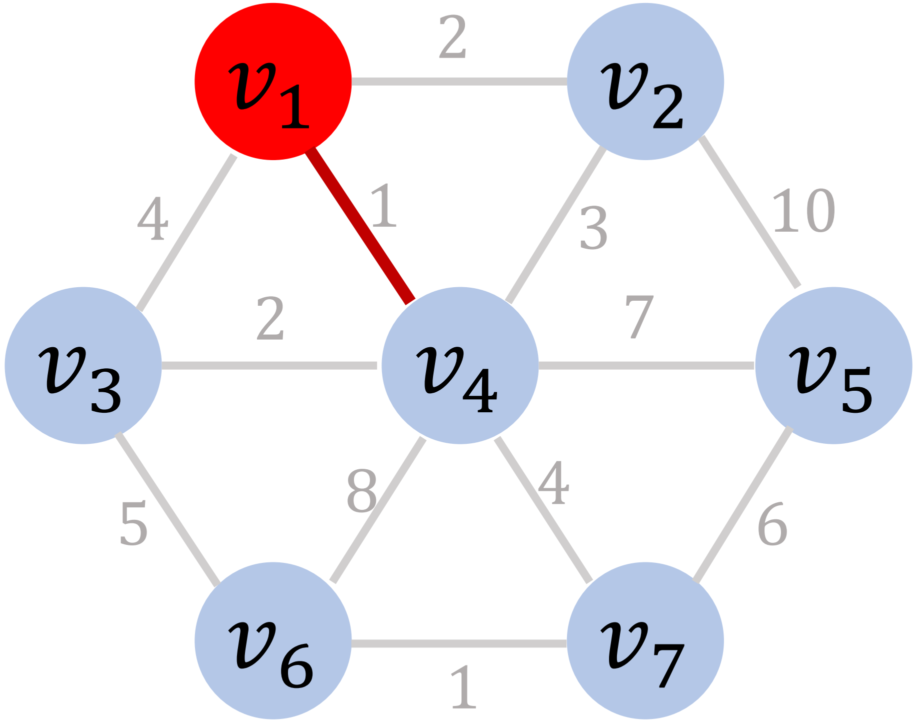
$\mathcal{U} = \{v_1\}$

- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

$$e_{1,2}, e_{1,3}, e_{1,4}.$$

- Among them, $e_{1,4}$ has the smallest weight.

Iteration 2



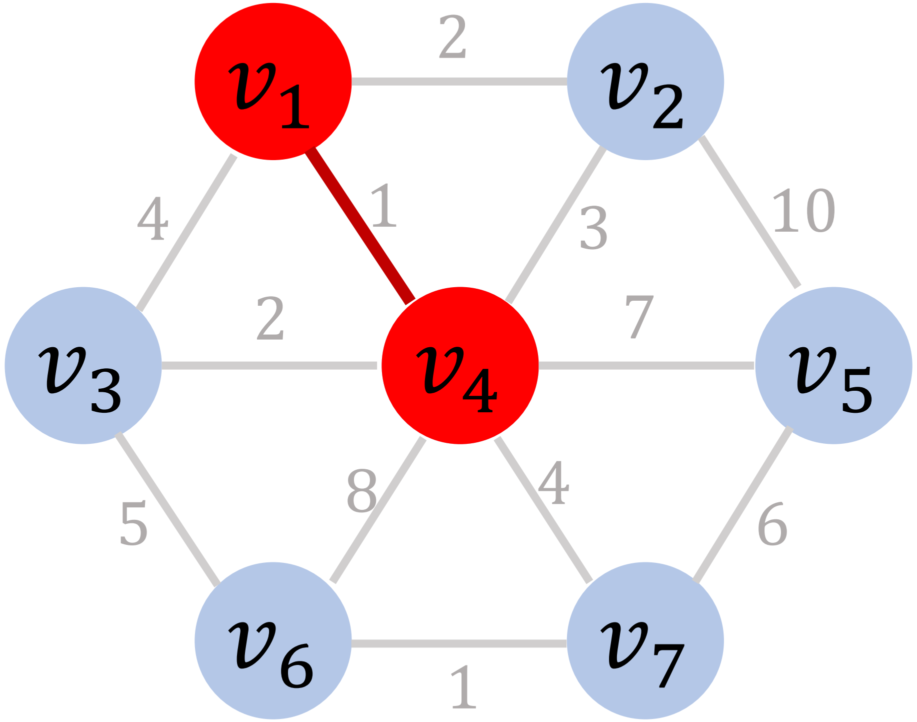
$\mathcal{U} = \{v_1\}$

- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

$$e_{1,2}, e_{1,3}, e_{1,4}.$$

- Among them, $e_{1,4}$ has the smallest weight.

Iteration 2



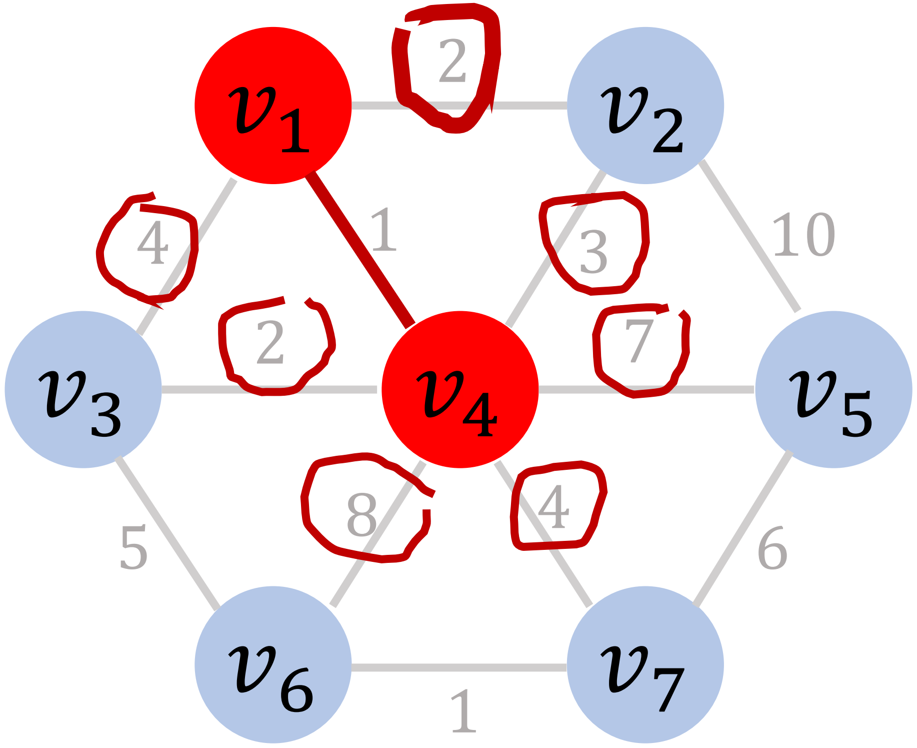
$\mathcal{U} = \{v_1, v_4\}$

- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

$$e_{1,2}, e_{1,3}, e_{1,4}.$$

- Among them, $e_{1,4}$ has the smallest weight.
- Add v_4 to $\mathcal{U}$.

Iteration 3



$\mathcal{U} = \{v_1, v_4\}$

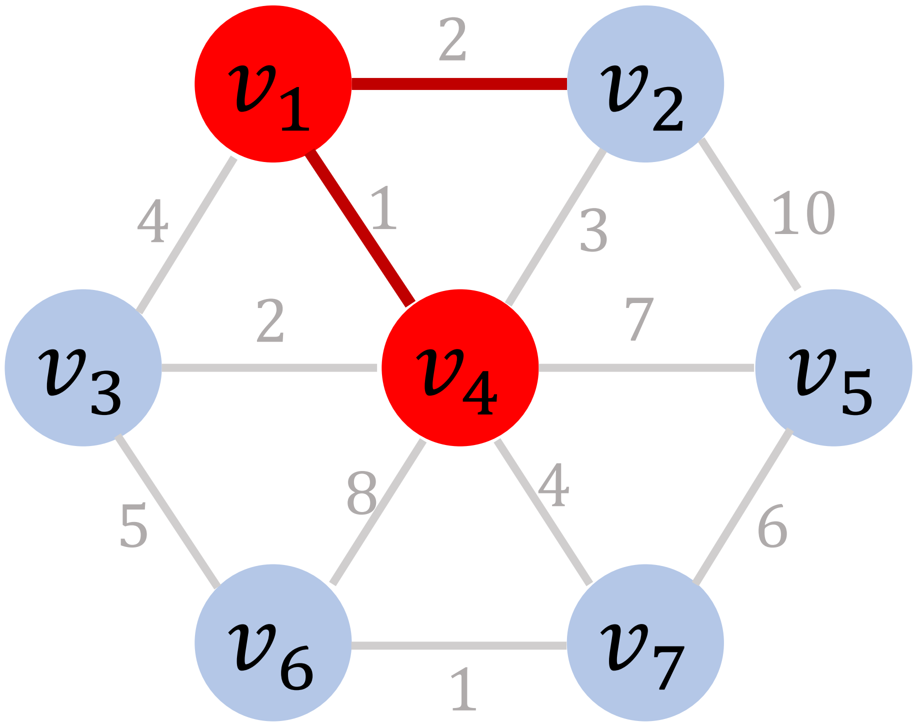
- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

$$e_{1,2}, e_{1,3},$$

$$e_{4,2}, e_{4,3}, e_{4,5}, e_{4,6}, e_{4,7}.$$

- Among them, $e_{1,2}$ has the smallest weight.

Iteration 3



$\mathcal{U} = \{v_1, v_4\}$

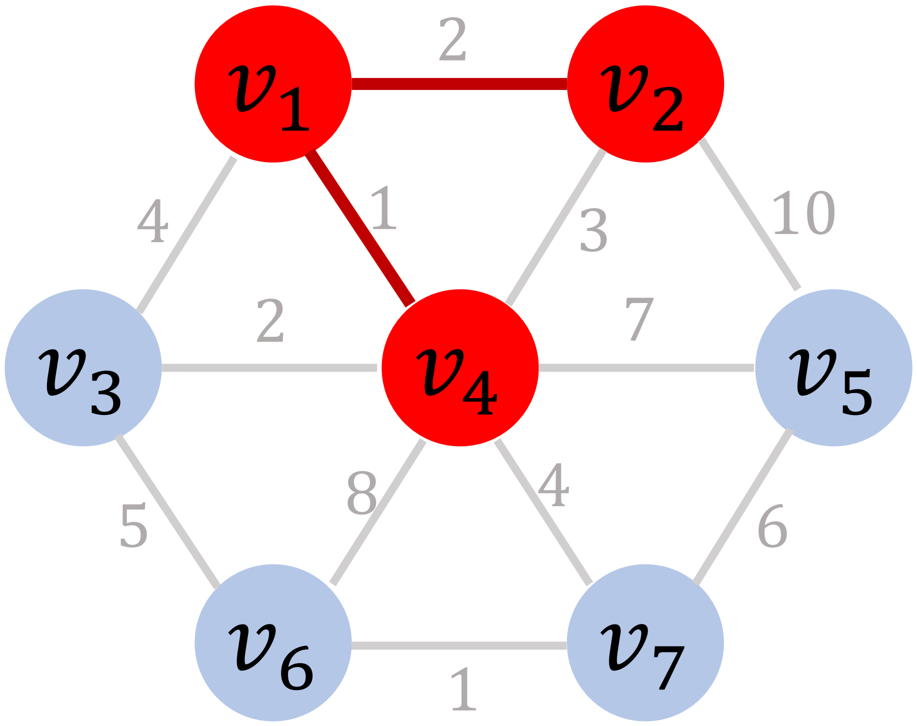
- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

$$e_{1,2}, e_{1,3},$$

$$e_{4,2}, e_{4,3}, e_{4,5}, e_{4,6}, e_{4,7}.$$

- Among them, $e_{1,2}$ has the smallest weight.

Iteration 3



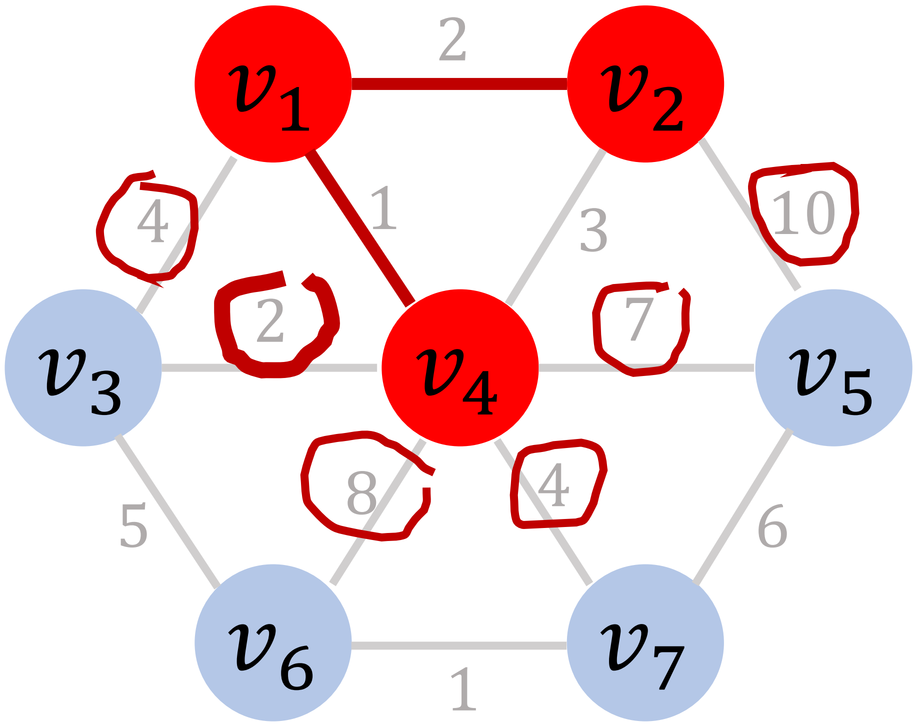
$\mathcal{U} = \{v_1, v_4, v_2\}$

- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

  $e_{1,2}, e_{1,3},$

  $e_{4,2}, e_{4,3}, e_{4,5}, e_{4,6}, e_{4,7}.$
- Among them, $e_{1,2}$ has the smallest weight.
- Add v_2 to $\mathcal{U}$.

Iteration 4



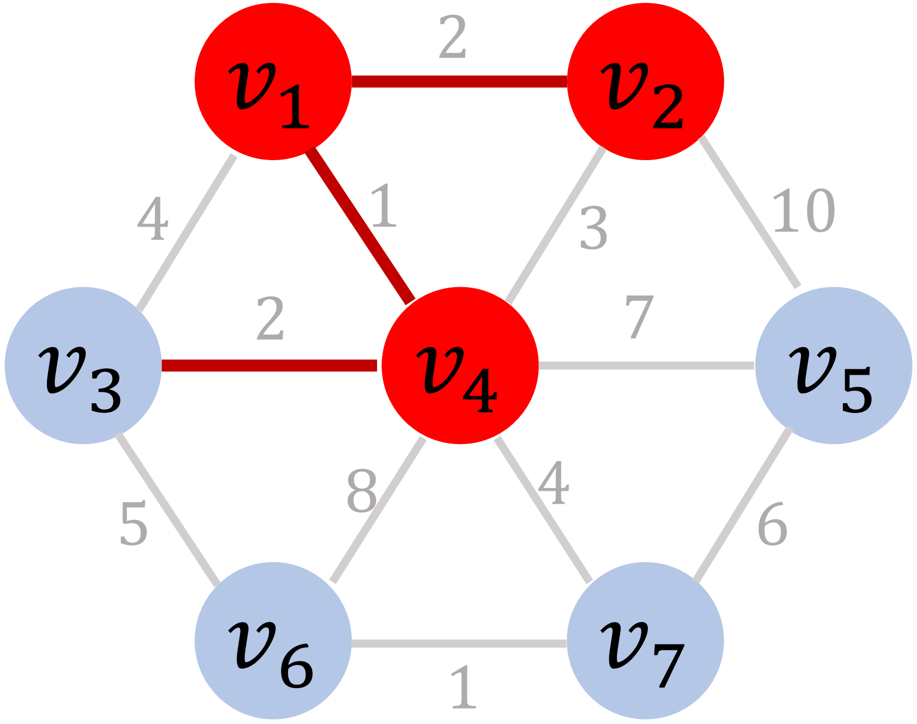
$\mathcal{U} = \{v_1, v_4, v_2\}$

- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

$$e_{1,3},$$
$$e_{4,3}, e_{4,5}, e_{4,6}, e_{4,7},$$
$$e_{2,5}.$$

- Among them, $e_{4,3}$ has the smallest weight.

Iteration 4



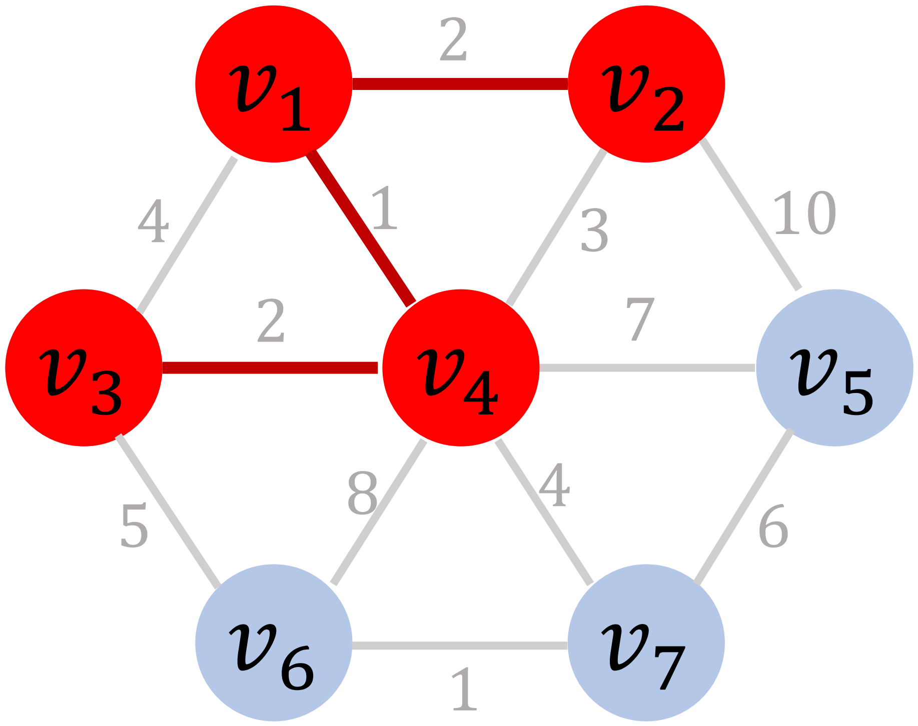
$\mathcal{U} = \{v_1, v_4, v_2\}$

- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

$$e_{1,3},$$
$$e_{4,3}, e_{4,5}, e_{4,6}, e_{4,7},$$
$$e_{2,5}.$$

- Among them, $e_{4,3}$ has the smallest weight.

Iteration 4



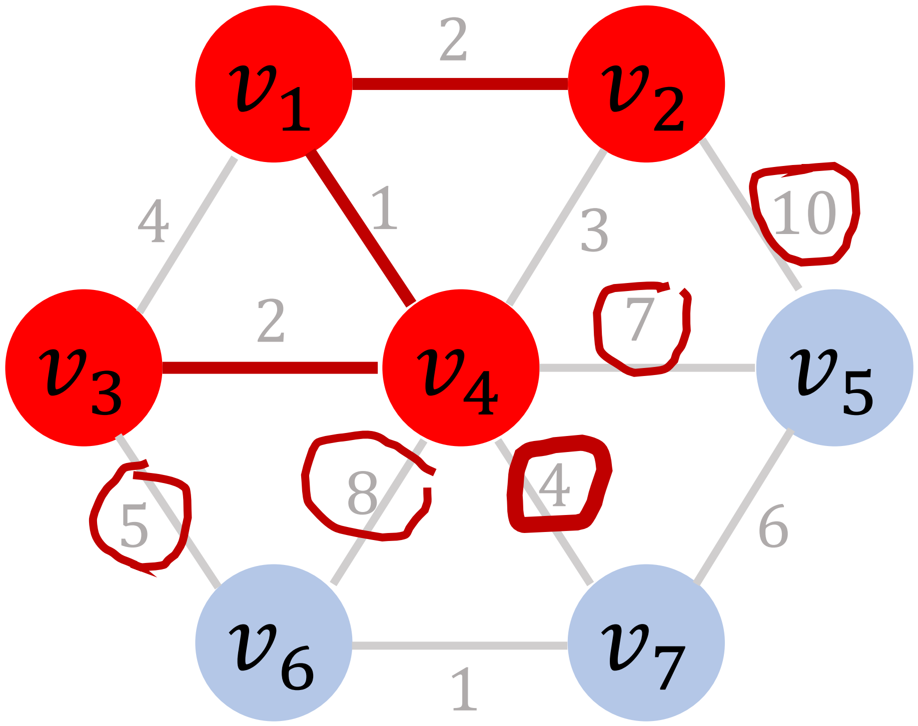
$\mathcal{U} = \{v_1, v_4, v_2, v_3\}$

- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

$$e_{1,3},$$
$$e_{4,3}, e_{4,5}, e_{4,6}, e_{4,7},$$
$$e_{2,5}.$$

- Among them, $e_{4,3}$ has the smallest weight.
- Add v_3 to $\mathcal{U}$.

Iteration 5



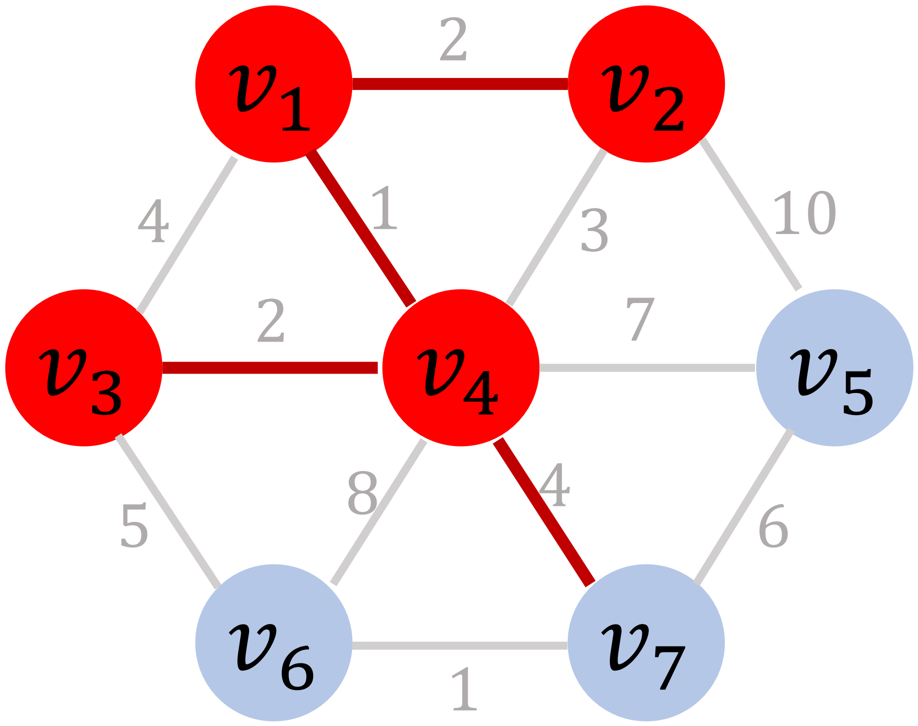
$\mathcal{U} = \{v_1, v_4, v_2, v_3\}$

- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

$$e_{4,5}, e_{4,6}, e_{4,7},$$
$$e_{2,5},$$
$$e_{3,6}.$$

- Among them, $e_{4,7}$ has the smallest weight.

Iteration 5



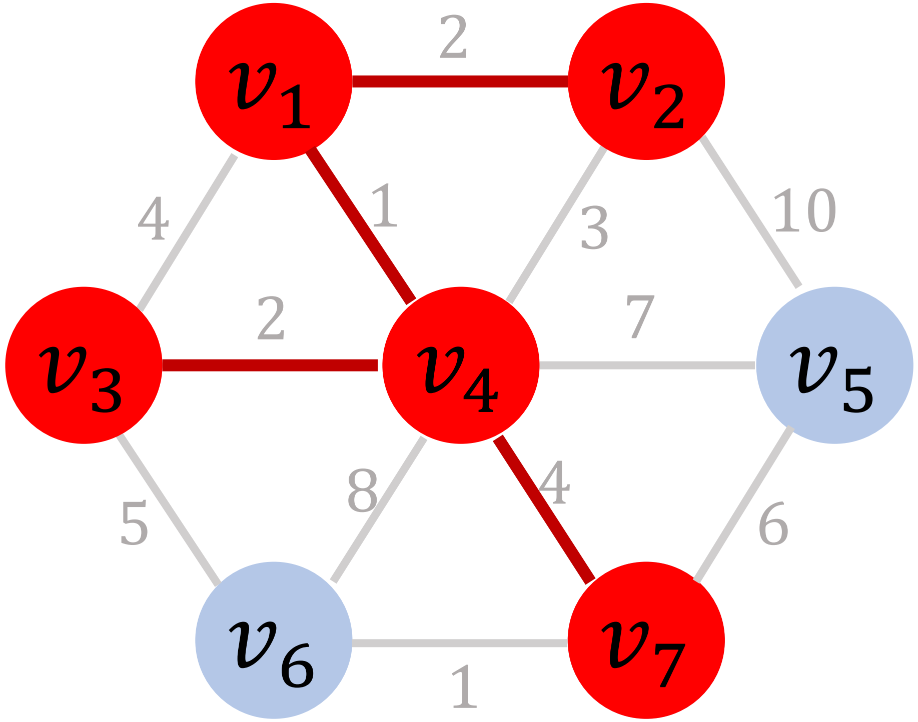
$\mathcal{U} = \{v_1, v_4, v_2, v_3\}$

- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

$$e_{4,5}, e_{4,6}, e_{4,7},$$
$$e_{2,5},$$
$$e_{3,6}.$$

- Among them, $e_{4,7}$ has the smallest weight.

Iteration 5



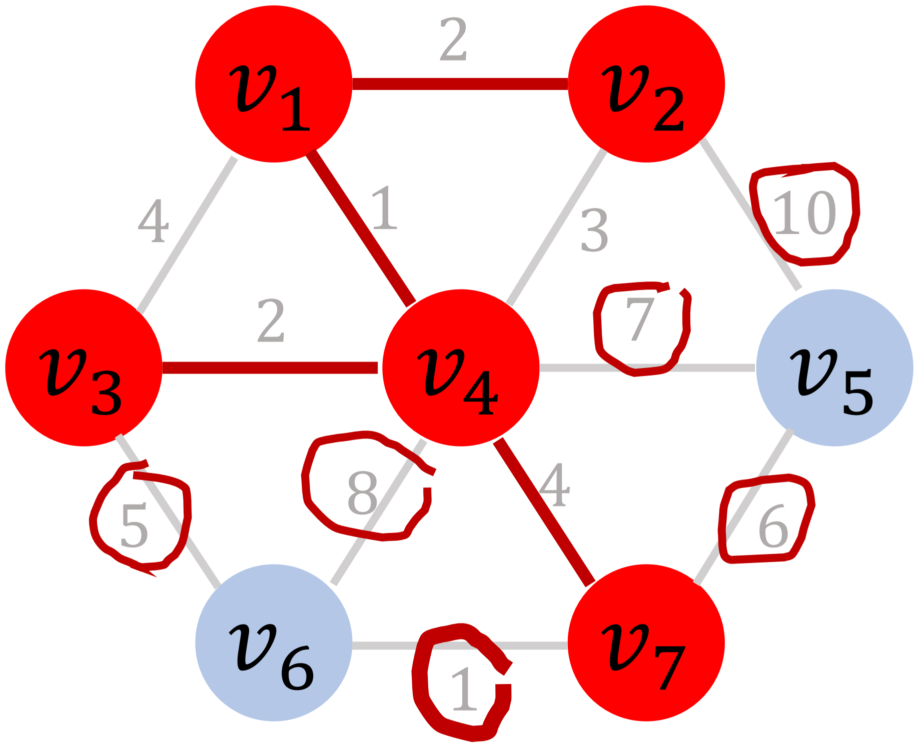
$\mathcal{U} = \{v_1, v_4, v_2, v_3, v_7\}$

- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

$$e_{4,5}, e_{4,6}, e_{4,7},$$
$$e_{2,5},$$
$$e_{3,6}.$$

- Among them, $e_{4,7}$ has the smallest weight.
- Add v_7 to $\mathcal{U}$.

Iteration 6



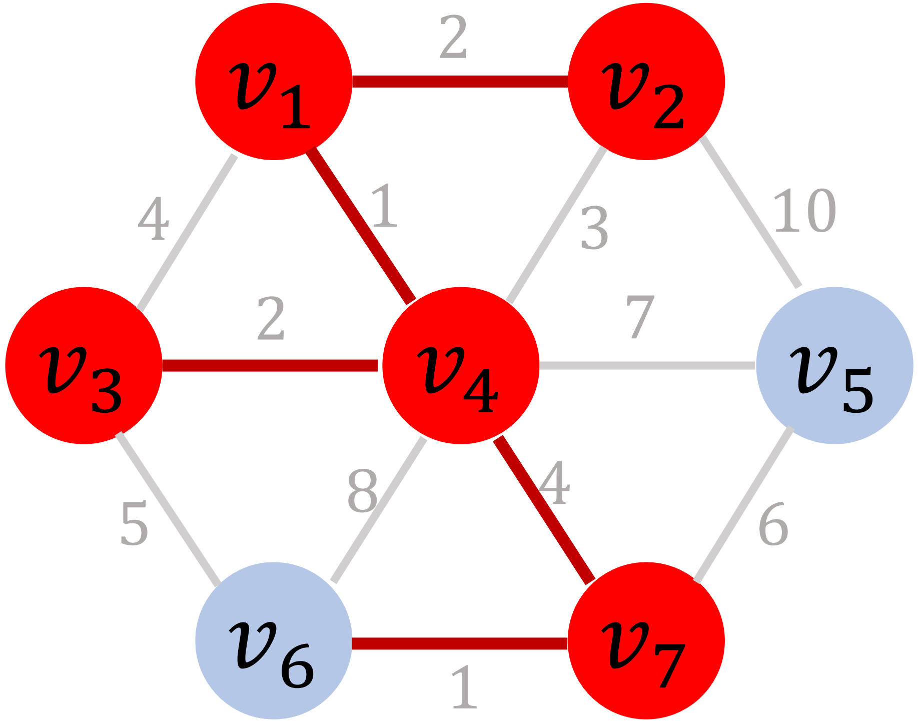
$\mathcal{U} = \{v_1, v_4, v_2, v_3, v_7\}$

- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

$$e_{4,5}, e_{4,6},$$
$$e_{2,5},$$
$$e_{3,6},$$
$$e_{75}, e_{7,6}.$$

- Among them, $e_{7,6}$ has the smallest weight.

Iteration 6



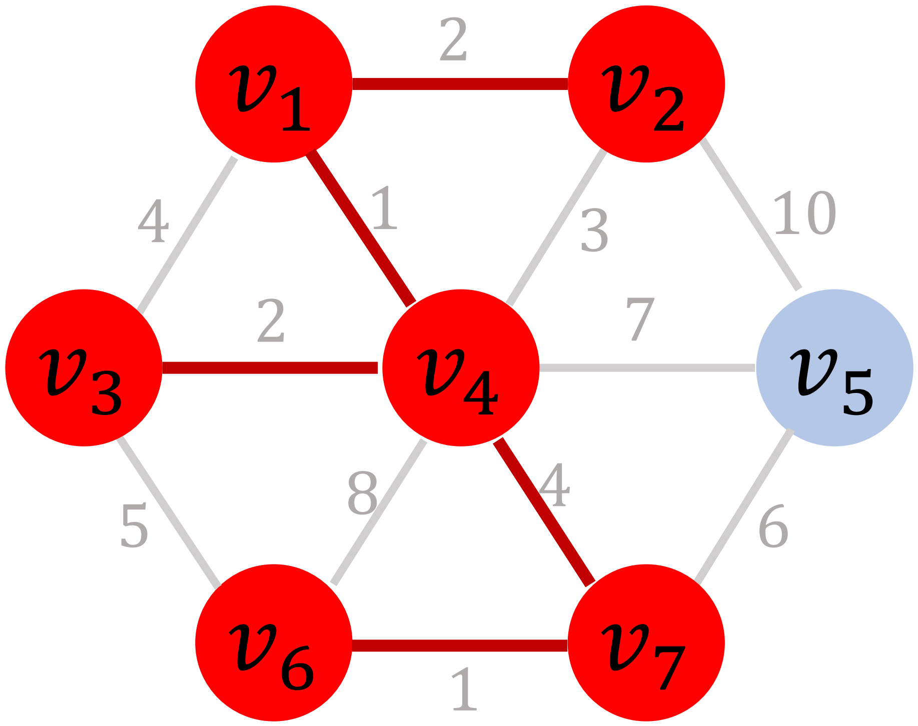
$\mathcal{U} = \{v_1, v_4, v_2, v_3, v_7\}$

- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

$$e_{4,5}, e_{4,6},$$
$$e_{2,5},$$
$$e_{3,6},$$
$$e_{75}, e_{7,6}.$$

- Among them, $e_{7,6}$ has the smallest weight.

Iteration 6



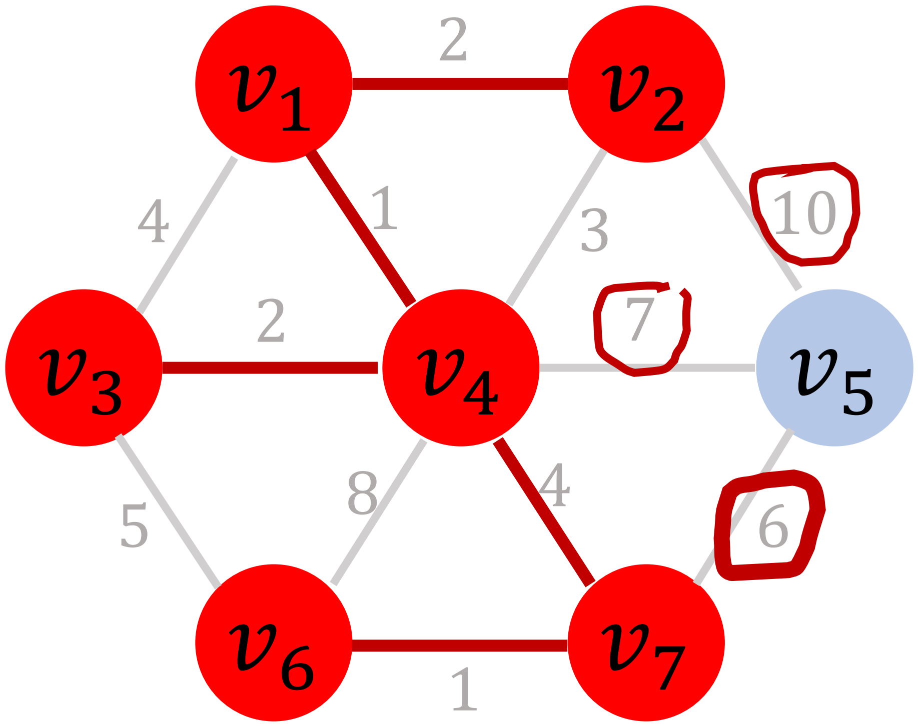
$\mathcal{U} = \{v_1, v_4, v_2, v_3, v_7, v_6\}$

- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

$$e_{4,5}, e_{4,6},$$
$$e_{2,5},$$
$$e_{3,6},$$
$$e_{75}, e_{7,6}.$$

- Among them, $e_{7,6}$ has the smallest weight.
- Add v_6 to $\mathcal{U}$.

Iteration 7



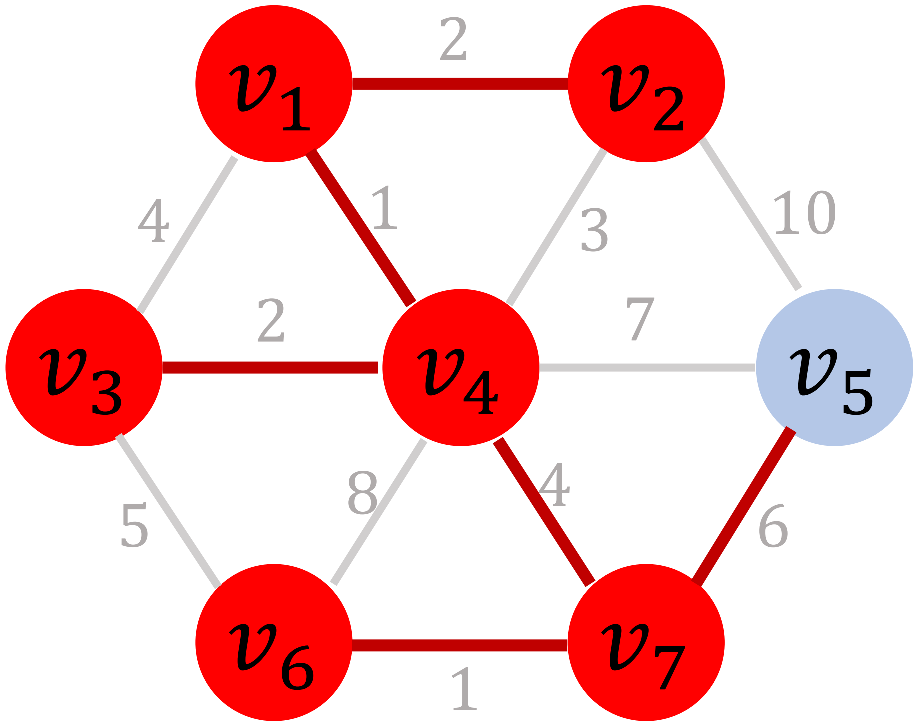
$\mathcal{U} = \{v_1, v_4, v_2, v_3, v_7, v_6\}$

- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

$$e_{4,5},$$
$$e_{2,5},$$
$$e_{7,5}.$$

- Among them, $e_{7,5}$ has the smallest weight.

Iteration 7



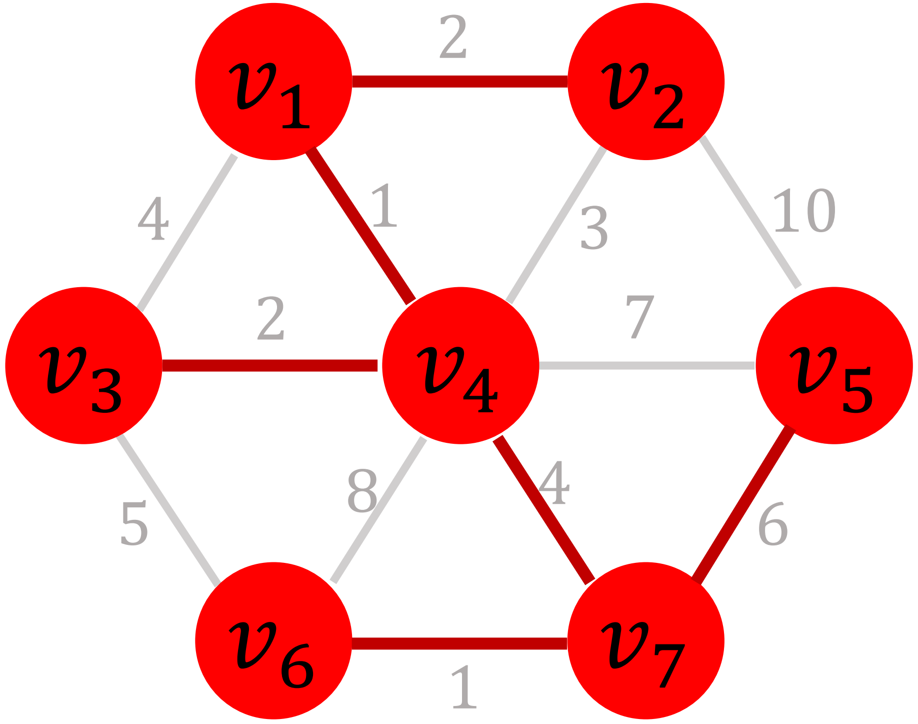
$\textcolor{red}{\mathcal{U}} = \{v_1, v_4, v_2, v_3, v_7, v_6\}$

- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

$$e_{4,5},$$
$$e_{2,5},$$
$$e_{7,5}.$$

- Among them, $e_{7,5}$ has the smallest weight.

Iteration 7



$\mathcal{U} = \{v_1, v_4, v_2, v_3, v_7, v_6, v_5\}$

- The edges connecting $\mathcal{U}$ to $\mathcal{V} \backslash \mathcal{U}$:

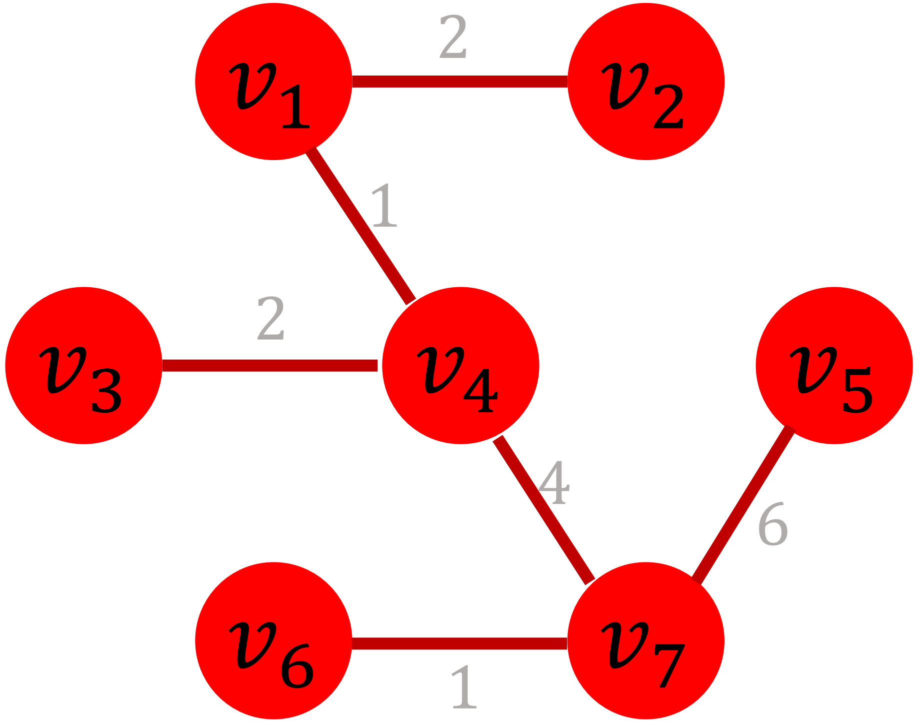
  $e_{4,5}$,

  $e_{2,5}$,

  $e_{7,5}$.

- Among them, $e_{7,5}$ has the smallest weight.
- Add v_5 to $\mathcal{U}$.

End of Procedure



$\mathcal{U} = \{v_1, v_4, v_2, v_3, v_7, v_6, v_5\}$

- Now $\mathcal{U}=\mathcal{V}$. (All the vertices have been added to $\mathcal{U}$.)
- Return the tree.

Prim's Algorithm

1. Initially, the tree has one vertex and no edge.
2. Let set $\mathcal{U}$ contain the vertices of the tree.
3. Let set $\mathcal{T}$ be the empty set.

Prim's Algorithm

1. Initially, the tree has one vertex and no edge.
2. Let set $\mathcal{U}$ contain the vertices of the tree.
3. Let set $\mathcal{T}$ be the empty set.
4. In each iteration, select one vertex and one edge:
   - Define the set $\mathcal{S} = \{e_{xy} \in \mathcal{E} \mid x \in \mathcal{U}\ and\ \ y \notin \mathcal{U}\}$.
   - Select from $\mathcal{S}$ the edge (denote e_{uv}) that has the smallest weight.
   - Add edge e_{uv} to $\mathcal{T}$.
   - Add vertex v to $\mathcal{U}$.
5. Return the selected edges, $\mathcal{T}$.

Thank You!