

# Approximation Algorithms

## Lecture 6: $k$ -CENTER via Parametric Pruning

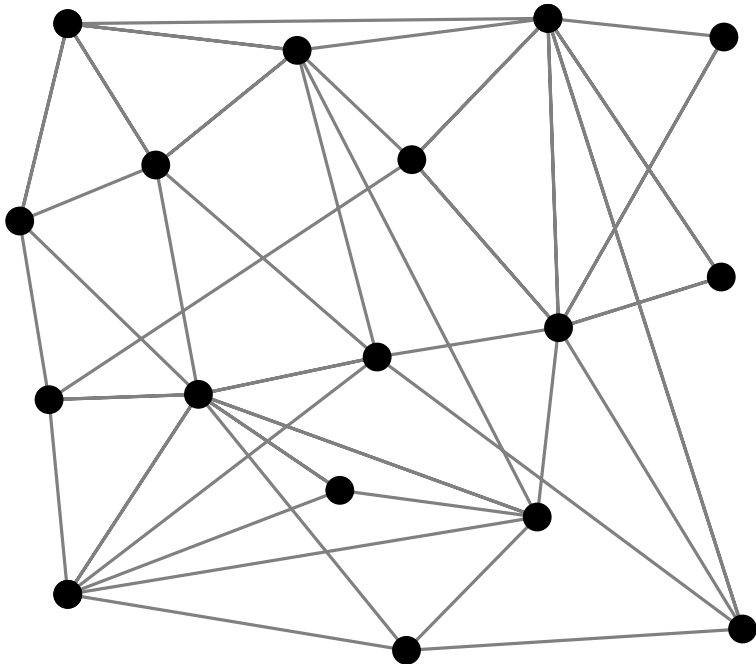
### Part I: Metric $k$ -CENTER

# Metric $k$ -CENTER

**Given:** A graph  $G = (V, E)$

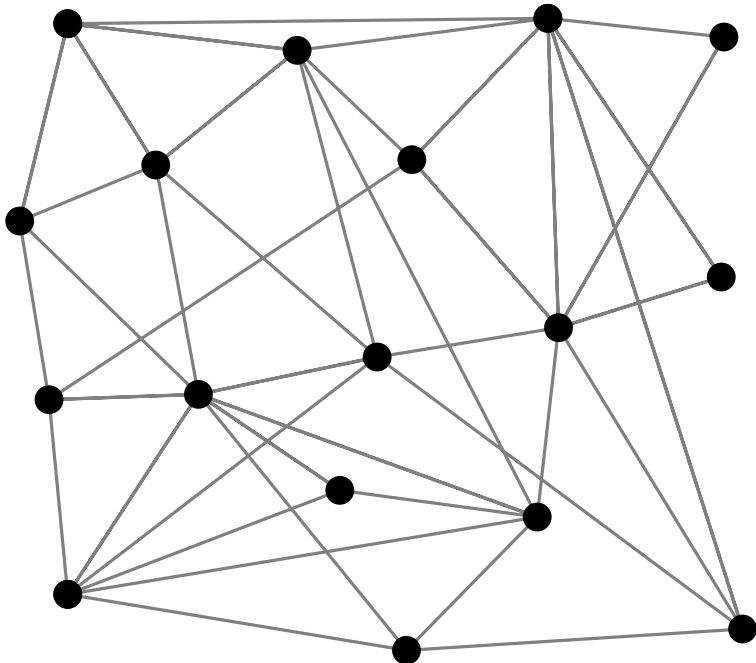
# Metric $k$ -CENTER

**Given:** A graph  $G = (V, E)$



# Metric $k$ -CENTER

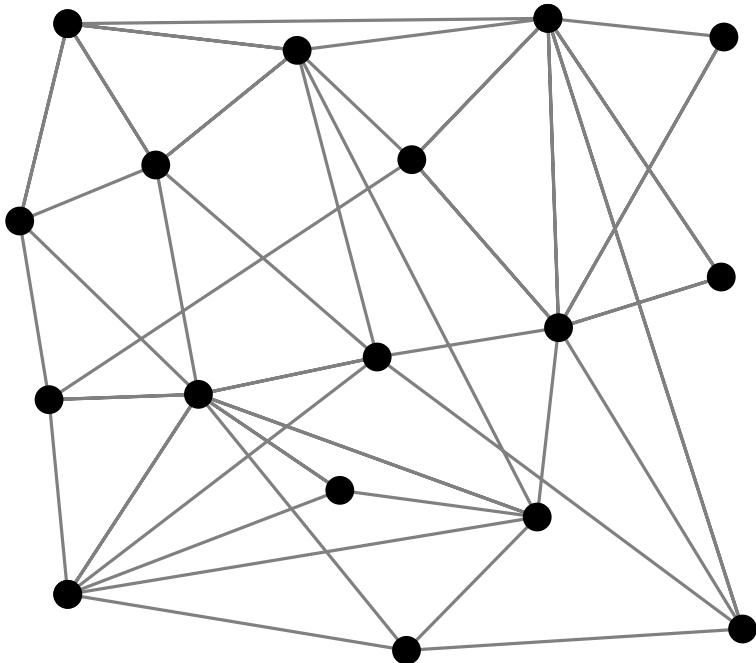
**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and



# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and

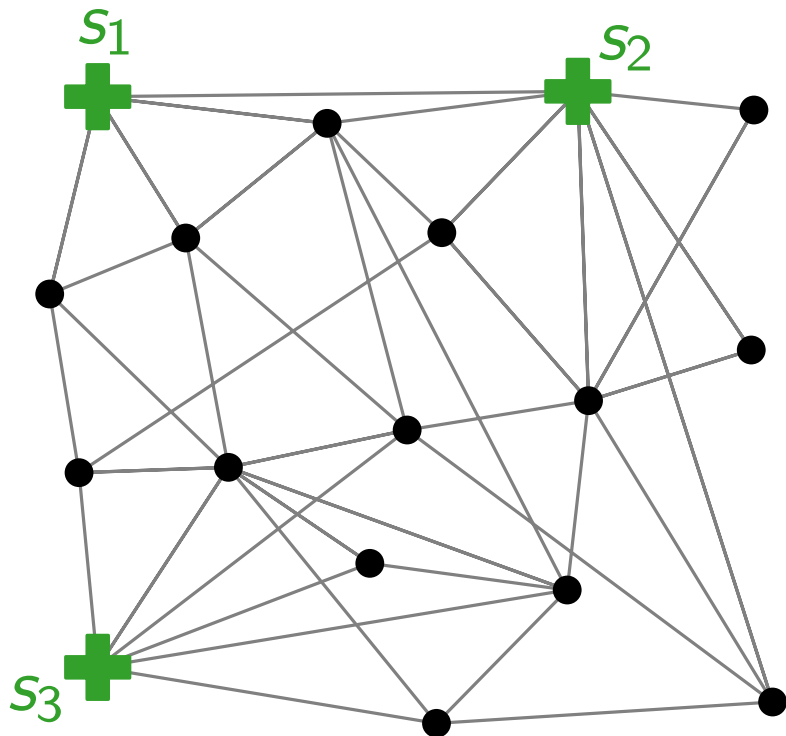
vertex set  $S \subseteq V$



# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and

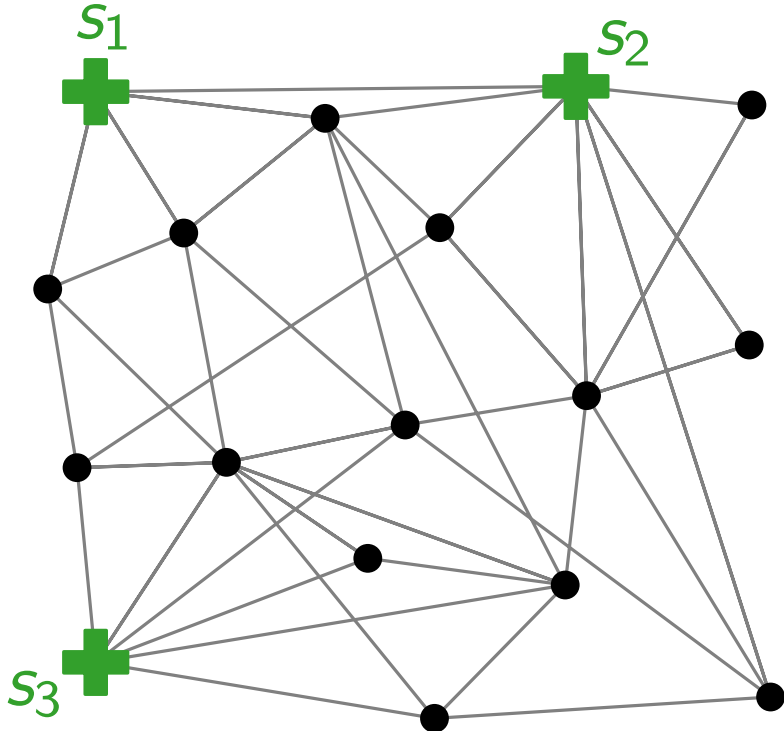
vertex set  $S \subseteq V$



# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and

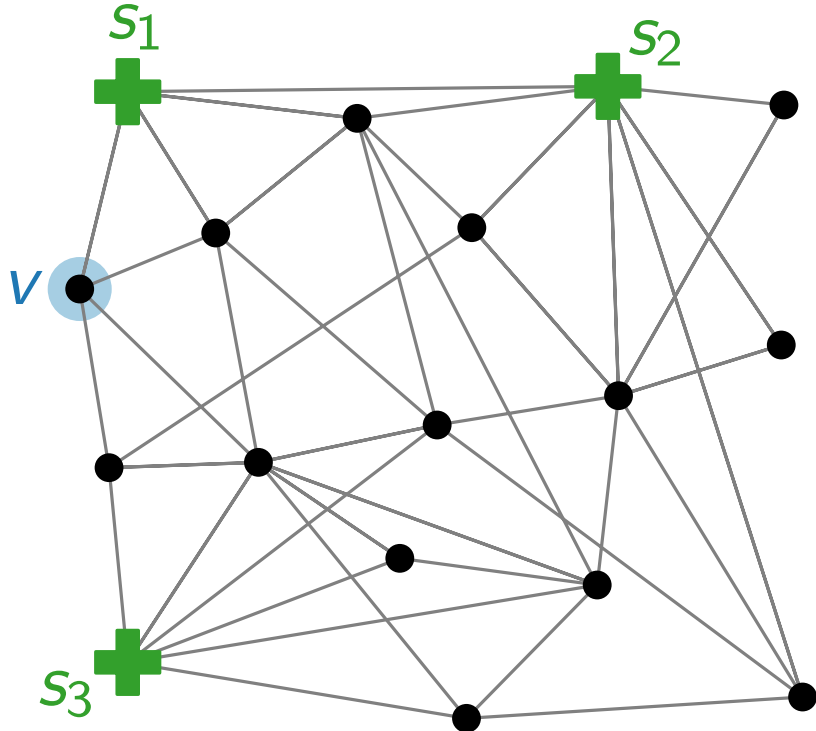
For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .



# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and

For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .

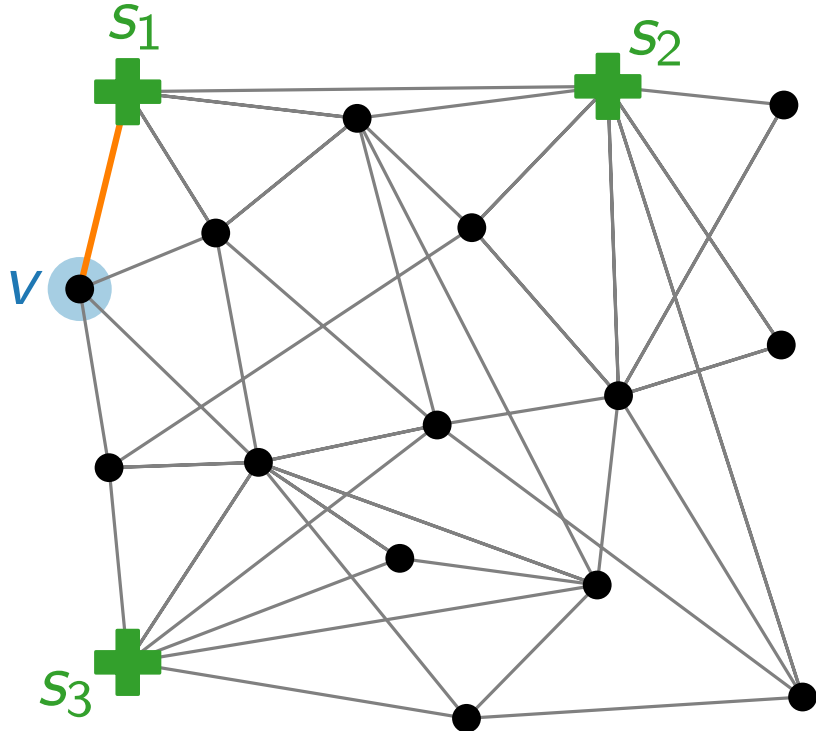




# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and

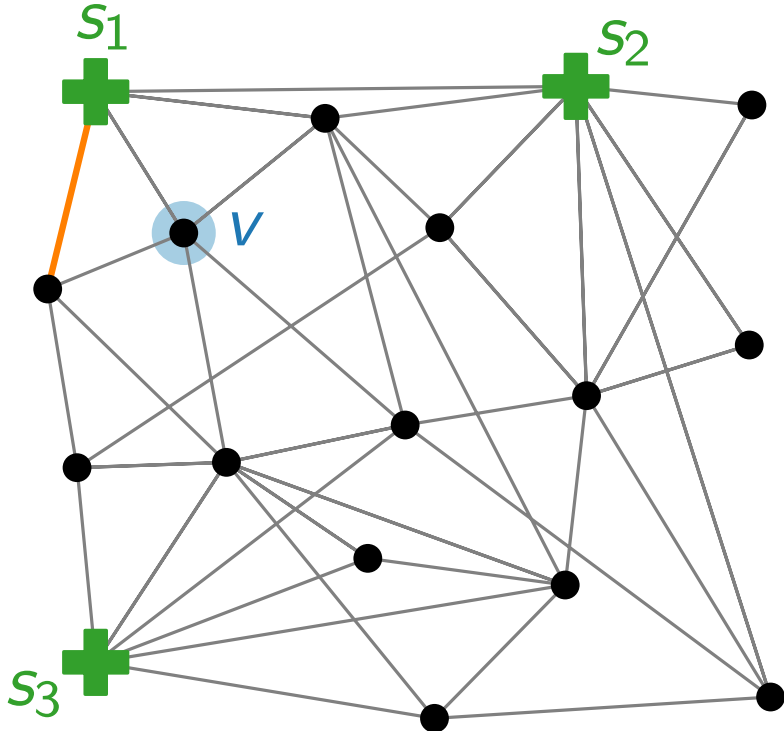
For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .



# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and

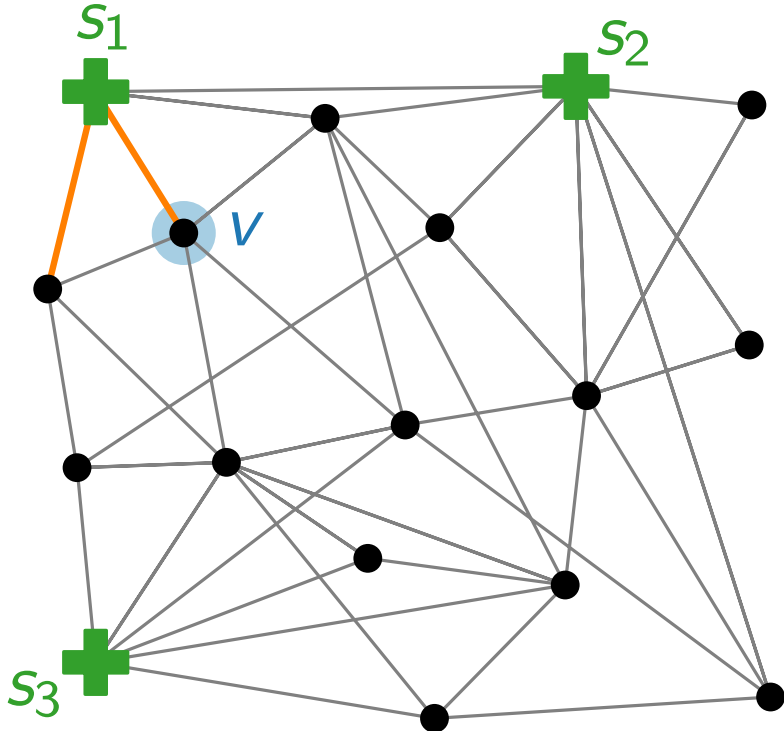
For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .



# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and

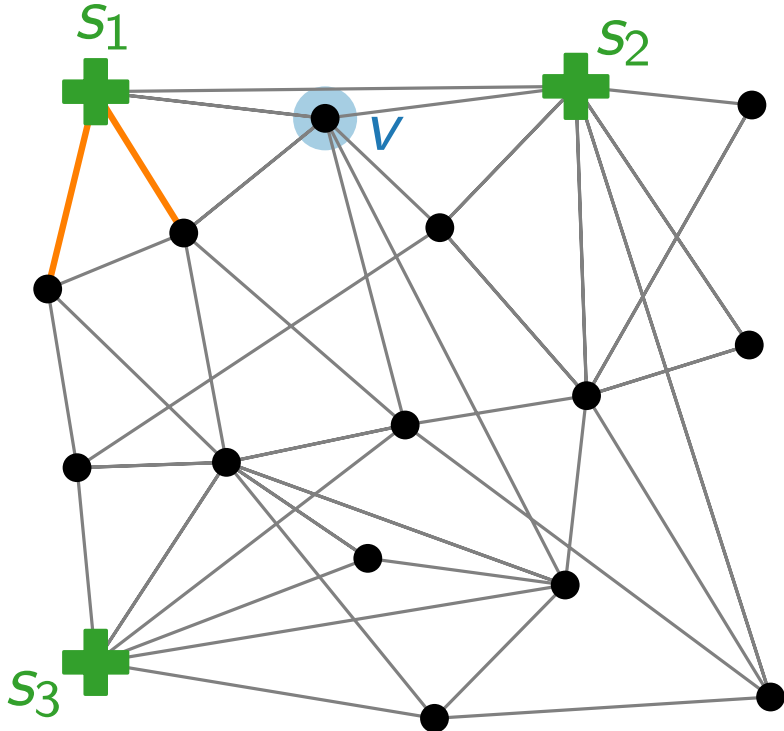
For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .



# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and

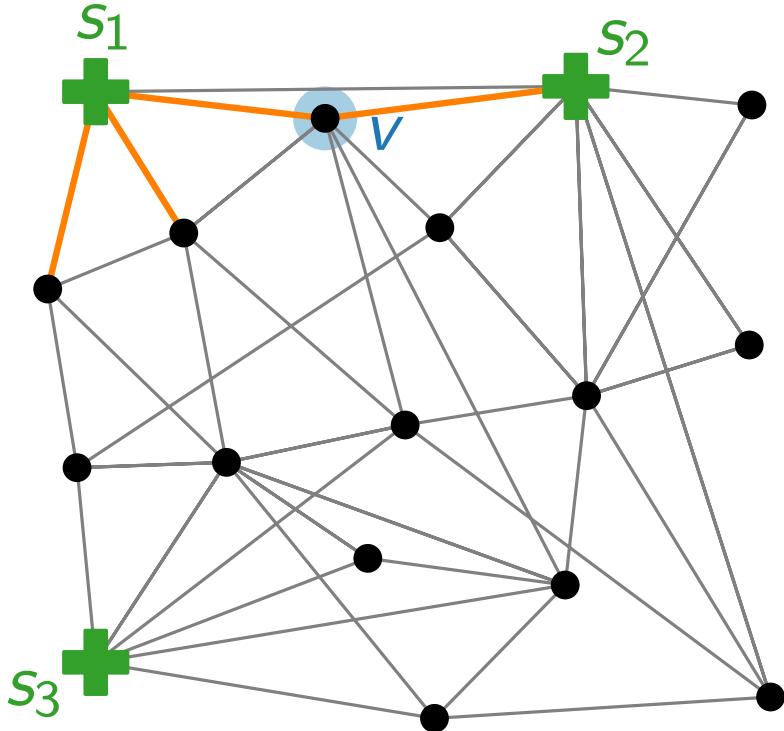
For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .



# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and

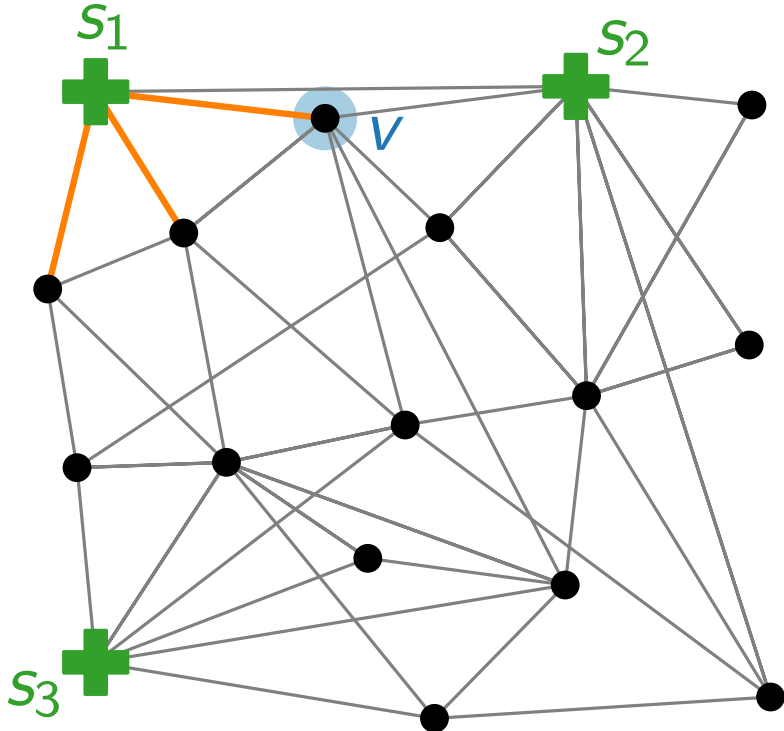
For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .



# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and

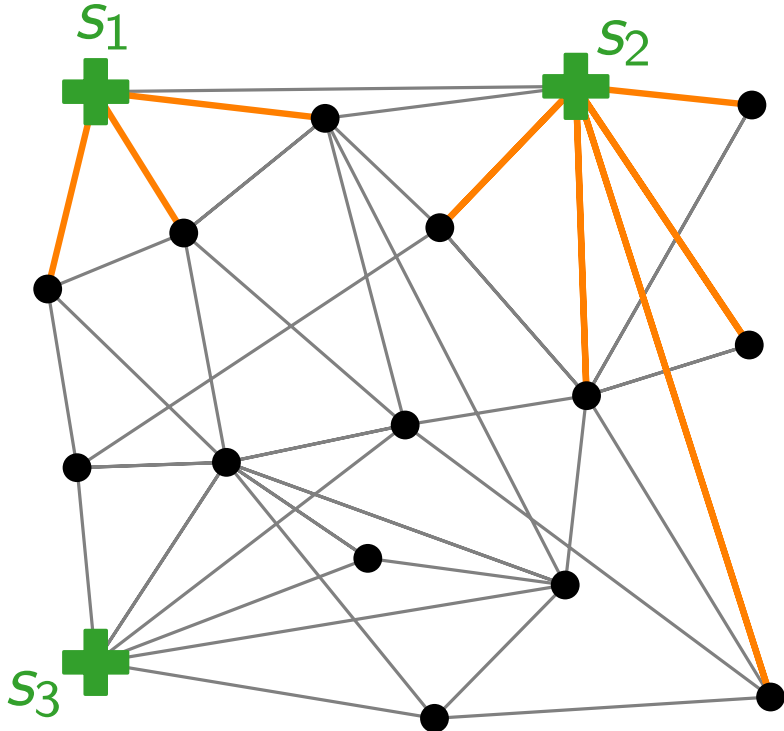
For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .



# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and

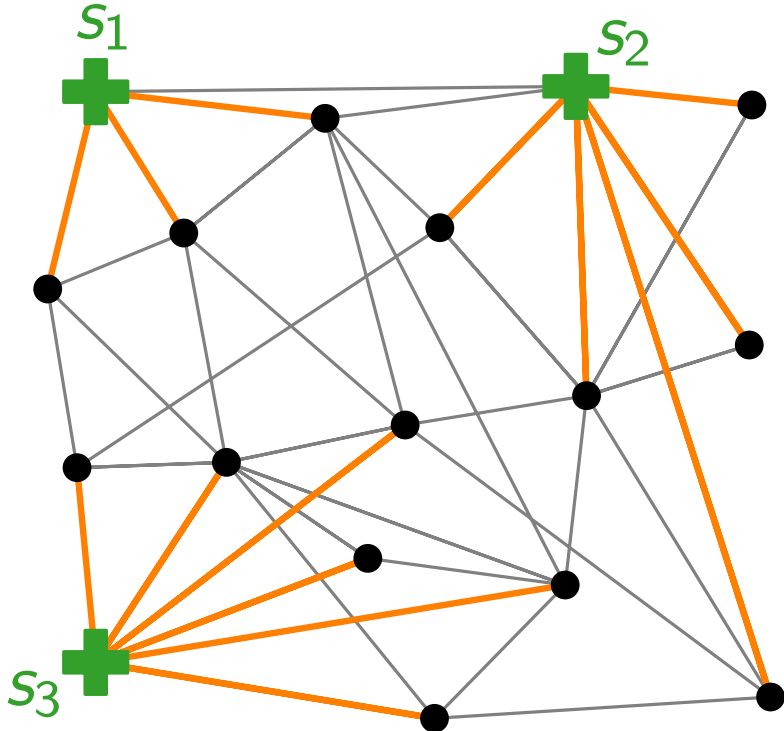
For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .



# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and

For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .



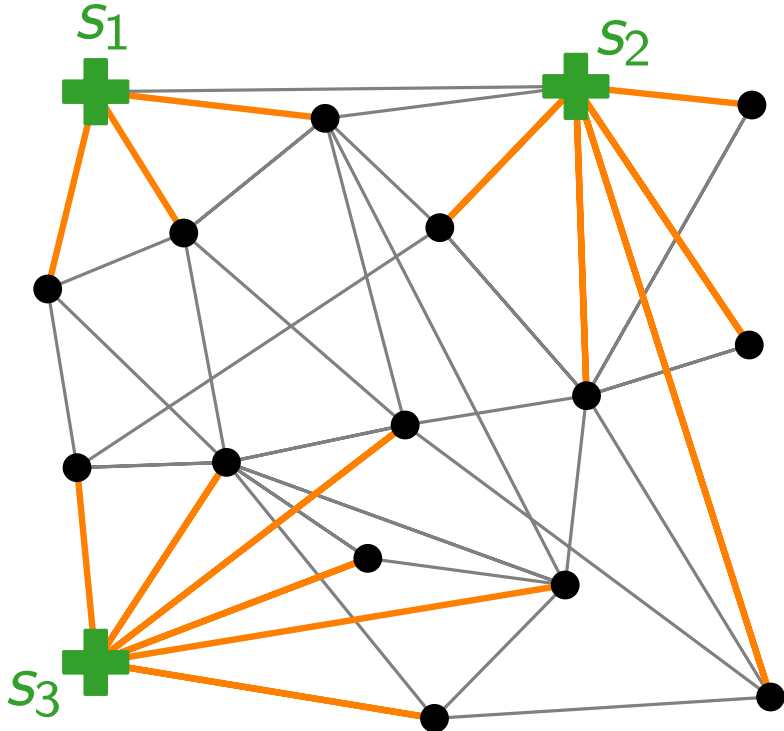


# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and

For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .

$$\text{cost}(S) := \max_{v \in V} c(v, S)$$

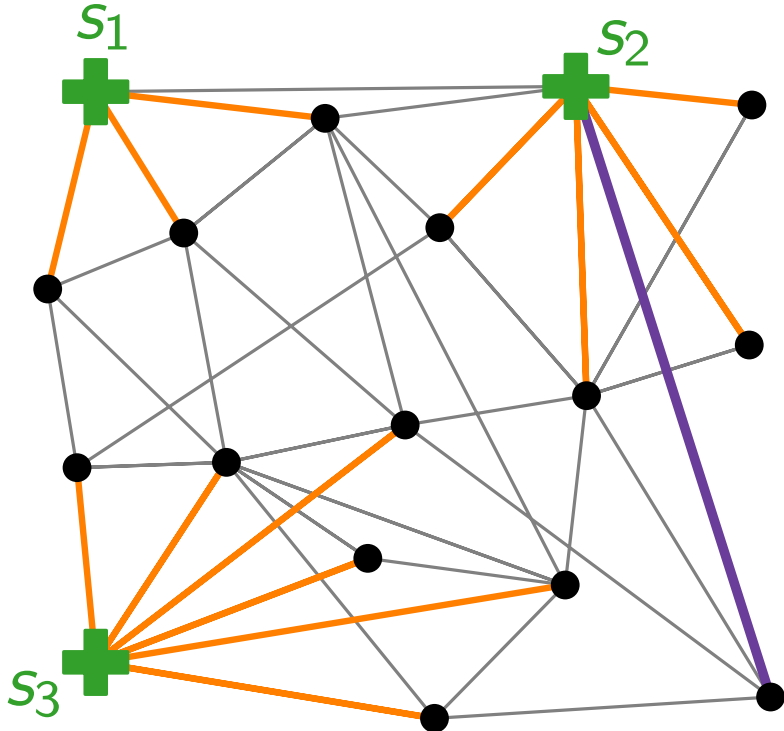


# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and

For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .

$$\text{cost}(S) := \max_{v \in V} c(v, S)$$

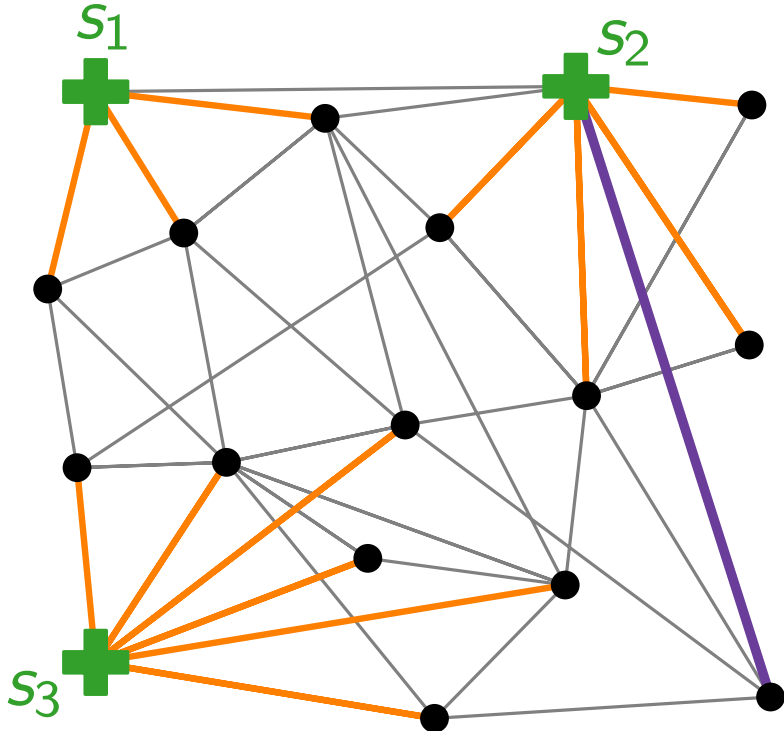


# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and

For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .

**Find:** A vertex set  $S$  such that  $\text{cost}(S) := \max_{v \in V} c(v, S)$  is minimized.

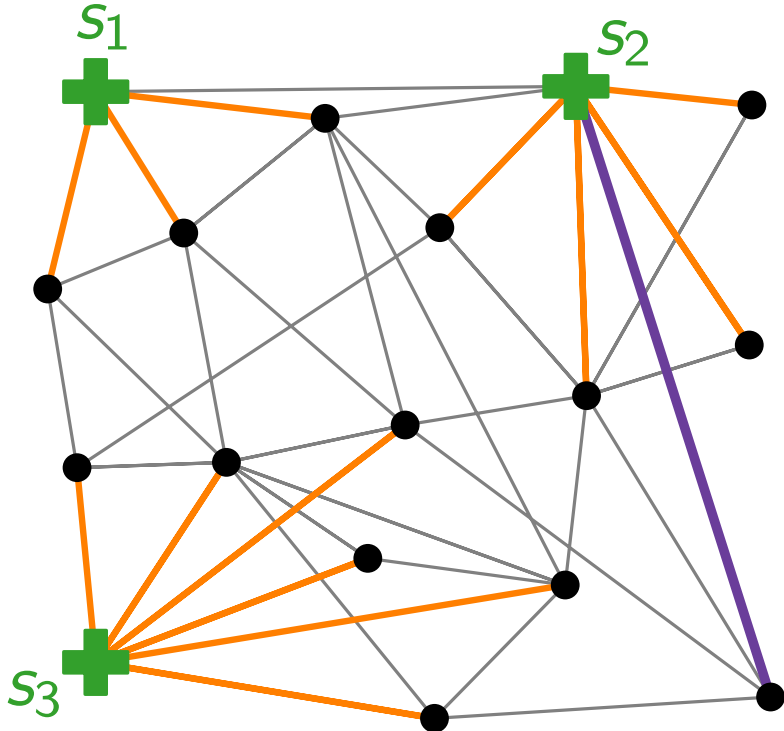


# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and a natural number  $k \leq |V|$ .

For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .

**Find:** A vertex set  $S$  such that  $\text{cost}(S) := \max_{v \in V} c(v, S)$  is minimized.

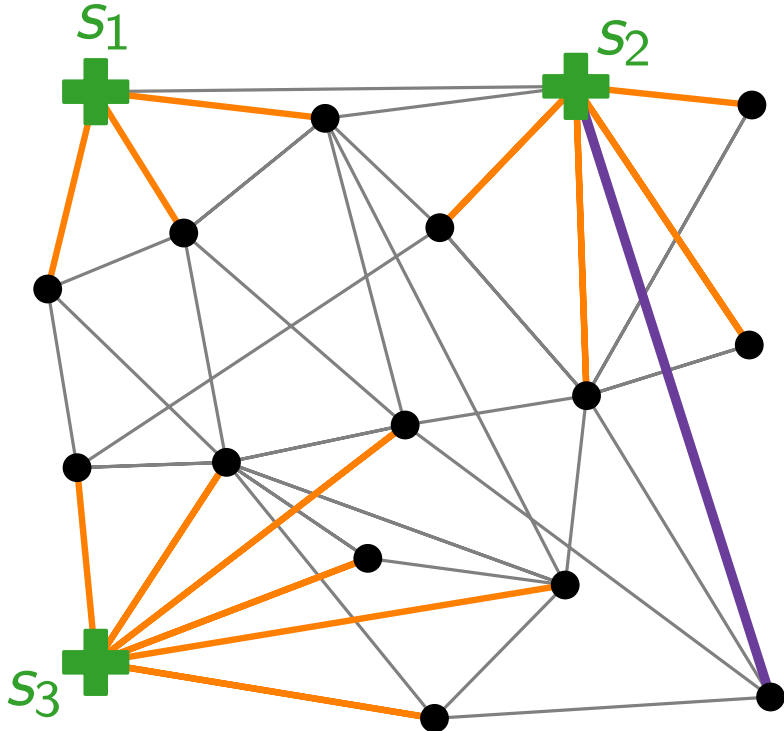


# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and a natural number  $k \leq |V|$ .

For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .

**Find:** A  $k$ -element vertex set  $S$  such that  $\text{cost}(S) := \max_{v \in V} c(v, S)$  is minimized.

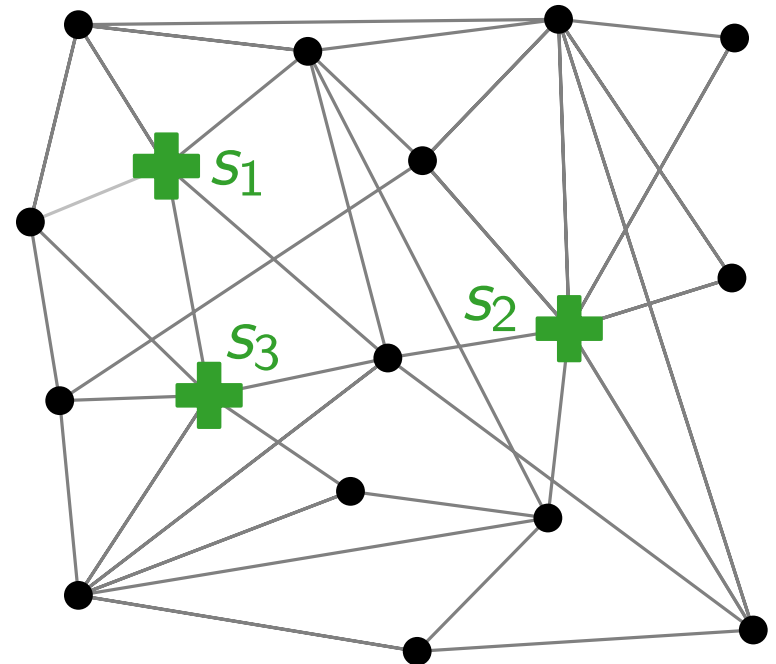
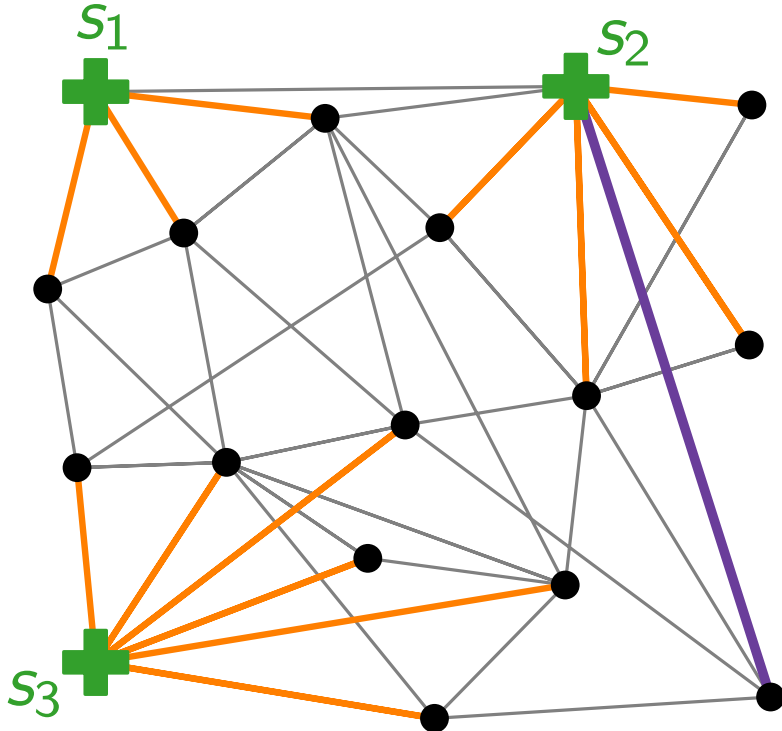


# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and a natural number  $k \leq |V|$ .

For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .

**Find:** A  $k$ -element vertex set  $S$  such that  $\text{cost}(S) := \max_{v \in V} c(v, S)$  is minimized.

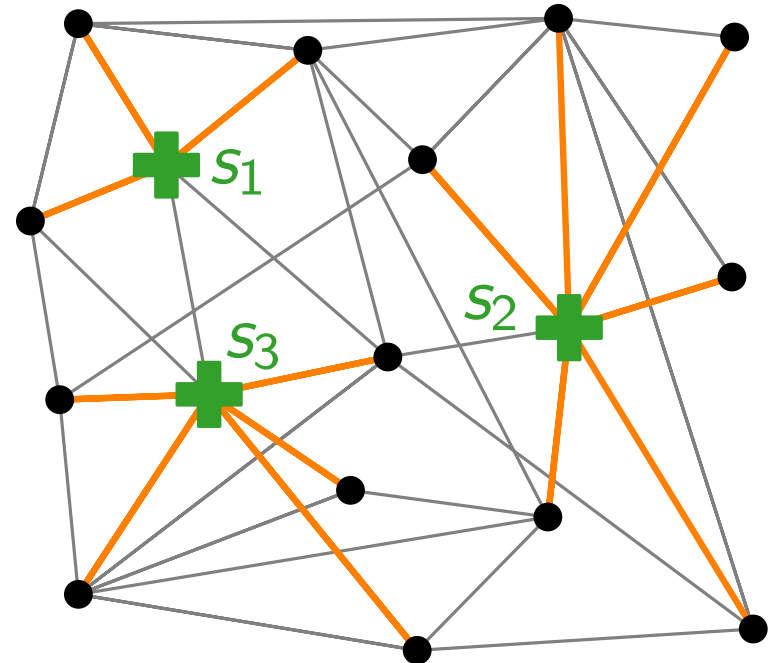
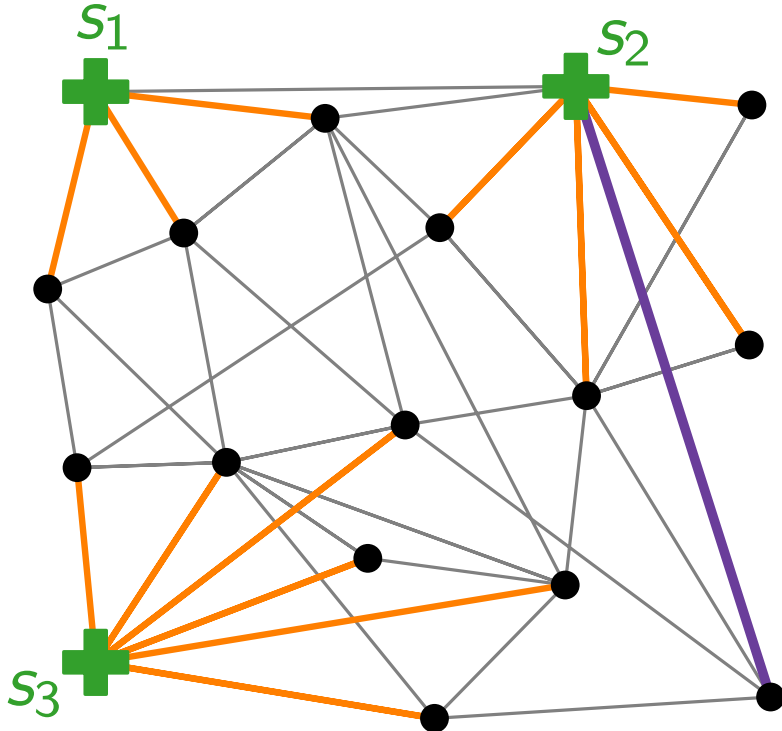


# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and a natural number  $k \leq |V|$ .

For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .

**Find:** A  $k$ -element vertex set  $S$  such that  $\text{cost}(S) := \max_{v \in V} c(v, S)$  is minimized.

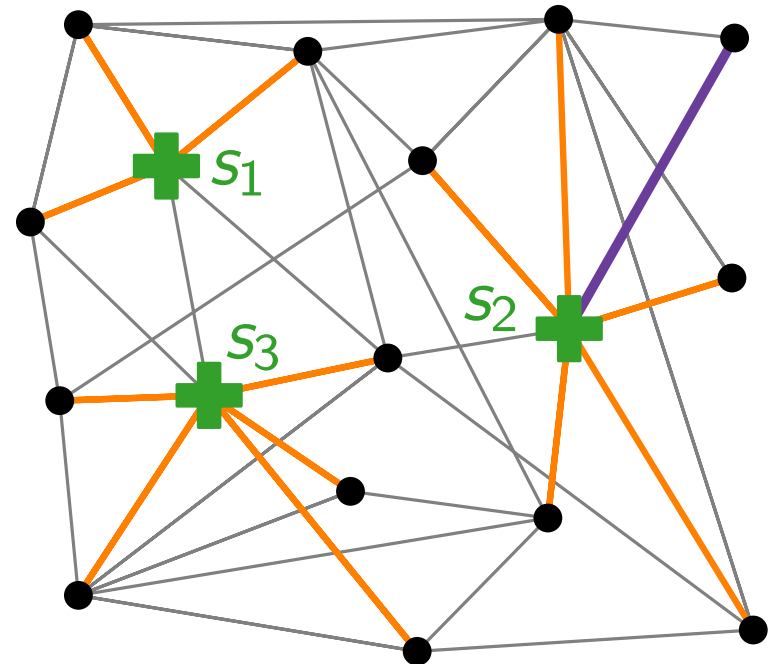
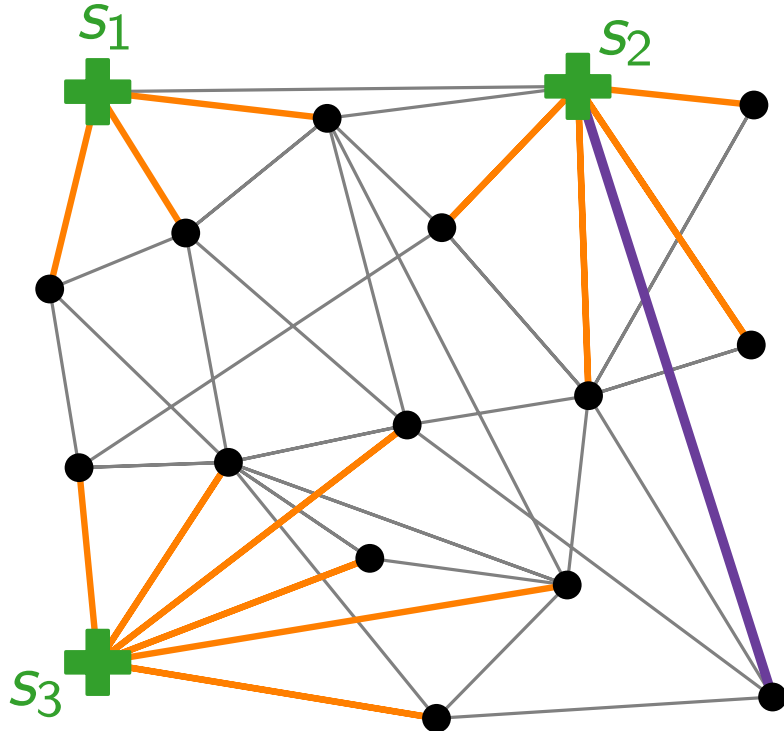


# Metric $k$ -CENTER

**Given:** A complete graph  $G = (V, E)$  with edge costs  $c: E \rightarrow \mathbb{Q}_{\geq 0}$  satisfying the triangle inequality and a natural number  $k \leq |V|$ .

For each vertex set  $S \subseteq V$ ,  $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .

**Find:** A  $k$ -element vertex set  $S$  such that  $\text{cost}(S) := \max_{v \in V} c(v, S)$  is minimized.





# Approximation Algorithms

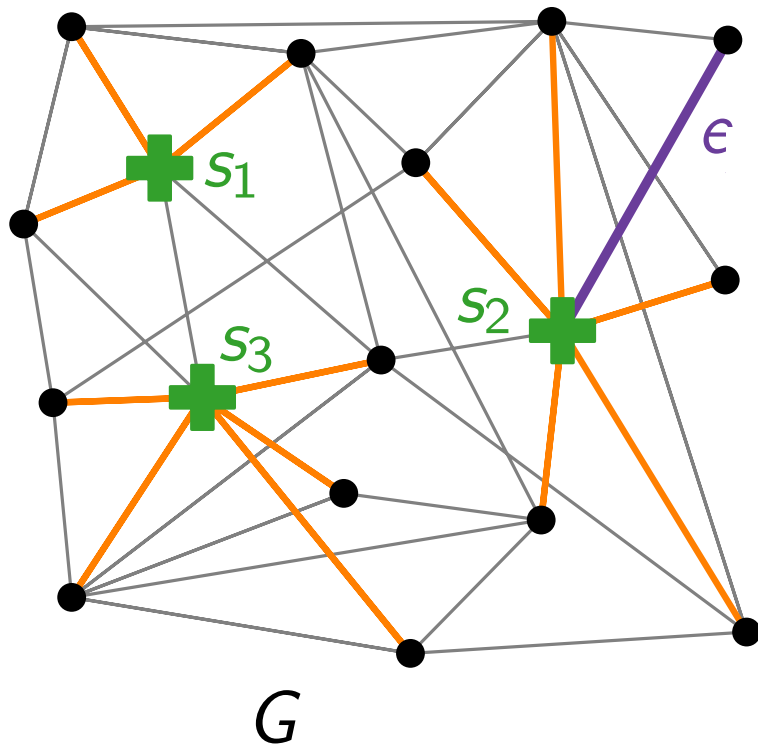
Lecture 6:

$k$ -CENTER via Parametric Pruning

Part II:

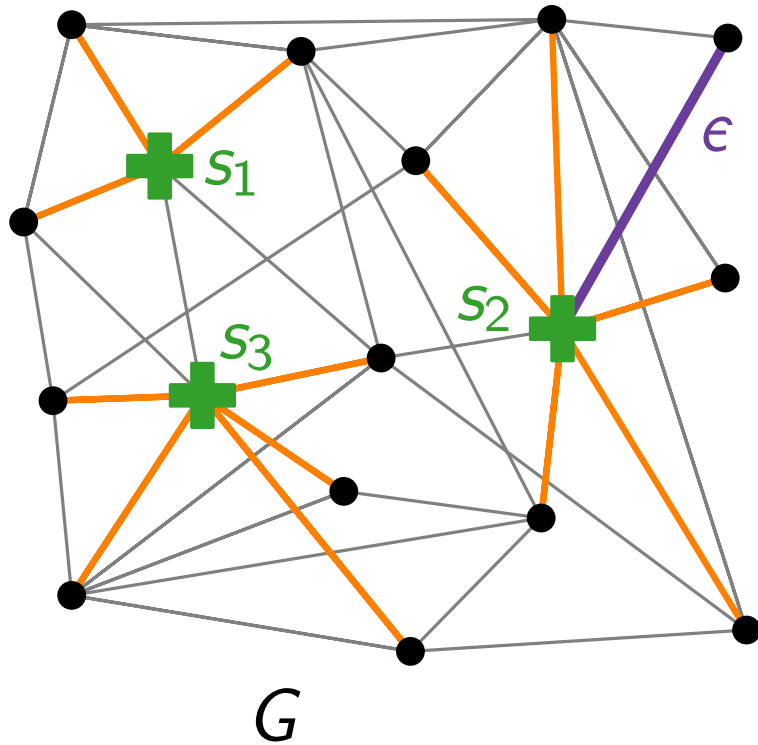
Parametric Pruning

# Parametric Pruning



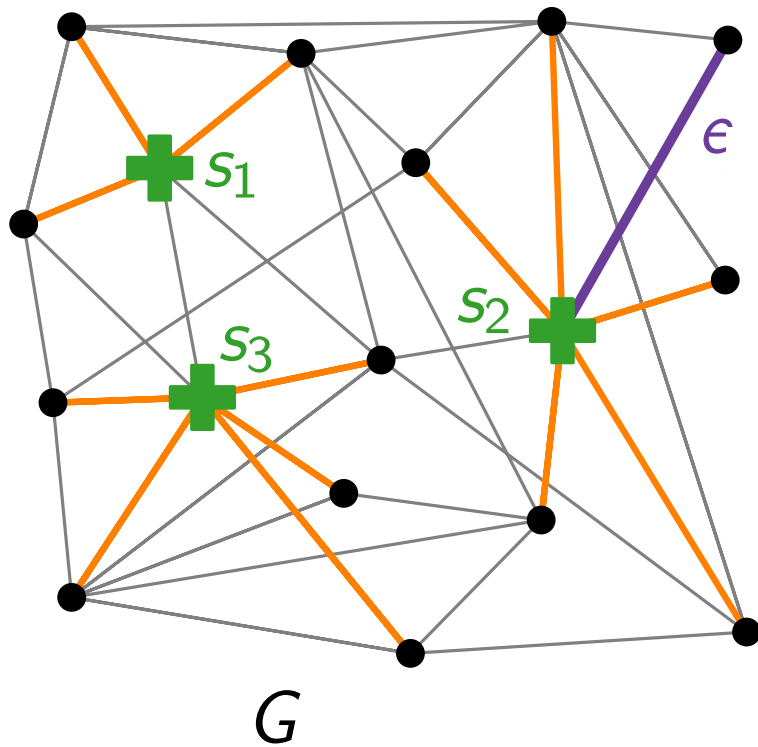
# Parametric Pruning

Let  $E = \{e_1, \dots, e_m\}$  with  $c(e_1) \leq \dots \leq c(e_m)$ .



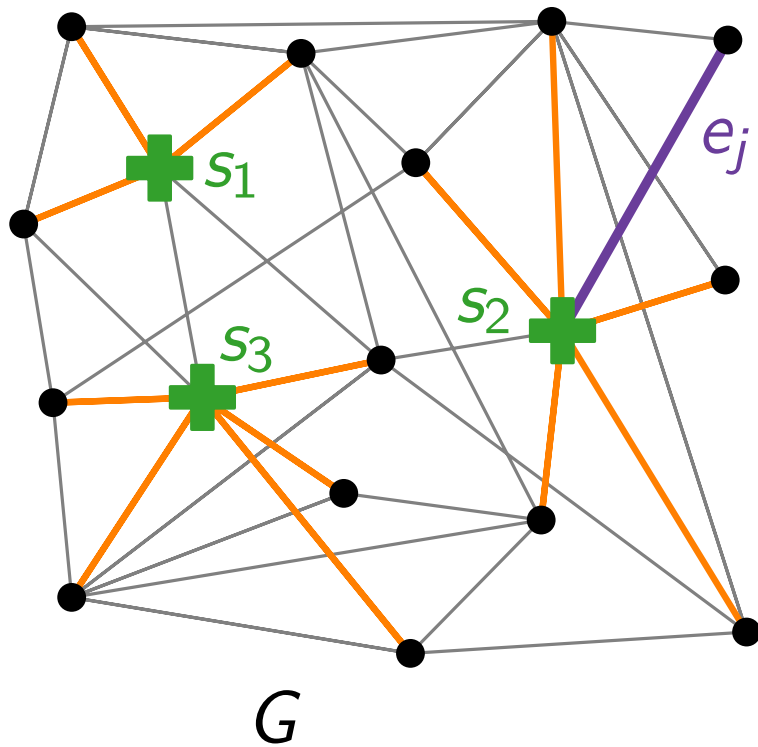
# Parametric Pruning

Let  $E = \{e_1, \dots, e_m\}$  with  $c(e_1) \leq \dots \leq c(e_m)$ .  
Suppose that we know that  $\text{OPT} = c(e_j)$ .



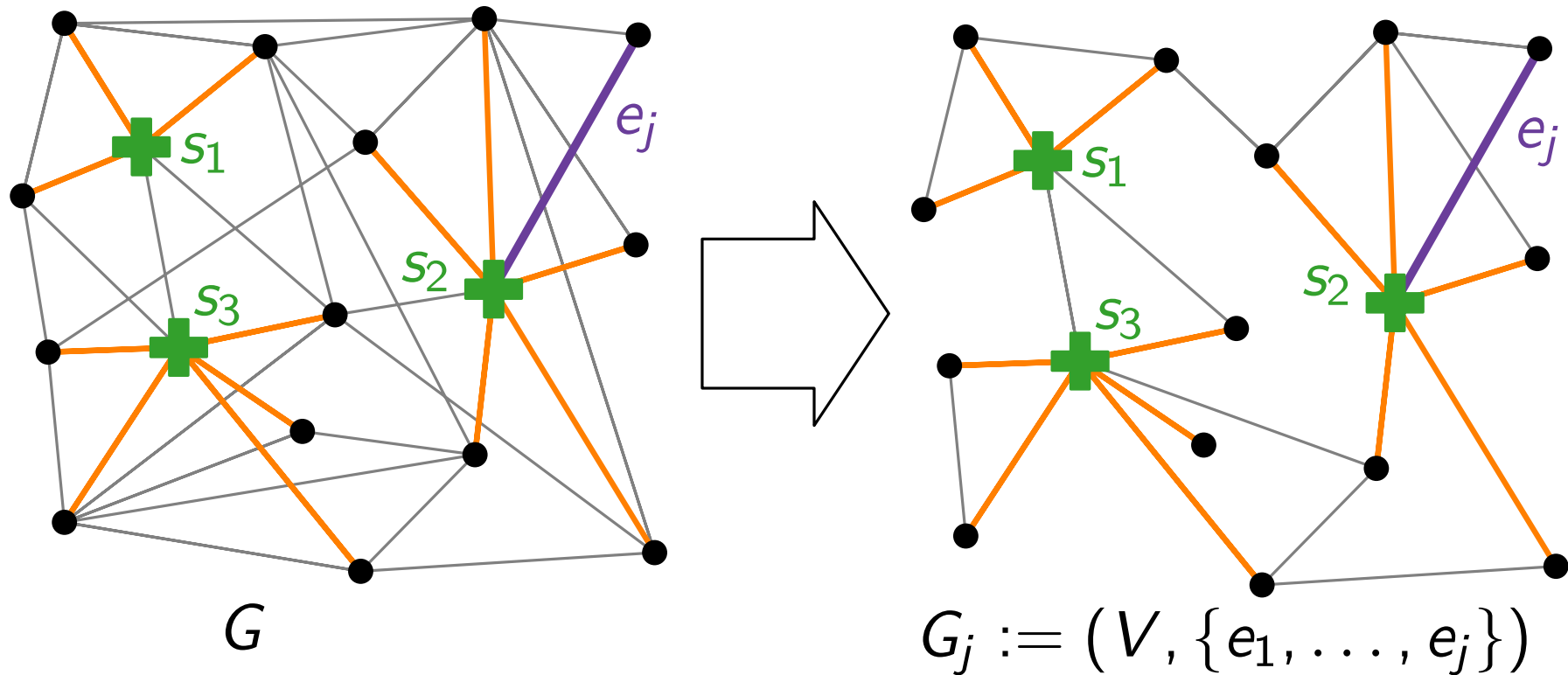
# Parametric Pruning

Let  $E = \{e_1, \dots, e_m\}$  with  $c(e_1) \leq \dots \leq c(e_m)$ .  
Suppose that we know that  $\text{OPT} = c(e_j)$ .



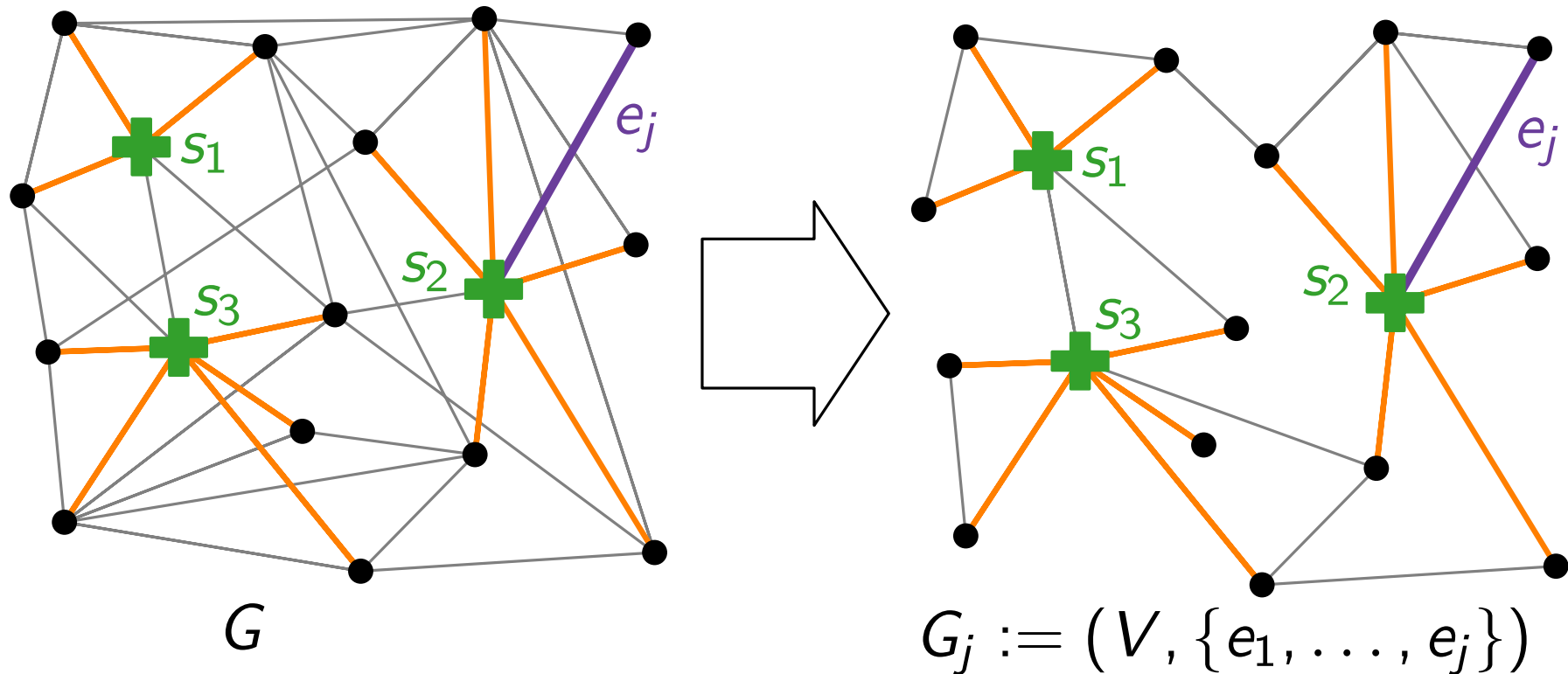
# Parametric Pruning

Let  $E = \{e_1, \dots, e_m\}$  with  $c(e_1) \leq \dots \leq c(e_m)$ .  
Suppose that we know that  $\text{OPT} = c(e_j)$ .



# Parametric Pruning

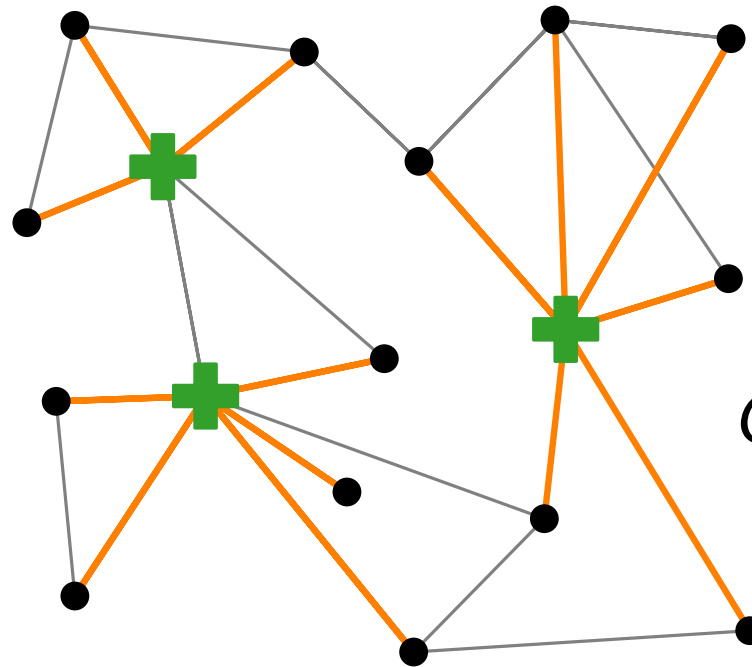
Let  $E = \{e_1, \dots, e_m\}$  with  $c(e_1) \leq \dots \leq c(e_m)$ .  
Suppose that we know that  $\text{OPT} = c(e_j)$ .



...try each  $G_j$ .

... try each  $G_j$ .

Def.

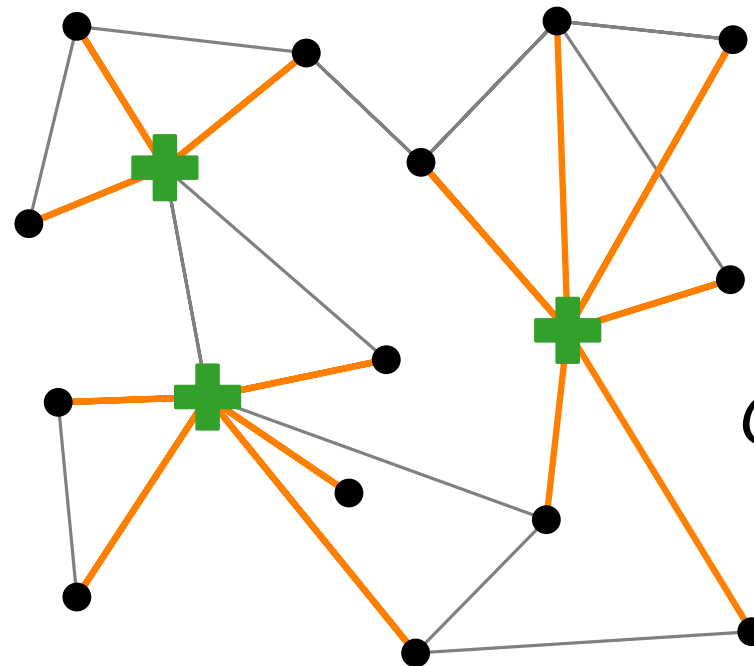


$$G_j := (V, \{e_1, \dots, e_j\})$$



...try each  $G_j$ .

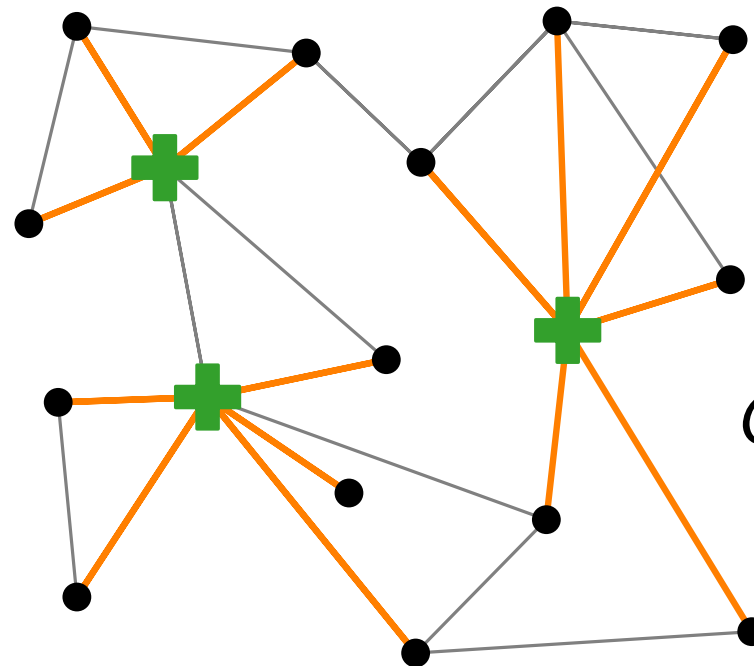
**Def.** A vertex set  $D$  of a graph  $H$  is **dominating** if each vertex is either in  $D$  or adjacent to a vertex in  $D$ .



$$G_j := (V, \{e_1, \dots, e_j\})$$

... try each  $G_j$ .

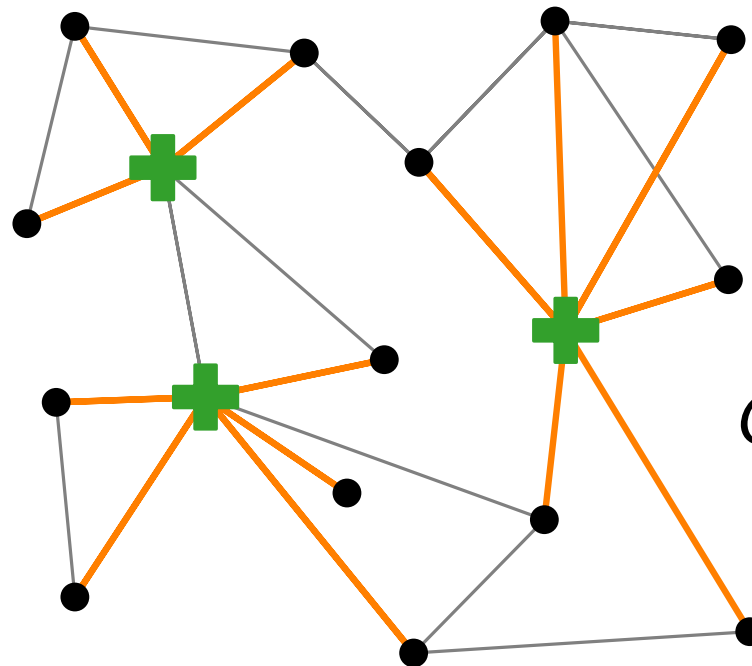
**Def.** A vertex set  $D$  of a graph  $H$  is **dominating** if each vertex is either in  $D$  or adjacent to a vertex in  $D$ .  
The cardinality of a smallest dominating set in  $H$  is denoted by  $\text{dom}(H)$ .



$$G_j := (V, \{e_1, \dots, e_j\})$$

... try each  $G_j$ .

**Def.** A vertex set  $D$  of a graph  $H$  is **dominating** if each vertex is either in  $D$  or adjacent to a vertex in  $D$ . The cardinality of a smallest dominating set in  $H$  is denoted by  $\text{dom}(H)$ .

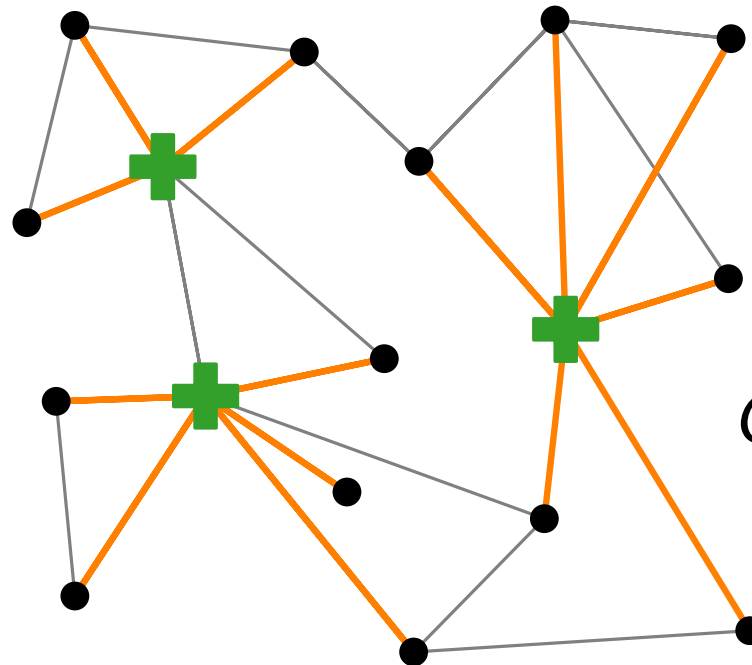


$$\text{dom}(G_j) = 3$$

$$G_j := (V, \{e_1, \dots, e_j\})$$

...try each  $G_j$ .

**Def.** A vertex set  $D$  of a graph  $H$  is **dominating** if each vertex is either in  $D$  or adjacent to a vertex in  $D$ . The cardinality of a smallest dominating set in  $H$  is denoted by  $\text{dom}(H)$ .



$$\text{dom}(G_j) = 3$$

$$G_j := (V, \{e_1, \dots, e_j\})$$



...but computing  $\text{dom}(H)$  is NP-hard.

# Approximation Algorithms

Lecture 6:

$k$ -CENTER via Parametric Pruning

Part III:

Square of a Graph

# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

# Square of a Graph

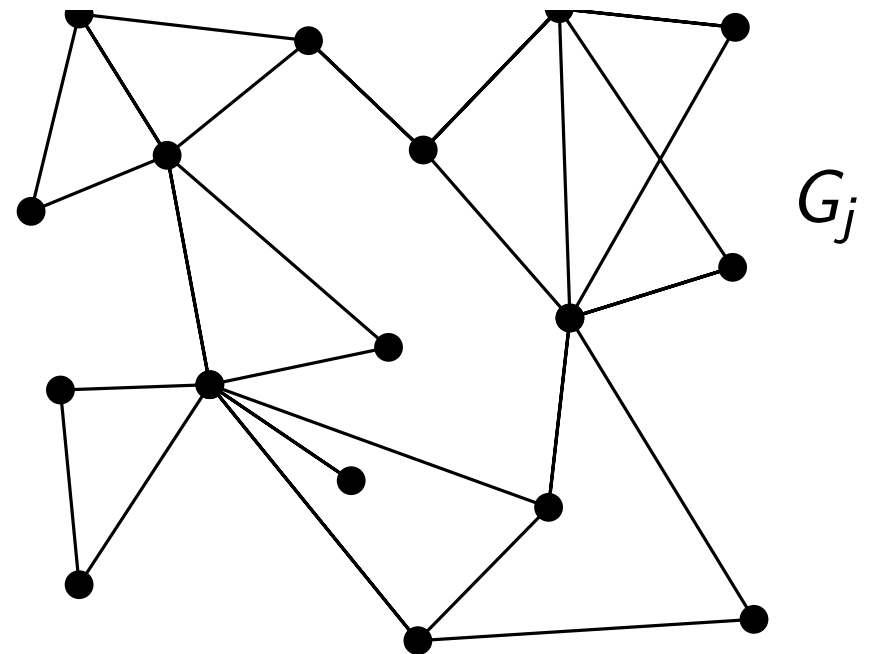
**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ .

# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ .

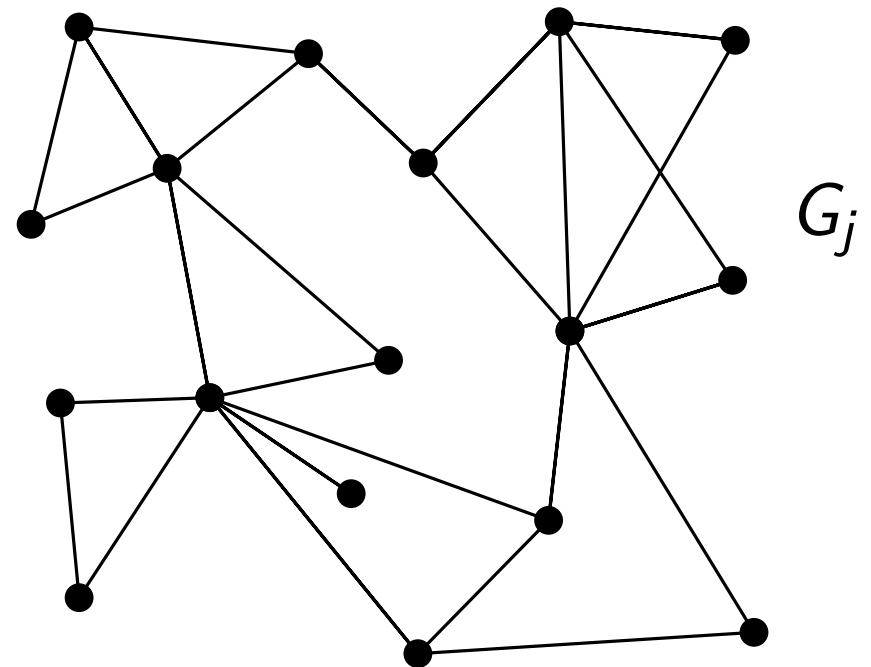




# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

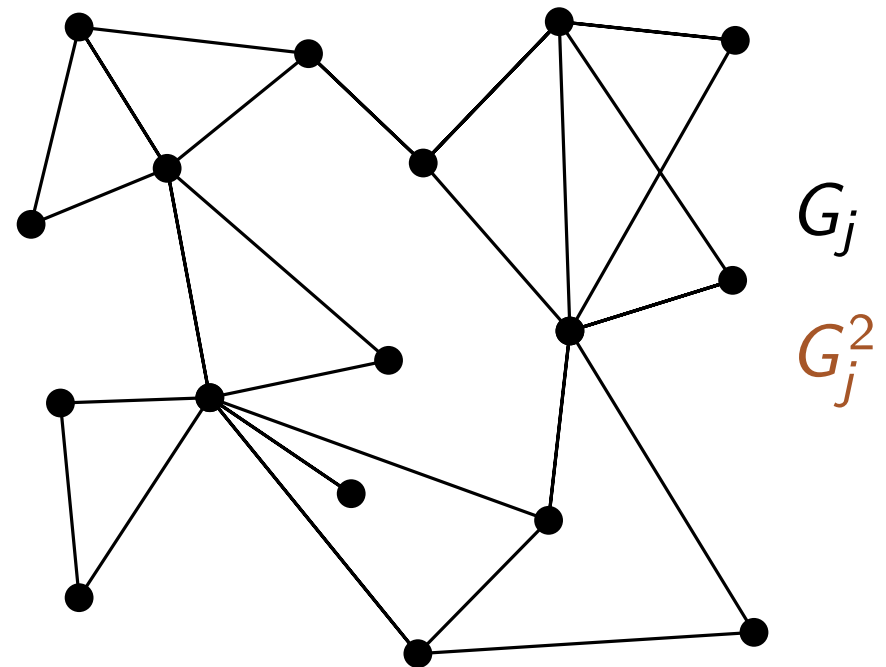
**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ . Two vertices  $u \neq v$  are adjacent in  $H^2$  iff they are within distance at most **two** in  $H$ .



# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

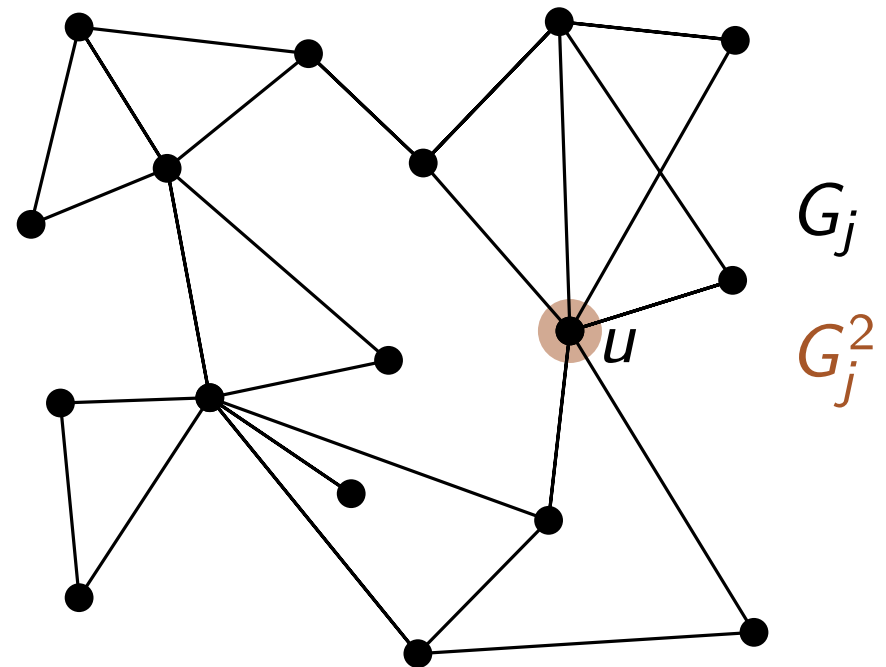
**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ . Two vertices  $u \neq v$  are adjacent in  $H^2$  iff they are within distance at most **two** in  $H$ .



# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

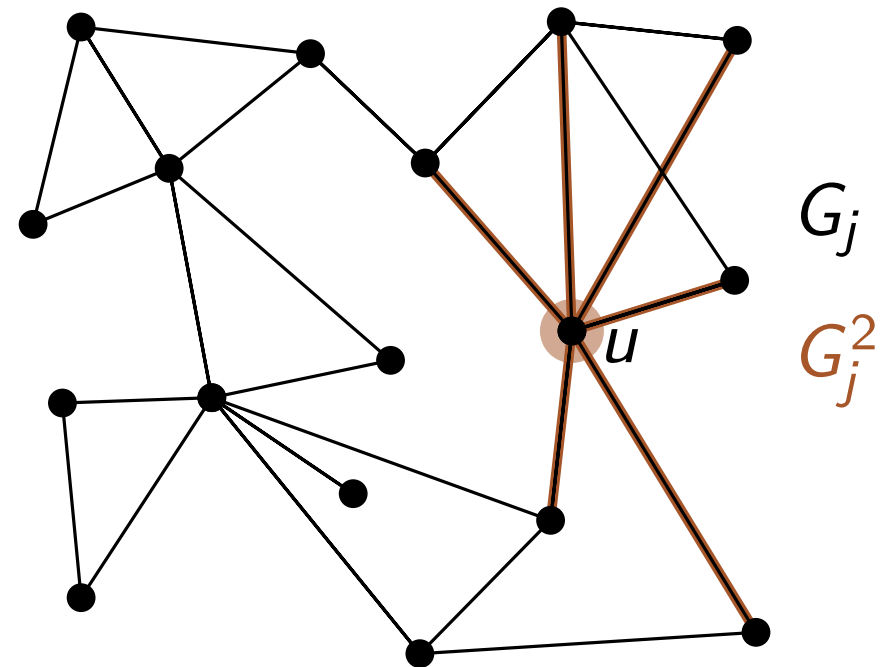
**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ . Two vertices  $u \neq v$  are adjacent in  $H^2$  iff they are within distance at most **two** in  $H$ .



# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

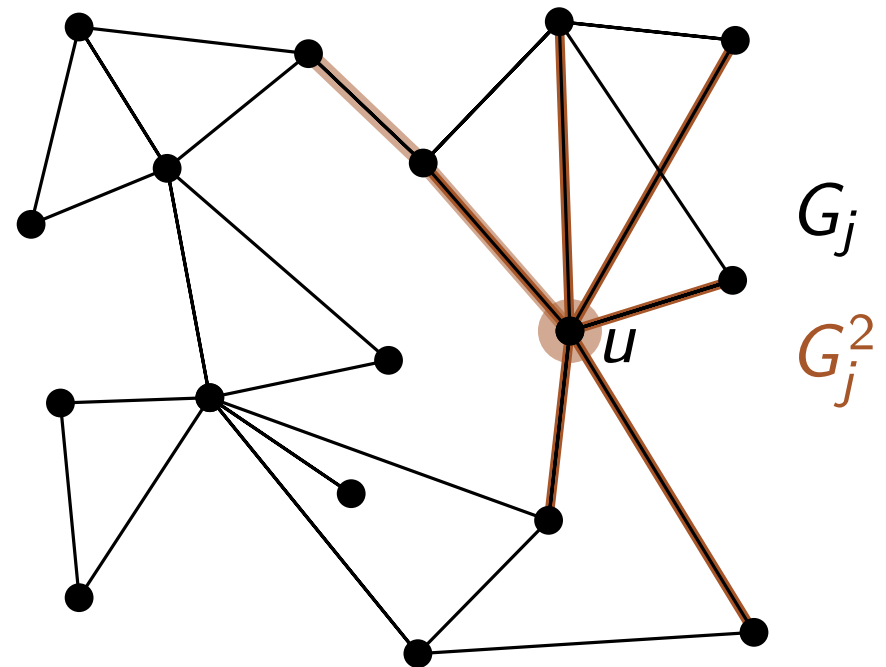
**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ . Two vertices  $u \neq v$  are adjacent in  $H^2$  iff they are within distance at most **two** in  $H$ .



# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

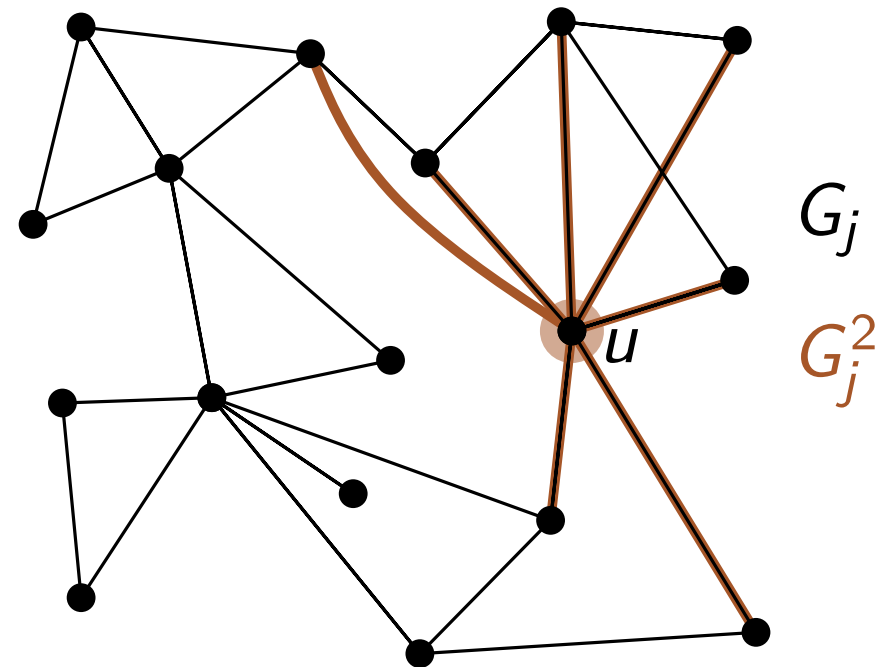
**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ . Two vertices  $u \neq v$  are adjacent in  $H^2$  iff they are within distance at most **two** in  $H$ .



# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

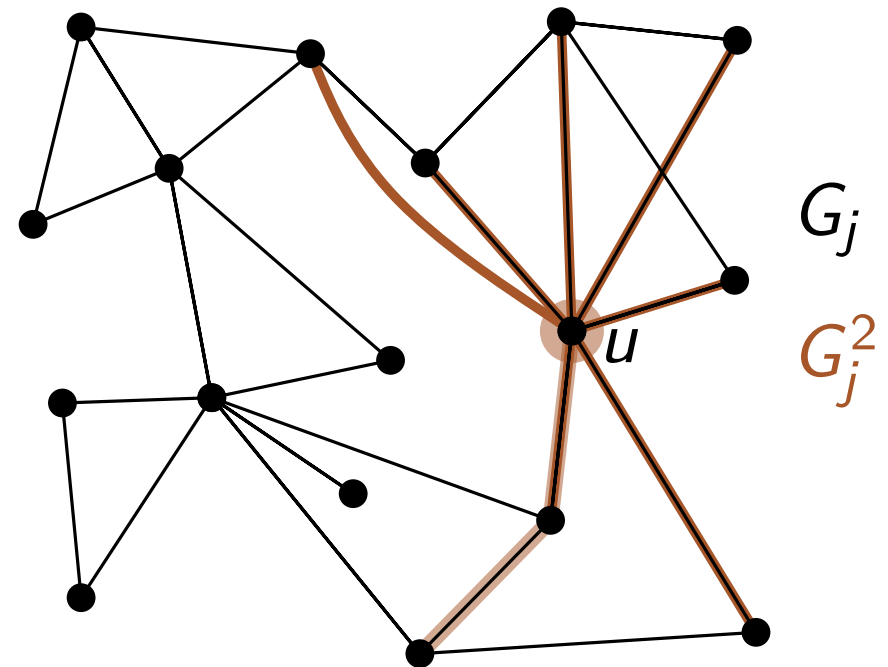
**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ . Two vertices  $u \neq v$  are adjacent in  $H^2$  iff they are within distance at most **two** in  $H$ .



# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

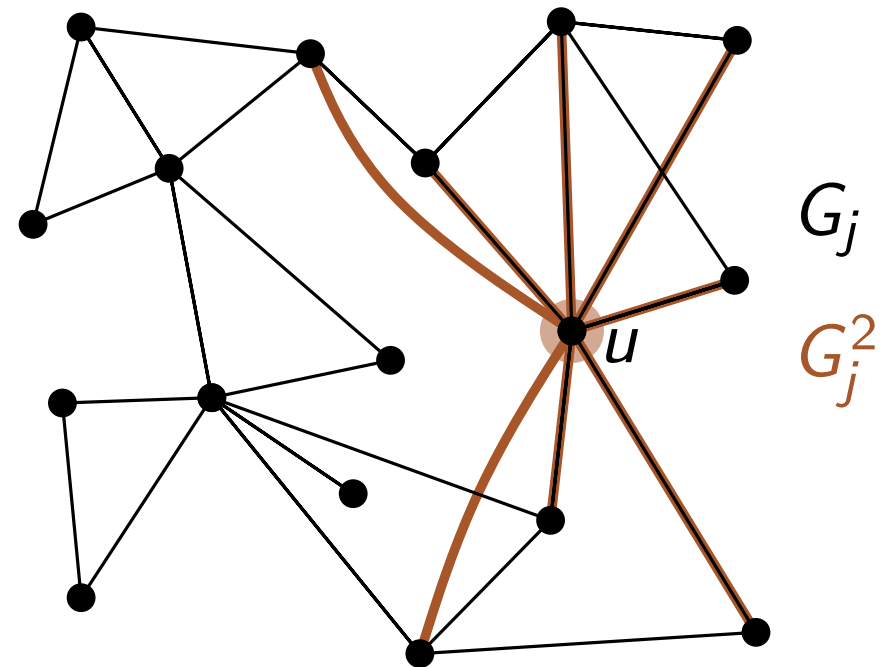
**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ . Two vertices  $u \neq v$  are adjacent in  $H^2$  iff they are within distance at most **two** in  $H$ .



# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ . Two vertices  $u \neq v$  are adjacent in  $H^2$  iff they are within distance at most **two** in  $H$ .

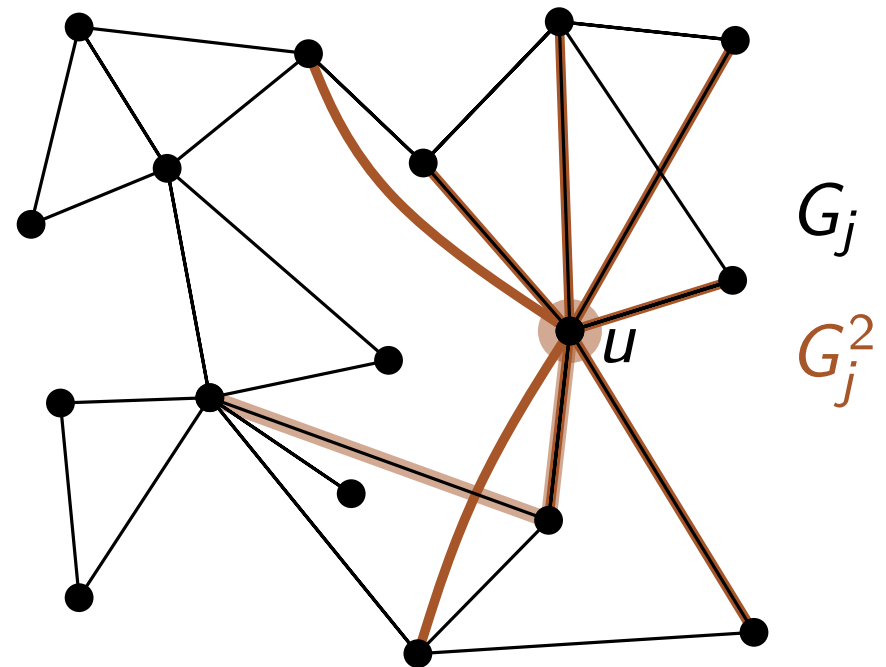




# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

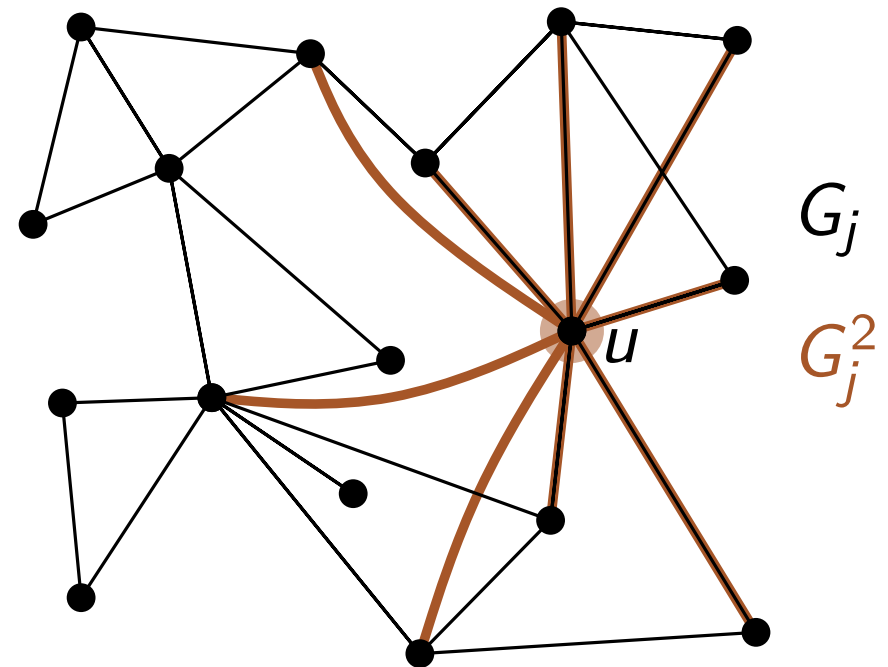
**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ . Two vertices  $u \neq v$  are adjacent in  $H^2$  iff they are within distance at most **two** in  $H$ .



# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

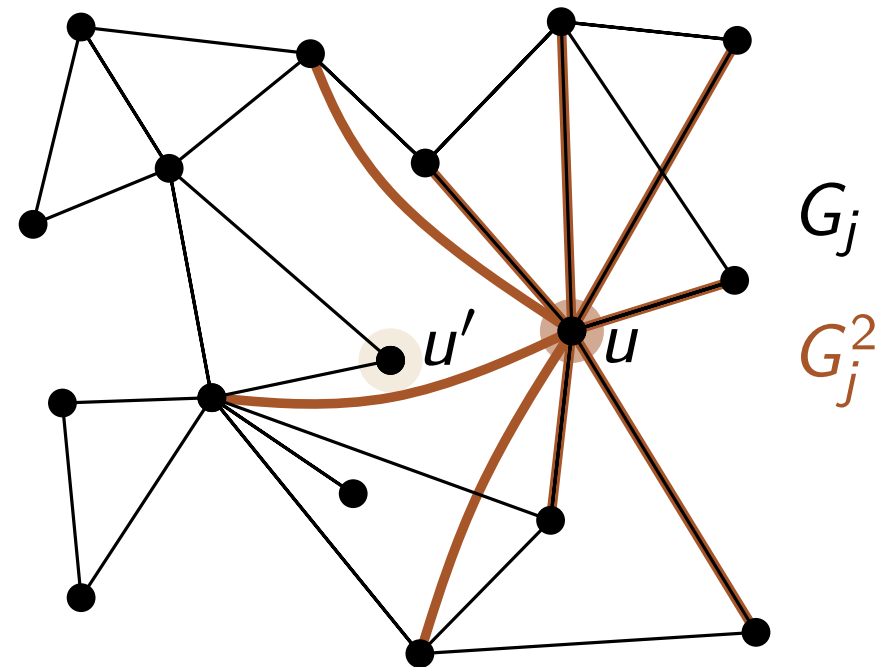
**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ . Two vertices  $u \neq v$  are adjacent in  $H^2$  iff they are within distance at most **two** in  $H$ .



# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

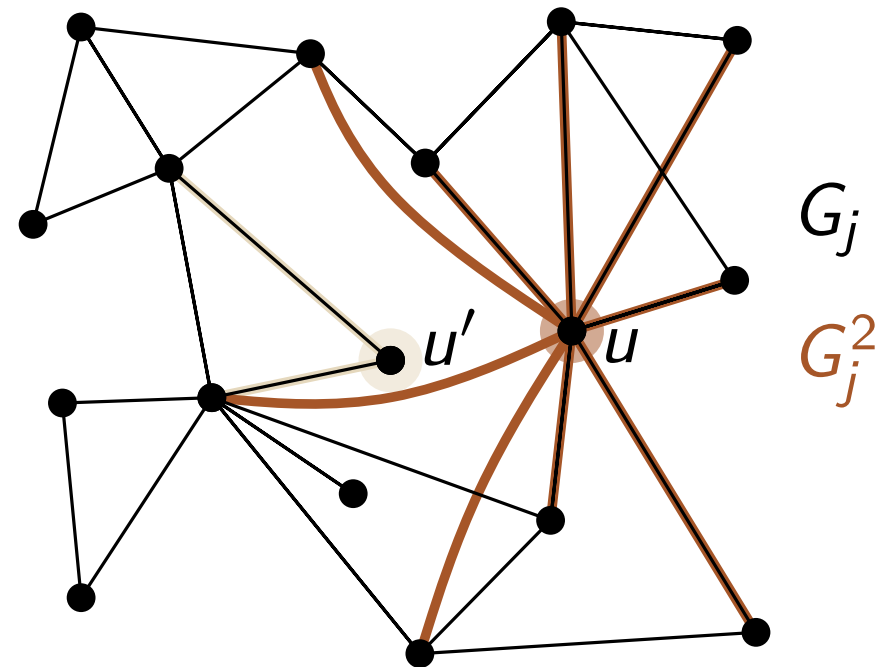
**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ . Two vertices  $u \neq v$  are adjacent in  $H^2$  iff they are within distance at most **two** in  $H$ .



# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

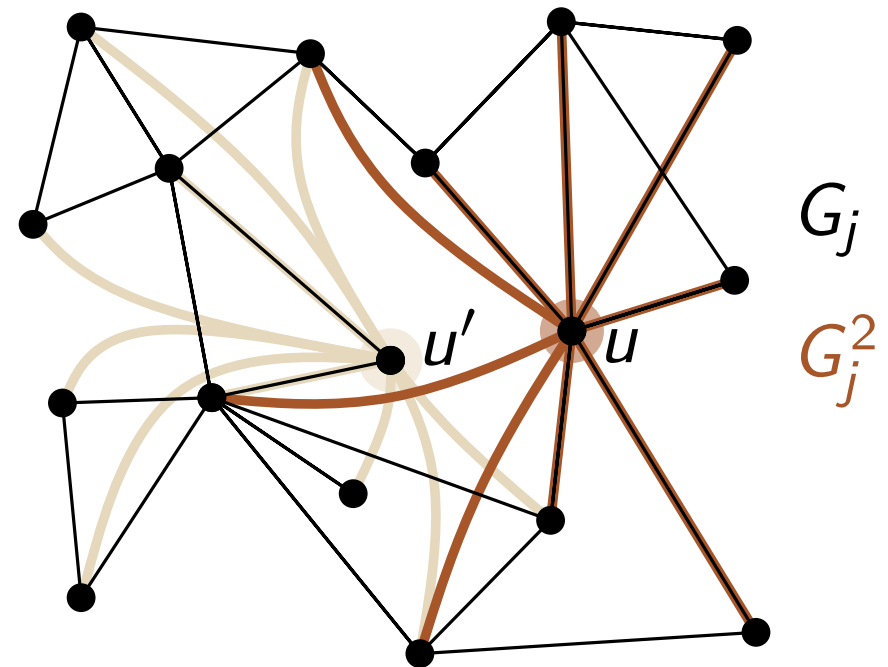
**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ . Two vertices  $u \neq v$  are adjacent in  $H^2$  iff they are within distance at most **two** in  $H$ .



# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ . Two vertices  $u \neq v$  are adjacent in  $H^2$  iff they are within distance at most **two** in  $H$ .

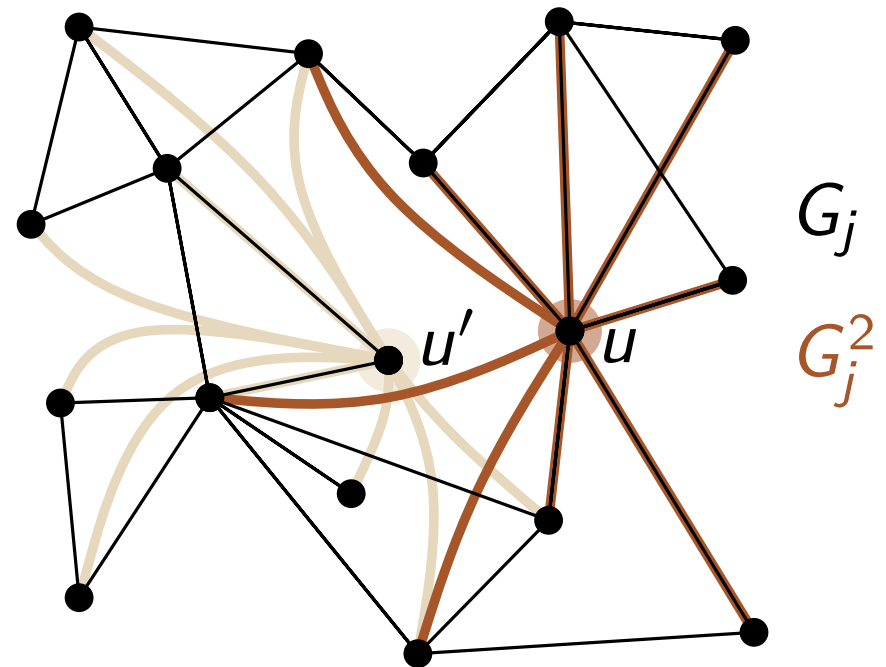


# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ . Two vertices  $u \neq v$  are adjacent in  $H^2$  iff they are within distance at most **two** in  $H$ .

**Obs.** A dominating set of size at most  $k$  in  $G_j^2$  is a  $\frac{1}{2}$ -approximation for metric  $k$ -CENTER.

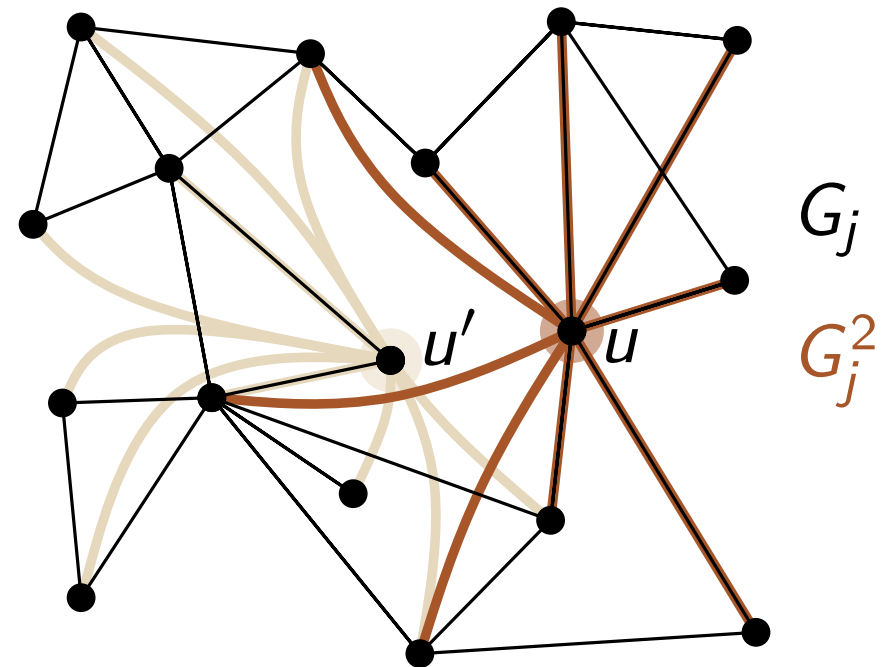


# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ . Two vertices  $u \neq v$  are adjacent in  $H^2$  iff they are within distance at most **two** in  $H$ .

**Obs.** A dominating set of size at most  $k$  in  $G_j^2$  is a 2-approximation for metric  $k$ -CENTER.



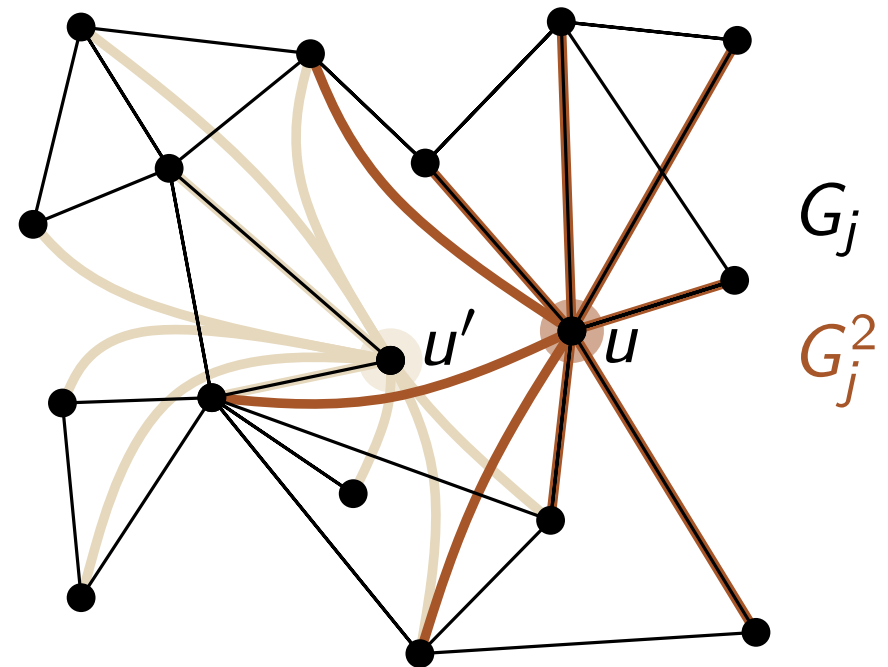
# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ . Two vertices  $u \neq v$  are adjacent in  $H^2$  iff they are within distance at most **two** in  $H$ .

**Obs.** A dominating set of size at most  $k$  in  $G_j^2$  is a 2-approximation for metric  $k$ -CENTER.

Why?





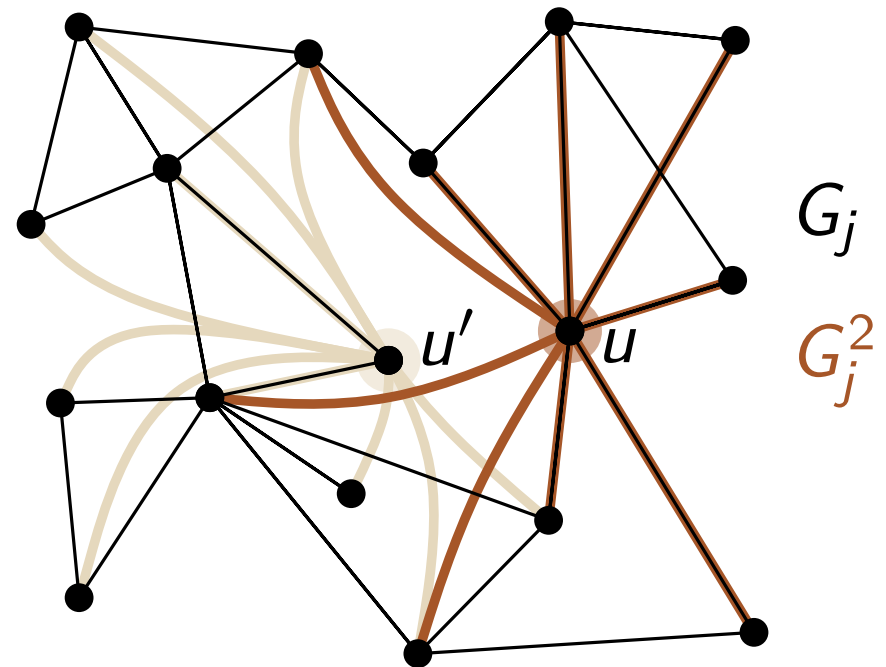
# Square of a Graph

**Idea:** Find a small dominating set in a “coarsened”  $G_j$ .

**Def.** The **square**  $H^2$  of a graph  $H$  has the same vertex set as  $H$ . Two vertices  $u \neq v$  are adjacent in  $H^2$  iff they are within distance at most **two** in  $H$ .

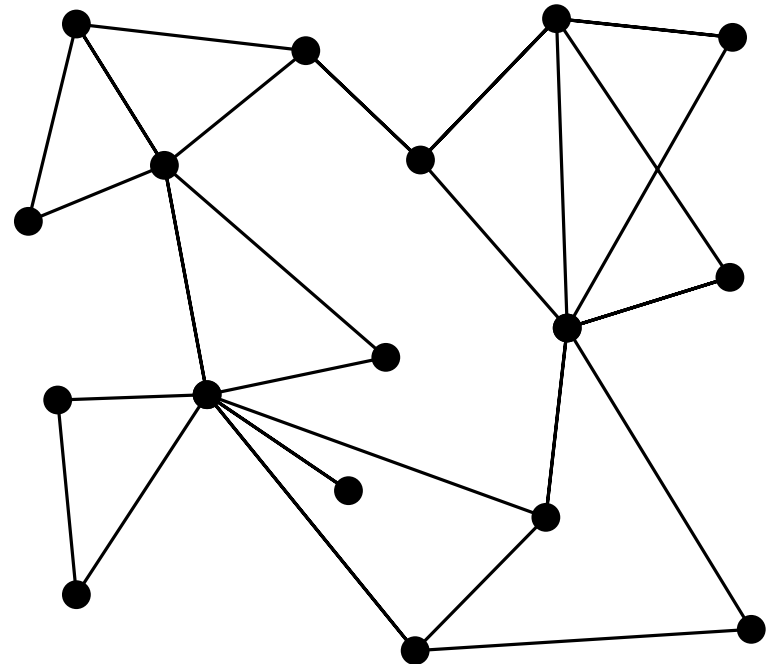
**Obs.** A dominating set of size at most  $k$  in  $G_j^2$  is a 2-approximation for metric  $k$ -CENTER.

**Why?**  $\max_{e \in E(G_j)} c(e) = \text{OPT} !$



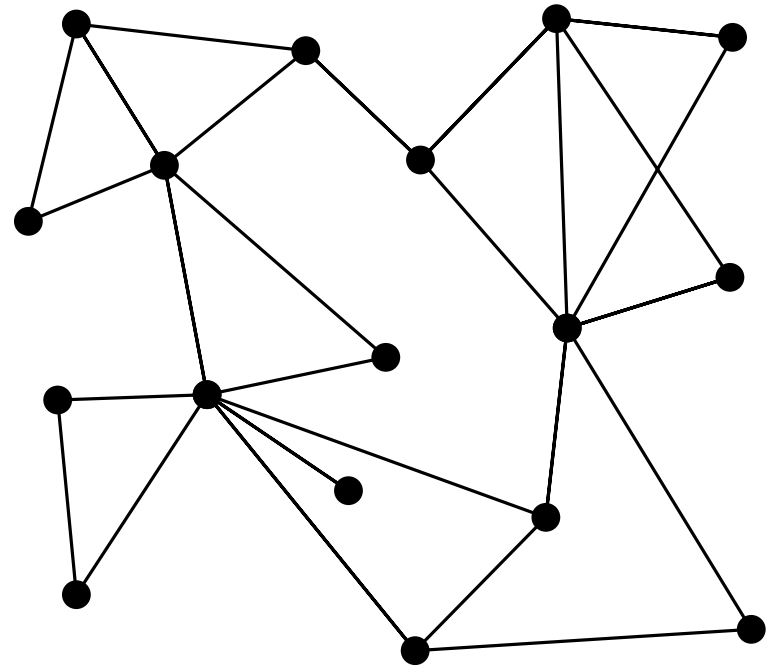
# Independent Sets

**Def.** A vertex set  $I$  in a graph is called **independent** (or **stable**) if no pair of vertices in  $I$  forms an edge.



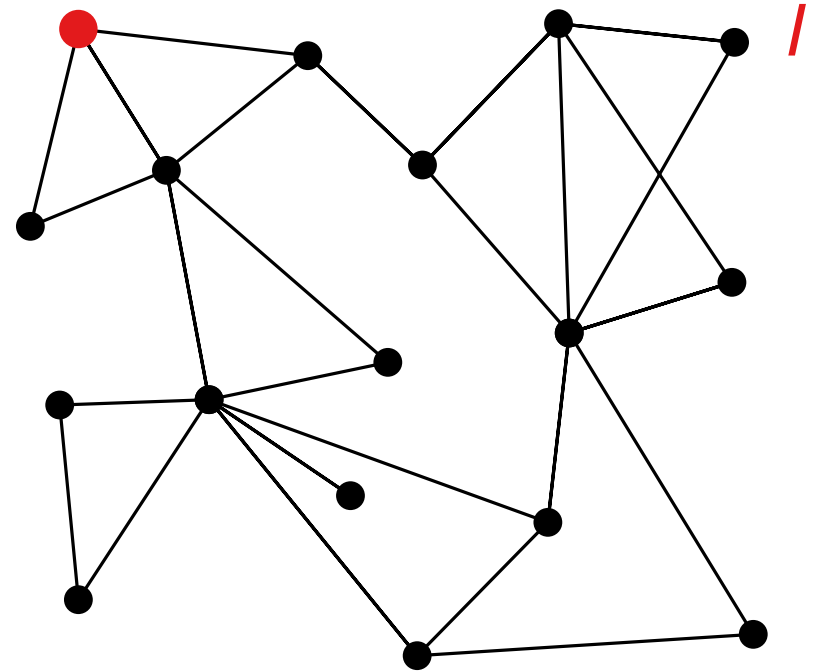
# Independent Sets

**Def.** A vertex set  $I$  in a graph is called **independent** (or **stable**) if no pair of vertices in  $I$  forms an edge. An independent set is called **maximal** if it does not have an independent superset.



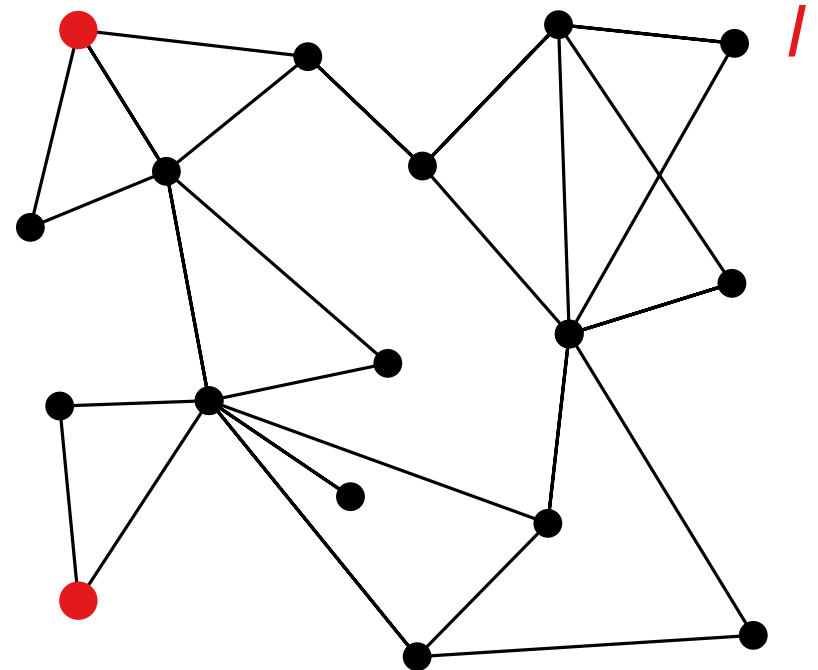
# Independent Sets

**Def.** A vertex set  $I$  in a graph is called **independent** (or **stable**) if no pair of vertices in  $I$  forms an edge. An independent set is called **maximal** if it does not have an independent superset.



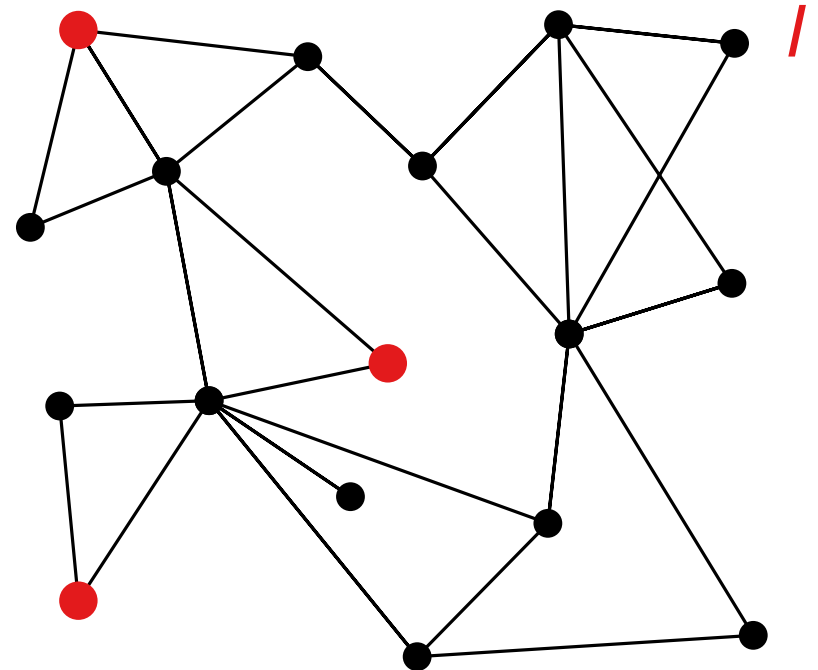
# Independent Sets

**Def.** A vertex set  $I$  in a graph is called **independent** (or **stable**) if no pair of vertices in  $I$  forms an edge. An independent set is called **maximal** if it does not have an independent superset.



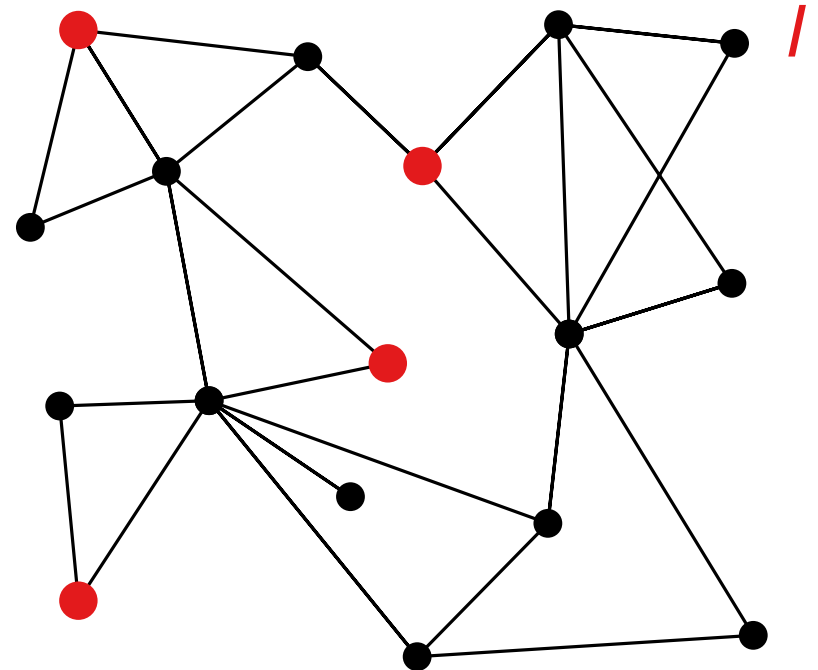
# Independent Sets

**Def.** A vertex set  $I$  in a graph is called **independent** (or **stable**) if no pair of vertices in  $I$  forms an edge. An independent set is called **maximal** if it does not have an independent superset.



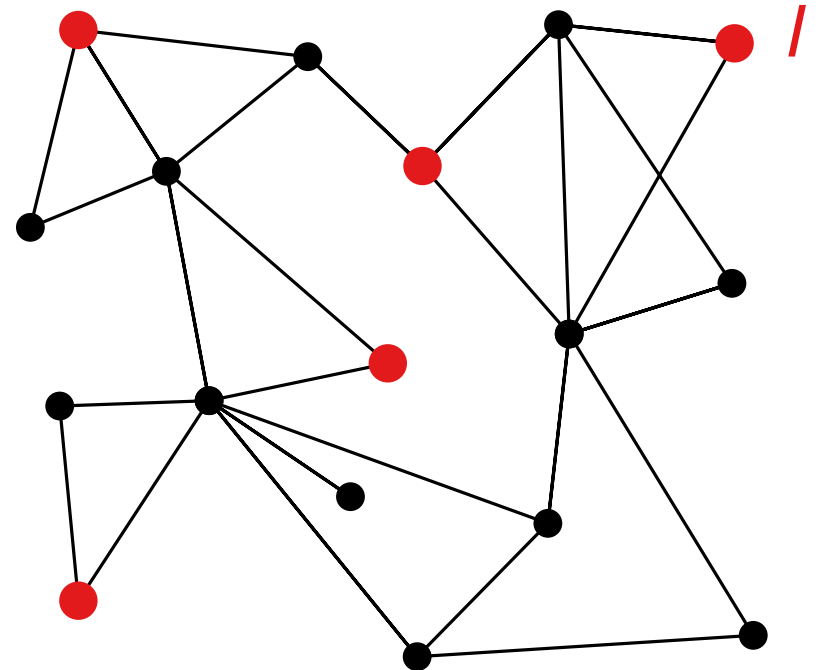
# Independent Sets

**Def.** A vertex set  $I$  in a graph is called **independent** (or **stable**) if no pair of vertices in  $I$  forms an edge. An independent set is called **maximal** if it does not have an independent superset.



# Independent Sets

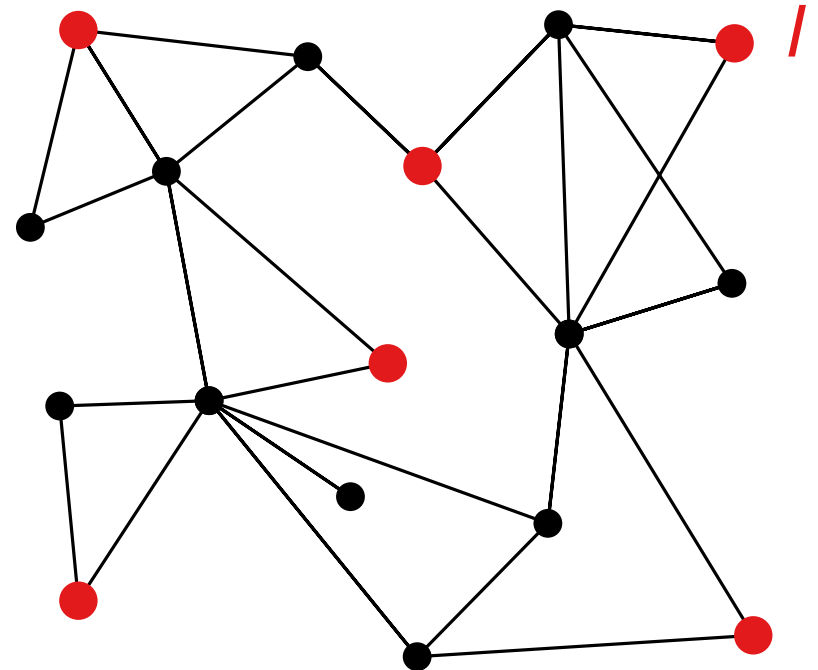
**Def.** A vertex set  $I$  in a graph is called **independent** (or **stable**) if no pair of vertices in  $I$  forms an edge. An independent set is called **maximal** if it does not have an independent superset.





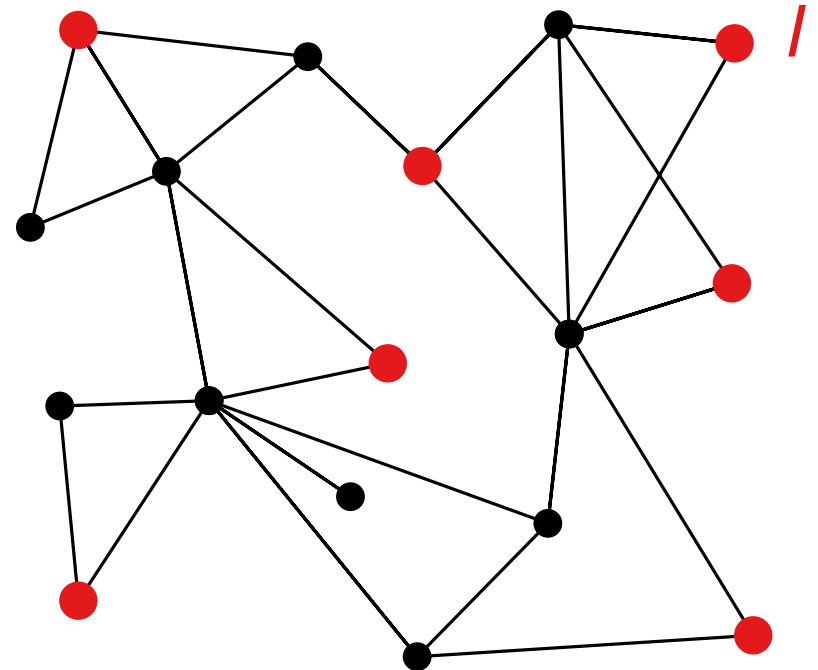
# Independent Sets

**Def.** A vertex set  $I$  in a graph is called **independent** (or **stable**) if no pair of vertices in  $I$  forms an edge. An independent set is called **maximal** if it does not have an independent superset.



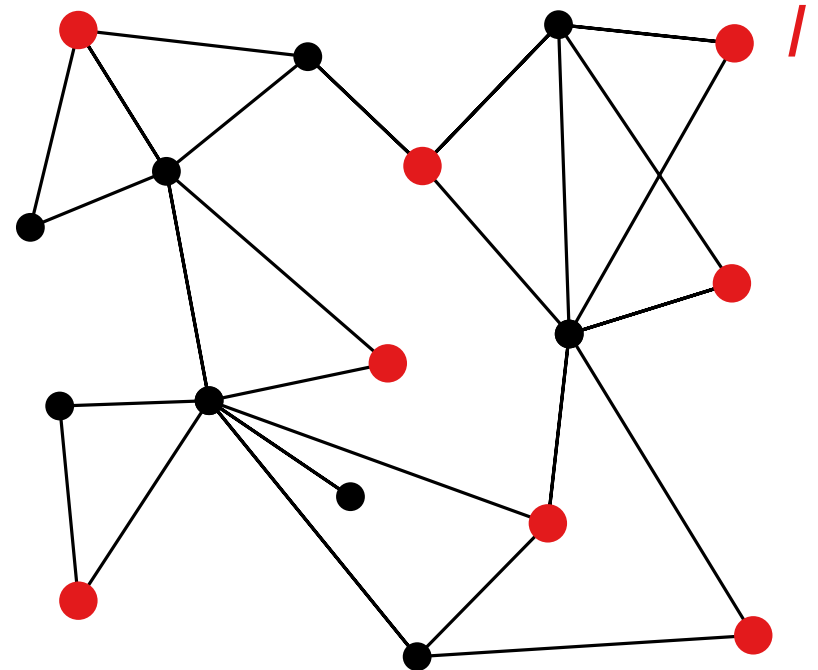
# Independent Sets

**Def.** A vertex set  $I$  in a graph is called **independent** (or **stable**) if no pair of vertices in  $I$  forms an edge. An independent set is called **maximal** if it does not have an independent superset.



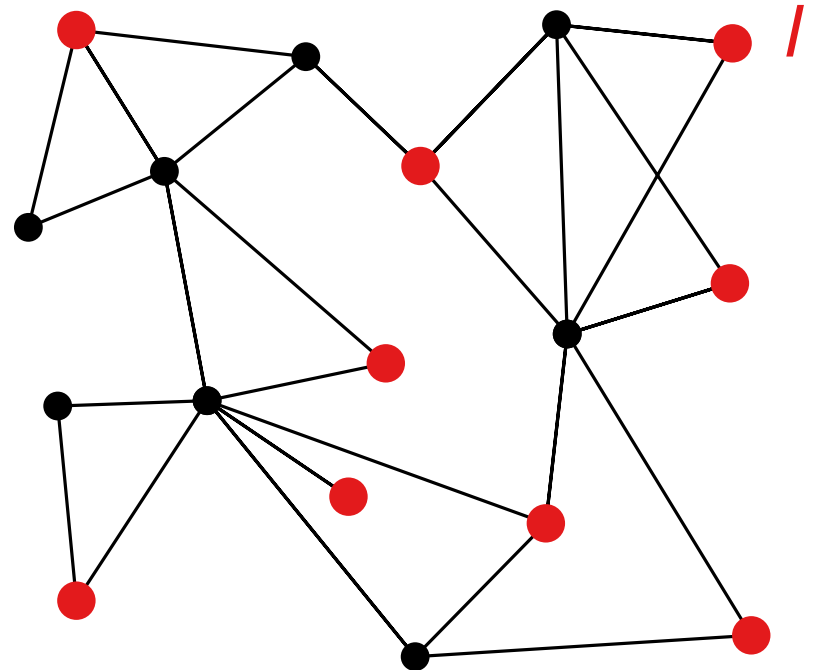
# Independent Sets

**Def.** A vertex set  $I$  in a graph is called **independent** (or **stable**) if no pair of vertices in  $I$  forms an edge. An independent set is called **maximal** if it does not have an independent superset.



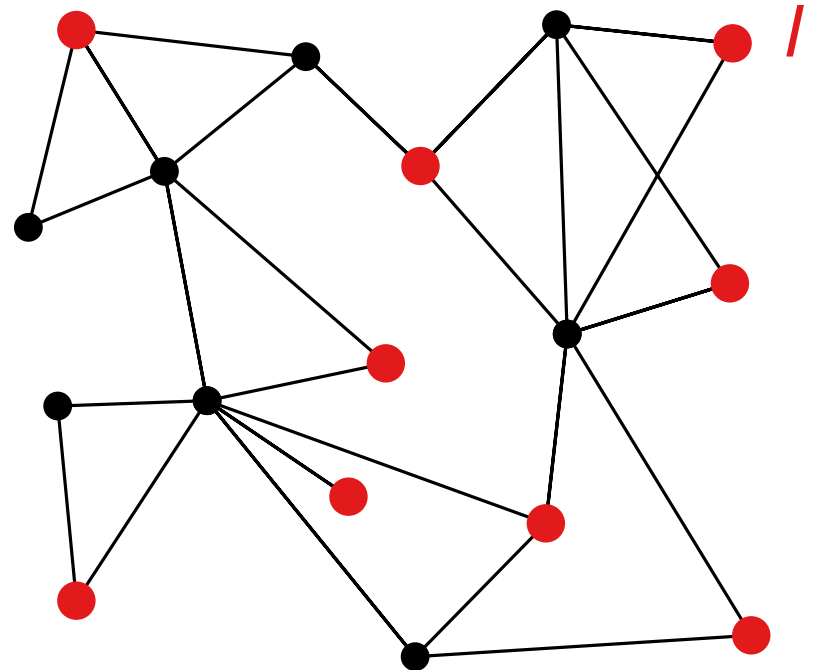
# Independent Sets

**Def.** A vertex set  $I$  in a graph is called **independent** (or **stable**) if no pair of vertices in  $I$  forms an edge. An independent set is called **maximal** if it does not have an independent superset.



# Independent Sets

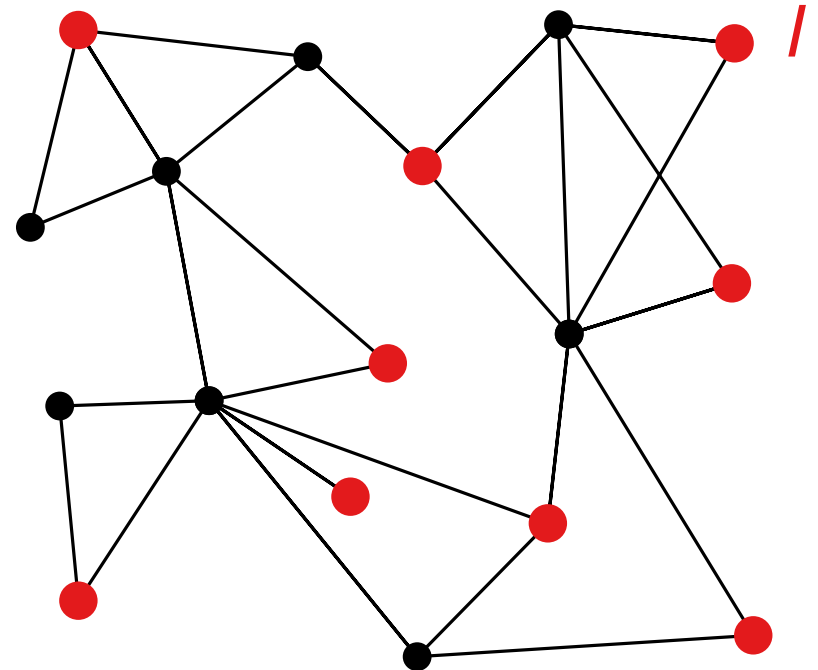
**Def.** A vertex set  $I$  in a graph is called **independent** (or **stable**) if no pair of vertices in  $I$  forms an edge. An independent set is called **maximal** if it does not have an independent superset.



# Independent Sets

**Def.** A vertex set  $I$  in a graph is called **independent** (or **stable**) if no pair of vertices in  $I$  forms an edge. An independent set is called **maximal** if it does not have an independent superset.

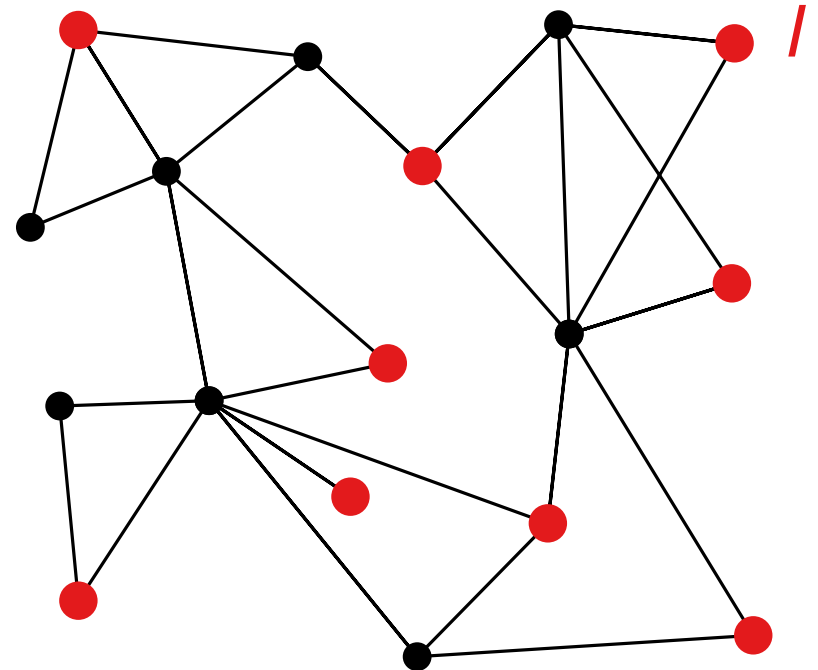
**Obs.** Maximal independent sets are



# Independent Sets

**Def.** A vertex set  $I$  in a graph is called **independent** (or **stable**) if no pair of vertices in  $I$  forms an edge. An independent set is called **maximal** if it does not have an independent superset.

**Obs.** Maximal independent sets are dominating. :-)



# Independent Sets in $H^2$

**Lemma.** For a graph  $H$  and an independent set  $I$  in  $H^2$ ,  
 $|I| \leq \frac{1}{2} |V(H)|$



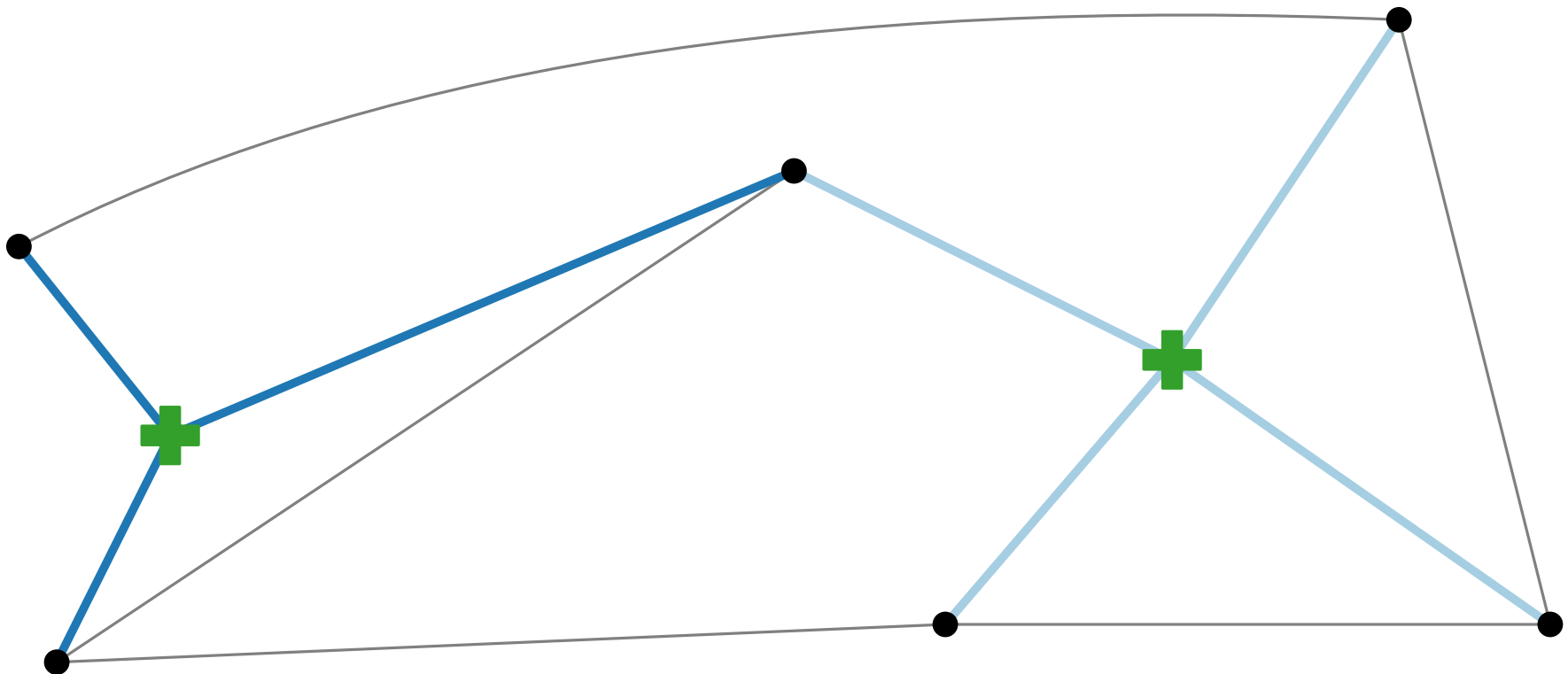
# Independent Sets in $H^2$

**Lemma.** For a graph  $H$  and an independent set  $I$  in  $H^2$ ,  
 $|I| \leq \text{dom}(H)$ .

# Independent Sets in $H^2$

**Lemma.** For a graph  $H$  and an independent set  $I$  in  $H^2$ ,  
 $|I| \leq \text{dom}(H)$ .

*Proof.* What does a dominating set of  $H$  look like in  $H^2$ ?

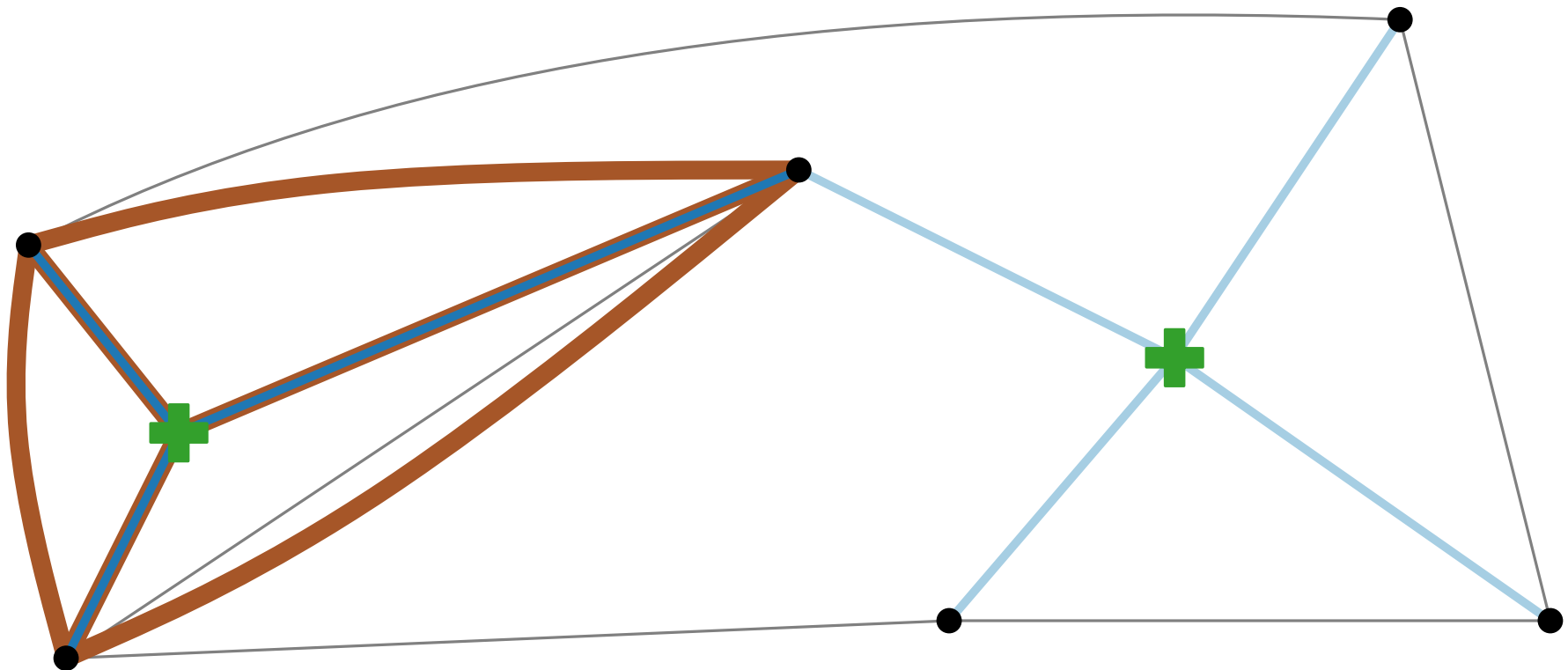


Star in  $H$

# Independent Sets in $H^2$

**Lemma.** For a graph  $H$  and an independent set  $I$  in  $H^2$ ,  
 $|I| \leq \text{dom}(H)$ .

*Proof.* What does a dominating set of  $H$  look like in  $H^2$ ?

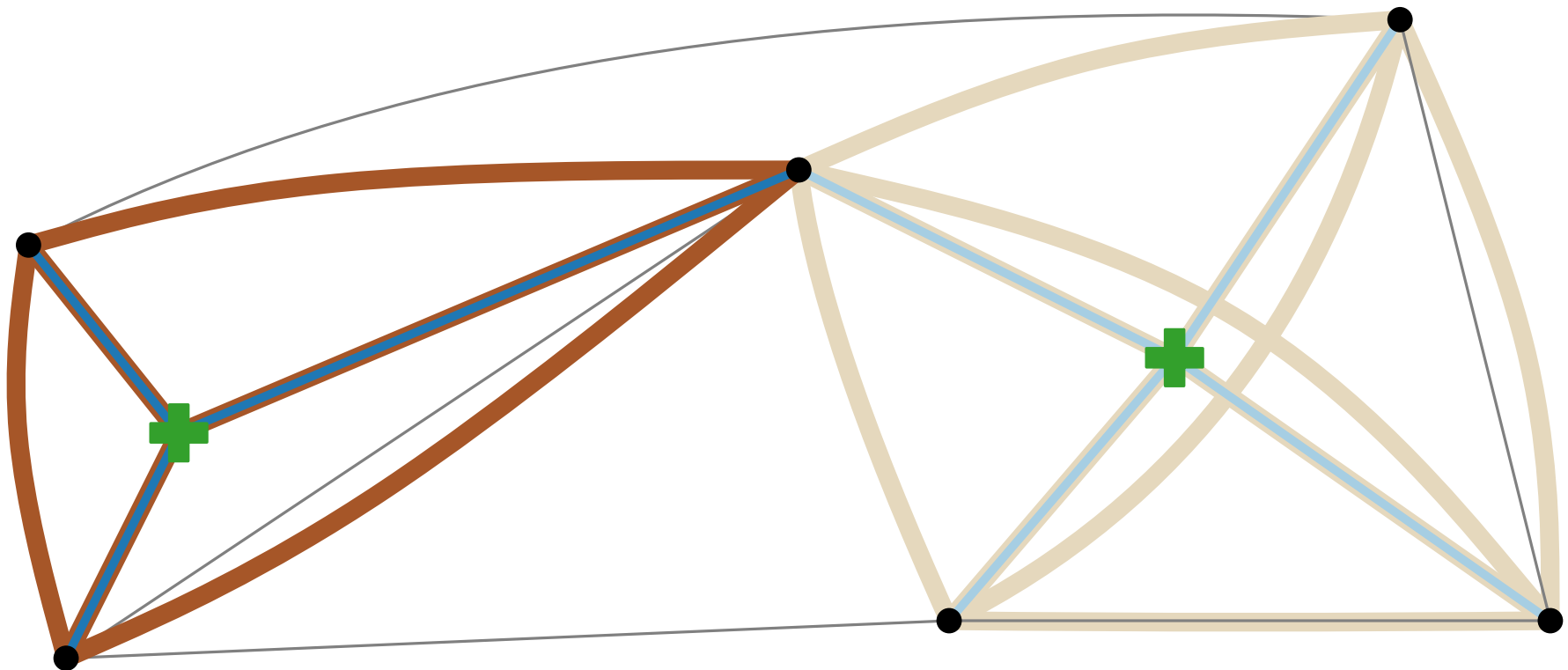


Star in  $H$

# Independent Sets in $H^2$

**Lemma.** For a graph  $H$  and an independent set  $I$  in  $H^2$ ,  
 $|I| \leq \text{dom}(H)$ .

*Proof.* What does a dominating set of  $H$  look like in  $H^2$ ?

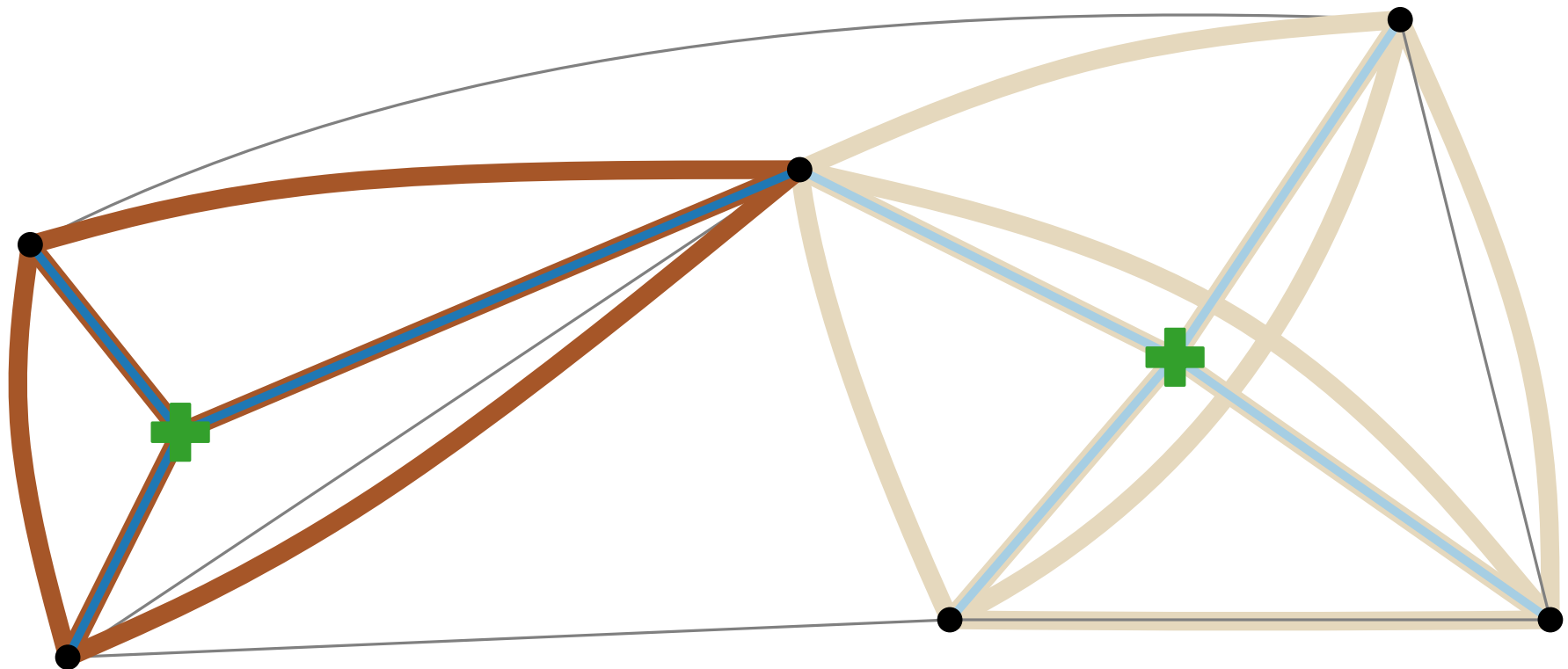


Star in  $H$

# Independent Sets in $H^2$

**Lemma.** For a graph  $H$  and an independent set  $I$  in  $H^2$ ,  
 $|I| \leq \text{dom}(H)$ .

*Proof.* What does a dominating set of  $H$  look like in  $H^2$ ?



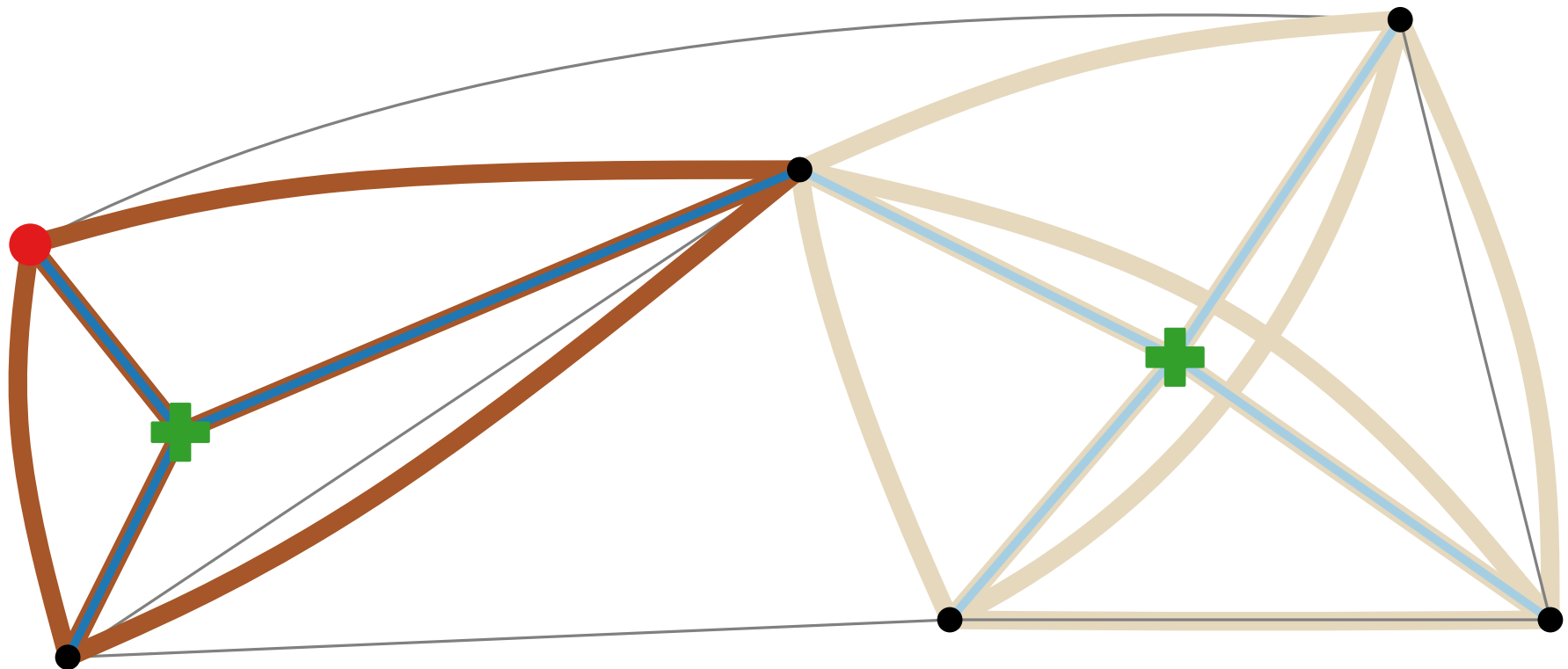
Star in  $H$

Clique in  $H^2$

# Independent Sets in $H^2$

**Lemma.** For a graph  $H$  and an independent set  $I$  in  $H^2$ ,  
 $|I| \leq \text{dom}(H)$ .

*Proof.* What does a dominating set of  $H$  look like in  $H^2$ ?



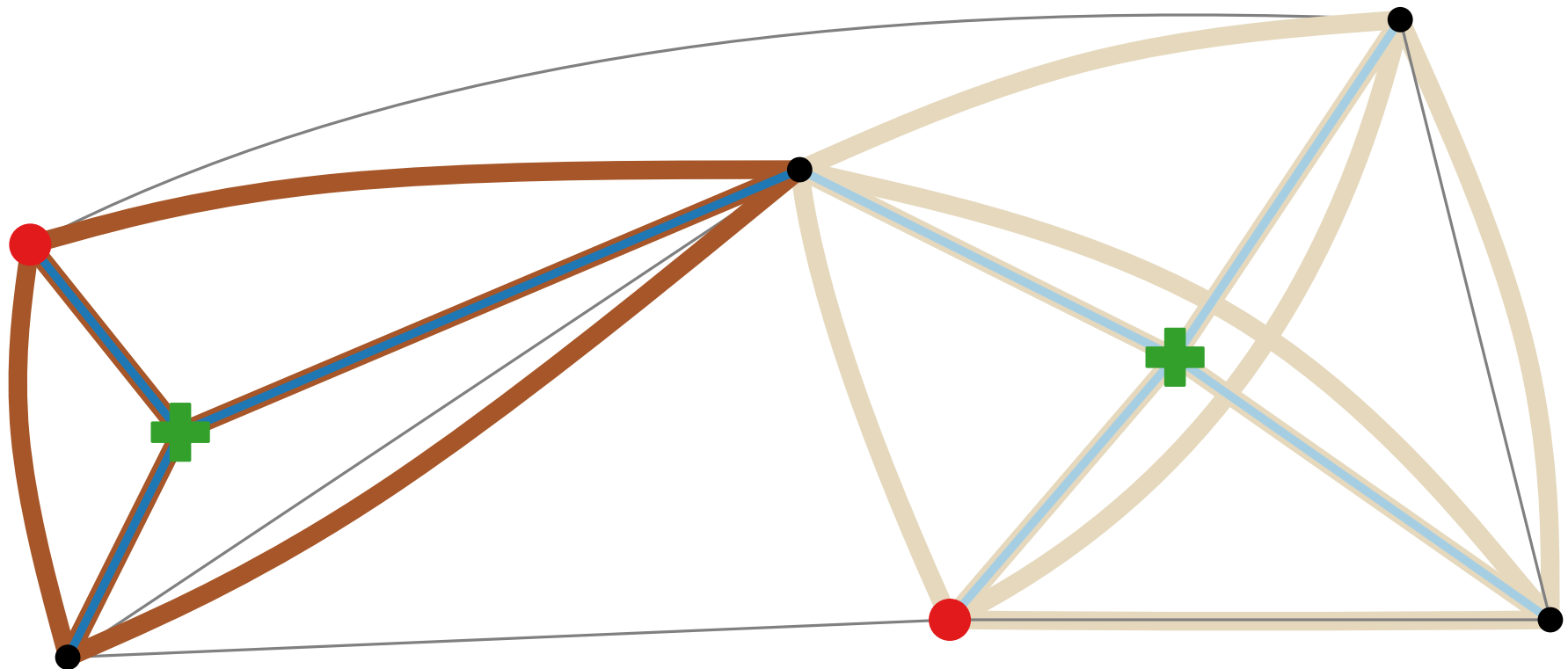
Star in  $H$

Clique in  $H^2$

# Independent Sets in $H^2$

**Lemma.** For a graph  $H$  and an independent set  $I$  in  $H^2$ ,  
 $|I| \leq \text{dom}(H)$ .

*Proof.* What does a dominating set of  $H$  look like in  $H^2$ ?



Star in  $H$

Clique in  $H^2$

# Approximation Algorithms

Lecture 6:

$k$ -CENTER via Parametric Pruning

Part IV:

Factor-2 Approximation for METRIC- $k$ -CENTER



# Factor-2 Approx. for Metric $k$ -CENTER

Metric- $k$ -CENTER( $G = (V, E; c), k$ )

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$ .

# Factor-2 Approx. for Metric $k$ -CENTER

Metric- $k$ -CENTER( $G = (V, E; c), k$ )

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$ .

**for**  $j = 1$  **to**  $m$  **do**

|

# Factor-2 Approx. for Metric $k$ -CENTER

Metric- $k$ -CENTER( $G = (V, E; c), k$ )

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$ .

**for**  $j = 1$  **to**  $m$  **do**

    Construct  $G_j^2$ .

# Factor-2 Approx. for Metric $k$ -CENTER

Metric- $k$ -CENTER( $G = (V, E; c), k$ )

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$ .

**for**  $j = 1$  **to**  $m$  **do**

    Construct  $G_j^2$ .

    Find a maximal independent set  $I_j$  in  $G_j^2$ .

# Factor-2 Approx. for Metric $k$ -CENTER

Metric- $k$ -CENTER( $G = (V, E; c)$ ,  $k$ )

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$ .

**for**  $j = 1$  **to**  $m$  **do**

    Construct  $G_j^2$ .

    Find a maximal independent set  $I_j$  in  $G_j^2$ .

**if**  $|I_j| \leq k$  **then**

**return**  $I_j$

# Factor-2 Approx. for Metric $k$ -CENTER

Metric- $k$ -CENTER( $G = (V, E; c)$ ,  $k$ )

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$ .

**for**  $j = 1$  **to**  $m$  **do**

    Construct  $G_j^2$ .

    Find a maximal independent set  $I_j$  in  $G_j^2$ .

**if**  $|I_j| \leq k$  **then**

**return**  $I_j$

**Lemma.** For  $j$  provided by the algorithm,  
it holds that  $c(e_j) \leq \text{OPT}$ .

# Factor-2 Approx. for Metric $k$ -CENTER

Metric- $k$ -CENTER( $G = (V, E; c), k$ )

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$ .

**for**  $j = 1$  **to**  $m$  **do**

    Construct  $G_j^2$ .

    Find a maximal independent set  $I_j$  in  $G_j^2$ .

**if**  $|I_j| \leq k$  **then**

**return**  $I_j$

**Lemma.** For  $j$  provided by the algorithm,  
it holds that  $c(e_j) \leq \text{OPT}$ .

**Theorem.** The above algorithm is a factor-2 approximation  
algorithm for the metric  $k$ -CENTER problem.

Can we do better ... ?

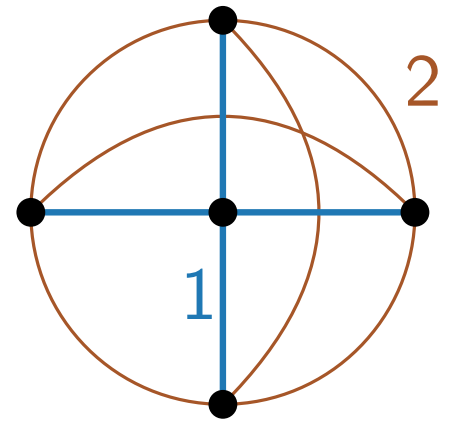


# Can we do better ... ?

What about a tight example?

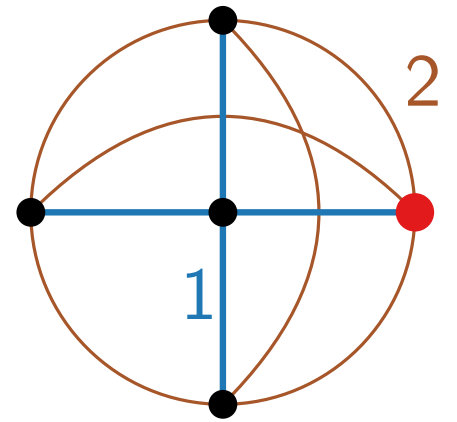
Can we do better ... ?

What about a tight example?



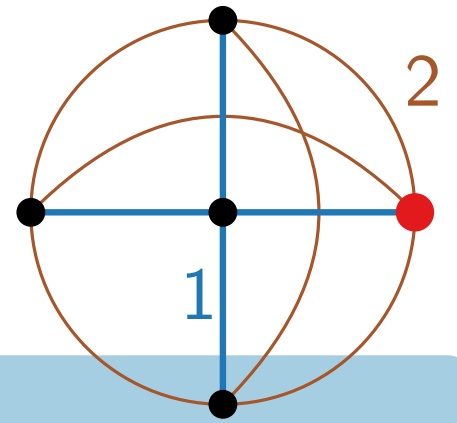
Can we do better ... ?

What about a tight example?



# Can we do better ...?

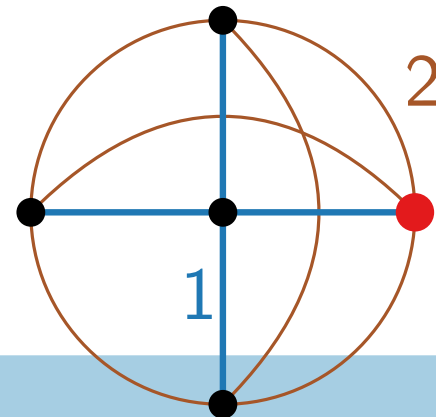
What about a tight example?



**Theorem.** Assuming  $P \neq NP$ , for no  $\varepsilon > 0$ , there is a  $(2 - \varepsilon)$ -approximation algorithm for the metric  $k$ -CENTER problem.

# Can we do better ...?

What about a tight example?

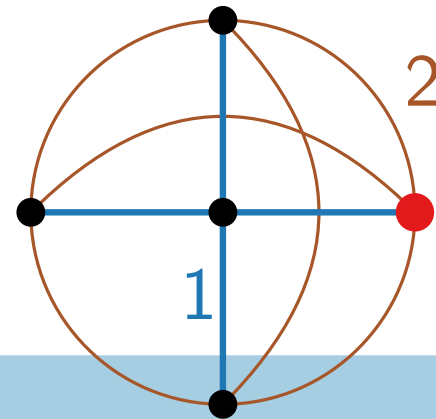


**Theorem.** Assuming  $P \neq NP$ , for no  $\varepsilon > 0$ , there is a  $(2 - \varepsilon)$ -approximation algorithm for the metric  $k$ -CENTER problem.

**Proof.** Reduce from dominating set to metric  $k$ -CENTER.

# Can we do better ...?

What about a tight example?

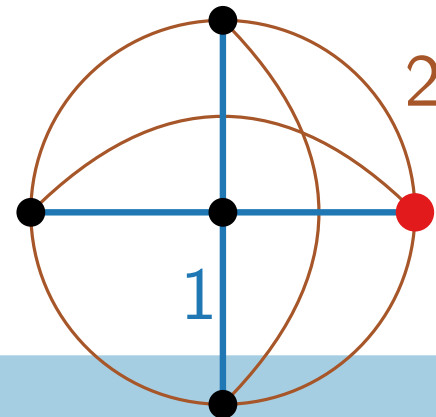


**Theorem.** Assuming  $P \neq NP$ , for no  $\varepsilon > 0$ , there is a  $(2 - \varepsilon)$ -approximation algorithm for the metric  $k$ -CENTER problem.

**Proof.** Reduce from dominating set to metric  $k$ -CENTER.  
Given graph  $G$  and integer  $k$ ,

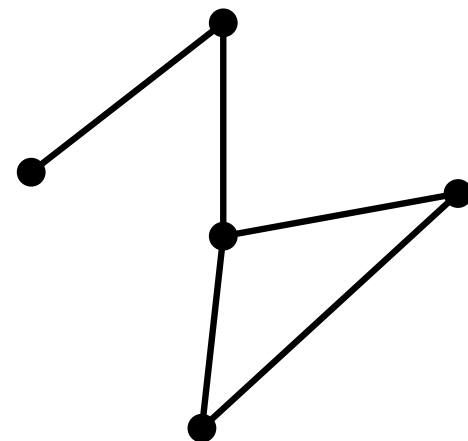
# Can we do better ...?

What about a tight example?



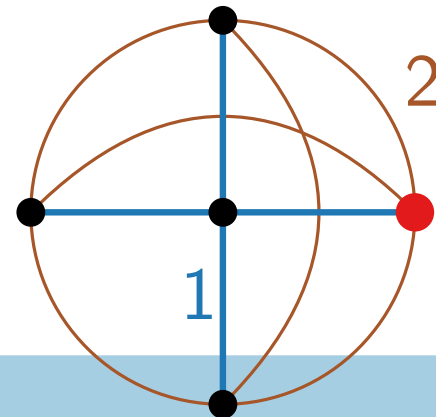
**Theorem.** Assuming  $P \neq NP$ , for no  $\varepsilon > 0$ , there is a  $(2 - \varepsilon)$ -approximation algorithm for the metric  $k$ -CENTER problem.

**Proof.** Reduce from dominating set to metric  $k$ -CENTER. Given graph  $G$  and integer  $k$ ,



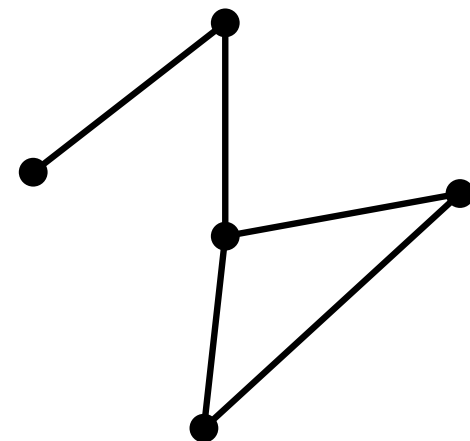
# Can we do better ... ?

What about a tight example?



**Theorem.** Assuming  $P \neq NP$ , for no  $\varepsilon > 0$ , there is a  $(2 - \varepsilon)$ -approximation algorithm for the metric  $k$ -CENTER problem.

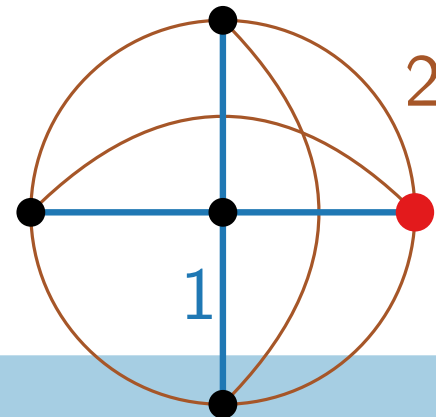
**Proof.** Reduce from dominating set to metric  $k$ -CENTER. Given graph  $G$  and integer  $k$ , construct complete graph  $G'$  with  $V(G') = V(G)$ ,  $E(G') = E(G) \cup E'$ .





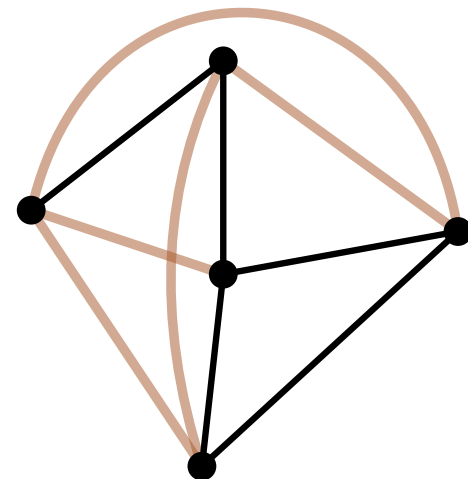
# Can we do better ... ?

What about a tight example?



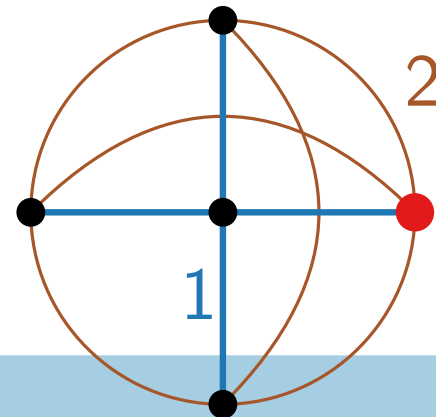
**Theorem.** Assuming  $P \neq NP$ , for no  $\varepsilon > 0$ , there is a  $(2 - \varepsilon)$ -approximation algorithm for the metric  $k$ -CENTER problem.

**Proof.** Reduce from dominating set to metric  $k$ -CENTER. Given graph  $G$  and integer  $k$ , construct complete graph  $G'$  with  $V(G') = V(G)$ ,  $E(G') = E(G) \cup E'$ .



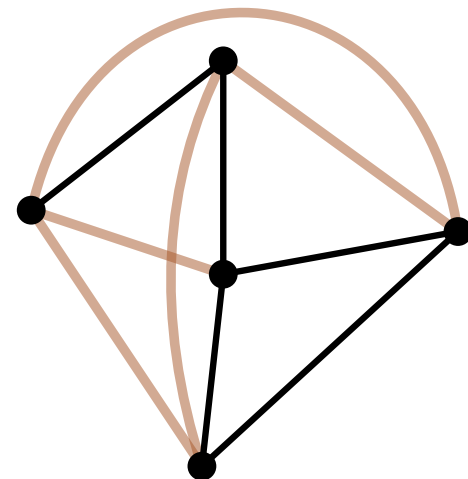
# Can we do better ...?

What about a tight example?



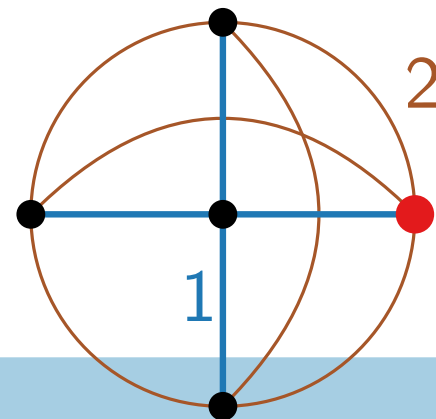
**Theorem.** Assuming  $P \neq NP$ , for no  $\varepsilon > 0$ , there is a  $(2 - \varepsilon)$ -approximation algorithm for the metric  $k$ -CENTER problem.

**Proof.** Reduce from dominating set to metric  $k$ -CENTER. Given graph  $G$  and integer  $k$ , construct complete graph  $G'$  with  $V(G') = V(G)$ ,  $E(G') = E(G) \cup E'$ .  
with  $c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$



# Can we do better ... ?

What about a tight example?

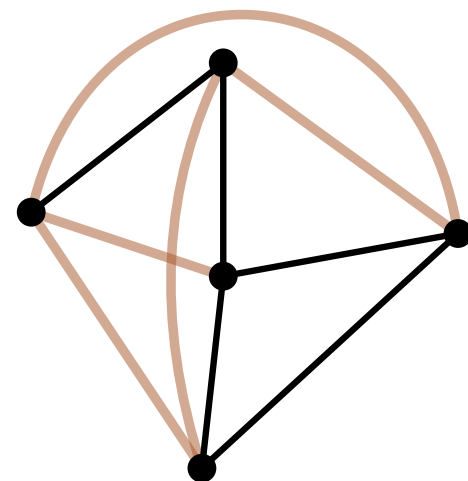


**Theorem.** Assuming  $P \neq NP$ , for no  $\varepsilon > 0$ , there is a  $(2 - \varepsilon)$ -approximation algorithm for the metric  $k$ -CENTER problem.

**Proof.** Reduce from dominating set to metric  $k$ -CENTER. Given graph  $G$  and integer  $k$ , construct complete graph  $G'$  with  $V(G') = V(G)$ ,  $E(G') = E(G) \cup E'$ .

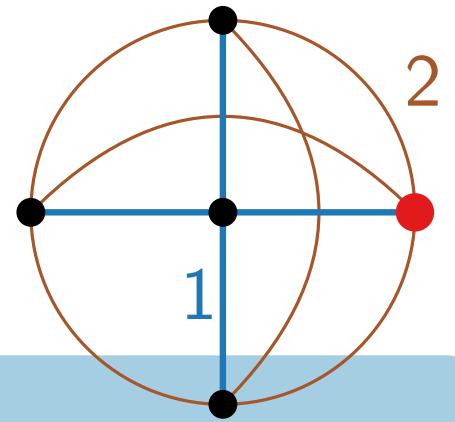
$$\text{with } c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$$

Let  $S$  be a metric  $k$ -center of  $G'$ .



# Can we do better ... ?

What about a tight example?

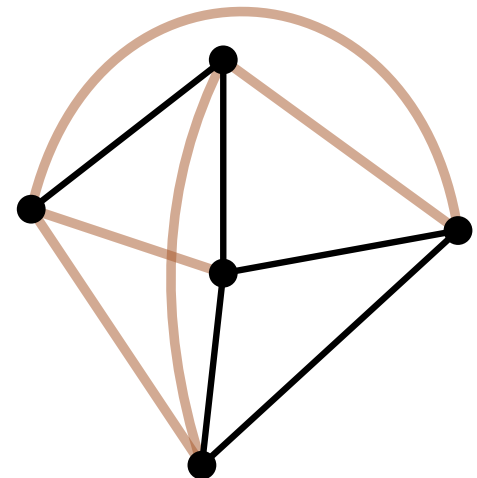


**Theorem.** Assuming  $P \neq NP$ , for no  $\varepsilon > 0$ , there is a  $(2 - \varepsilon)$ -approximation algorithm for the metric  $k$ -CENTER problem.

**Proof.** Reduce from dominating set to metric  $k$ -CENTER. Given graph  $G$  and integer  $k$ , construct complete graph  $G'$  with  $V(G') = V(G)$ ,  $E(G') = E(G) \cup E'$ .

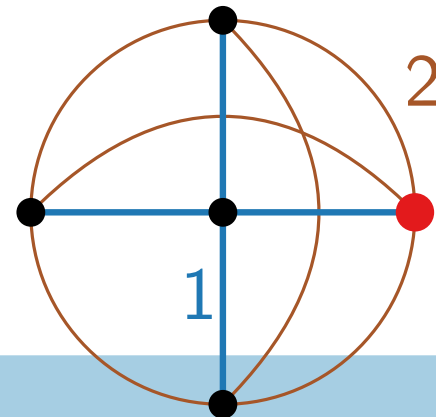
$$\text{with } c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$$

Let  $S$  be a metric  $k$ -center of  $G'$ .  
If  $\text{dom}(G) \leq k$ , then  $\text{cost}(S) = 1$ .



# Can we do better ... ?

What about a tight example?

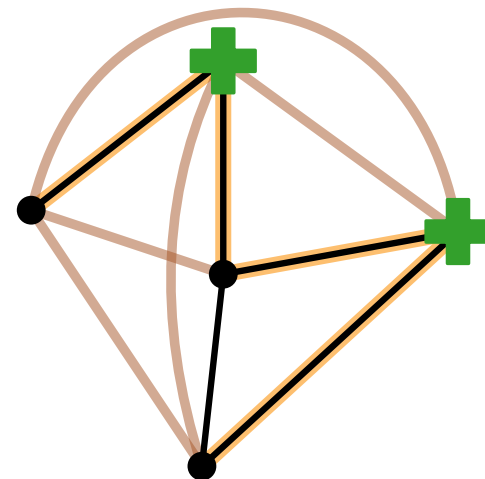


**Theorem.** Assuming  $P \neq NP$ , for no  $\varepsilon > 0$ , there is a  $(2 - \varepsilon)$ -approximation algorithm for the metric  $k$ -CENTER problem.

**Proof.** Reduce from dominating set to metric  $k$ -CENTER. Given graph  $G$  and integer  $k$ , construct complete graph  $G'$  with  $V(G') = V(G)$ ,  $E(G') = E(G) \cup E'$ .

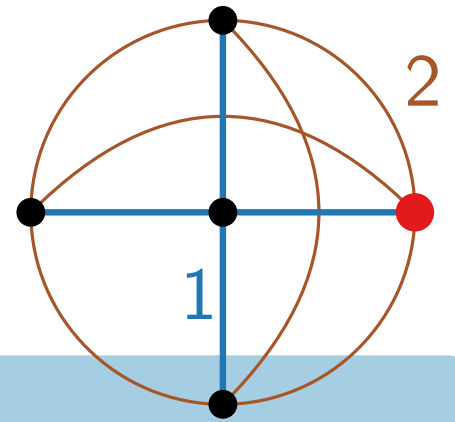
$$\text{with } c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$$

Let  $S$  be a metric  $k$ -center of  $G'$ .  
If  $\text{dom}(G) \leq k$ , then  $\text{cost}(S) = 1$ .



# Can we do better ... ?

What about a tight example?

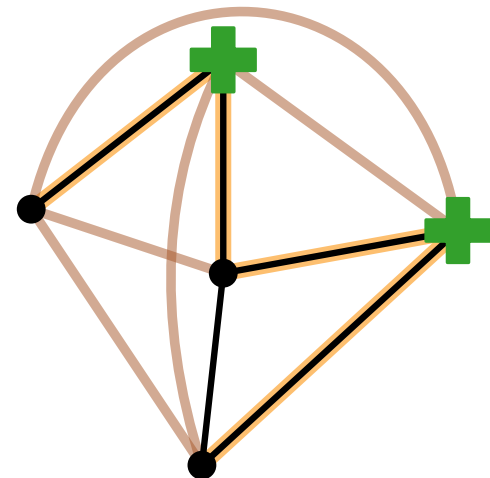


**Theorem.** Assuming  $P \neq NP$ , for no  $\varepsilon > 0$ , there is a  $(2 - \varepsilon)$ -approximation algorithm for the metric  $k$ -CENTER problem.

**Proof.** Reduce from dominating set to metric  $k$ -CENTER. Given graph  $G$  and integer  $k$ , construct complete graph  $G'$  with  $V(G') = V(G)$ ,  $E(G') = E(G) \cup E'$ .

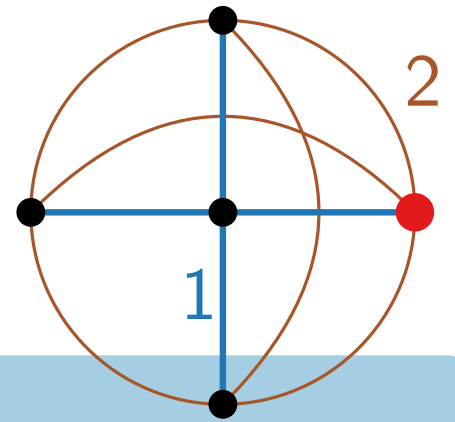
$$\text{with } c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$$

Let  $S$  be a metric  $k$ -center of  $G'$ .  
If  $\text{dom}(G) \leq k$ , then  $\text{cost}(S) = 1$ .  
If  $\text{dom}(G) > k$ , then  $\text{cost}(S) = 2$ .



# Can we do better ... ?

What about a tight example?

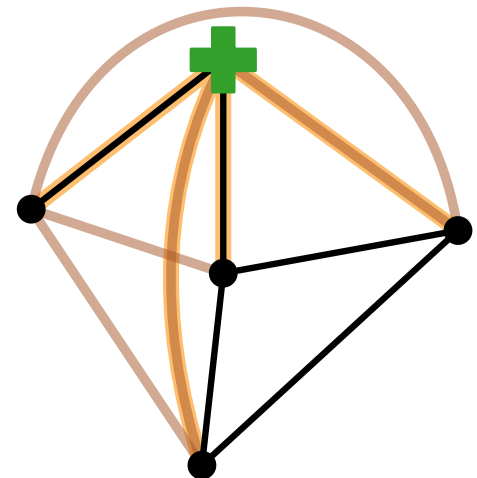


**Theorem.** Assuming  $P \neq NP$ , for no  $\varepsilon > 0$ , there is a  $(2 - \varepsilon)$ -approximation algorithm for the metric  $k$ -CENTER problem.

**Proof.** Reduce from dominating set to metric  $k$ -CENTER. Given graph  $G$  and integer  $k$ , construct complete graph  $G'$  with  $V(G') = V(G)$ ,  $E(G') = E(G) \cup E'$ .

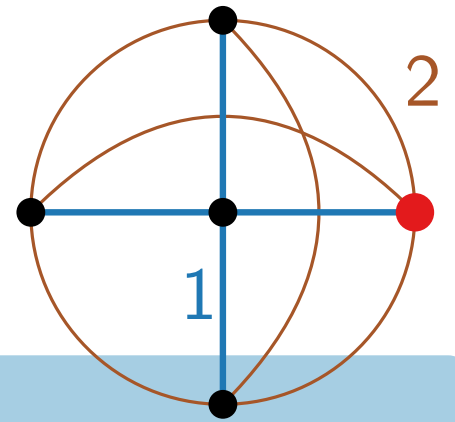
$$\text{with } c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$$

Let  $S$  be a metric  $k$ -center of  $G'$ .  
If  $\text{dom}(G) \leq k$ , then  $\text{cost}(S) = 1$ .  
If  $\text{dom}(G) > k$ , then  $\text{cost}(S) = 2$ .



# Can we do better ... ?

What about a tight example?

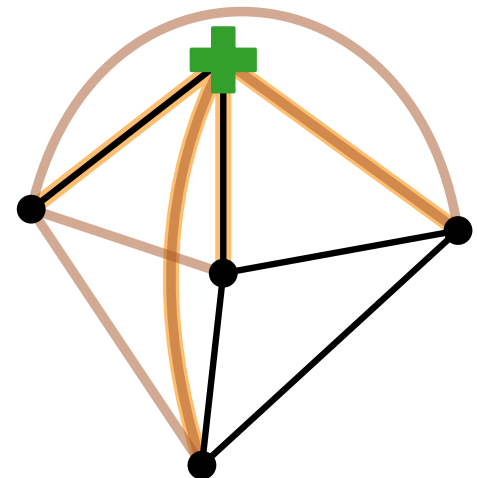


**Theorem.** Assuming  $P \neq NP$ , for no  $\varepsilon > 0$ , there is a  $(2 - \varepsilon)$ -approximation algorithm for the metric  $k$ -CENTER problem.

**Proof.** Reduce from dominating set to metric  $k$ -CENTER. Given graph  $G$  and integer  $k$ , construct complete graph  $G'$  with  $V(G') = V(G)$ ,  $E(G') = E(G) \cup E'$ .

$$\text{with } c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$$

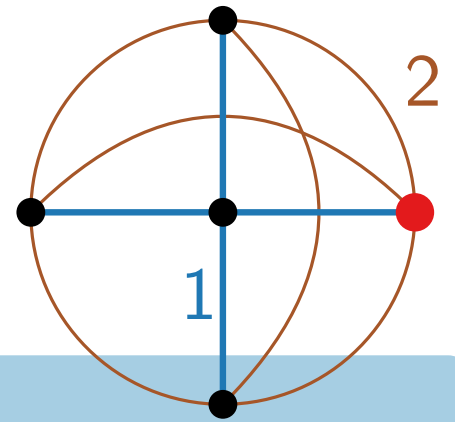
Let  $S$  be a metric  $k$ -center of  $G'$ .  
If  $\text{dom}(G) \leq k$ , then  $\text{cost}(S) = 1$ .  
If  $\text{dom}(G) > k$ , then  $\text{cost}(S) = 2$ .





# Can we do better ... ?

What about a tight example?



**Theorem.** Assuming  $P \neq NP$ , for no  $\varepsilon > 0$ , there is a  $(2 - \varepsilon)$ -approximation algorithm for the **metric**  $k$ -CENTER problem.

**Proof.** Reduce from dominating set to metric  $k$ -CENTER. Given graph  $G$  and integer  $k$ , construct complete graph  $G'$  with  $V(G') = V(G)$ ,  $E(G') = E(G) \cup E'$ .

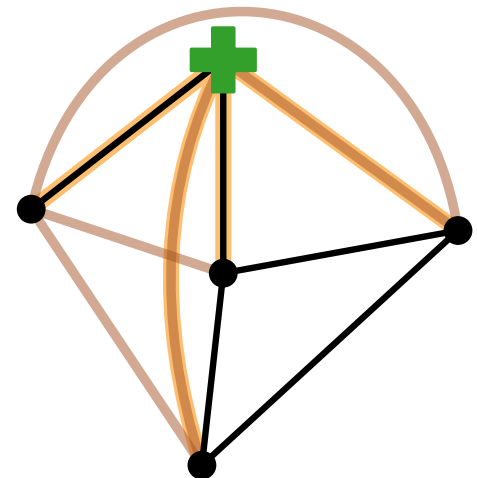
with  $c(e) = \begin{cases} 1, & \text{if } e \in E \\ 2, & \text{if } e \in E' \end{cases}$

$\Delta$ -inequality holds

Let  $S$  be a metric  $k$ -center of  $G'$ .

If  $\text{dom}(G) \leq k$ , then  $\text{cost}(S) = 1$ .

If  $\text{dom}(G) > k$ , then  $\text{cost}(S) = 2$ .



# Approximation Algorithms

Lecture 6:

$k$ -CENTER via Parametric Pruning

Part V:

METRIC-WEIGHTED-CENTER

# METRIC- $k$ -CENTER

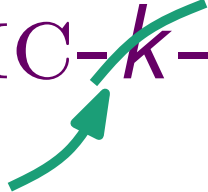
**Given:** A complete graph  $G$  with metric edge costs  $c: E(G) \rightarrow \mathbb{Q}_{\geq 0}$  and an integer  $k \leq |V|$ .

For  $S \subseteq V(G)$ ,  
 $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .

**Find:** A  $k$ -element vertex set  $S$  such that  $\text{cost}(S) := \max_{v \in V(G)} c(v, S)$  is minimized.

# METRIC- ~~$k$~~ -CENTER

WEIGHTED



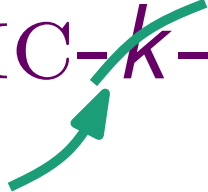
**Given:** A complete graph  $G$  with metric edge costs  $c: E(G) \rightarrow \mathbb{Q}_{\geq 0}$  and an integer  $k \leq |V|$ .

For  $S \subseteq V(G)$ ,  
 $c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .

**Find:** A  $k$ -element vertex set  $S$  such that  $\text{cost}(S) := \max_{v \in V(G)} c(v, S)$  is minimized.

# METRIC- ~~$k$~~ -CENTER

WEIGHTED



**Given:** A complete graph  $G$  with metric edge costs  $c: E(G) \rightarrow \mathbb{Q}_{\geq 0}$  and ~~an integer  $k \leq |V|$~~ , vertex weights  $w: V \rightarrow \mathbb{Q}_{\geq 0}$ , and a budget  $W \in \mathbb{Q}_+$

For  $S \subseteq V(G)$ ,

$c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .

**Find:** A  $k$ -element vertex set  $S$  such that  $\text{cost}(S) := \max_{v \in V(G)} c(v, S)$  is minimized.

# METRIC- ~~$k$~~ -CENTER

WEIGHTED

**Given:** A complete graph  $G$  with metric edge costs  $c: E(G) \rightarrow \mathbb{Q}_{\geq 0}$  and ~~an integer  $k \leq |V|$~~ , vertex weights  $w: V \rightarrow \mathbb{Q}_{\geq 0}$ , and a budget  $W \in \mathbb{Q}_+$

For  $S \subseteq V(G)$ ,

$c(v, S)$  is the cost of the cheapest edge from  $v$  to a vertex in  $S$ .

vertex set  $S$  of weight at most  $W$

**Find:** A  ~~$k$ -element~~ vertex set  $S$  such that  $\text{cost}(S) := \max_{v \in V(G)} c(v, S)$  is minimized.

# Algorithm for the Weighted Version

Algorithm Metric- $k$

CENTER

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$

**for**  $j = 1$  **to**  $m$  **do**

    Construct  $G_j^2$

    Find a maximal independent set  $I_j$  in  $G_j^2$

**if**  $|I_j| \leq k$  **then**

**return**  $I_j$

# Algorithm for the Weighted Version

Algorithm Metric-**Weighted**-CENTER

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$

**for**  $j = 1$  **to**  $m$  **do**

    Construct  $G_j^2$

    Find a maximal independent set  $I_j$  in  $G_j^2$

**if**  $|I_j| \leq k$  **then**

**return**  $I_j$



# Algorithm for the Weighted Version

Algorithm Metric-**Weighted**-CENTER

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$

**for**  $j = 1$  **to**  $m$  **do**


    Construct  $G_j^2$

    Find a maximal independent set  $I_j$  in  $G_j^2$

**if**  $|I_j| \leq k$  **then**

**return**  $I_j$

what about the weights?



# Algorithm for the Weighted Version

Algorithm Metric-**Weighted**-CENTER

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$

**for**  $j = 1$  **to**  $m$  **do**

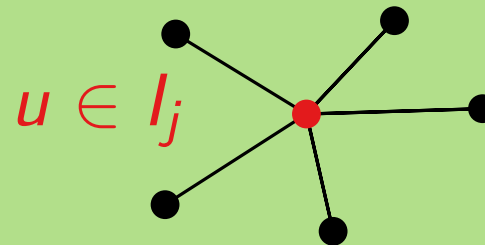
Construct  $G_j^2$

Find a maximal independent set  $I_j$  in  $G_j^2$

**if**  $|I_j| \leq k$  **then**

**return**  $I_j$

what about the weights?



# Algorithm for the Weighted Version

Algorithm Metric-**Weighted**-CENTER

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$

**for**  $j = 1$  **to**  $m$  **do**

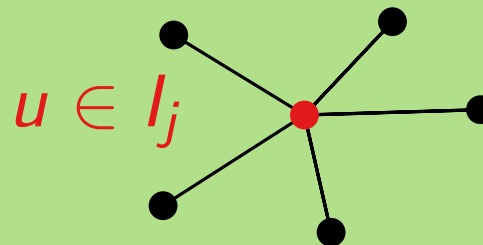
    Construct  $G_j^2$

    Find a maximal independent set  $I_j$  in  $G_j^2$

**if**  $|I_j| \leq k$  **then**

**return**  $I_j$

what about the weights?



$s_j(u) :=$  lightest node in  $N_{G_j}(u) \cup \{u\}$

# Algorithm for the Weighted Version

Algorithm Metric-**Weighted**-CENTER

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$

**for**  $j = 1$  **to**  $m$  **do**

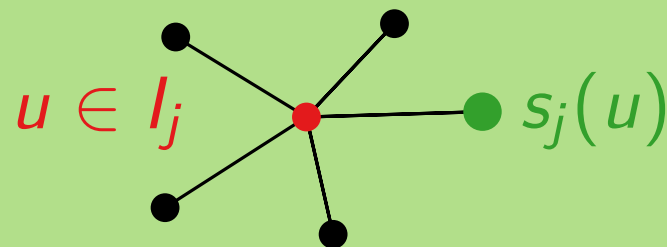
Construct  $G_j^2$

Find a maximal independent set  $I_j$  in  $G_j^2$

**if**  $|I_j| \leq k$  **then**

**return**  $I_j$

what about the weights?



$s_j(u) := \text{lightest node in } N_{G_j}(u) \cup \{u\}$

# Algorithm for the Weighted Version

Algorithm Metric-**Weighted**-CENTER

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$

**for**  $j = 1$  **to**  $m$  **do**

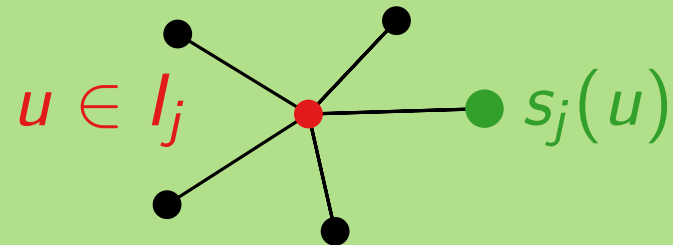
    Construct  $G_j^2$

    Find a maximal independent set  $I_j$  in  $G_j^2$

    Compute  $S_j := \{s_j(u) \mid u \in I_j\}$

**if**  $|I_j| \leq k$  **then**

**return**  $I_j$



$s_j(u) := \text{lightest node in } N_{G_j}(u) \cup \{u\}$

# Algorithm for the Weighted Version

Algorithm Metric-**Weighted**-CENTER

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$

**for**  $j = 1$  **to**  $m$  **do**

Construct  $G_j^2$

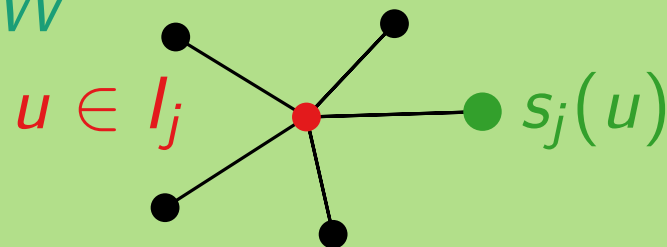
Find a maximal independent set  $I_j$  in  $G_j^2$

Compute  $S_j := \{s_j(u) \mid u \in I_j\}$

**if**  $|I_j| \leq k$  **then**

**return**  $I_j$

$w(S_j) \leq W$



$s_j(u) :=$  lightest node in  $N_{G_j}(u) \cup \{u\}$

# Algorithm for the Weighted Version

Algorithm Metric-**Weighted**-CENTER

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$

**for**  $j = 1$  **to**  $m$  **do**

Construct  $G_j^2$

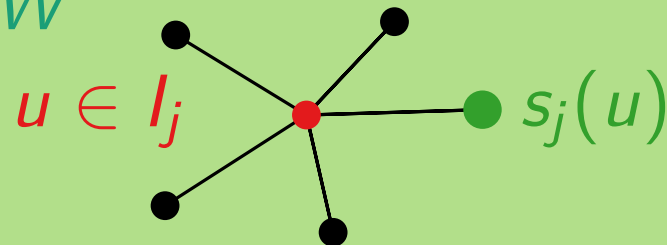
Find a maximal independent set  $I_j$  in  $G_j^2$

Compute  $S_j := \{s_j(u) \mid u \in I_j\}$

**if**  $|I_j| \leq k$  **then**

**return**  $I_j$

$w(S_j) \leq W$



$s_j(u) :=$  lightest node in  $N_{G_j}(u) \cup \{u\}$

# Algorithm for the Weighted Version

Algorithm Metric-**Weighted**-CENTER

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$

**for**  $j = 1$  **to**  $m$  **do**

Construct  $G_j^2$

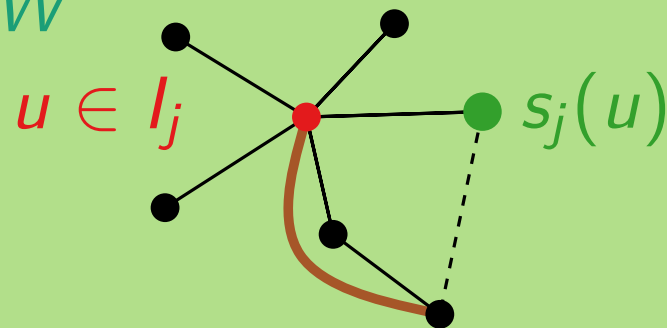
Find a maximal independent set  $I_j$  in  $G_j^2$

Compute  $S_j := \{s_j(u) \mid u \in I_j\}$

**if**  $|I_j| \leq k$  **then**

**return**  $I_j, S_j$

$w(S_j) \leq W$



$s_j(u) := \text{lightest node in } N_{G_j}(u) \cup \{u\}$



# Algorithm for the Weighted Version

Algorithm Metric-**Weighted**-CENTER

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$

**for**  $j = 1$  **to**  $m$  **do**

Construct  $G_j^2$

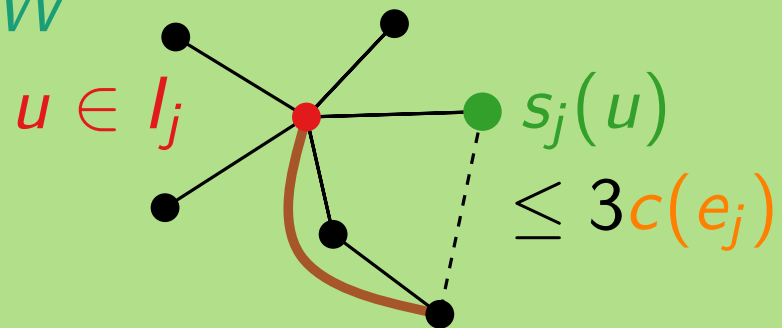
Find a maximal independent set  $I_j$  in  $G_j^2$

Compute  $S_j := \{s_j(u) \mid u \in I_j\}$

**if**  $|I_j| \leq k$  **then**

**return**  $I_j, S_j$

$w(S_j) \leq W$



$s_j(u) :=$  lightest node in  $N_{G_j}(u) \cup \{u\}$

# Algorithm for the Weighted Version

Algorithm Metric-**Weighted**-CENTER

Sort the edges of  $G$  by cost:  $c(e_1) \leq \dots \leq c(e_m)$

**for**  $j = 1$  **to**  $m$  **do**

Construct  $G_j^2$

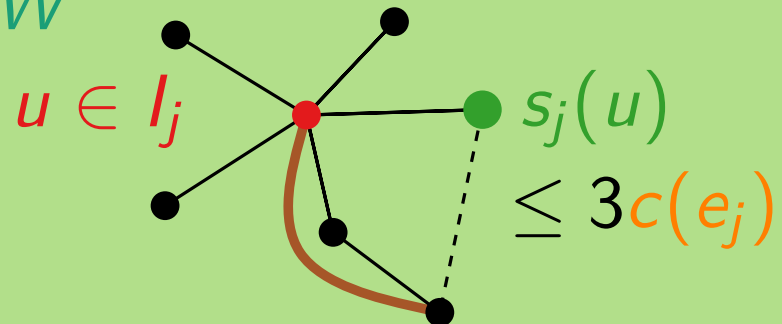
Find a maximal independent set  $I_j$  in  $G_j^2$

Compute  $S_j := \{s_j(u) \mid u \in I_j\}$

**if**  $|I_j| \leq k$  **then**

**return**  $I_j, S_j$

$w(S_j) \leq W$



$s_j(u) :=$  lightest node in  $N_{G_j}(u) \cup \{u\}$

**Theorem.** The above is a factor-3 approximation algorithm for METRIC-WEIGHTED-CENTER.

# Tight Example...?

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.

# Tight Example...?

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.

Consider  $W = 3$ .

# Tight Example...?

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.

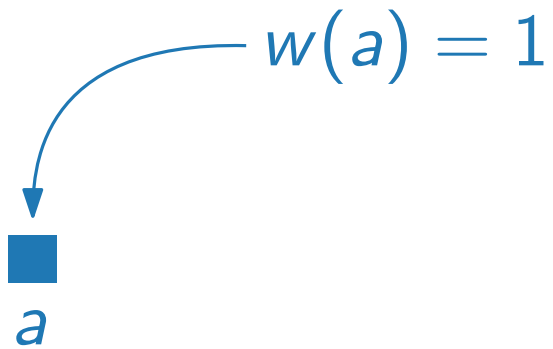
Consider  $W = 3$ .

■  
*a*

# Tight Example...?

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.

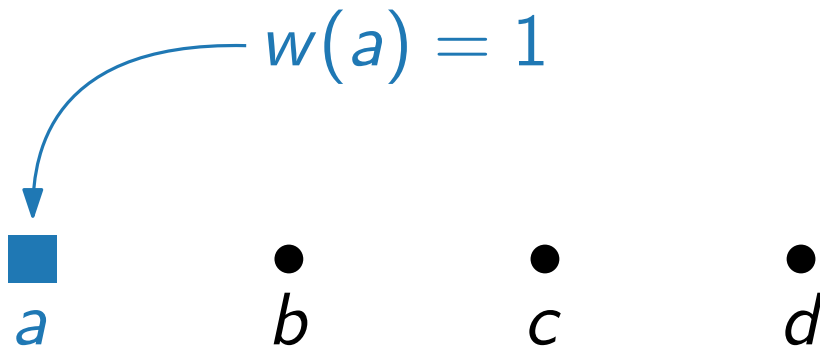
Consider  $W = 3$ .



# Tight Example...?

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.

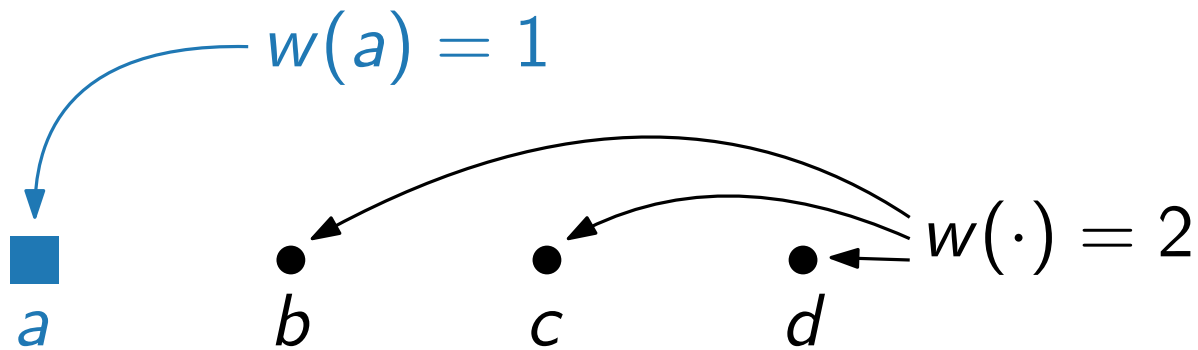
Consider  $W = 3$ .



# Tight Example...?

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.

Consider  $W = 3$ .

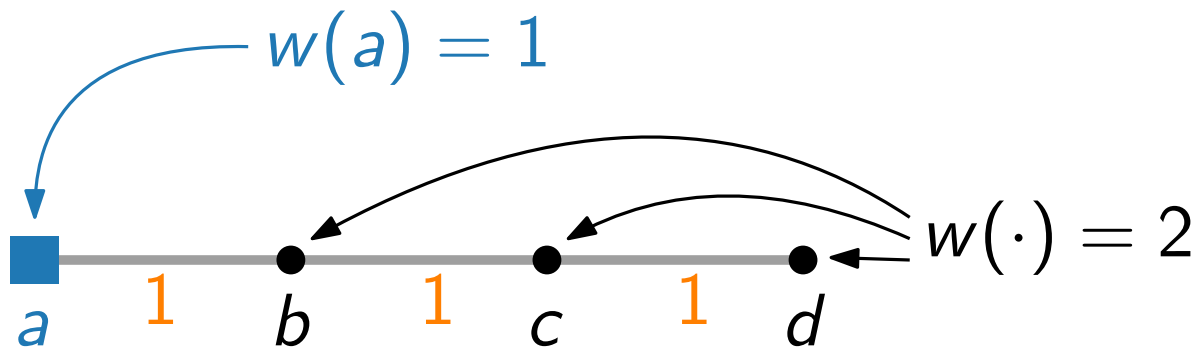




# Tight Example...?

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.

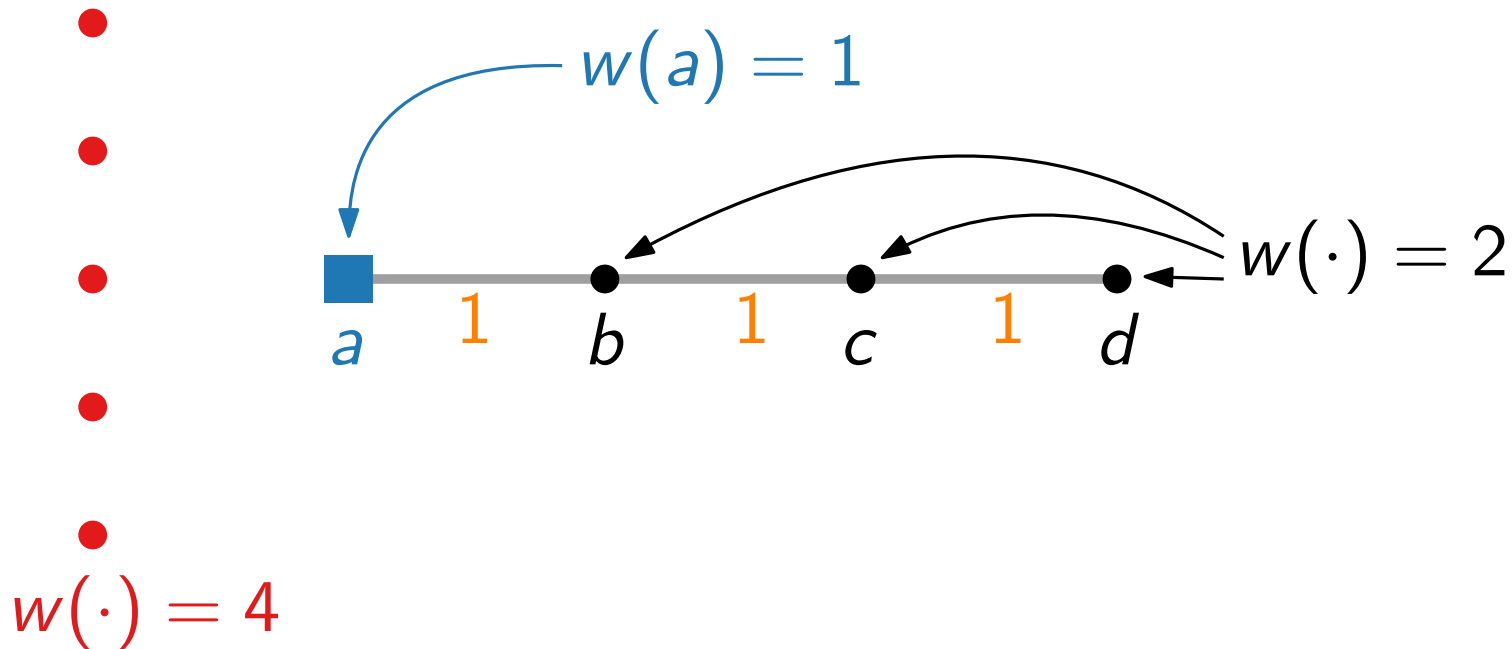
Consider  $W = 3$ .



# Tight Example...?

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.

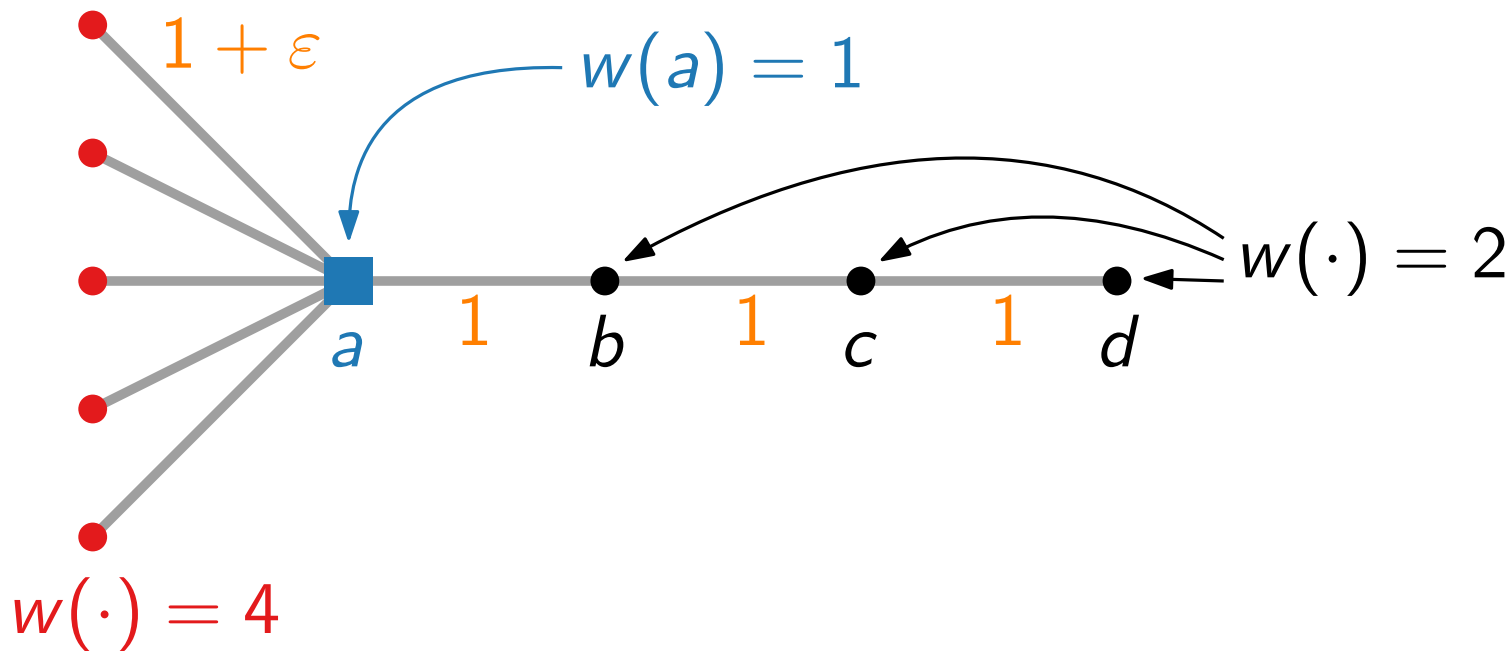
Consider  $W = 3$ .



# Tight Example...?

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.

Consider  $W = 3$ .

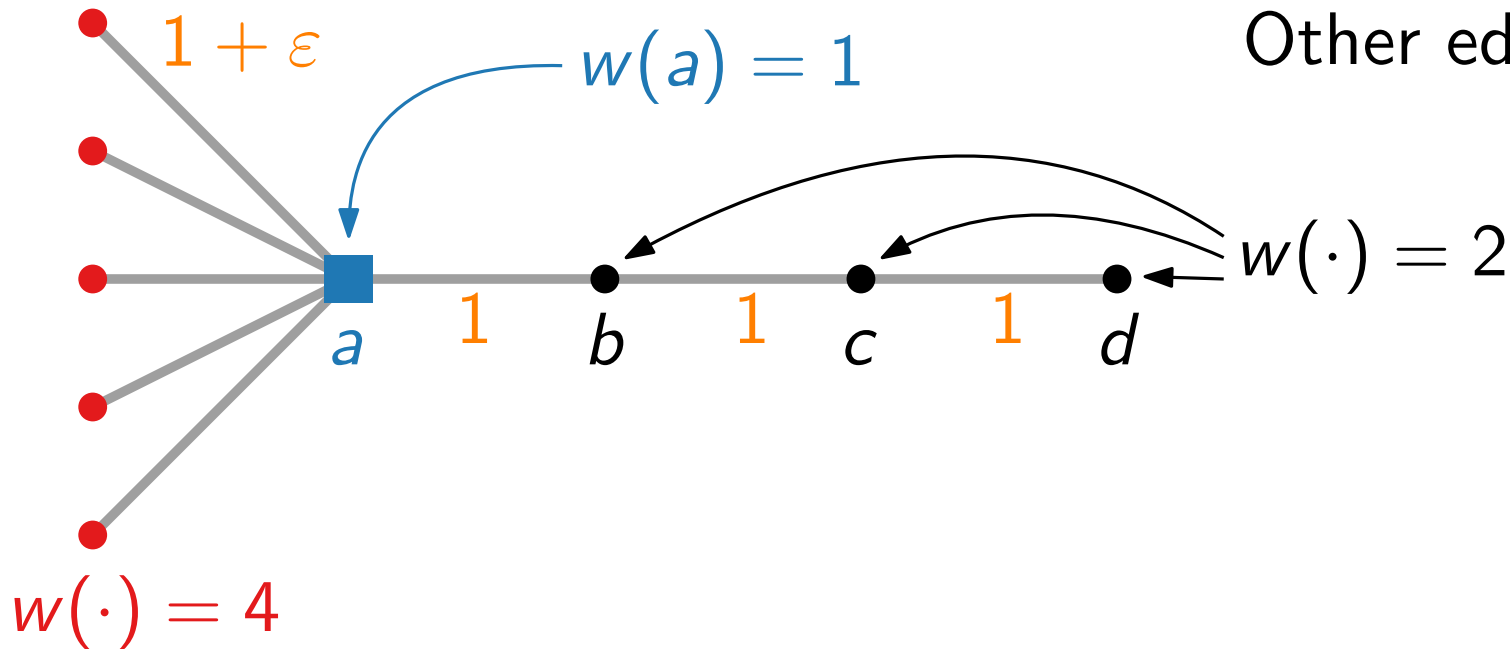


# Tight Example...?

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.

Consider  $W = 3$ .

Other edge costs?



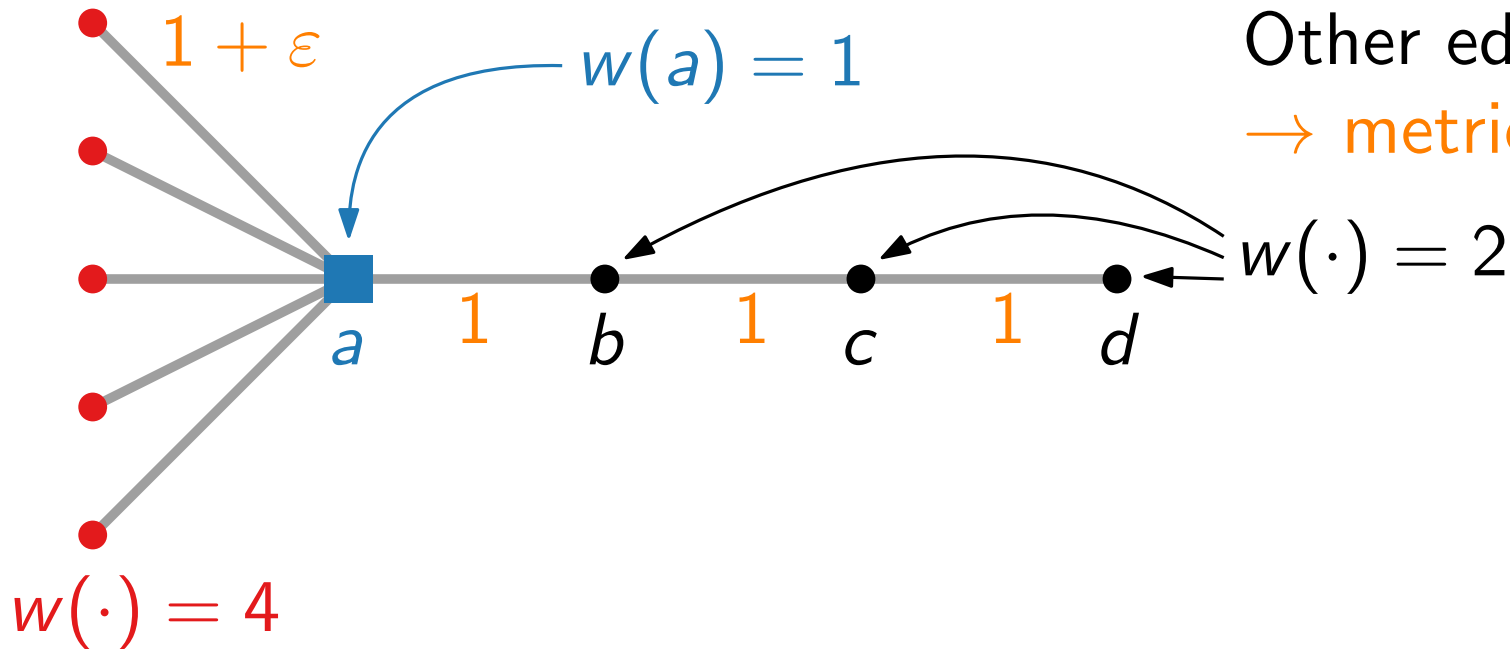
# Tight Example...?

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.

Consider  $W = 3$ .

Other edge costs?

→ metric completion!



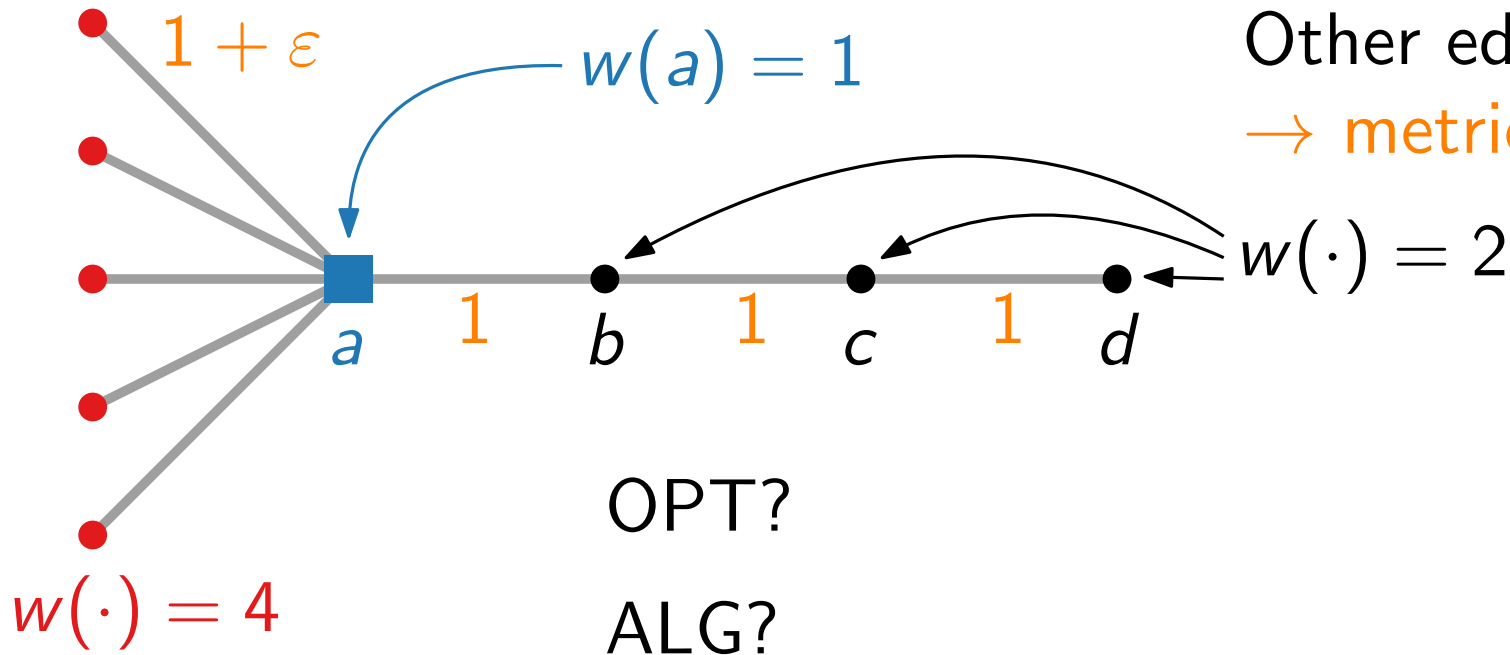
# Tight Example...?

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.

Consider  $W = 3$ .

Other edge costs?

→ metric completion!



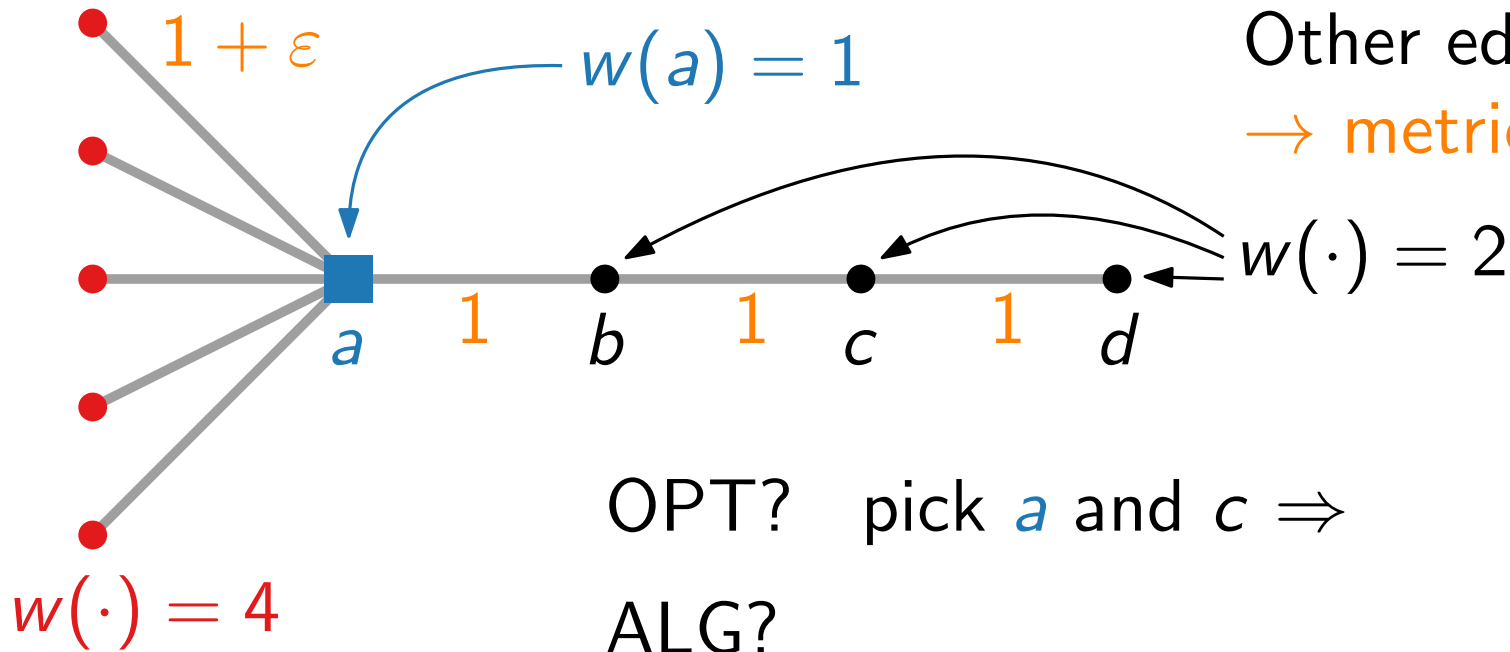
# Tight Example...?

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.

Consider  $W = 3$ .

Other edge costs?

→ metric completion!



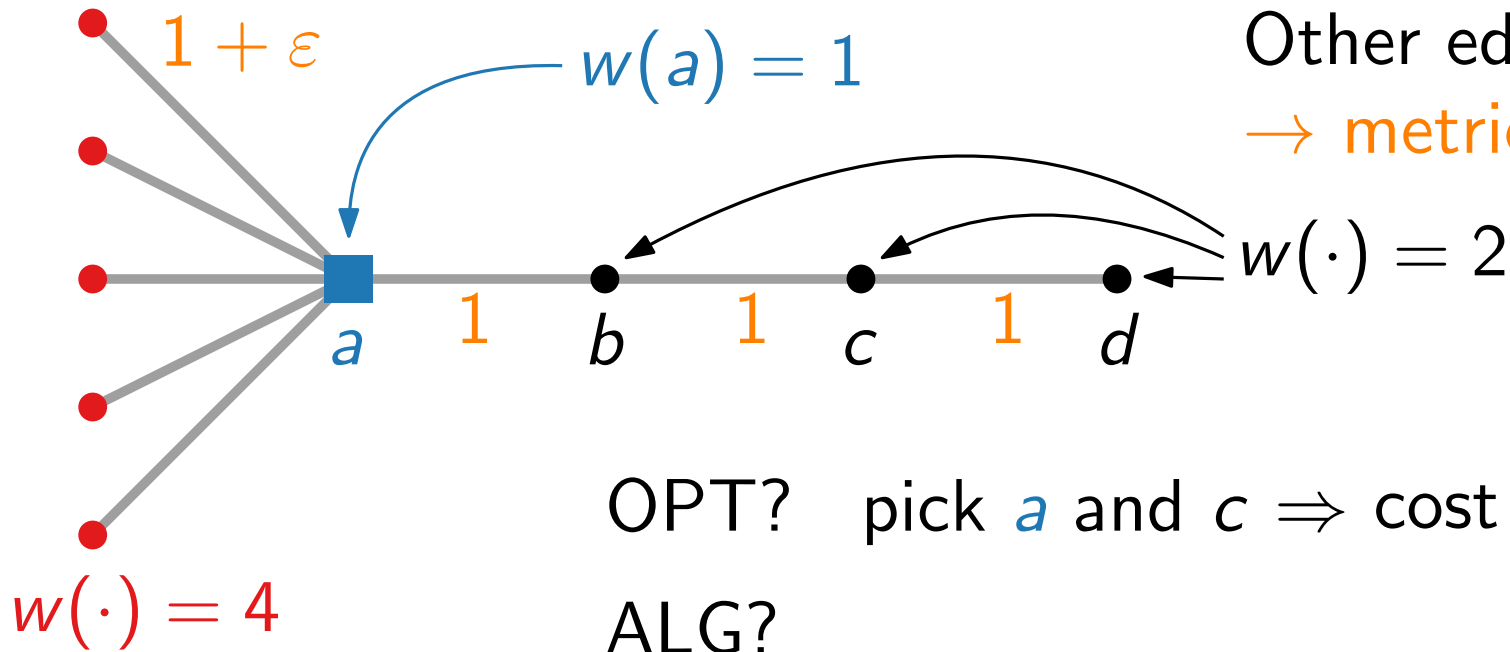
# Tight Example...

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.

Consider  $W = 3$ .

Other edge costs?

→ metric completion!





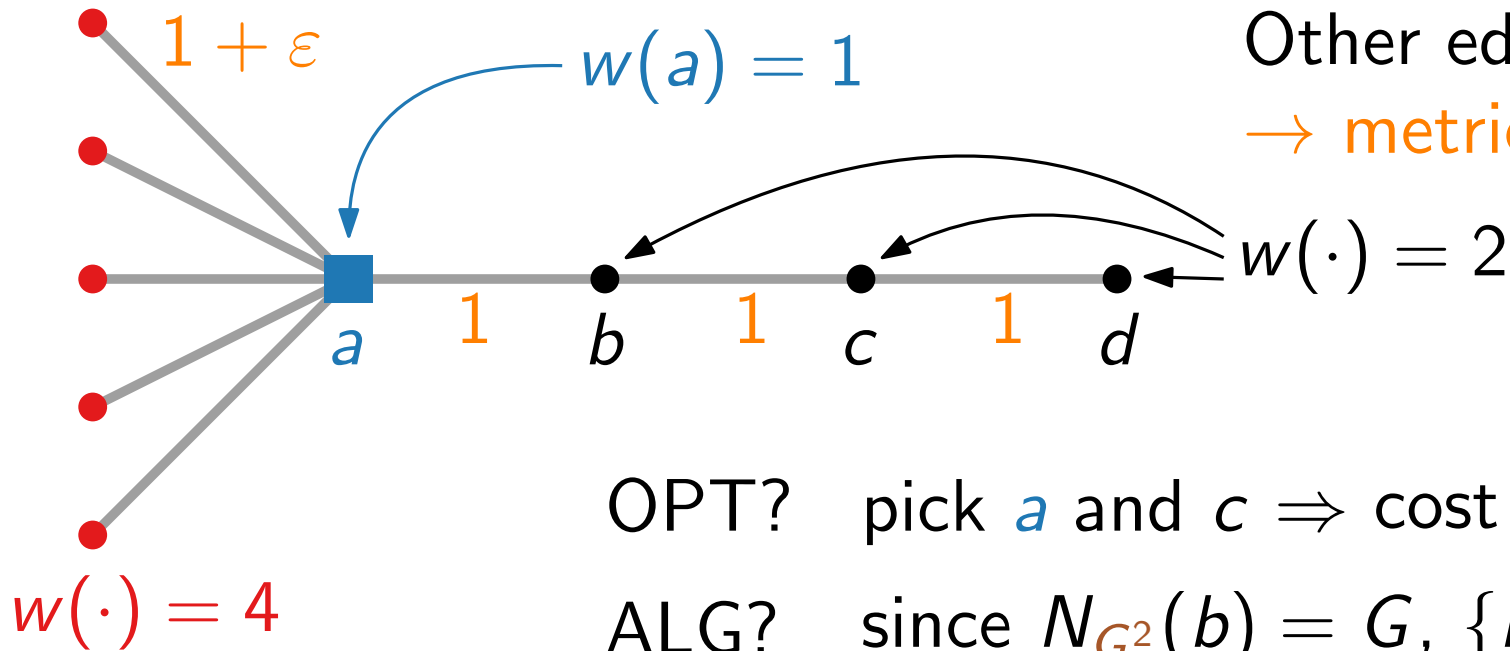
# Tight Example...?

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.

Consider  $W = 3$ .

Other edge costs?

→ metric completion!



OPT? pick  $a$  and  $c \Rightarrow$  cost  $1 + \varepsilon$ .

ALG? since  $N_{G^2}(b) = G$ ,  $\{b\}$  is a  
maximal independent set in  $G^2$

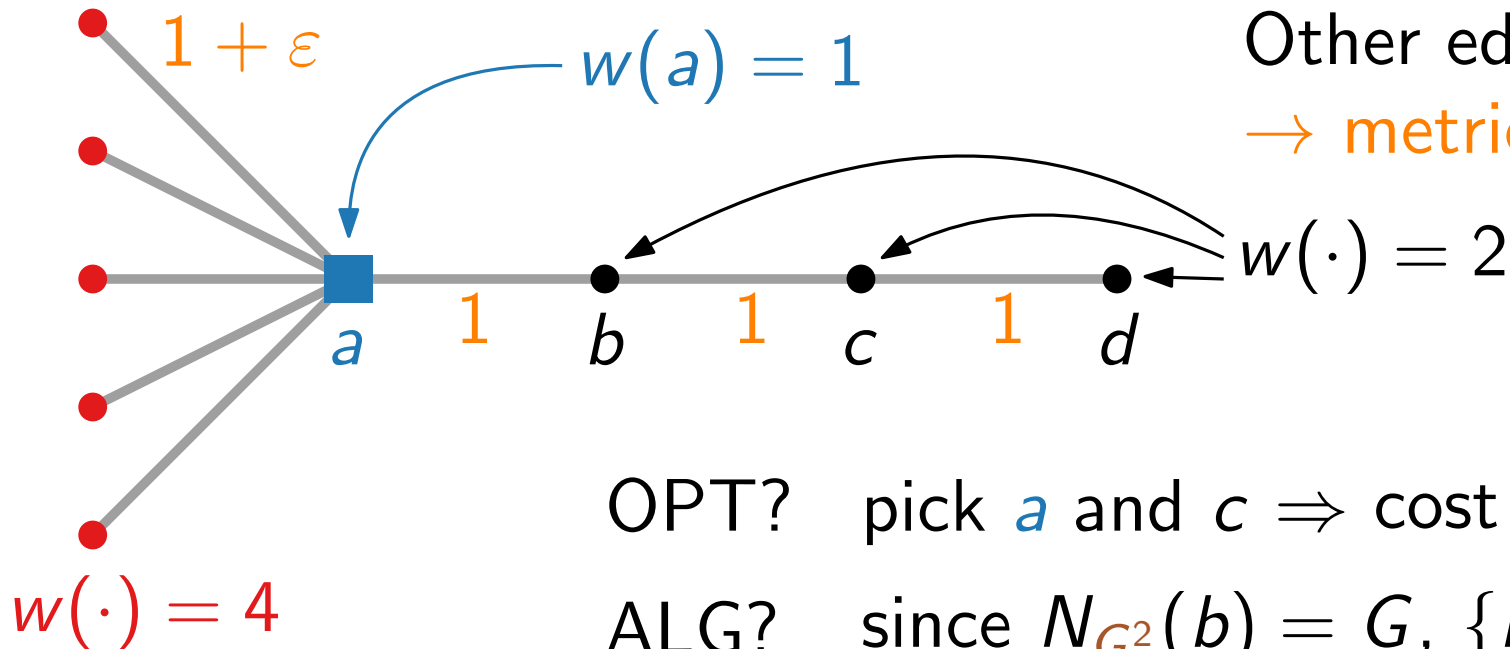
# Tight Example...

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.

Consider  $W = 3$ .

Other edge costs?

→ metric completion!

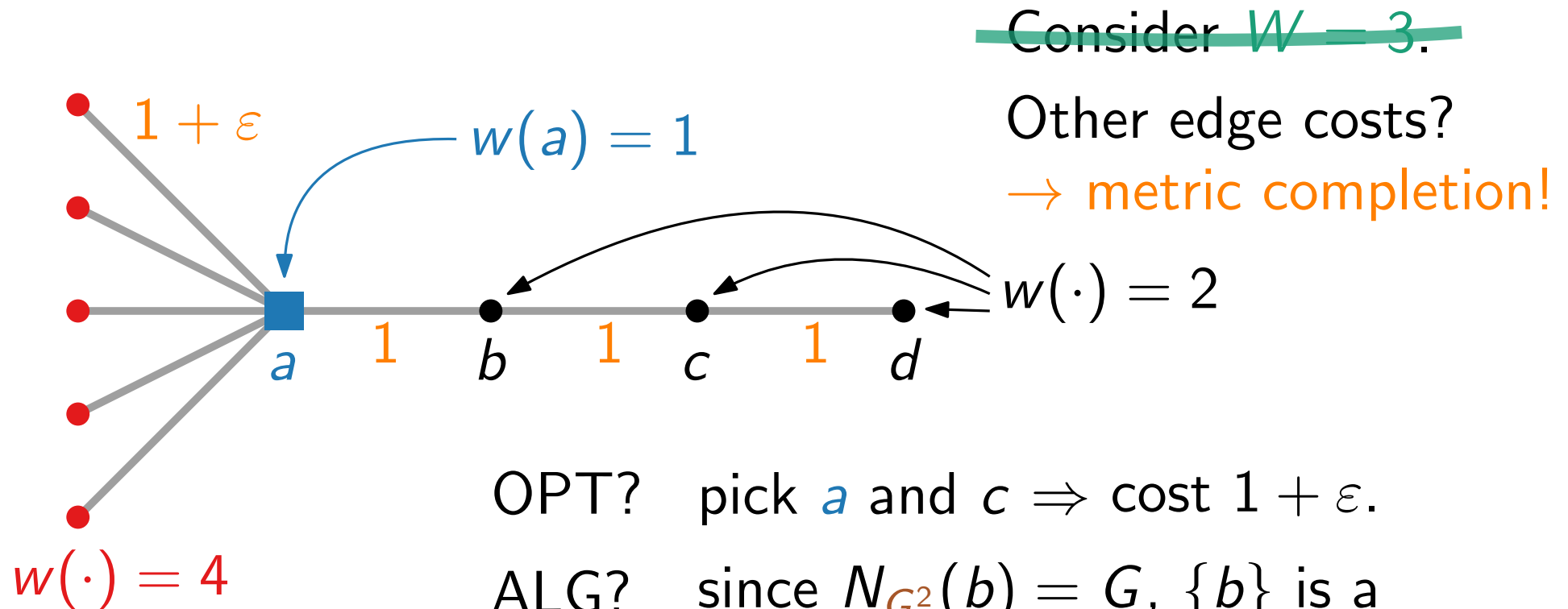


OPT? pick  $a$  and  $c \Rightarrow$  cost  $1 + \varepsilon$ .

ALG? since  $N_{G^2}(b) = G$ ,  $\{b\}$  is a  
maximal independent set in  $G^2$   
Thus, alg. picks only  $a \Rightarrow$  cost 3.

# Tight Example...?

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.



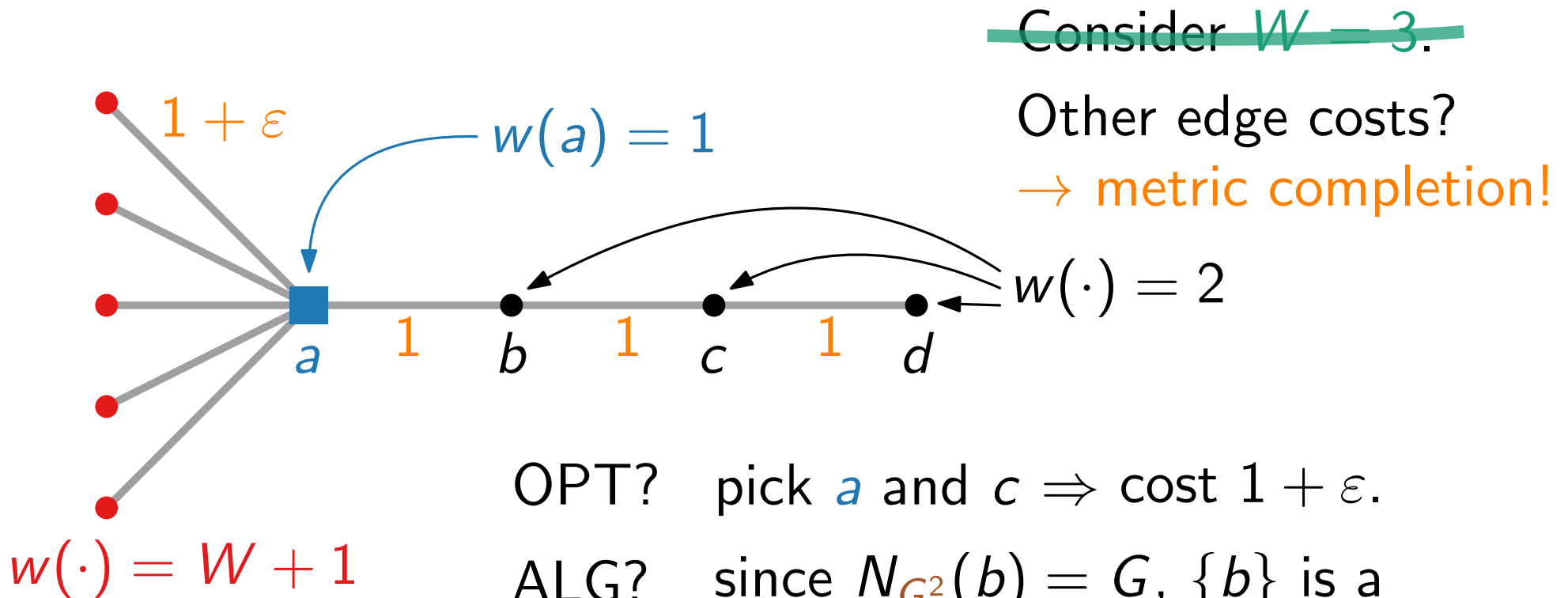
OPT? pick  $a$  and  $c \Rightarrow$  cost  $1 + \varepsilon$ .

ALG? since  $N_{G^2}(b) = G$ ,  $\{b\}$  is a  
maximal independent set in  $G^2$   
Thus, alg. picks only  $a \Rightarrow$  cost 3.

How can we generalize this to larger  $W$ ?

# Tight Example...?

Here, we need to have a budget  $W$ ,  
and edge costs satisfying the triangle inequality.



OPT? pick  $a$  and  $c \Rightarrow$  cost  $1 + \varepsilon$ .

ALG? since  $N_{G^2}(b) = G$ ,  $\{b\}$  is a  
maximal independent set in  $G^2$   
Thus, alg. picks only  $a \Rightarrow$  cost 3.

How can we generalize this to larger  $W$ ?