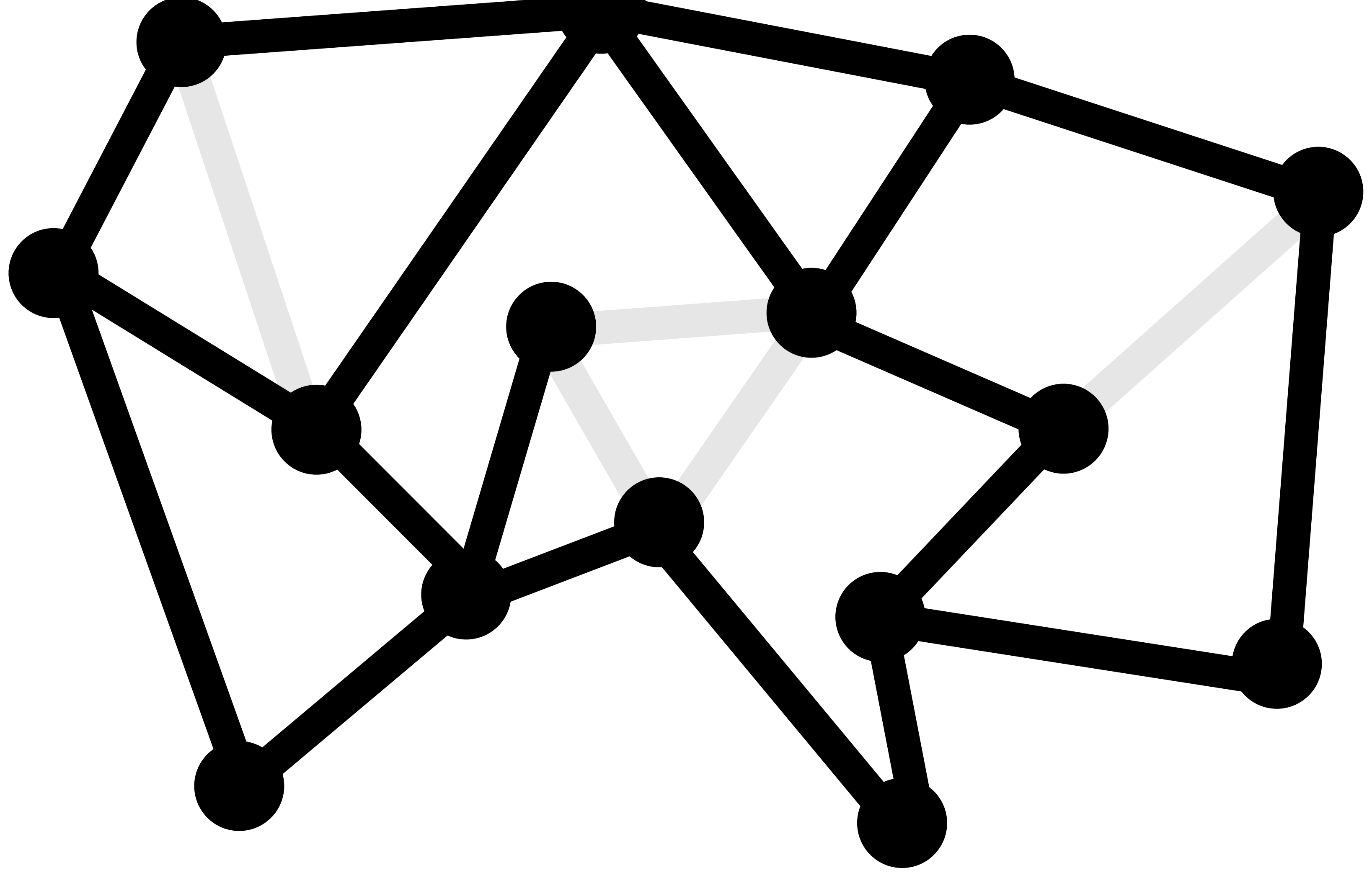
Spanners Mini-Talk

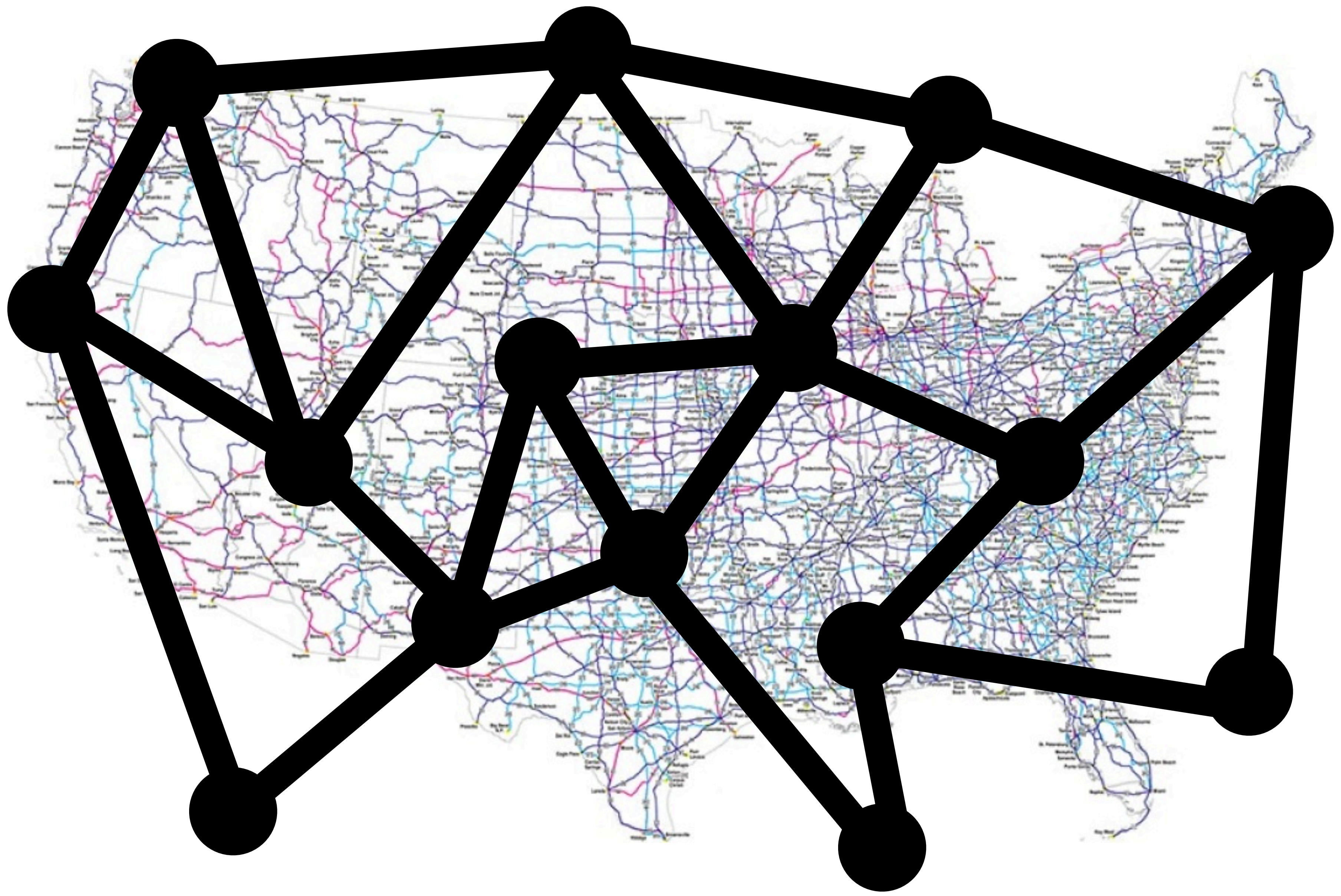
Fall 2023
Brown University



D Ellis Hershkowitz (Ellis)

Motivation: Distance Oracles

Computing Distances

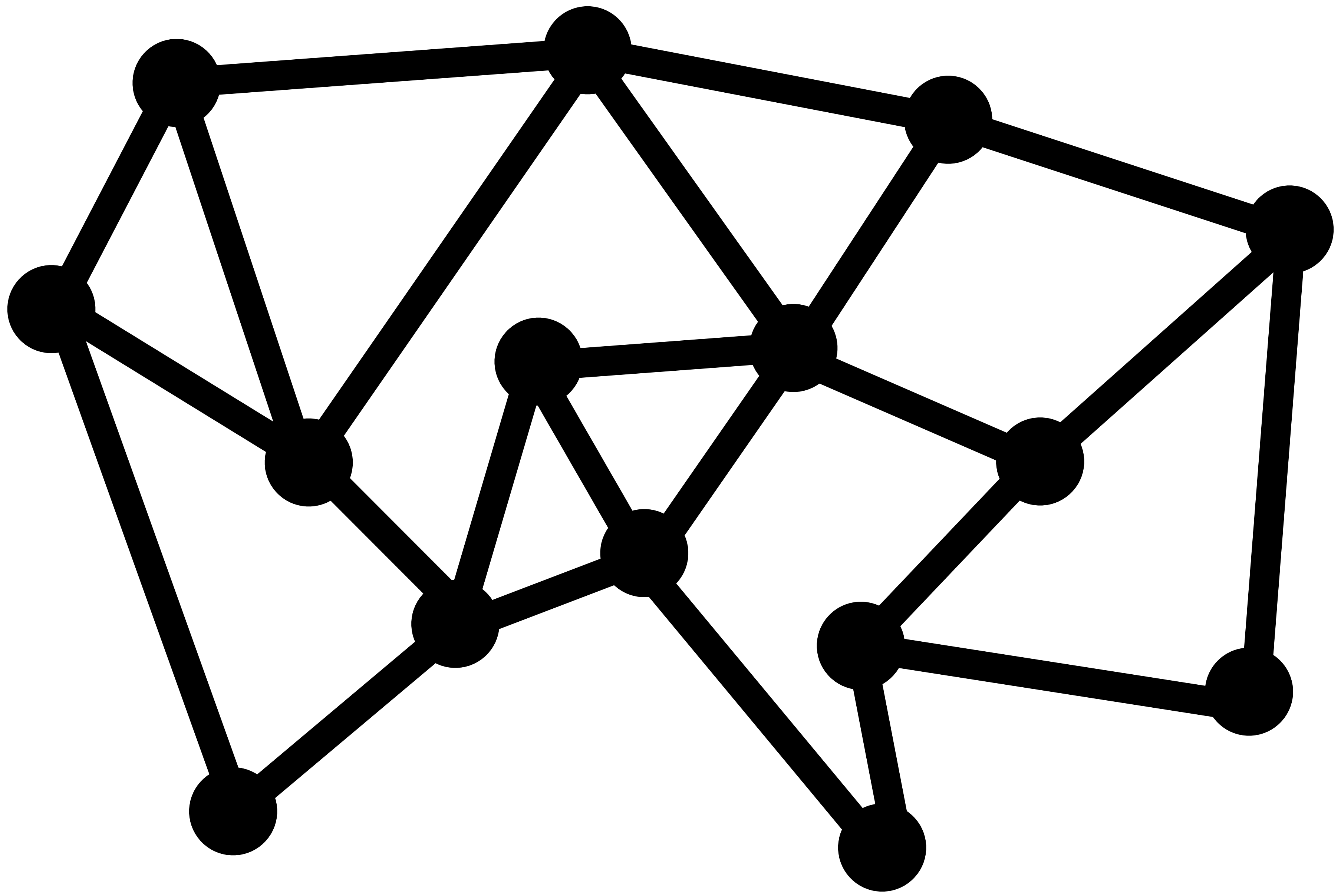


(Transportation) Network

Some notion of distance

Motivation: Distance Oracles

Computing Distances



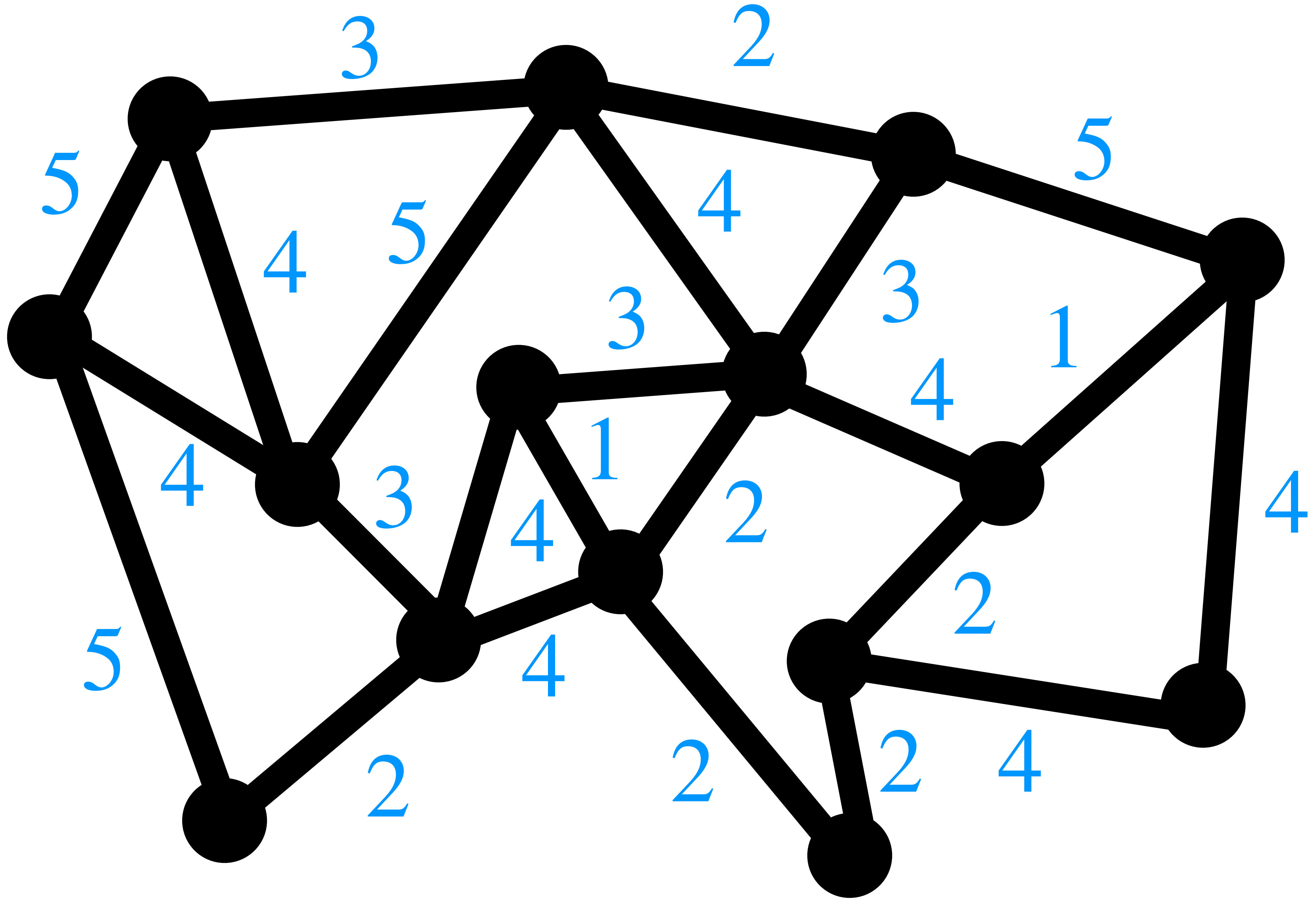
Graph $G = (V, E)$

$$d_G(u, v) := \min\{\,|P|\, : \text{path } P \text{ from } u \text{ to } v\}$$

$$w(P) := \sum_{e \in P} w(e)$$

Motivation: Distance Oracles

Computing Distances

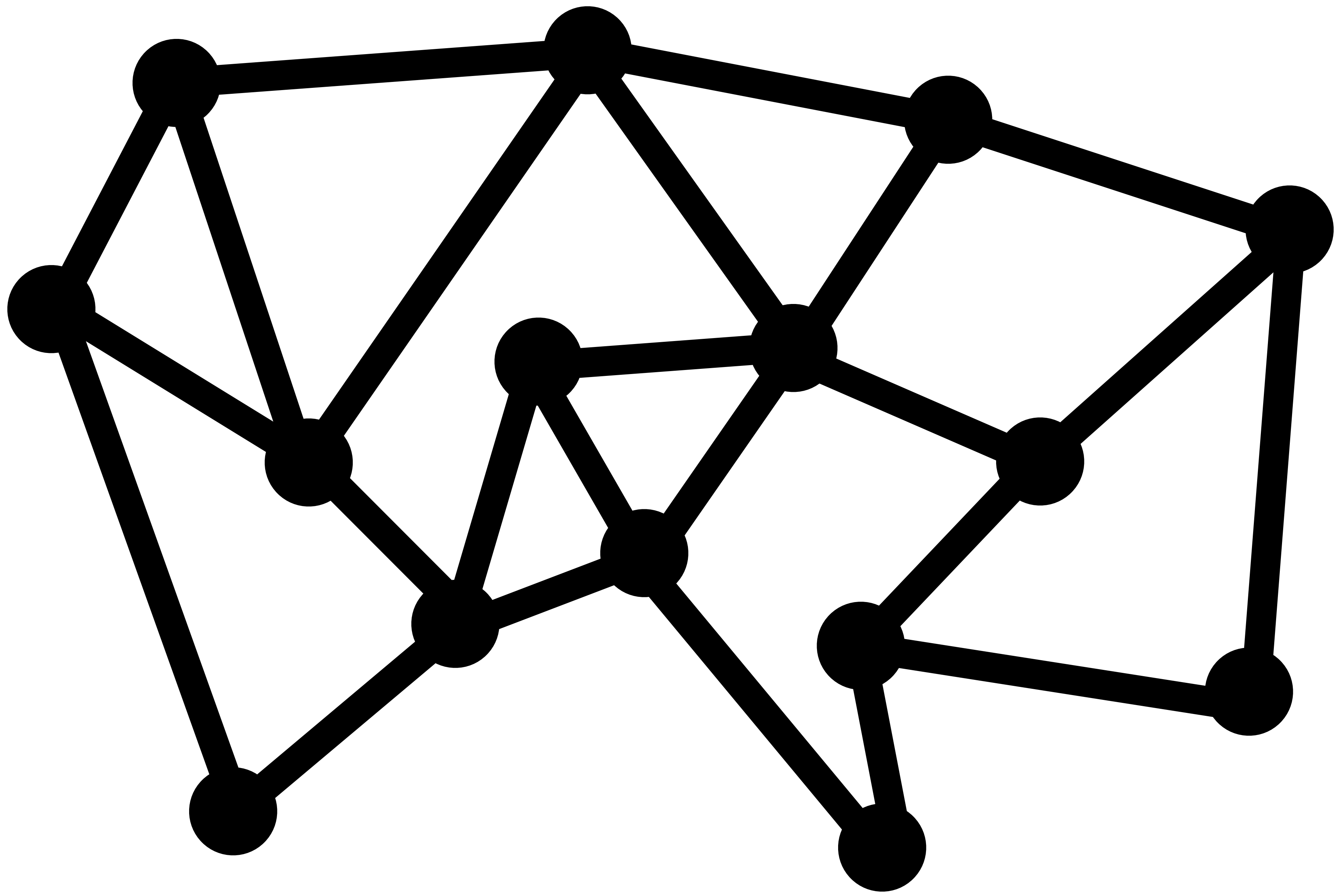


Graph $G = (V, E, w)$

$$d_G(u, v) := \min\{w(P)\text{: path } P \text{ from } u \text{ to } v\}$$

Motivation: Distance Oracles

Computing Distances



Graph $G = (V, E)$

$$d_G(u, v) := \min\{\,|P| : \text{path } P \text{ from } u \text{ to } v\}$$

Motivation: Distance Oracles

Computing Distances

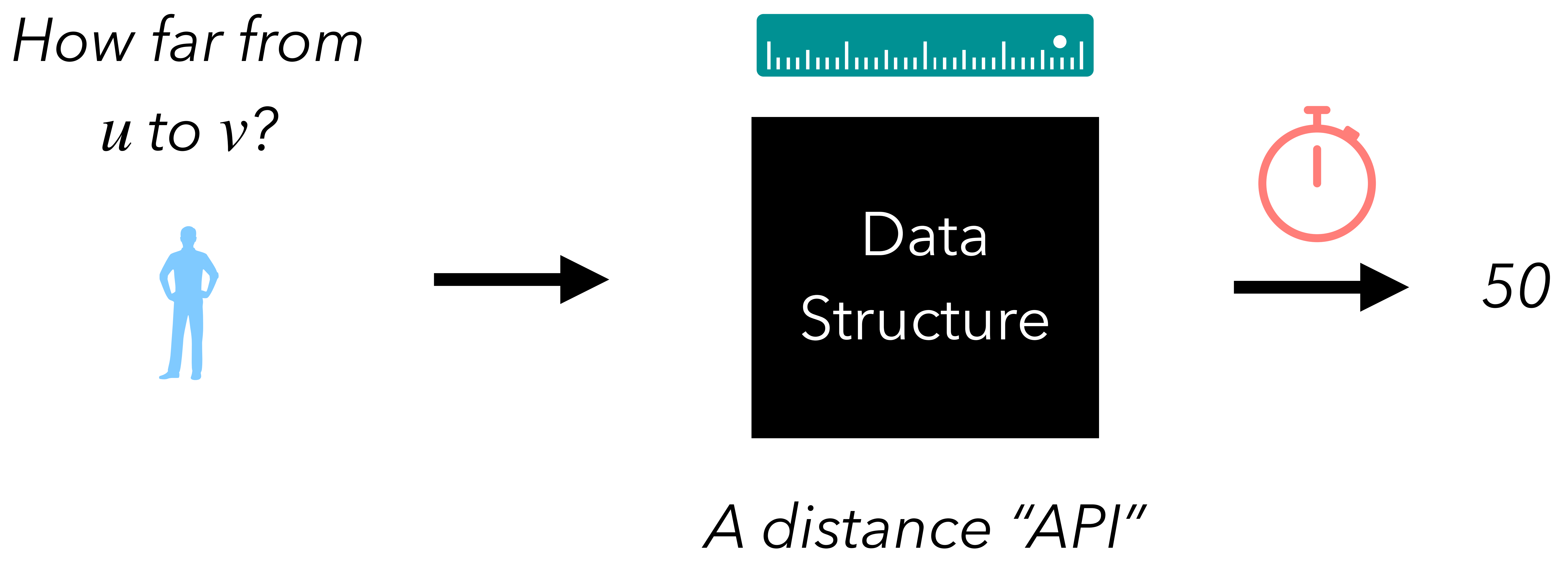
How far from

u *to* v*?*

A distance "API"

Motivation: Distance Oracles

Computing Distances

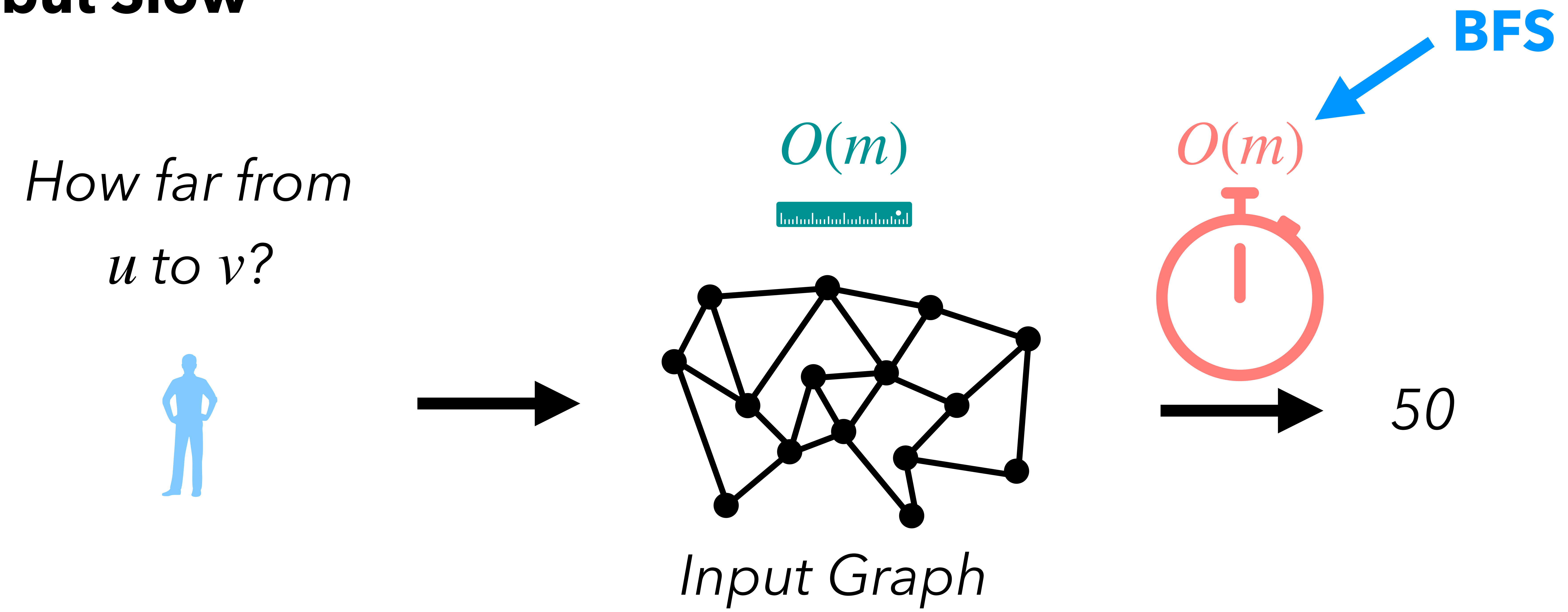


Tradeoff: **space** (of data structure) vs (response) **time**

$n := |V|$ and $m := |E|$

Motivation: Distance Oracles

Small but Slow



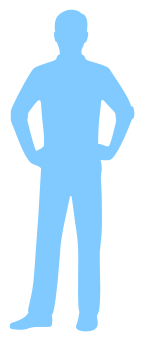
Tradeoff: space (of data structure) vs (response) **time**

Motivation: Distance Oracles

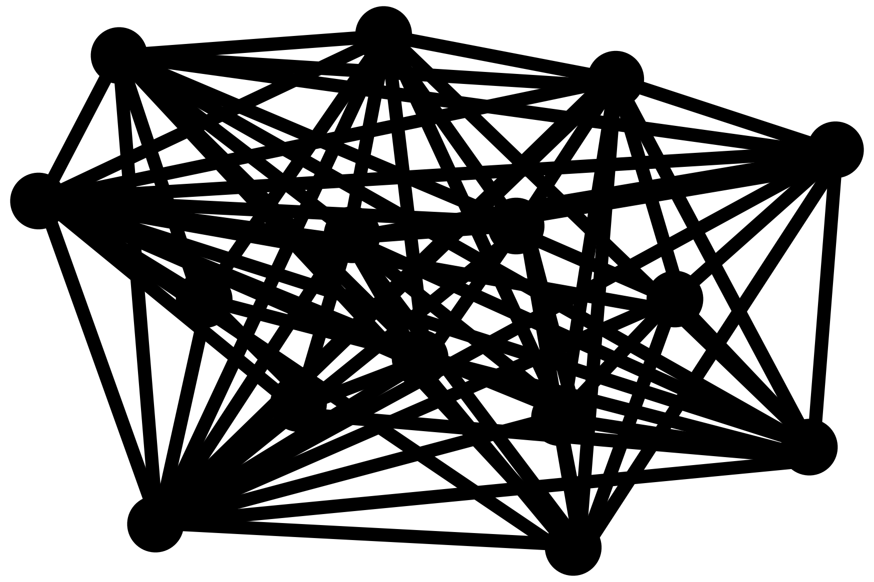
Fast but Large

How far from

u to v?



$O(n^2)$



$O(1)$

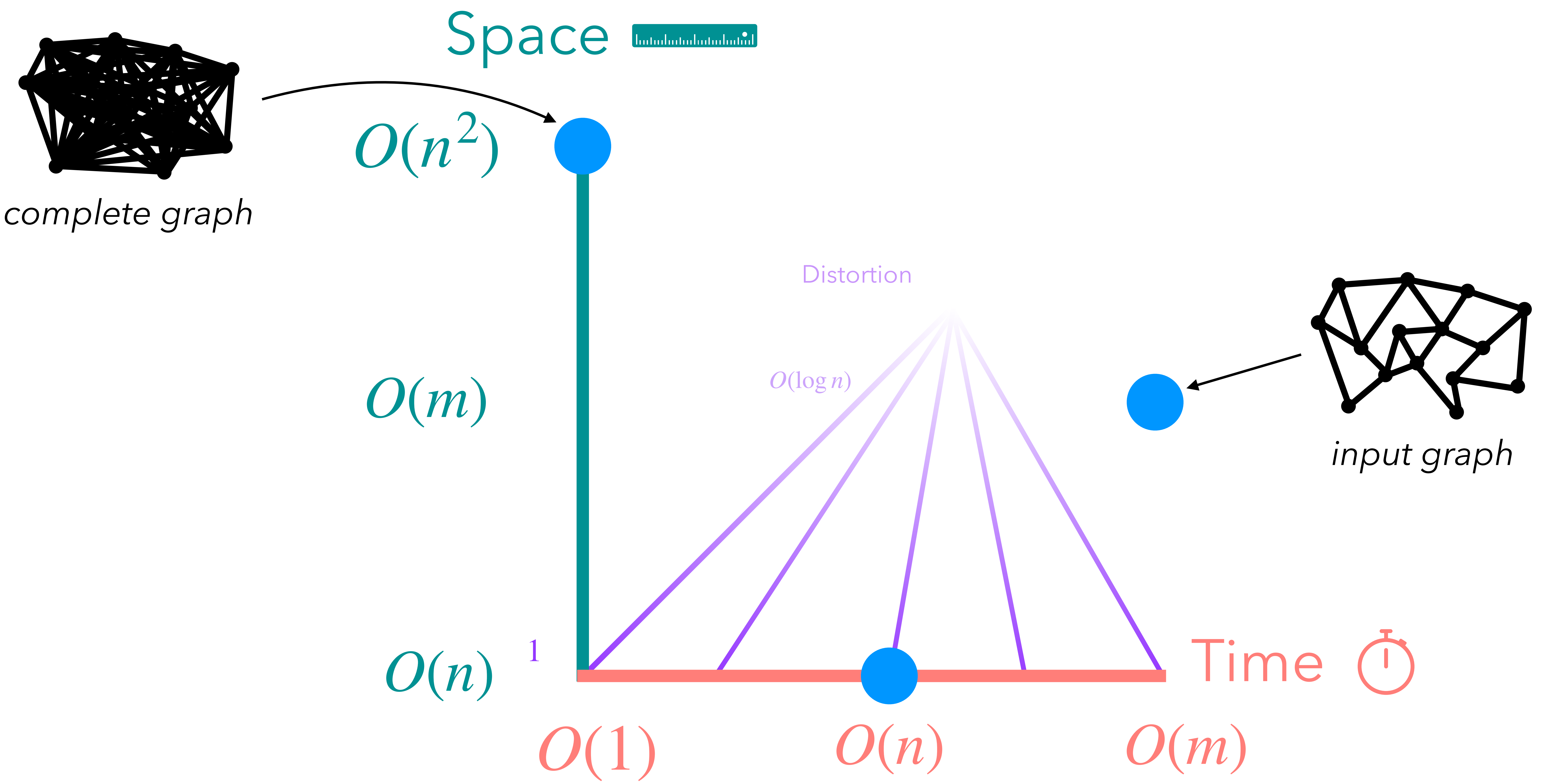
50

Complete Graph with $w = d_G$

Tradeoff: space (of data structure) vs (response) **time**

Motivation: Distance Oracles

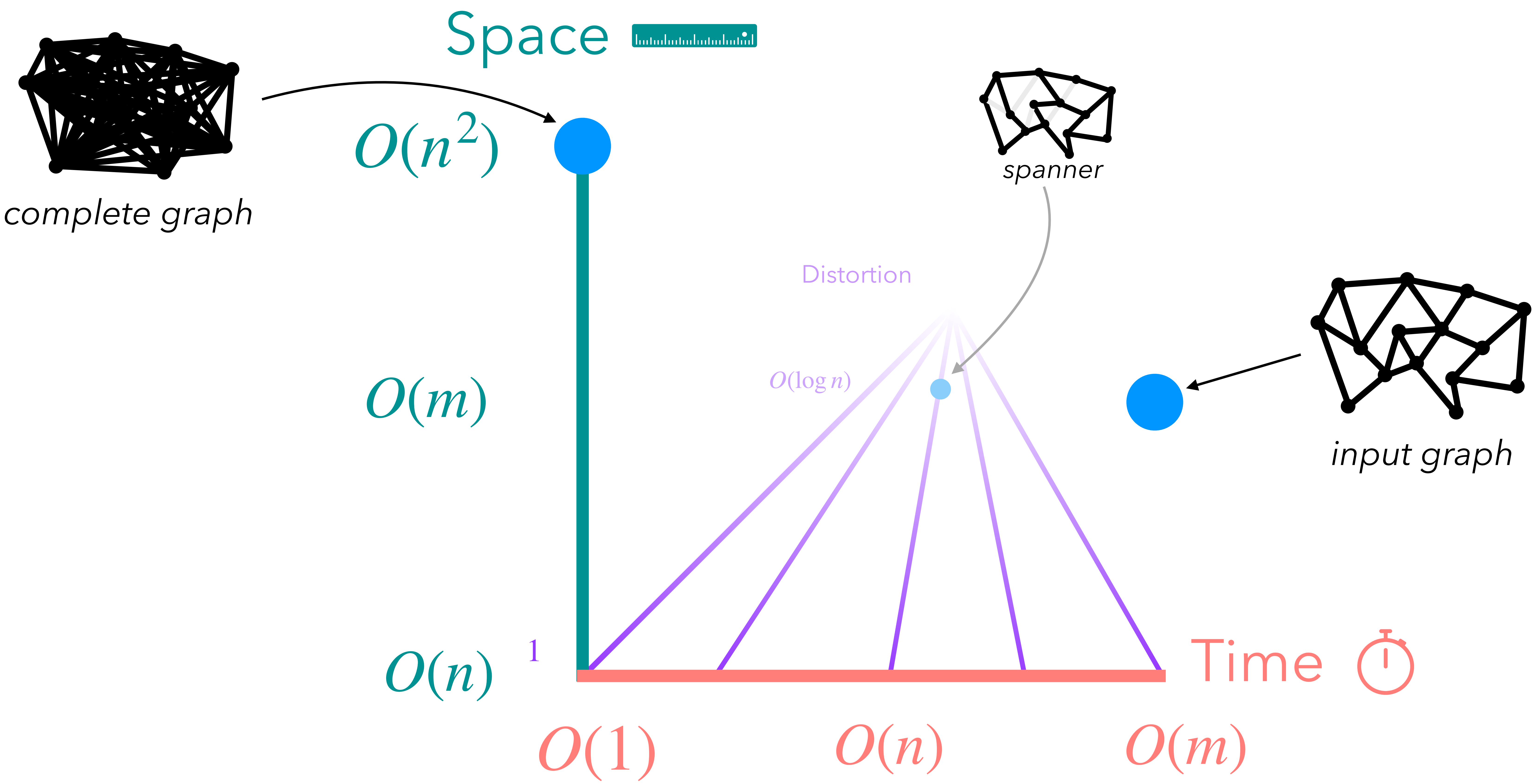
Plotting Tradeoffs



$n := |V|$ and $m := |E|$

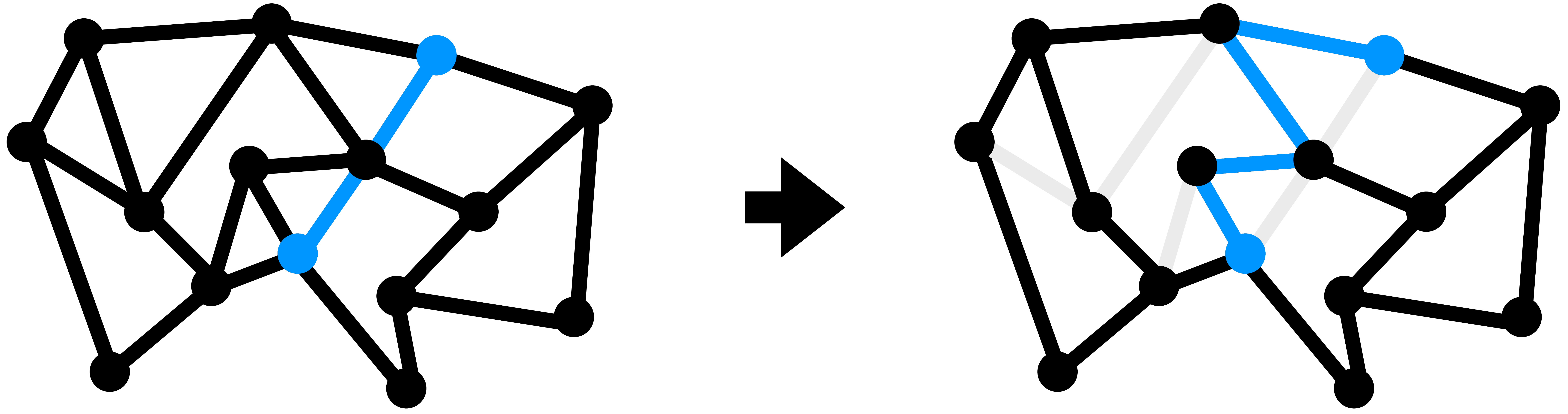
Motivation: Distance Oracles

Plotting Tradeoffs



Spanners

Observe: $d_G(u,v) \le d_H(u,v)$ $\forall u,v \in V$

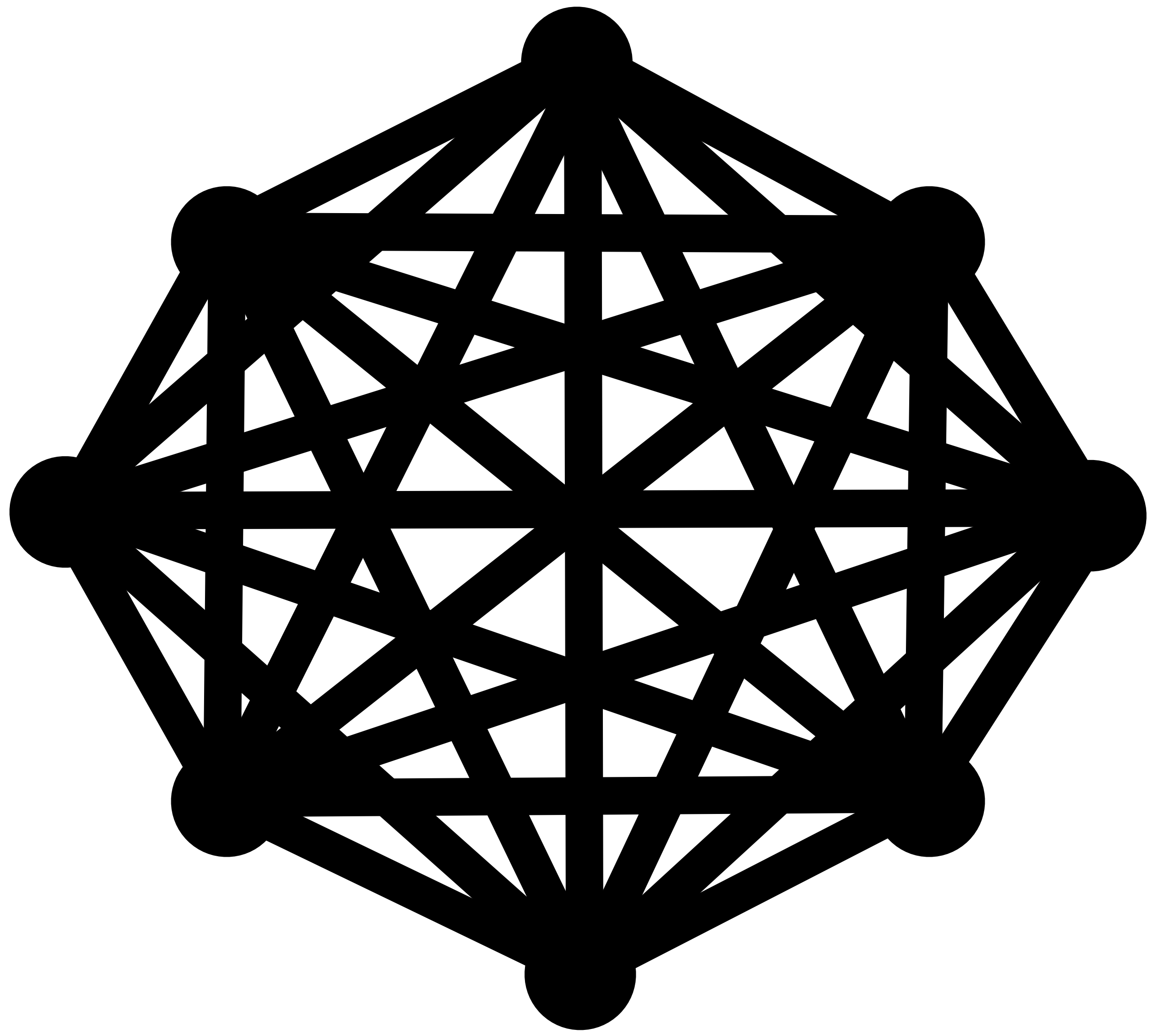


graph G | 2-*spanner* H *of* G

Definition (spanner): given graph $G = (V, E)$ and $t \ge 1$, a t-spanner H is a subgraph of G satisfying

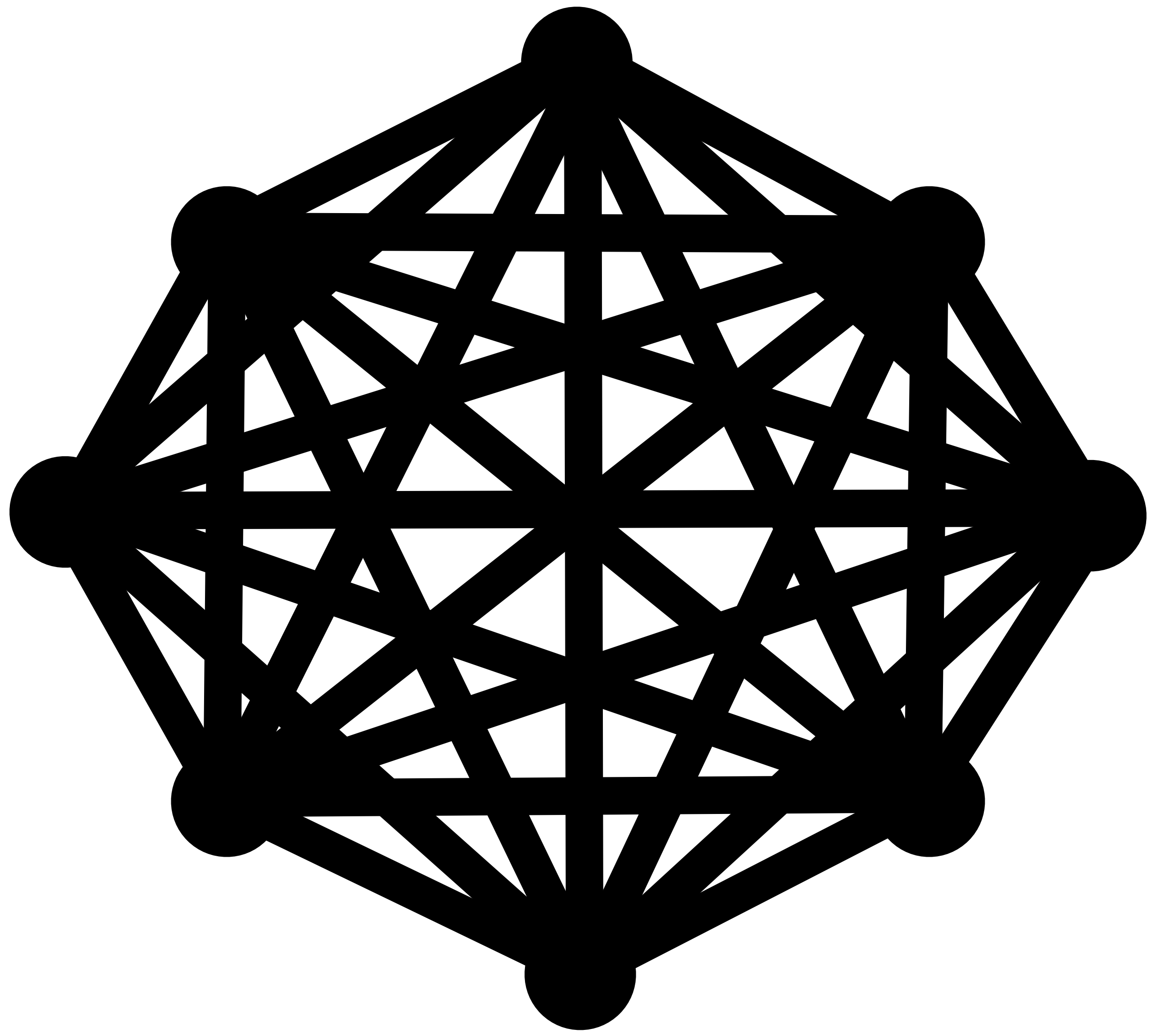
$$d_H(u,v) \le t \cdot d_G(u,v) \qquad \forall u,v \in V$$

Spanners



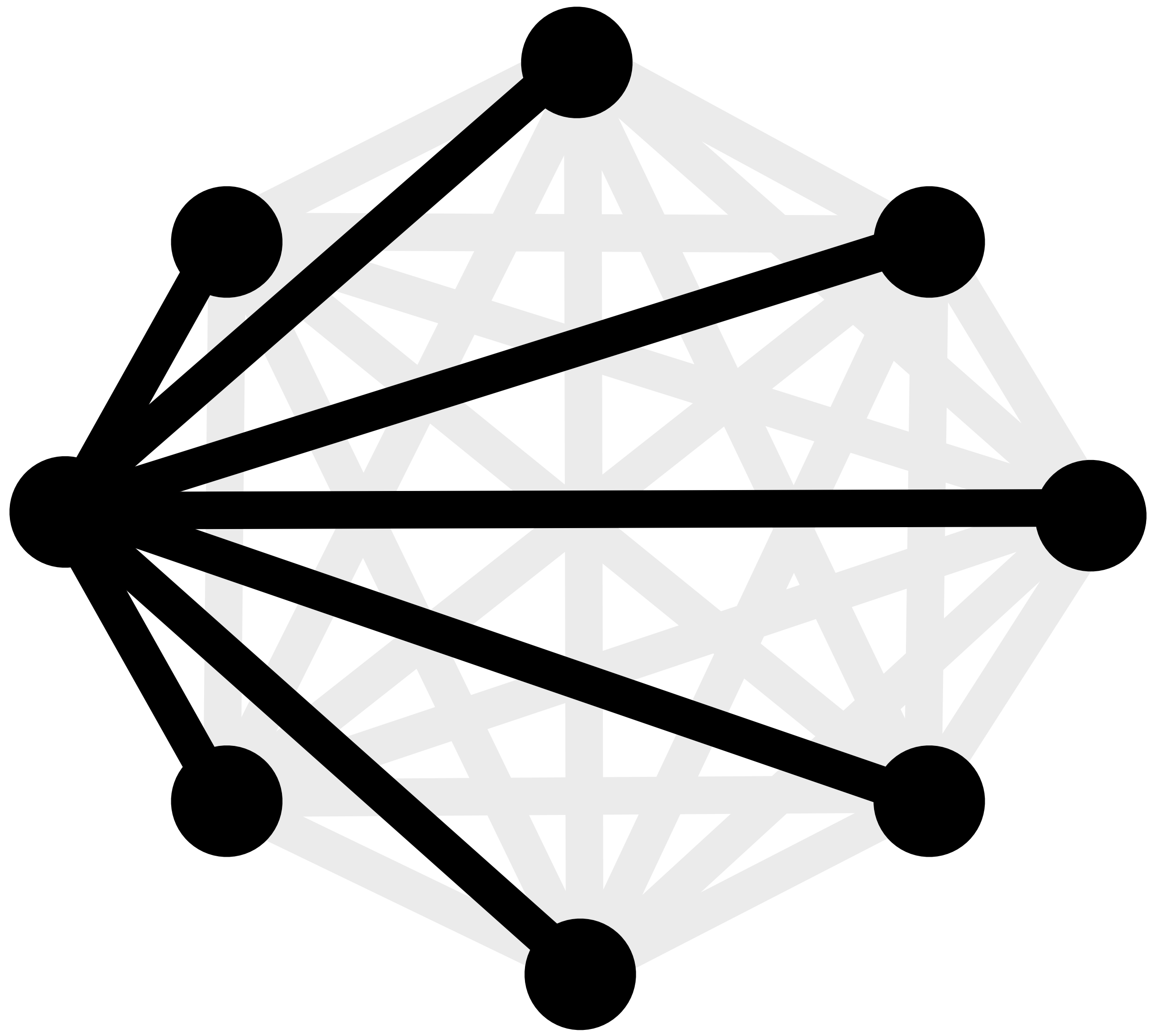
Question: smallest 1-spanner of complete graph?

Spanners



Question: smallest 2-spanner of complete graph?

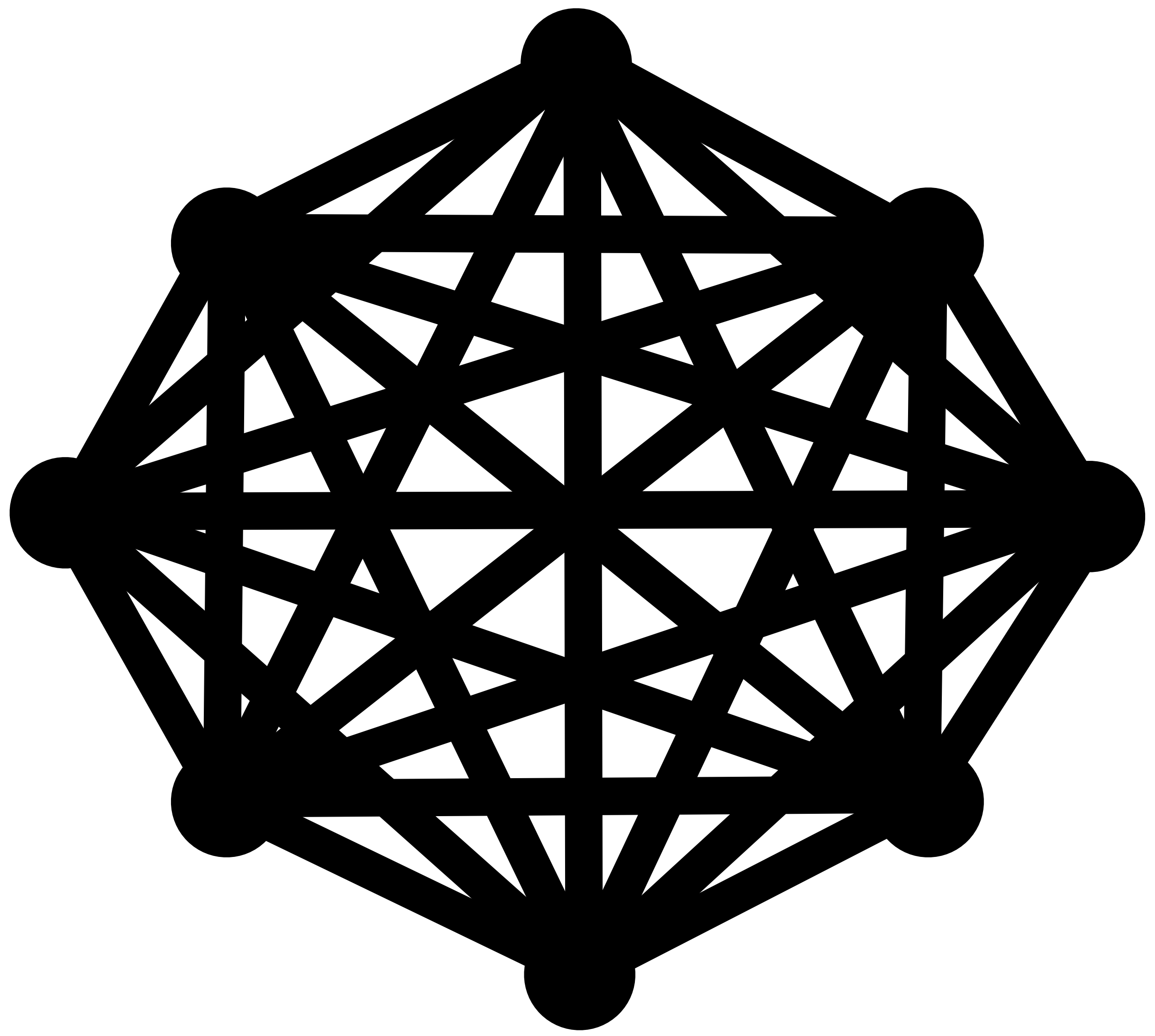
Spanners



Question: smallest 2-spanner of complete graph?

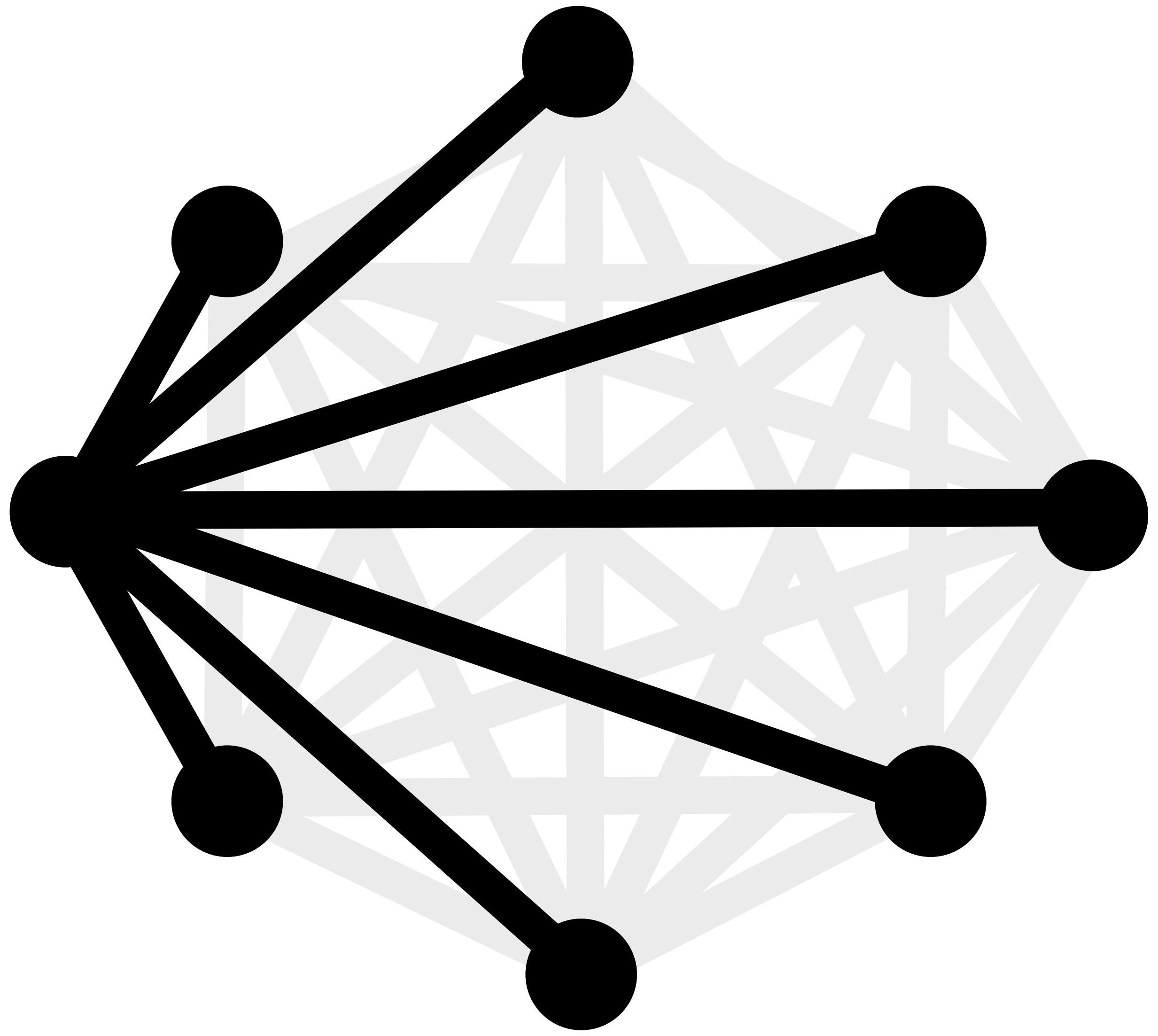
Spanners

$O(n^2)$ edges



smallest 1*-spanner*
(of complete graph)

$O(n)$ edges

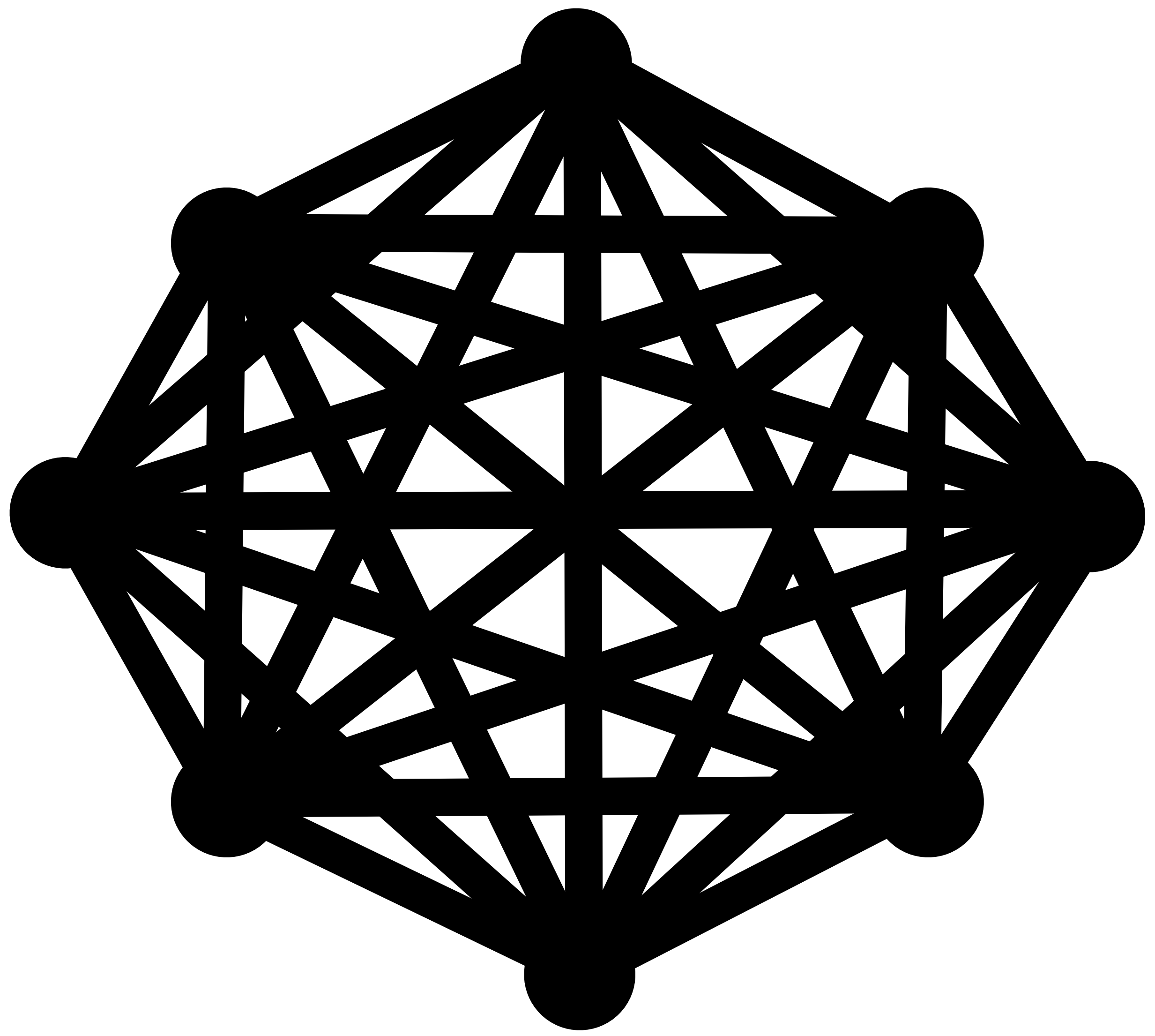


smallest 2*-spanner*
(of complete graph)

Moral: larger **distortion** allows smaller **size** (of spanner)

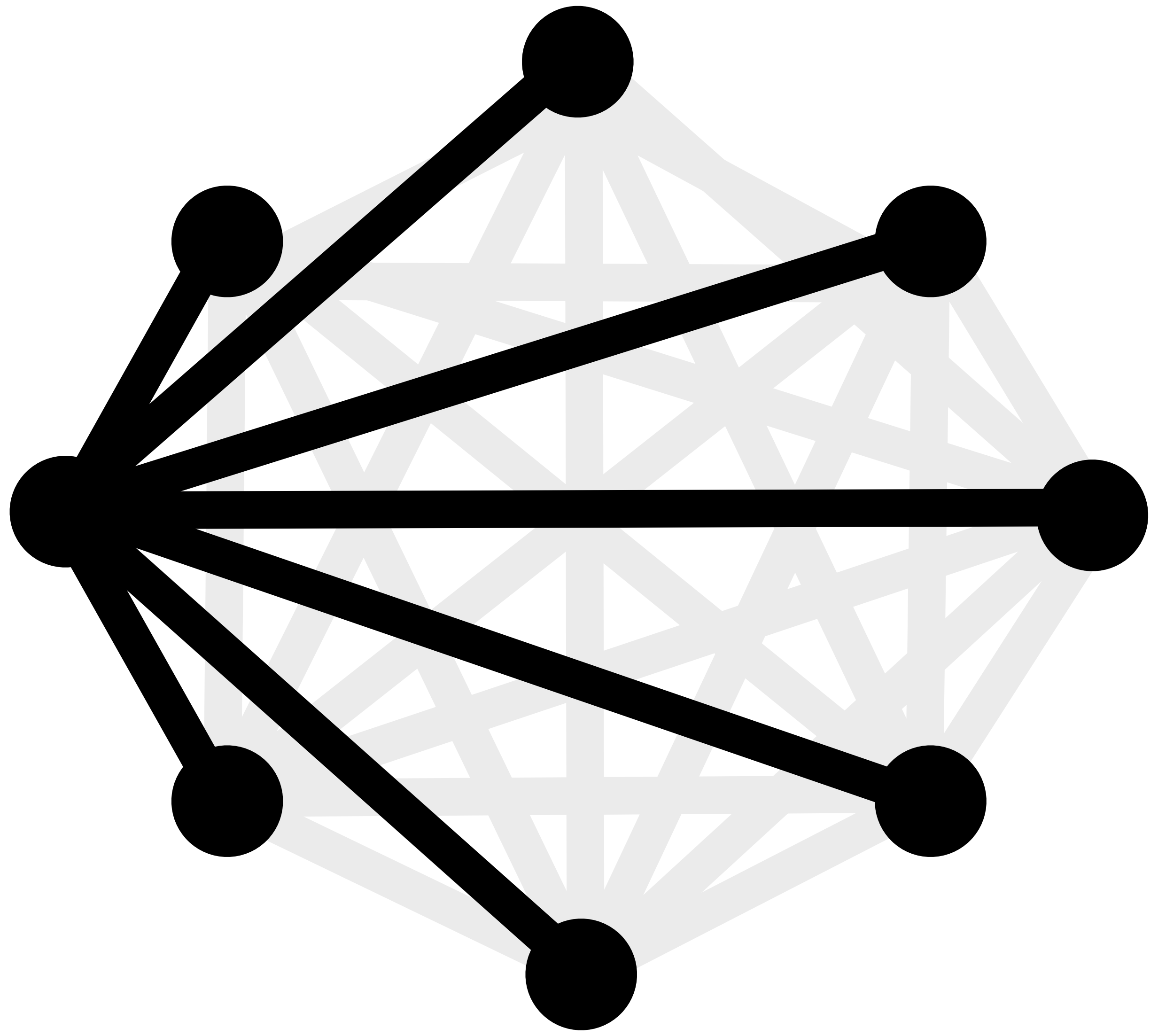
Spanners

$O(n^2)$ edges



smallest 1*-spanner*
(of complete graph)

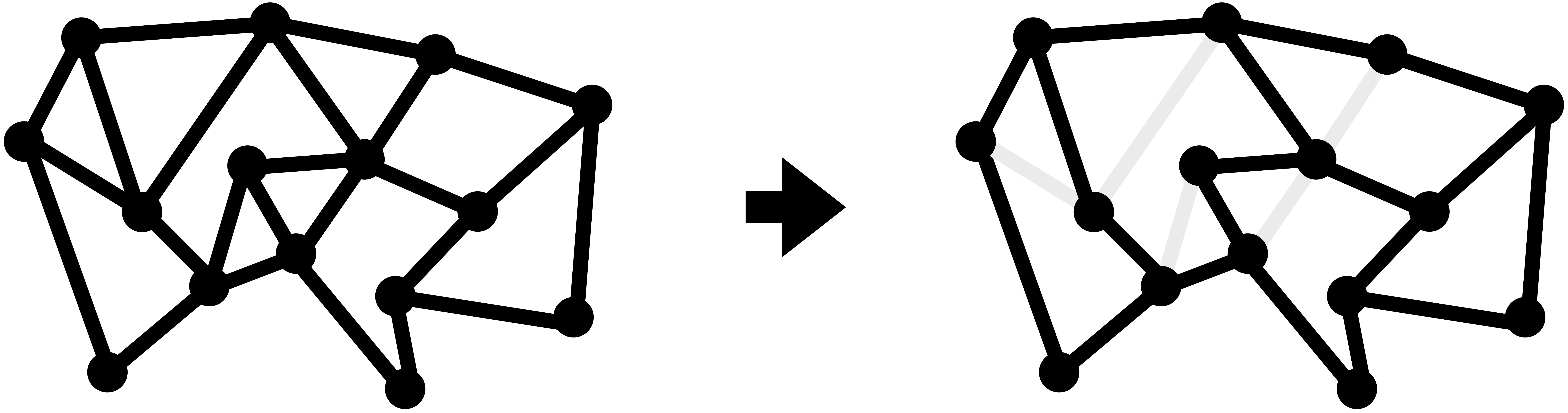
$O(n)$ edges



smallest 2*-spanner*
(of complete graph)

Main Question: how large of **distortion** for $O(n)$ edges in general?

Main Result Today



Theorem: every graph G has a t-spanner H w/

- **Distortion:** $t = O(\log n)$
- **Size:** $|H| = O(n)$

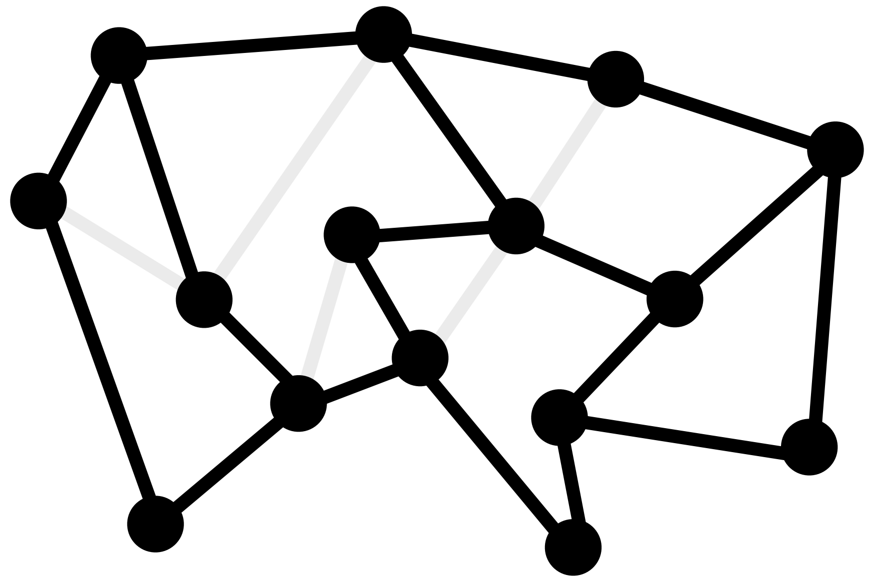
Distance Oracles with Spanners

BFS

$O(n)$

How far from

u *to* v*?*



$O(n)$

between

$50/t$ *and* 50

Roadmap of Proof

1. **Simple Observation**

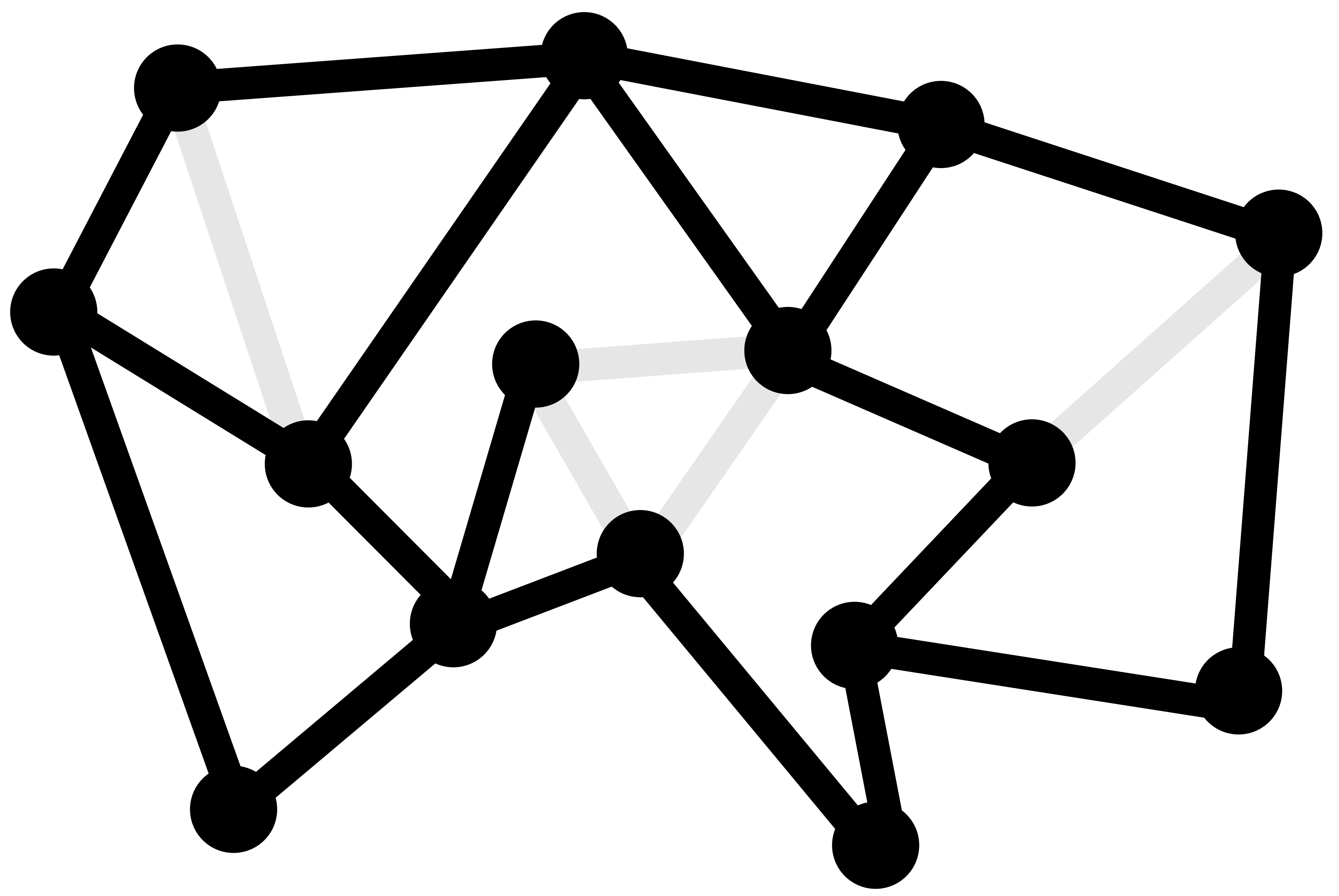
   edge spanners suffice

2. **Greedy Algorithm**

   suggested by observation

3. **Distortion Analysis**
4. **Size Analysis**

   by "Moore Bounds"

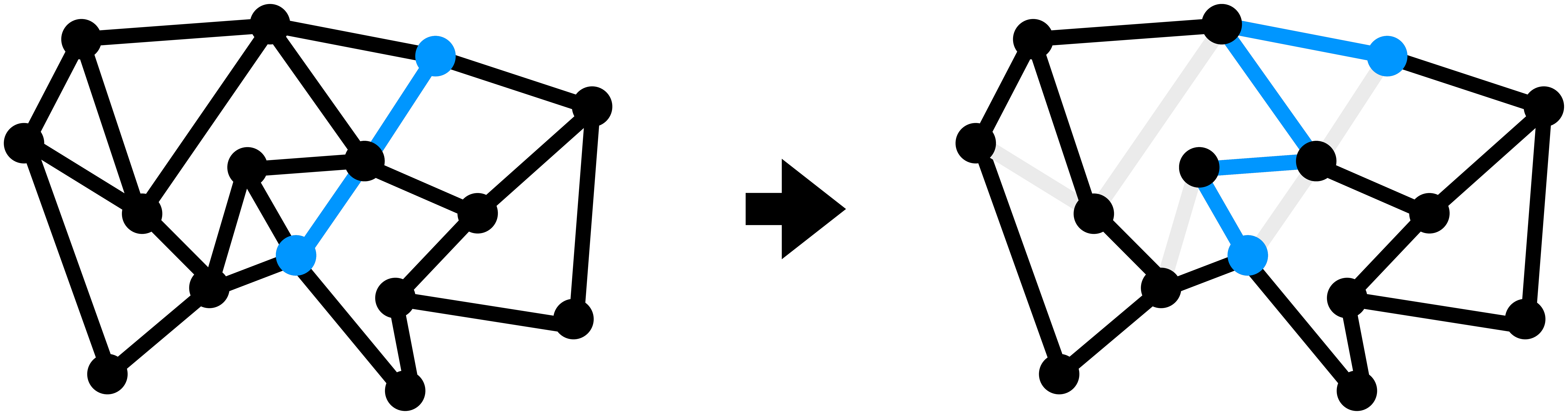


Theorem: every graph G has a t-spanner H w/

- **Distortion:** $t = O(\log n)$
- **Size:** $|H| = O(n)$

Simple Observation

Edge Spanners



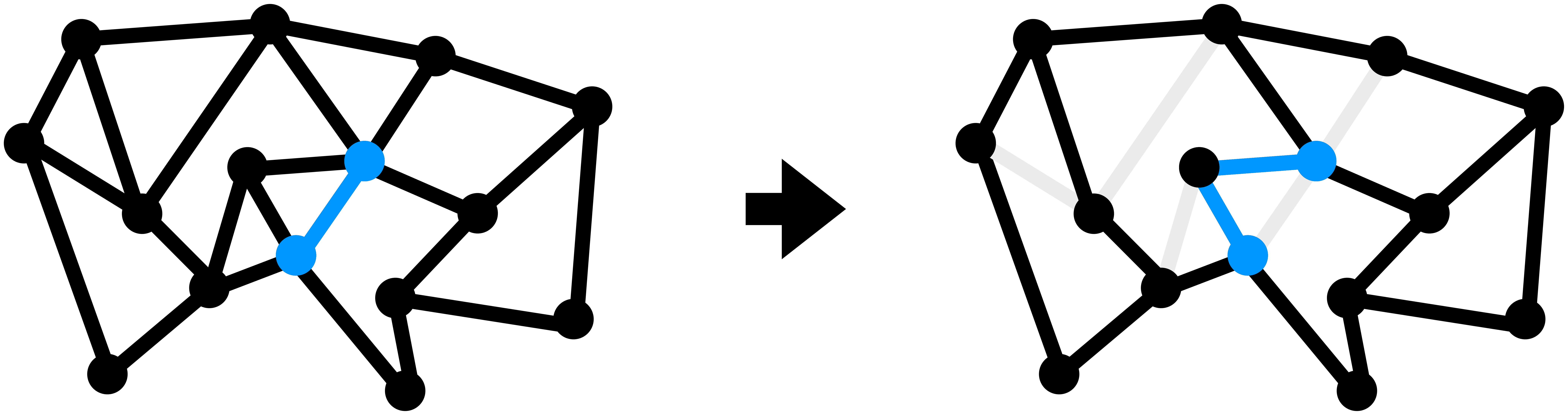
Definition (spanner): given graph $G = (V, E)$ and $t \geq 1$,

a t-spanner H is a subgraph of G satisfying

$$d_H(u, v) \leq t \cdot d_G(u, v) \qquad \forall u, v \in V$$

Simple Observation

Edge Spanners



Definition (edge spanner): given graph $G = (V, E)$ and $t \geq 1$, a t-edge-spanner H is a subgraph of G satisfying

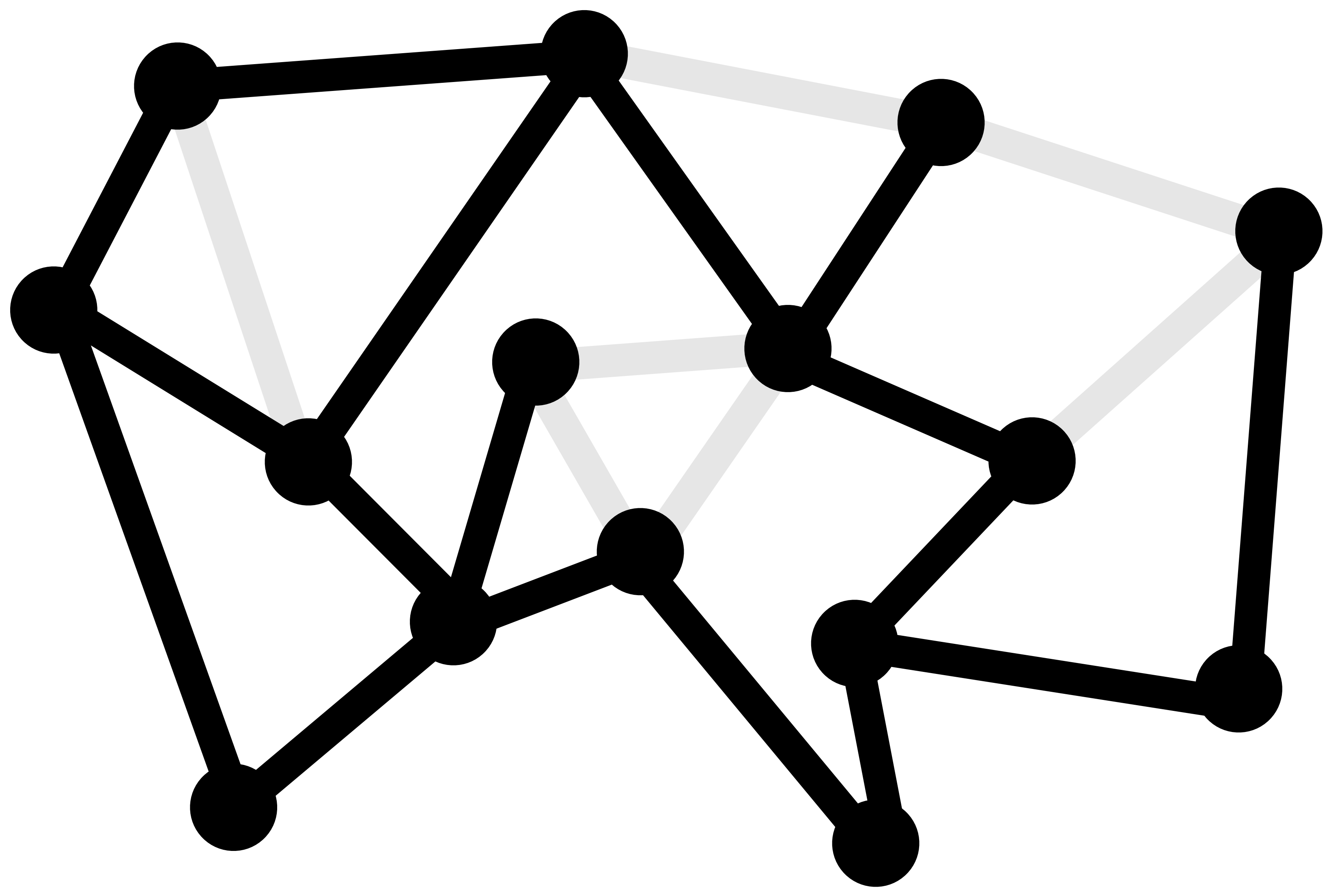
$$d_H(u, v) \leq t \qquad \forall \{u, v\} \in E$$

Simple Observation

Edge Spanners

all pairs distorted $\leq t$ all edges distorted $\leq t$

Claim: H is a t-spanner iff it is a t-edge-spanner



t-spanner

(trivially a t-edge-spanner)

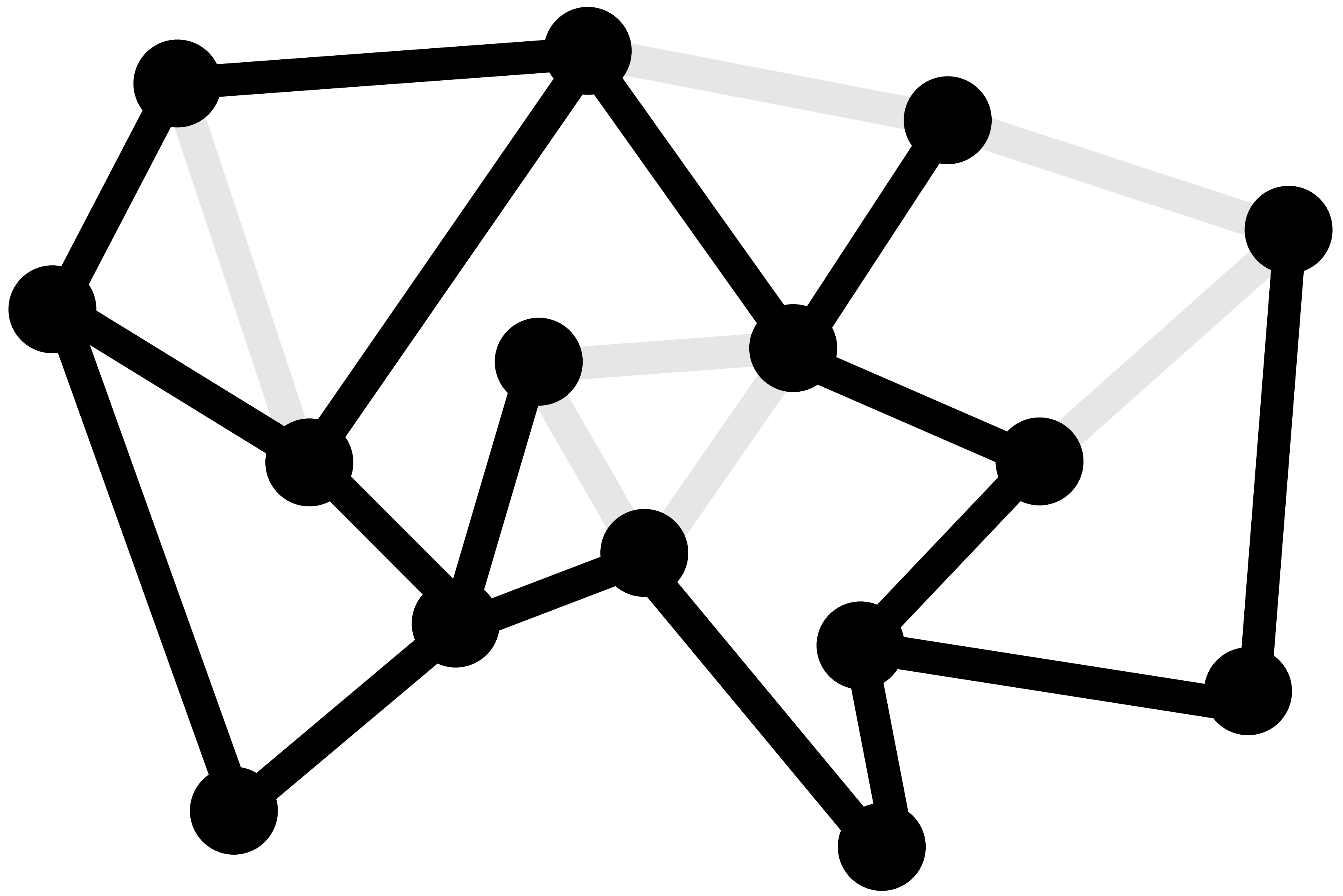
Simple Observation

Edge Spanners

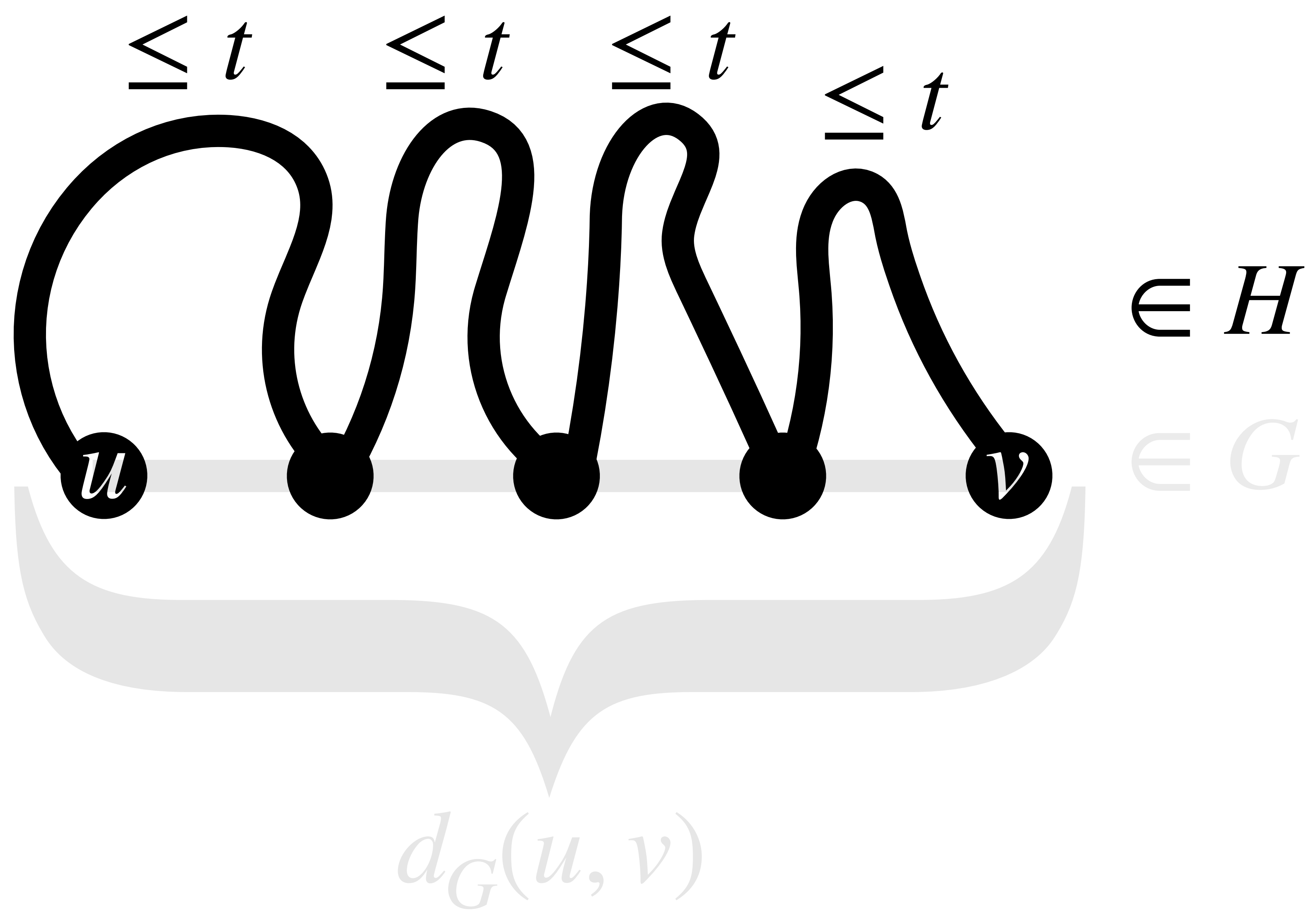
all pairs distorted $\leq t$

all edges distorted $\leq t$

Claim: H is a t-spanner iff it is a t-edge-spanner



t-edge-spanner H



So $d_H(u, v) \leq t \cdot d_G(u, v)$

Roadmap of Proof

1. **Simple Observation**

   edge spanners suffice

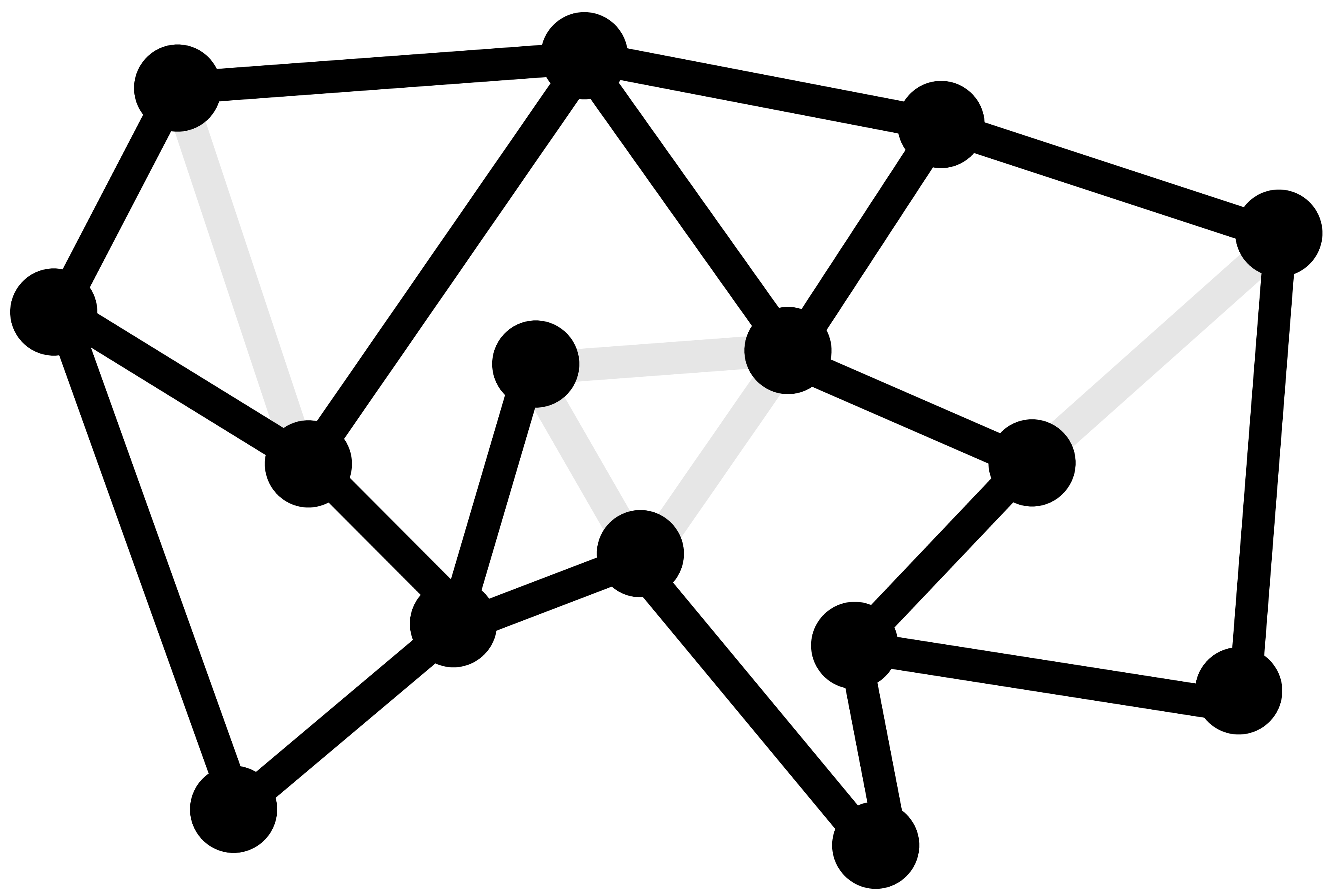
2. **Greedy Algorithm**

   suggested by observation

3. **Distortion Analysis**

4. **Size Analysis**

   by "Moore Bounds"



Theorem: every graph G has a t-spanner H w/

- **Distortion:** $t = O(\log n)$
- **Size:** $|H| = O(n)$

Roadmap of Proof

1. **Simple Observation** ✓

   edge spanners suffice

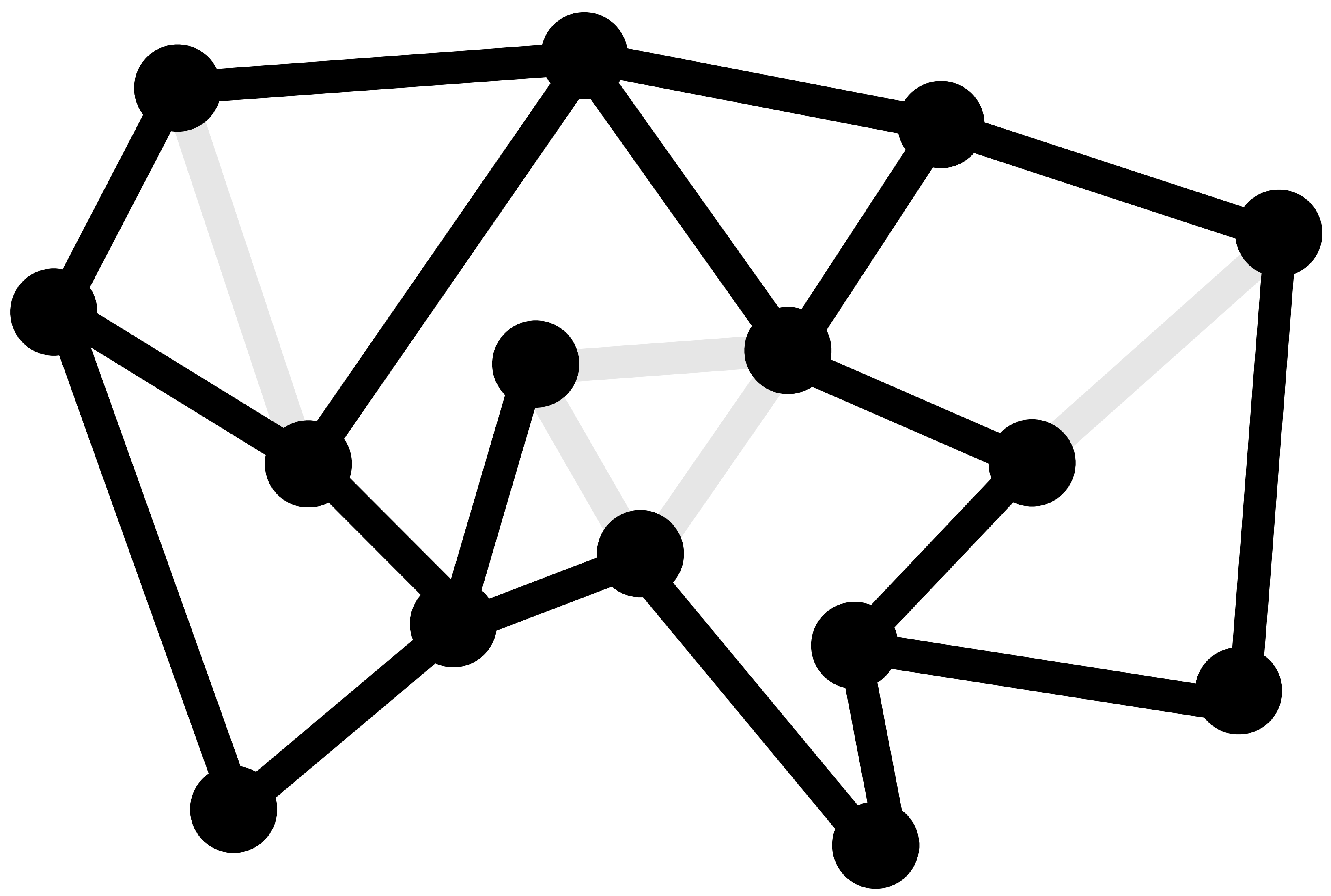
2. **Greedy Algorithm**

   suggested by observation

3. **Distortion Analysis**

4. **Size Analysis**

   by "Moore Bounds"



Theorem: every graph G has a t-spanner H w/

- **Distortion:** $t = O(\log n)$
- **Size:** $|H| = O(n)$

Greedy Algorithm

Claim: H is a t-spanner iff it is a t-edge-spanner

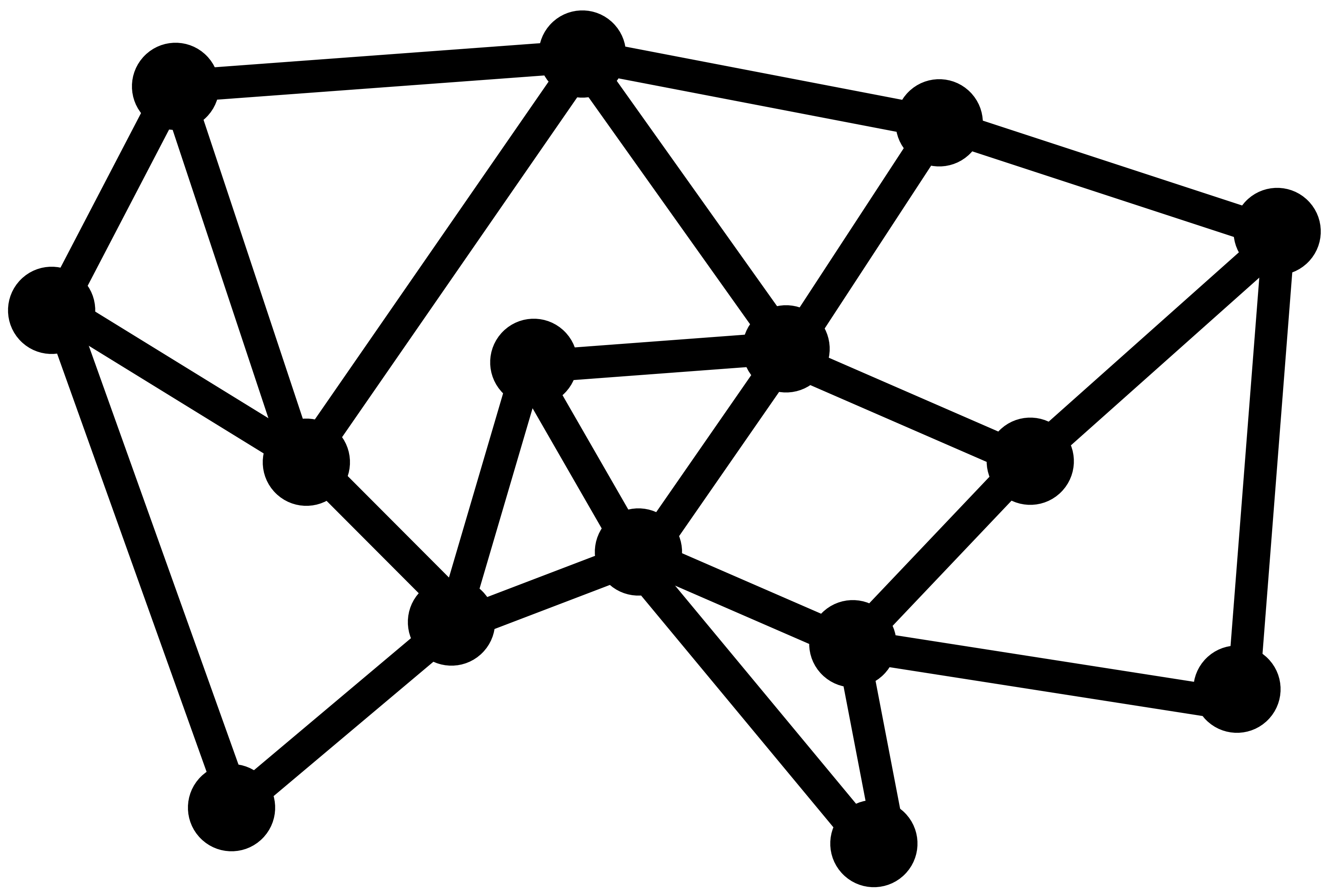
suggests

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



$t = 3$

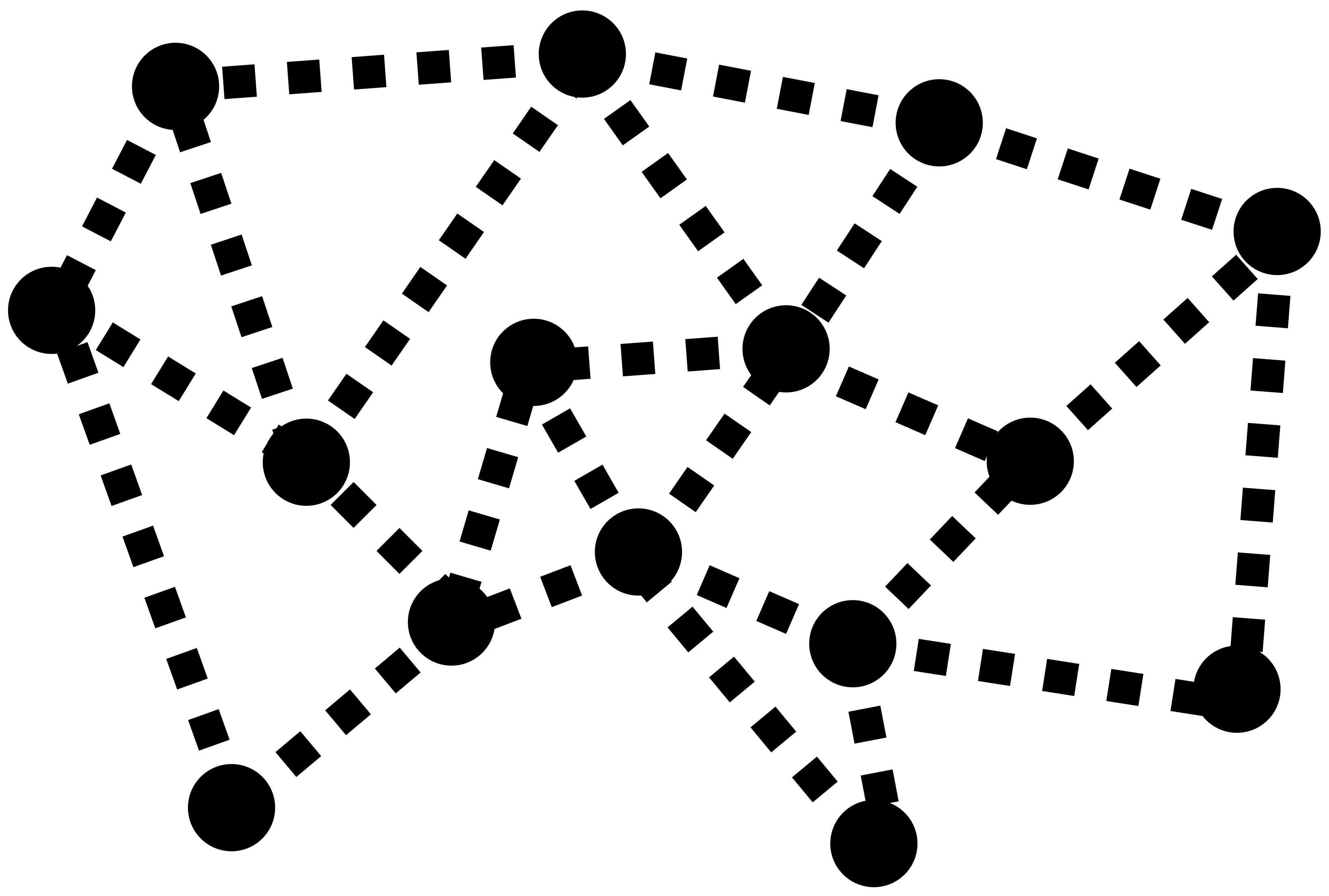
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



$t = 3$

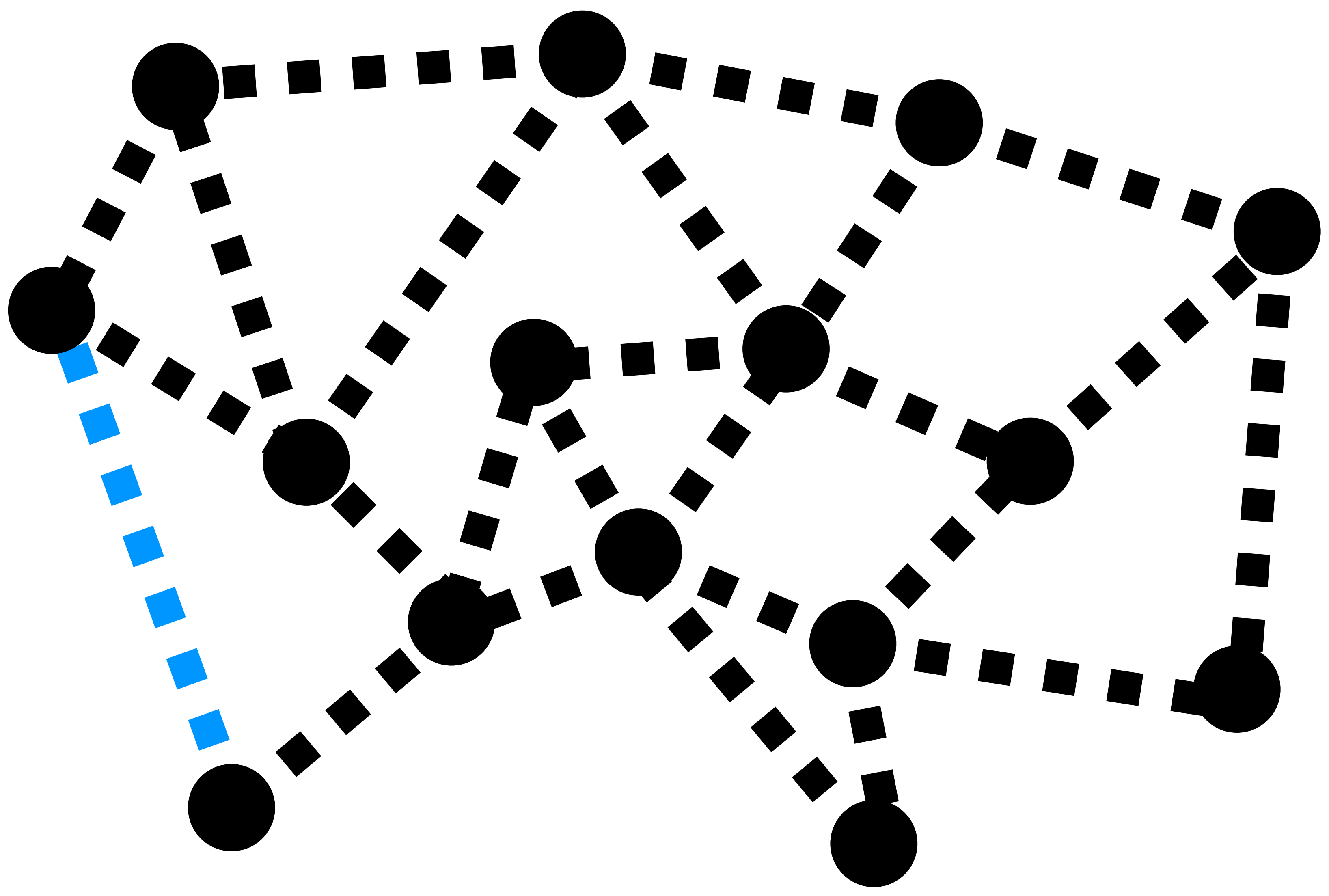
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$

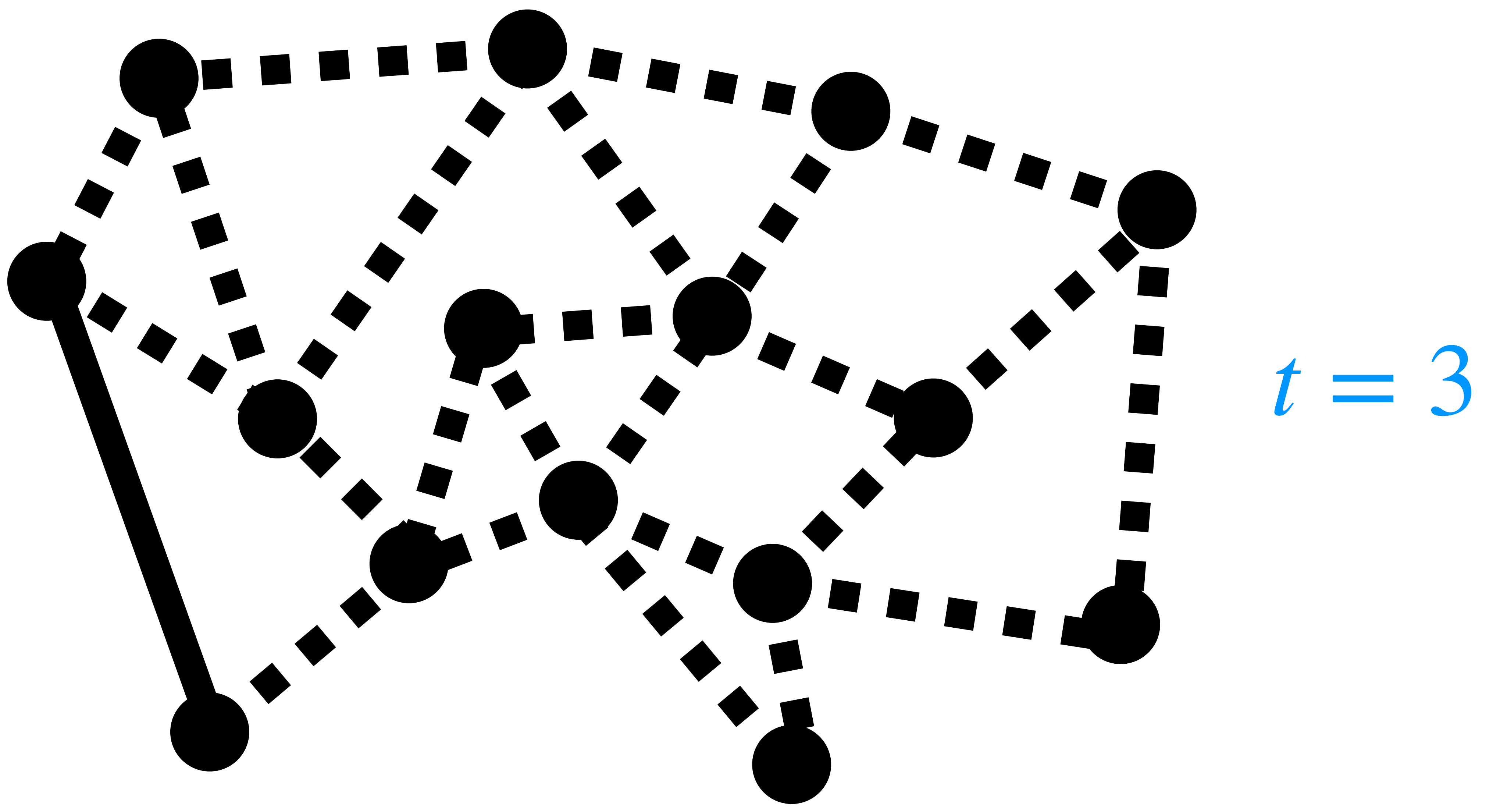


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



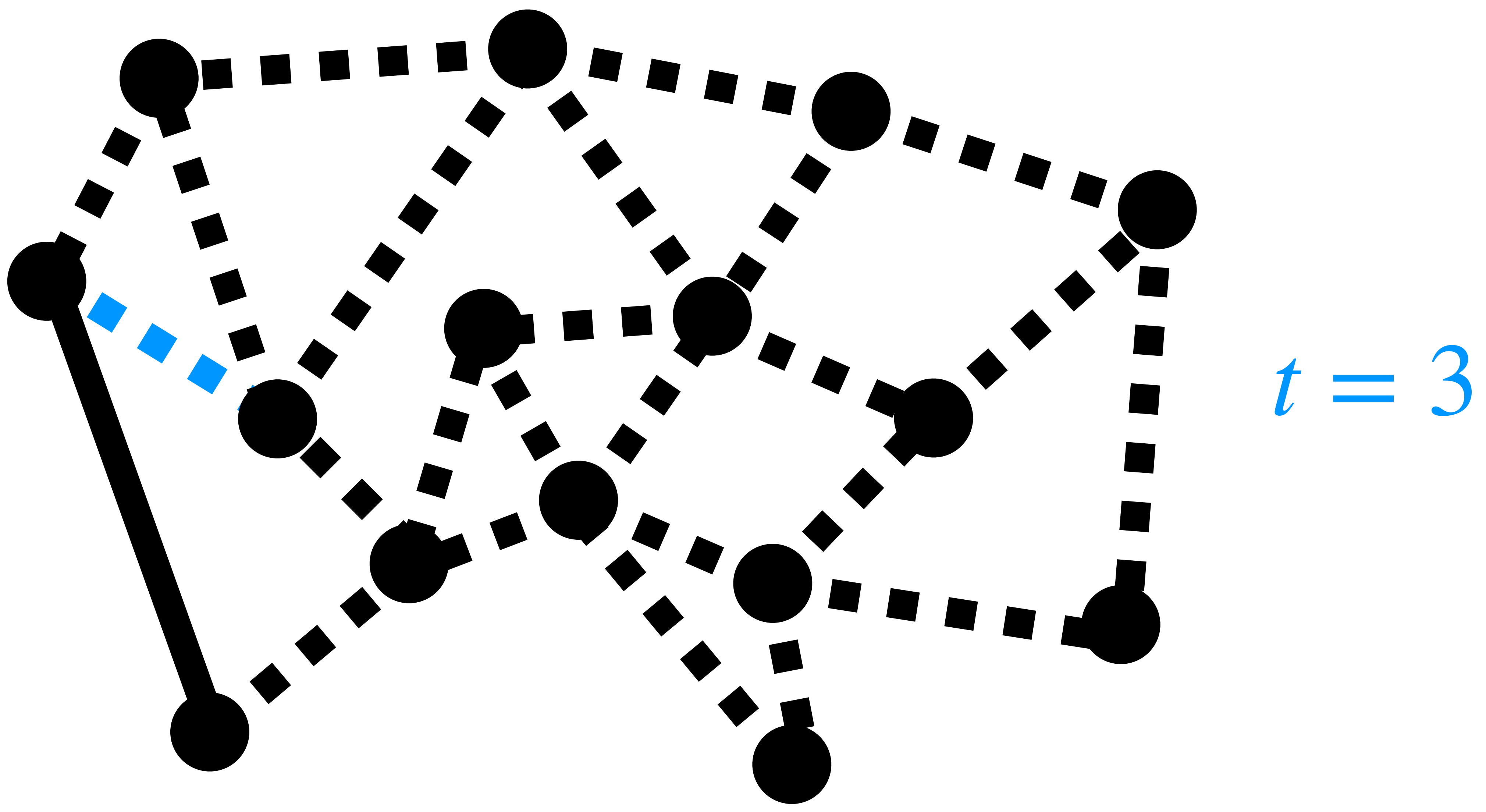
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



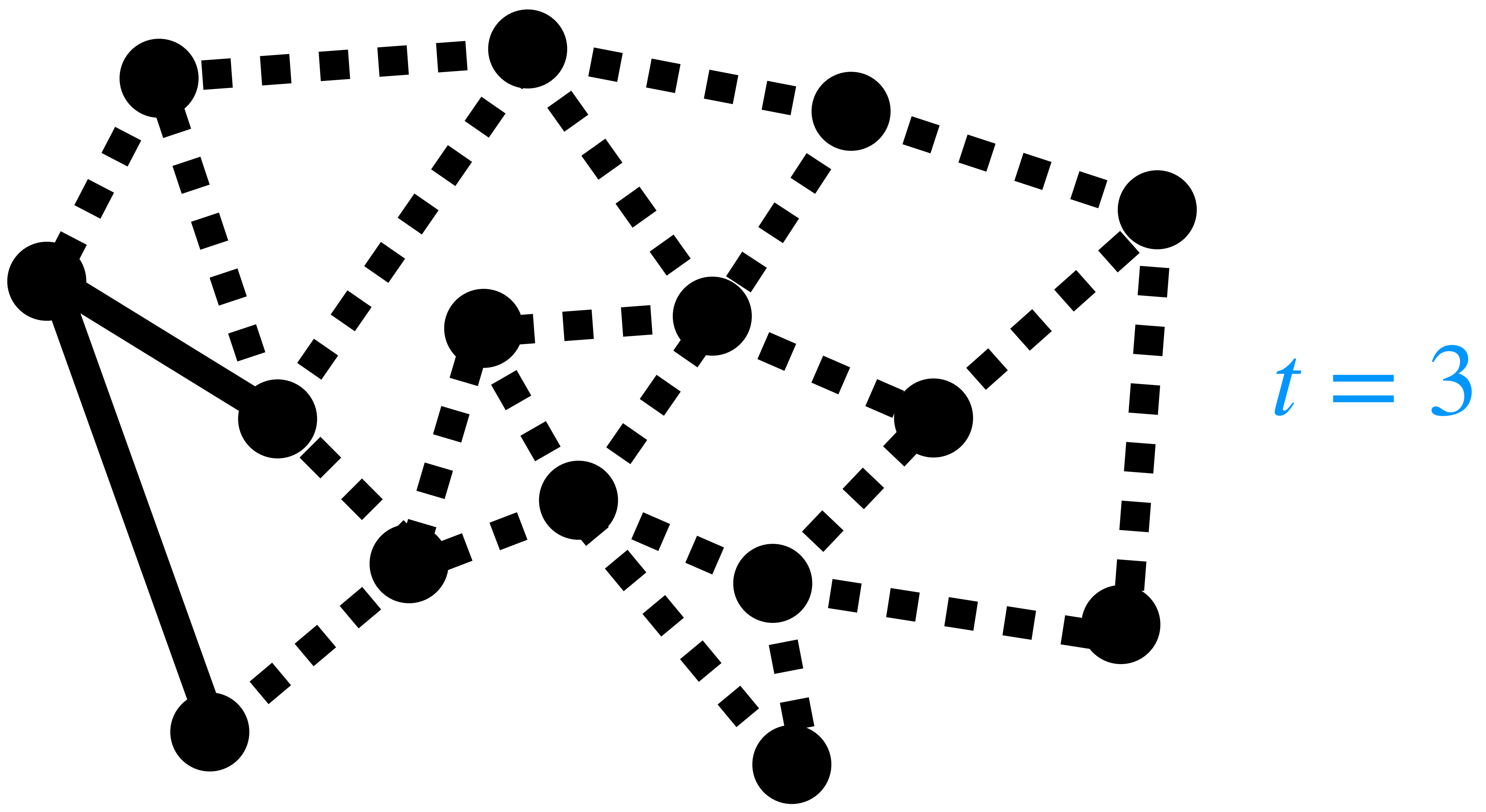
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



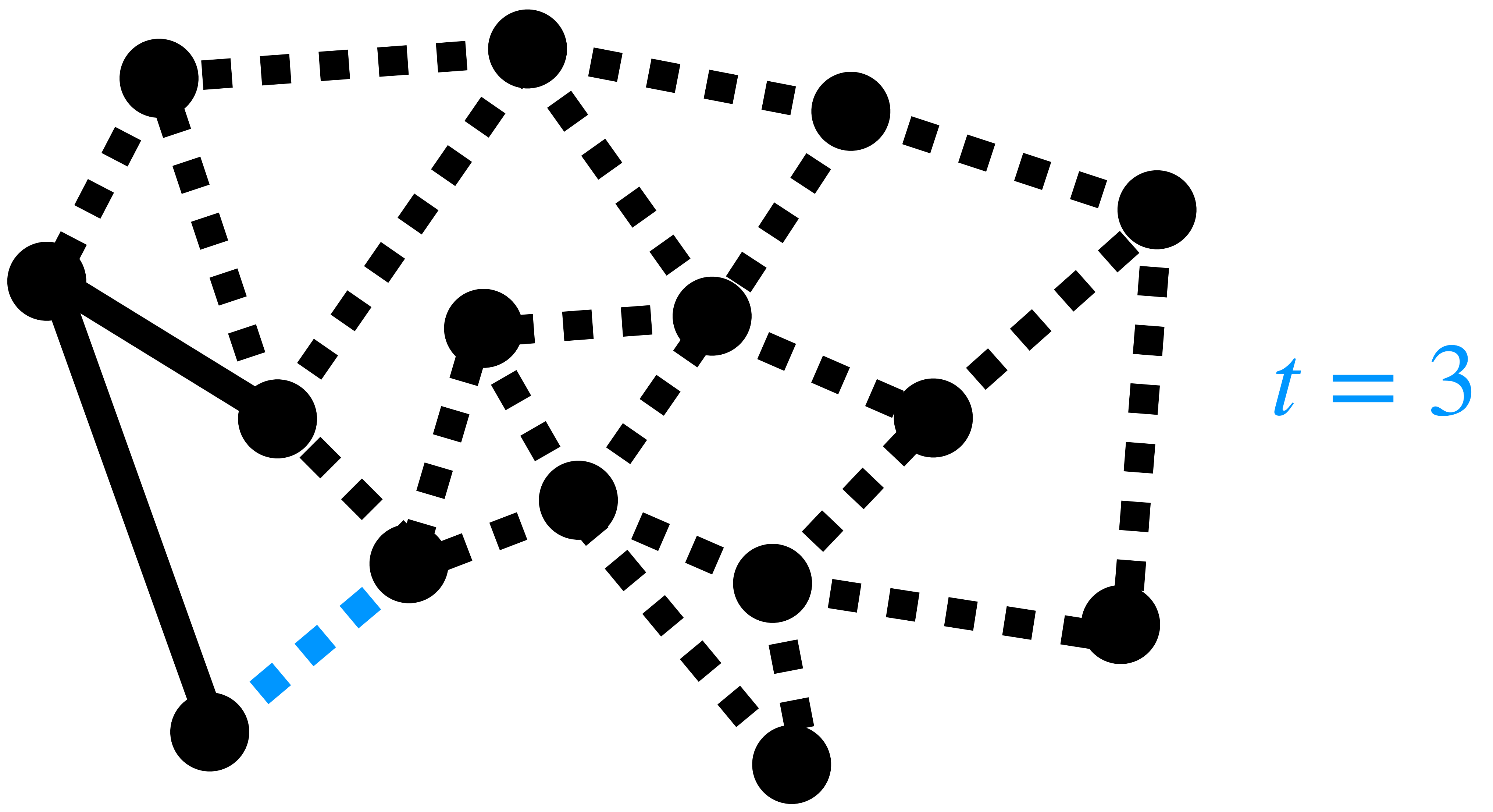
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



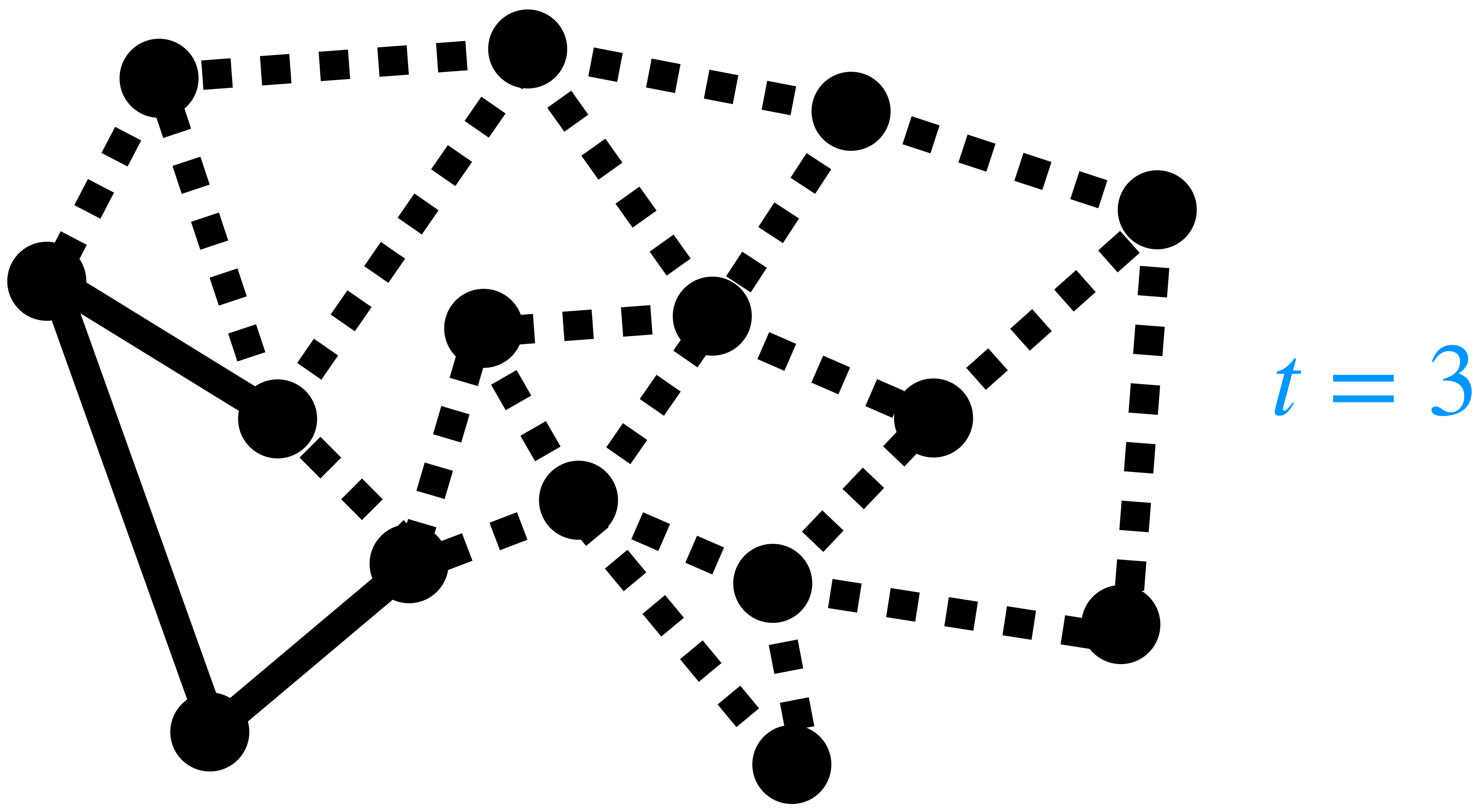
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \varnothing$
- For $\{u, v\} \in E$:
    - If $d_H(u, v) > t$ then

      $H \leftarrow H + \{u, v\}$



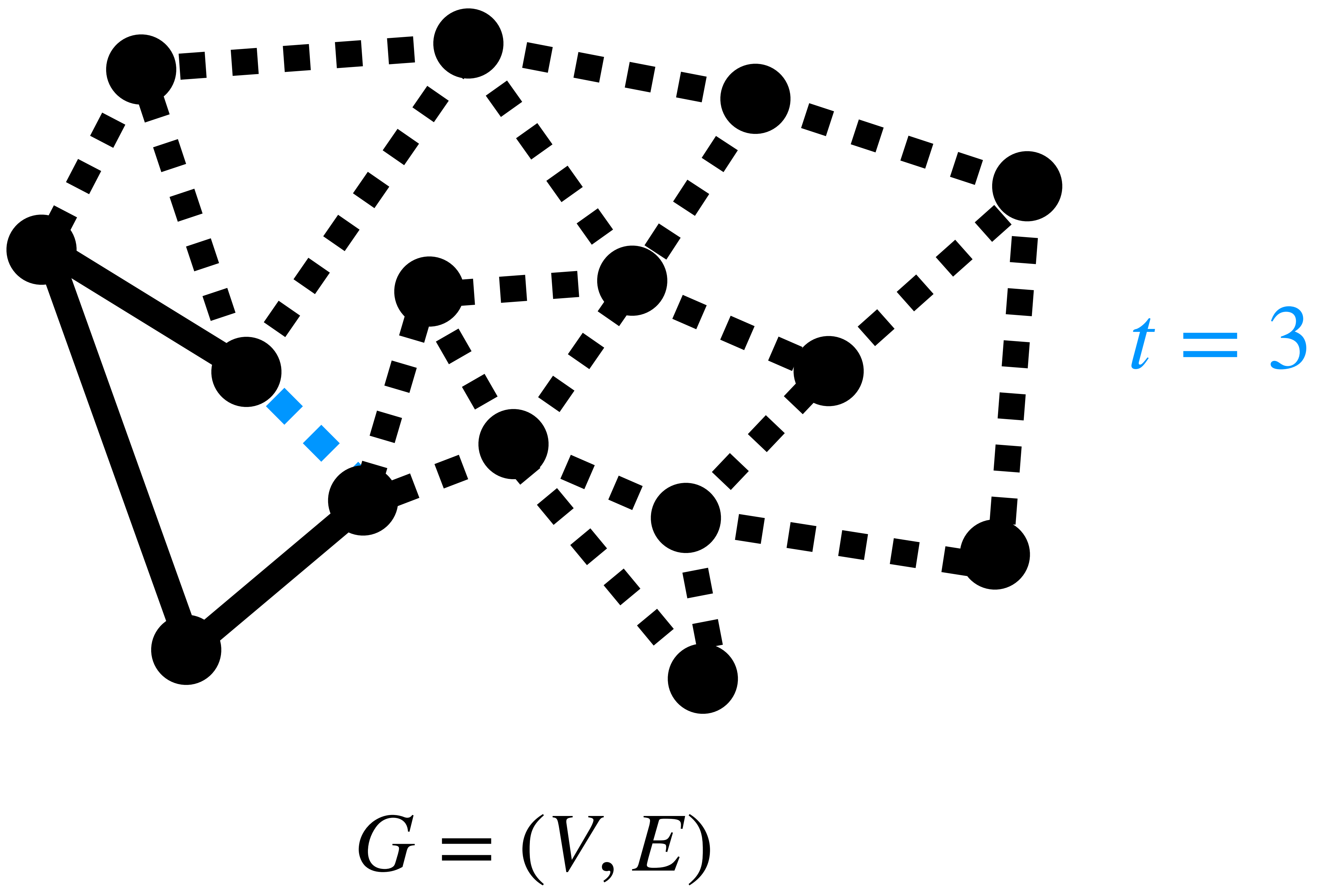
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$

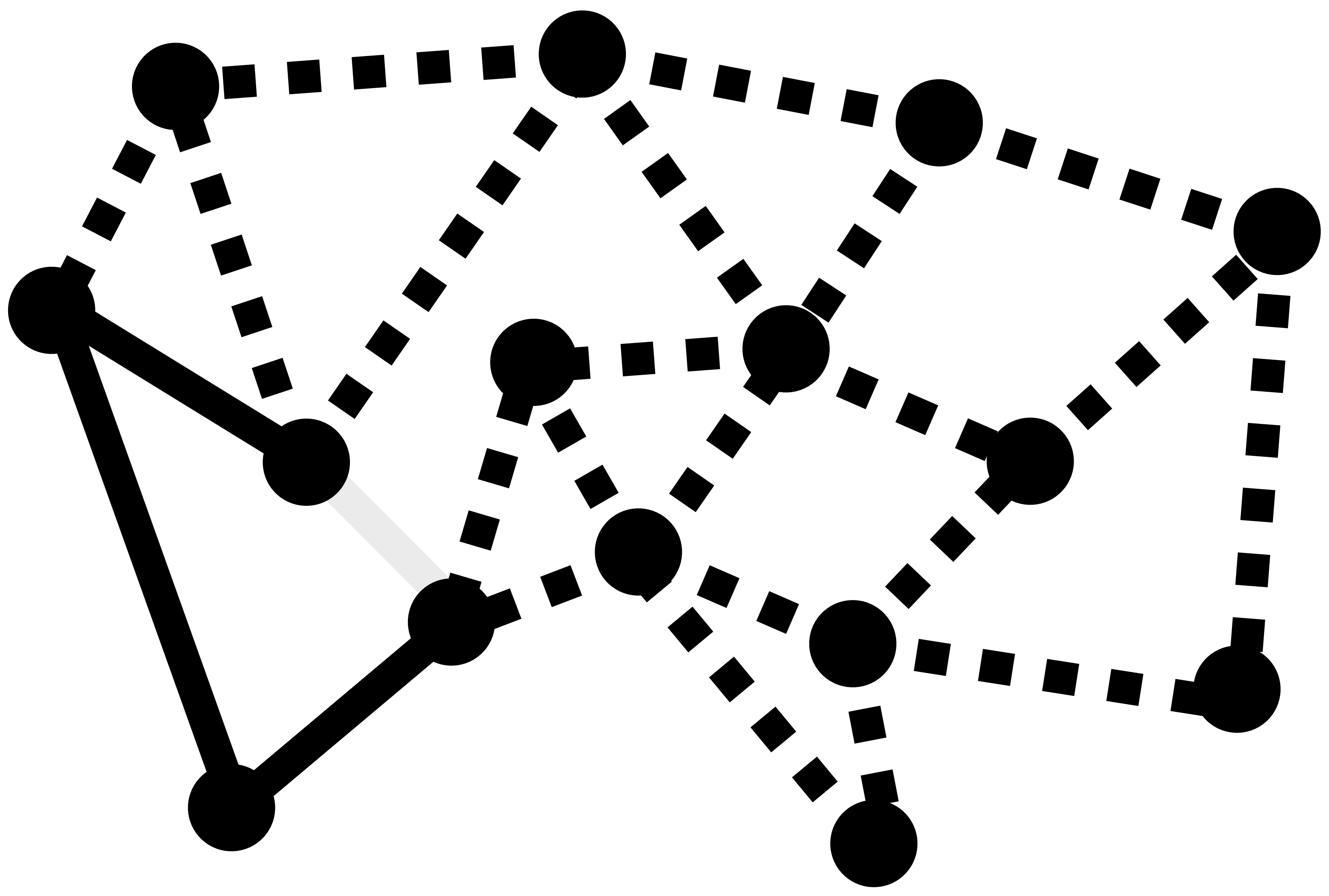


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$

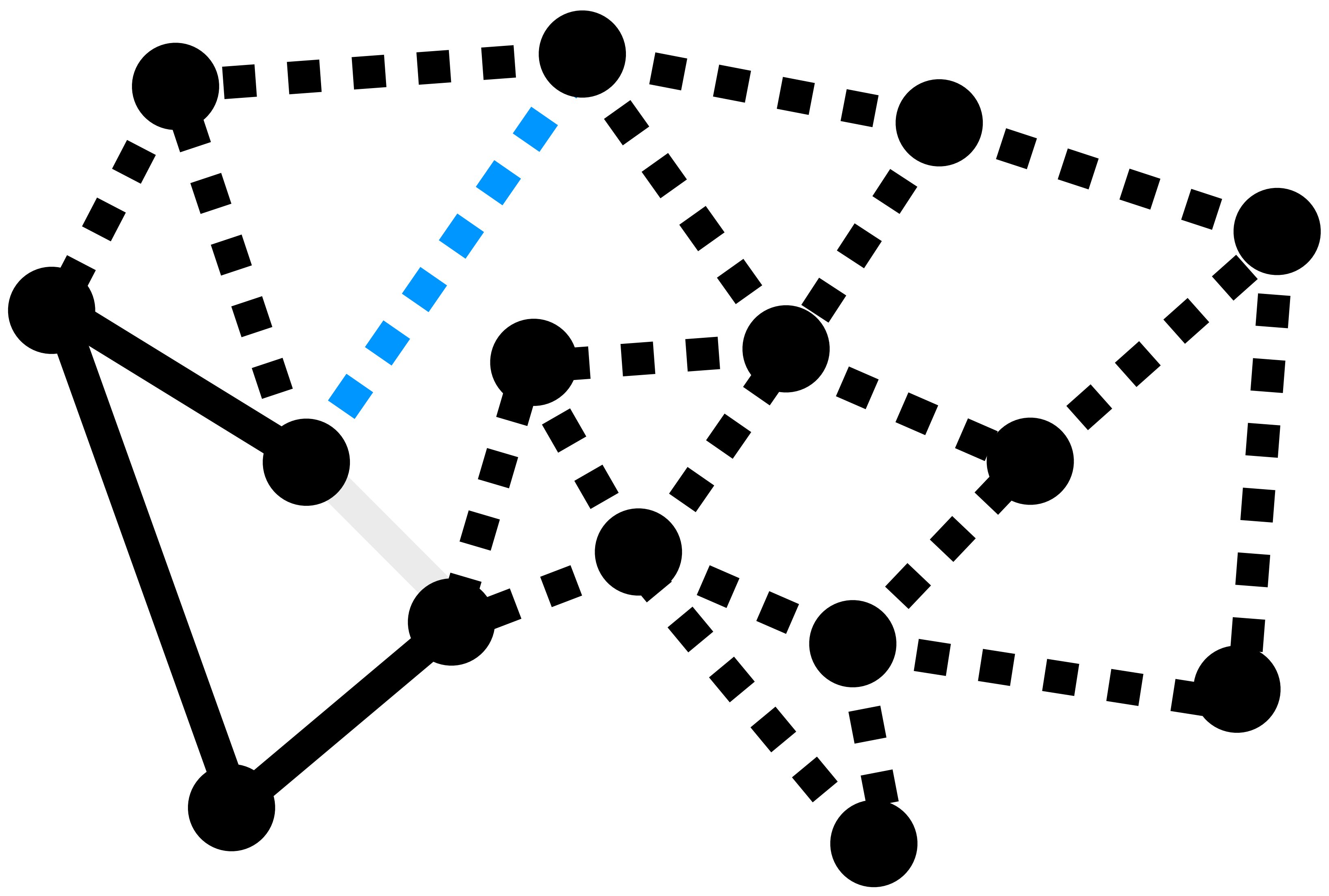


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



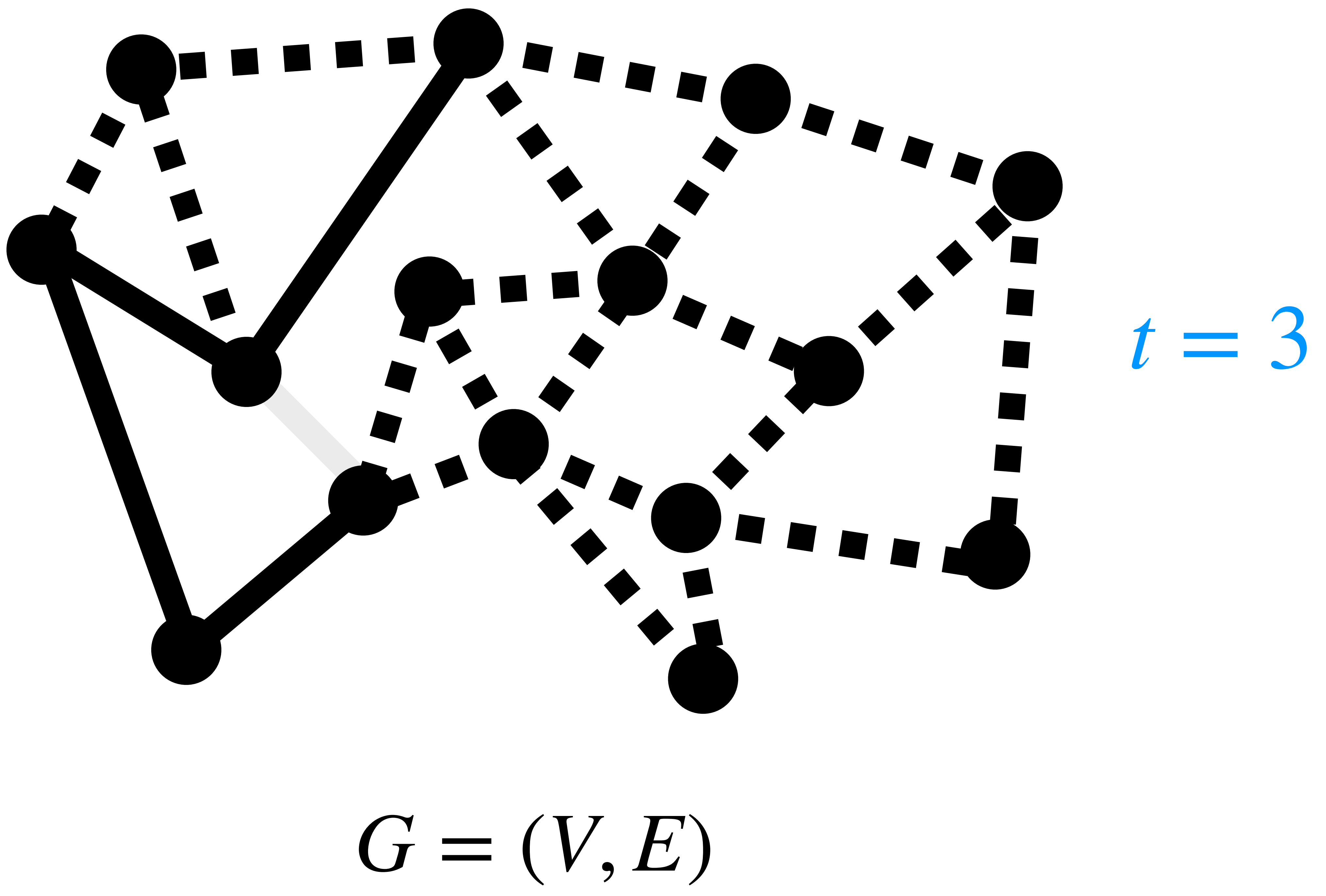
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$

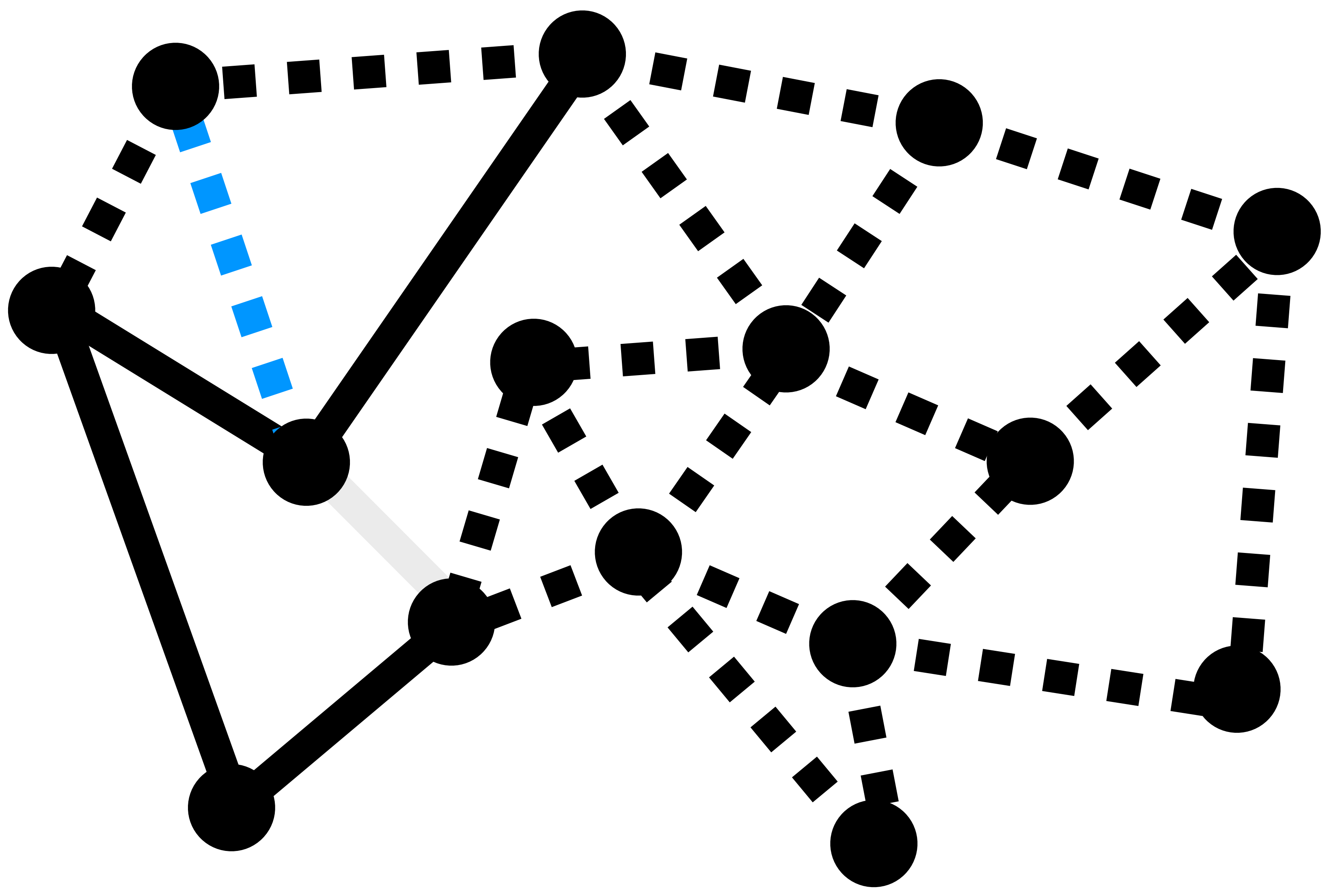


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \varnothing$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



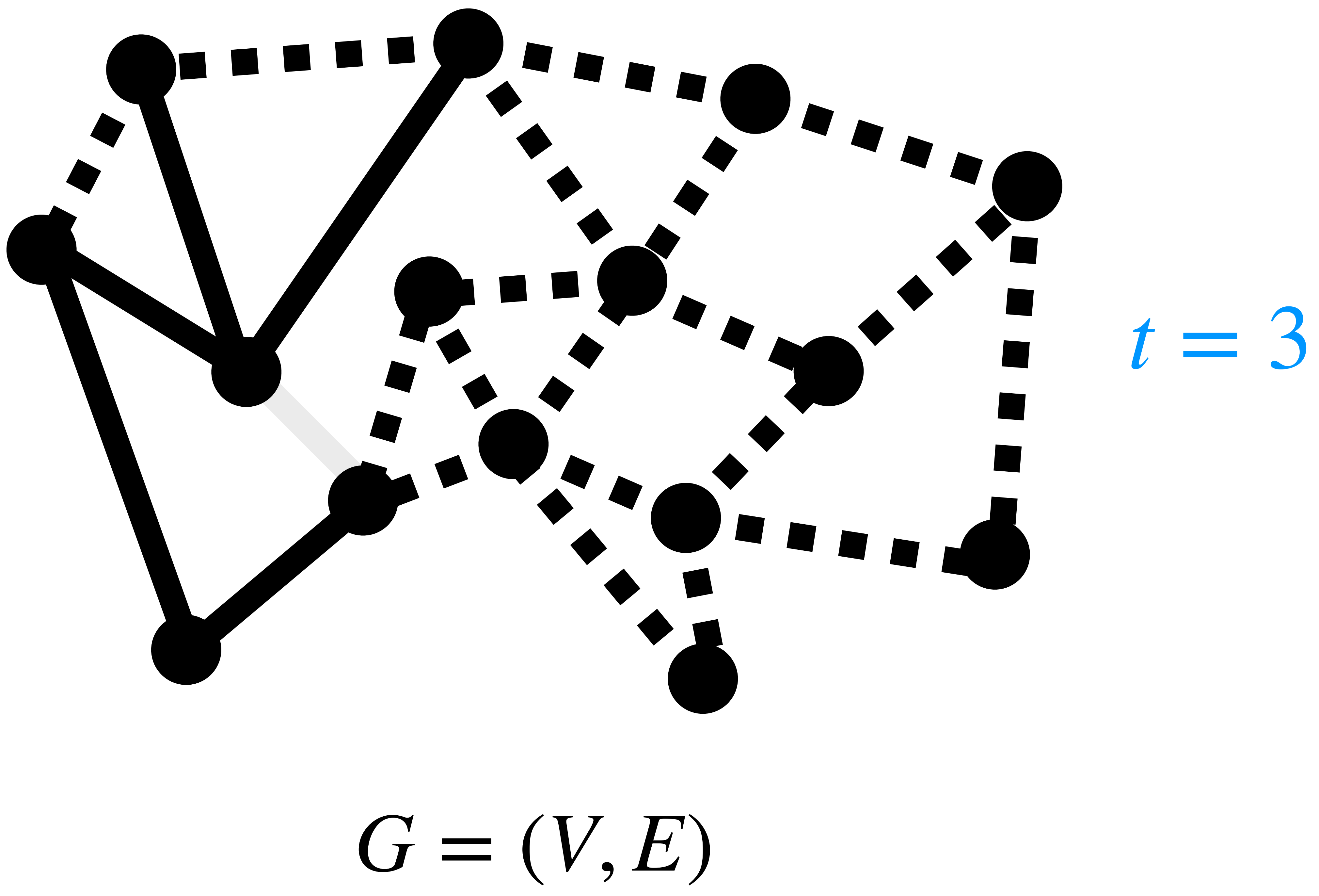
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$

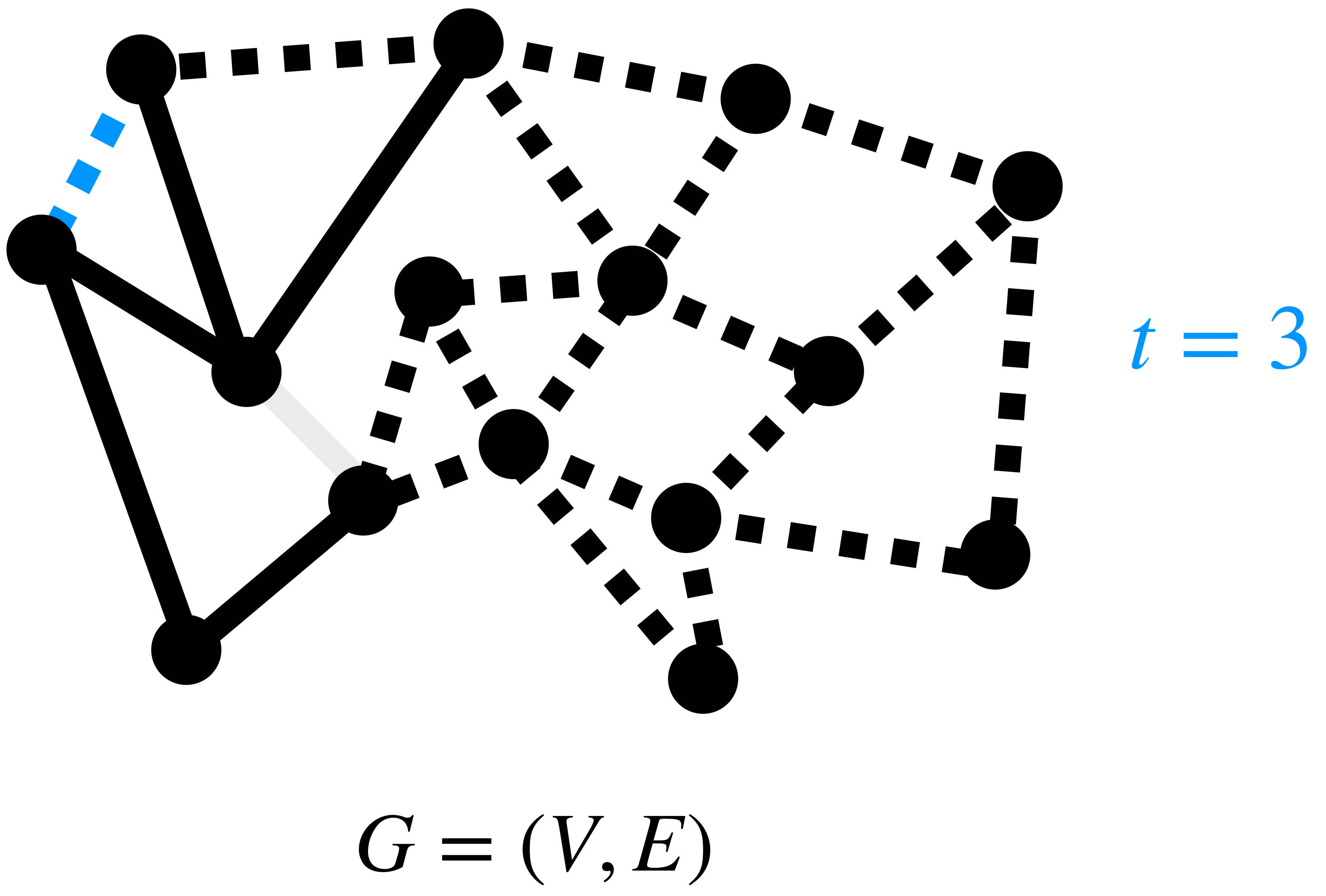


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$

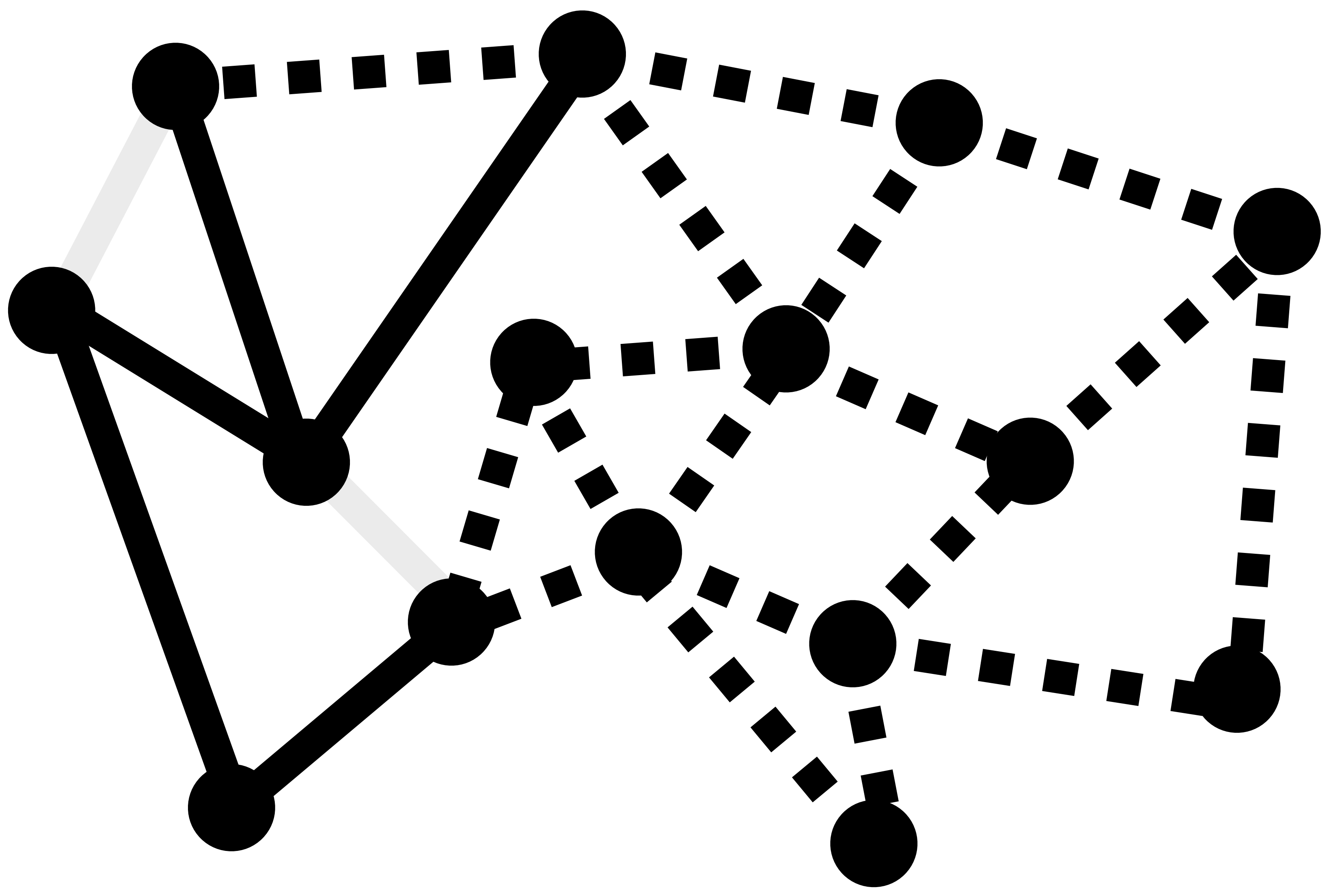


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$

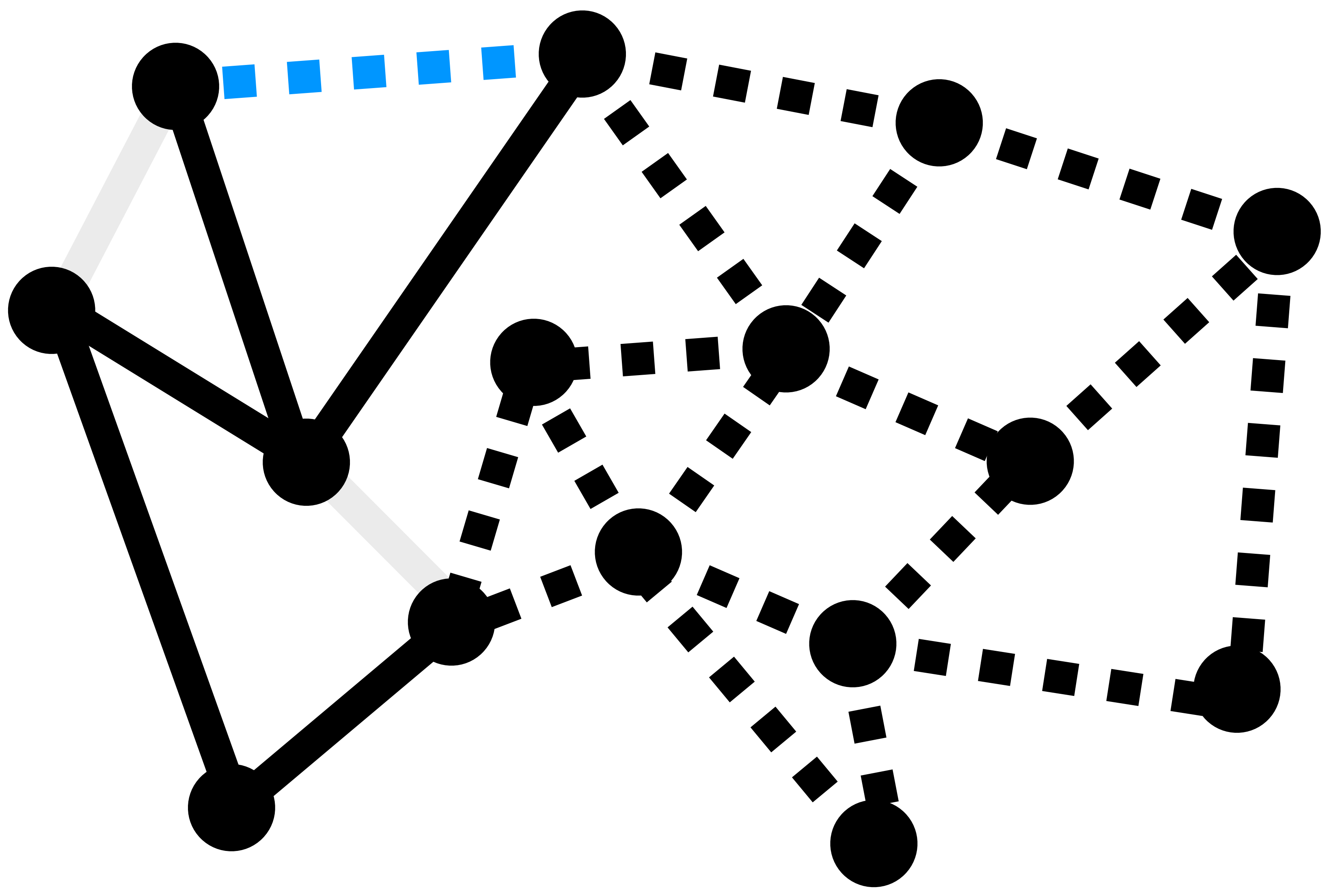


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



$t = 3$

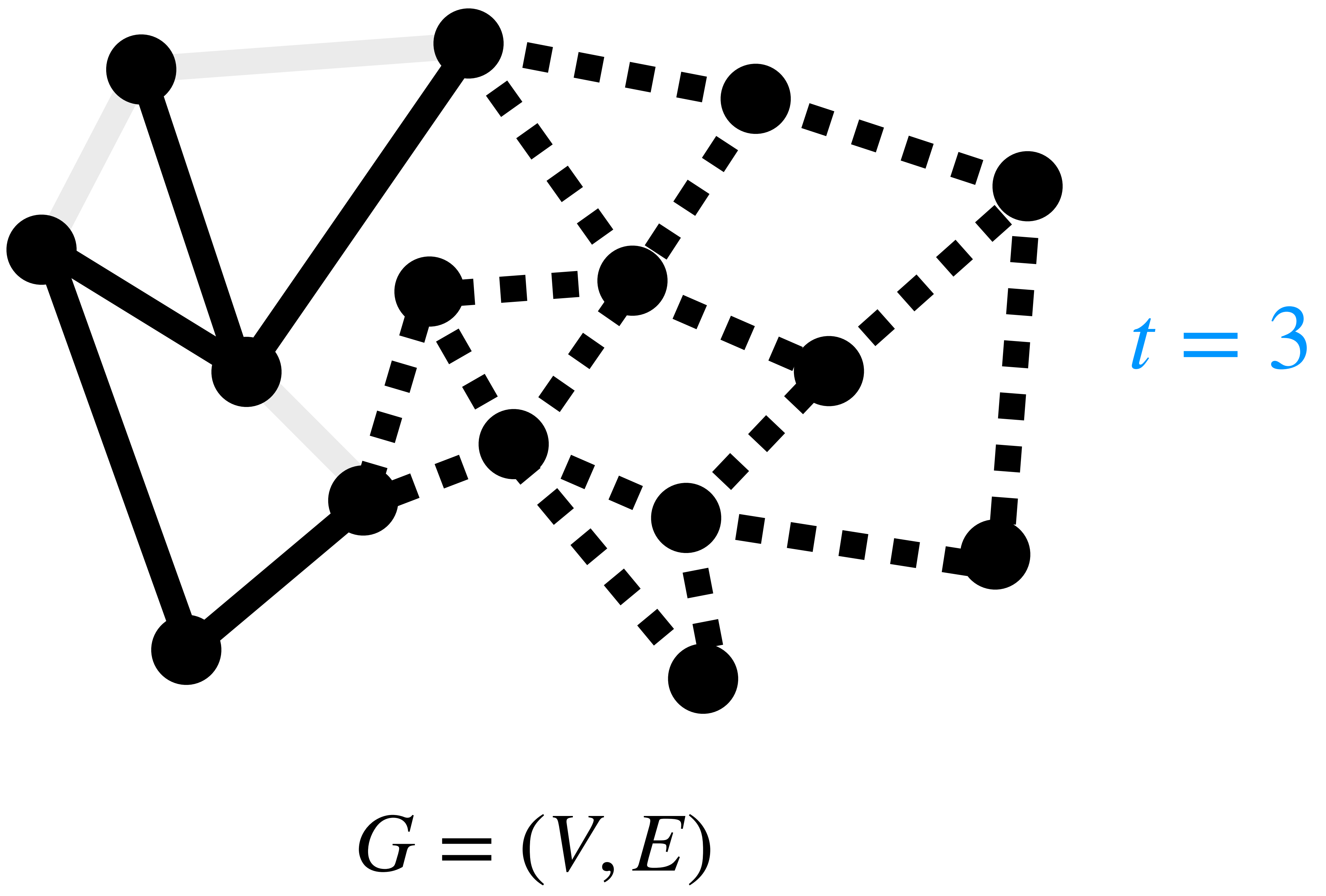
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$

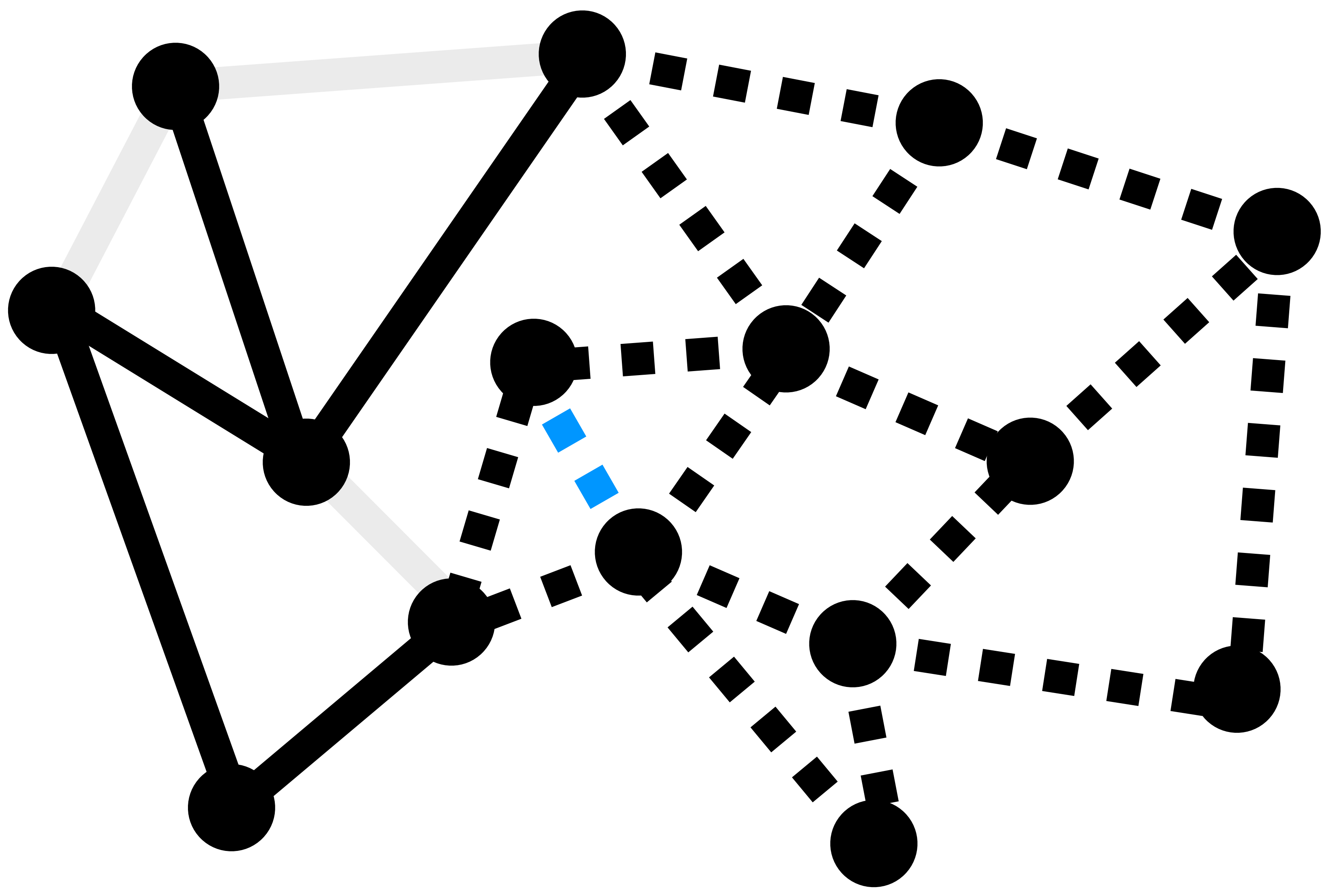


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \varnothing$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



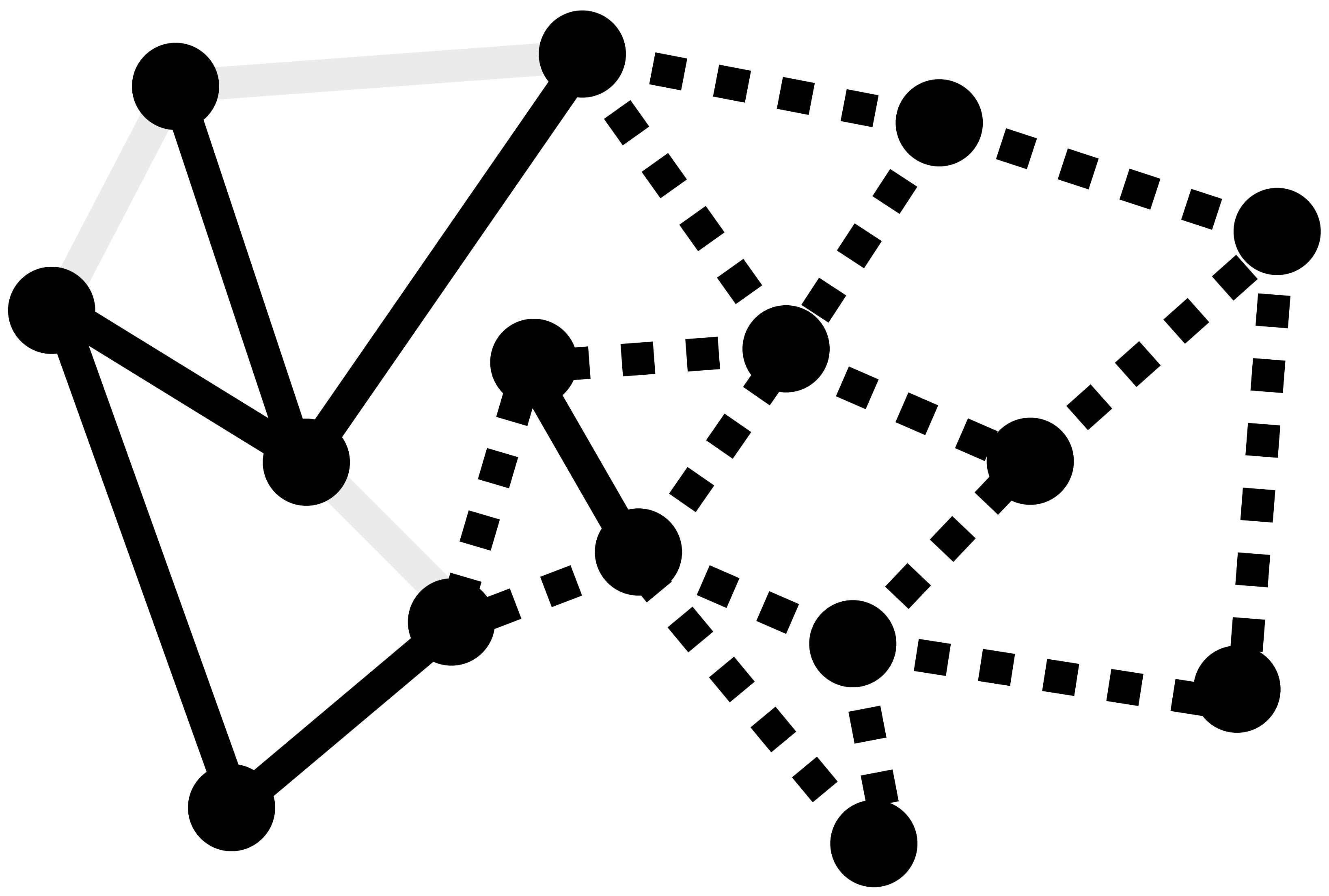
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



$t = 3$

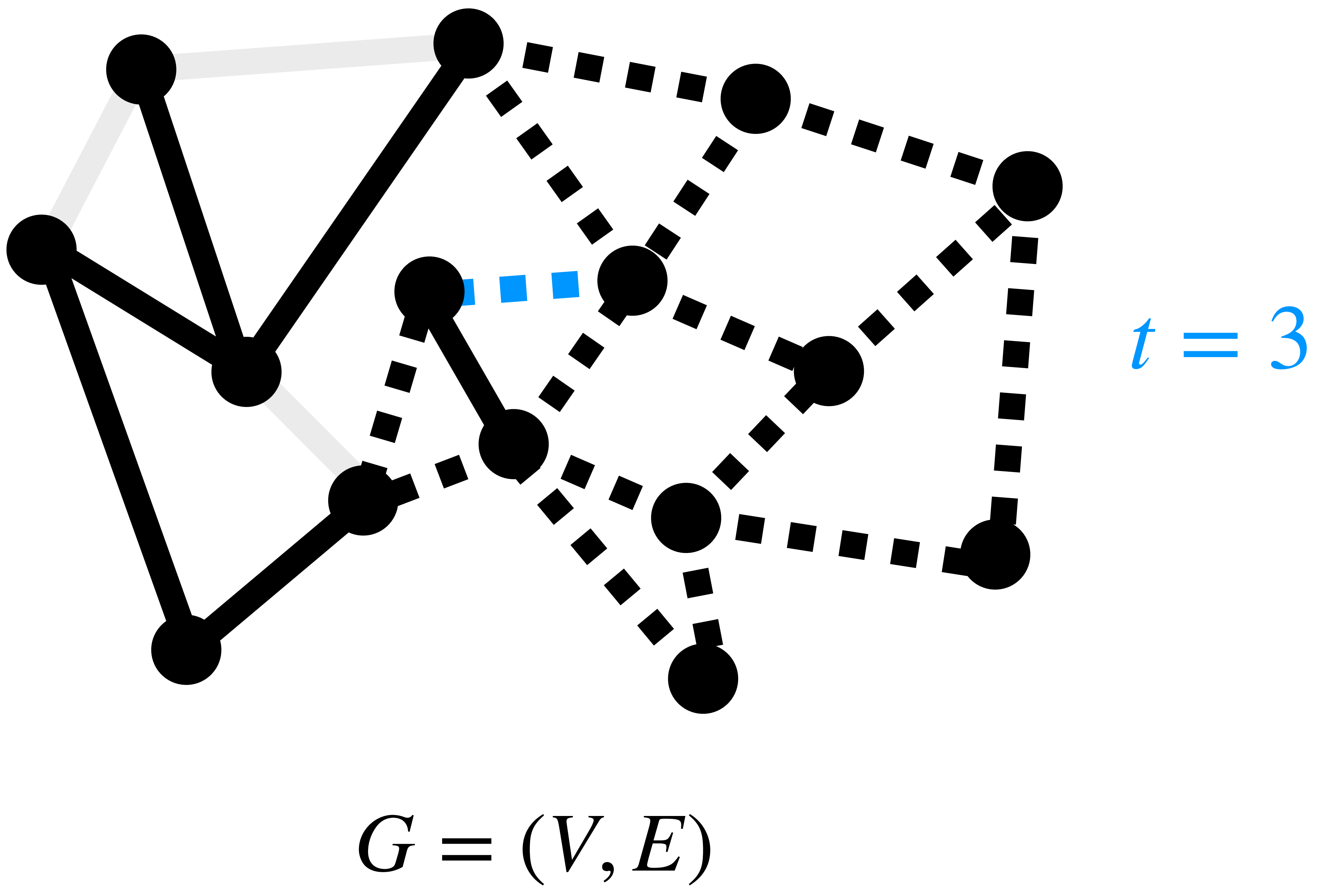
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $$H \leftarrow H + \{u, v\}$$

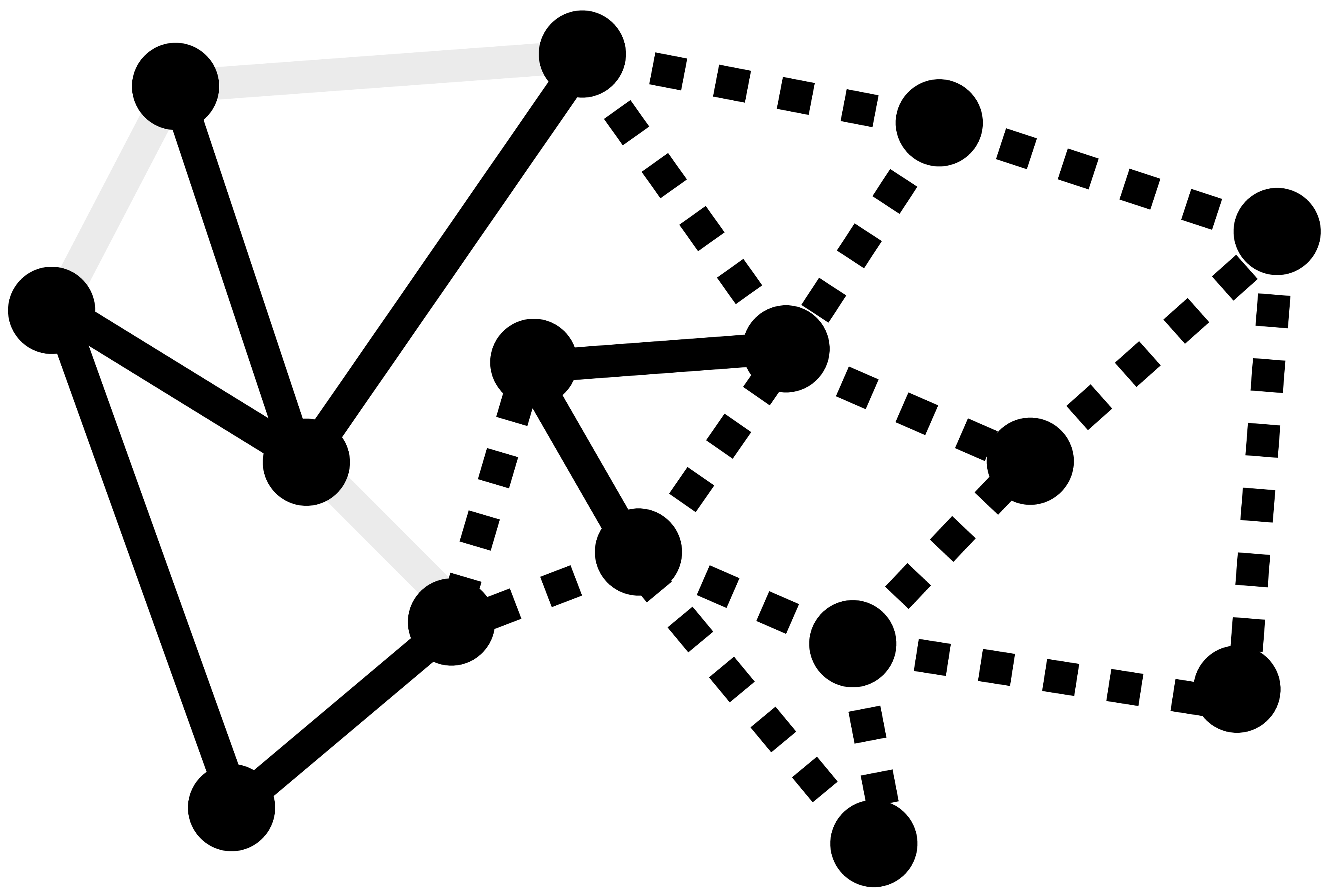


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \varnothing$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



$t = 3$

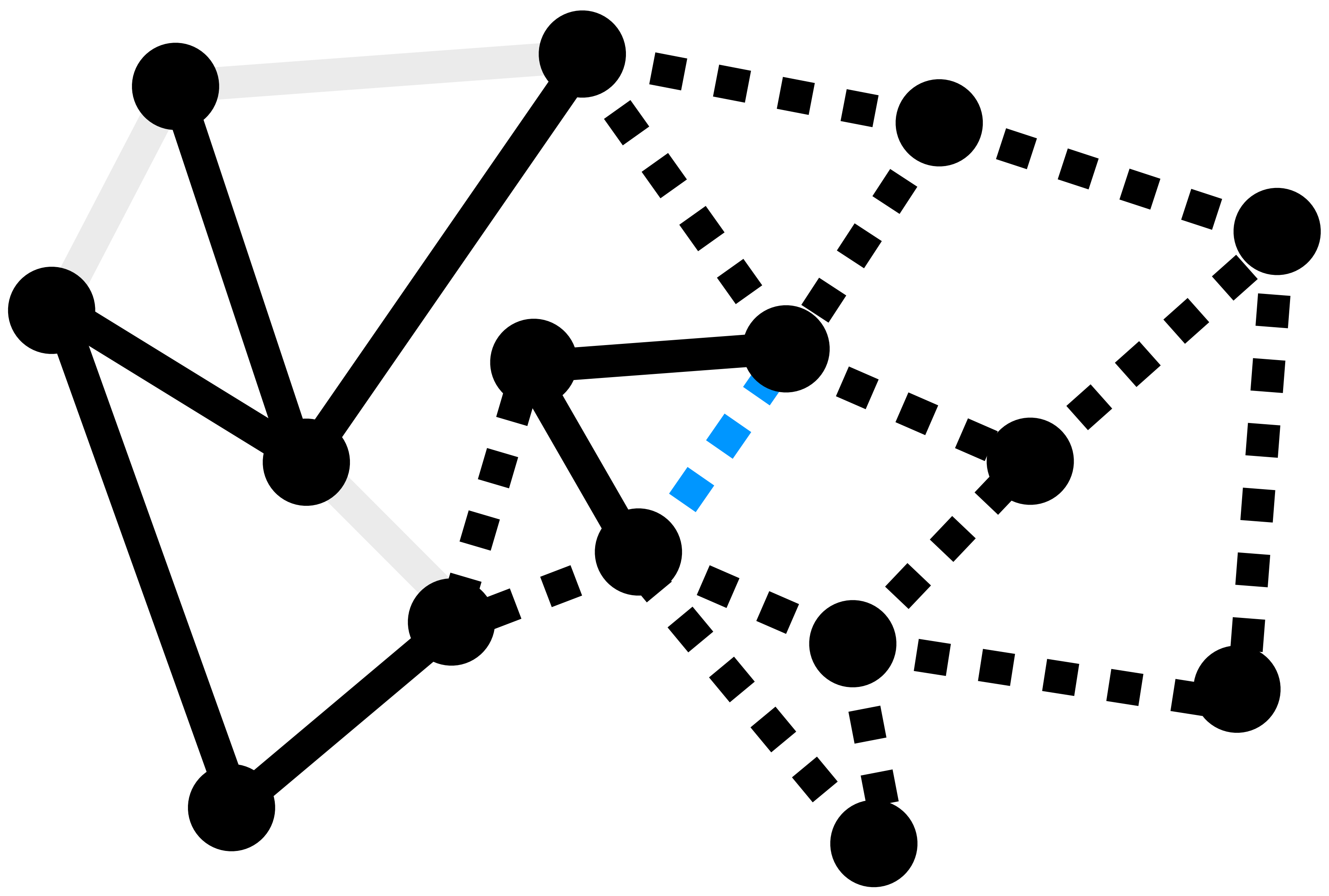
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \varnothing$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $$H \leftarrow H + \{u, v\}$$

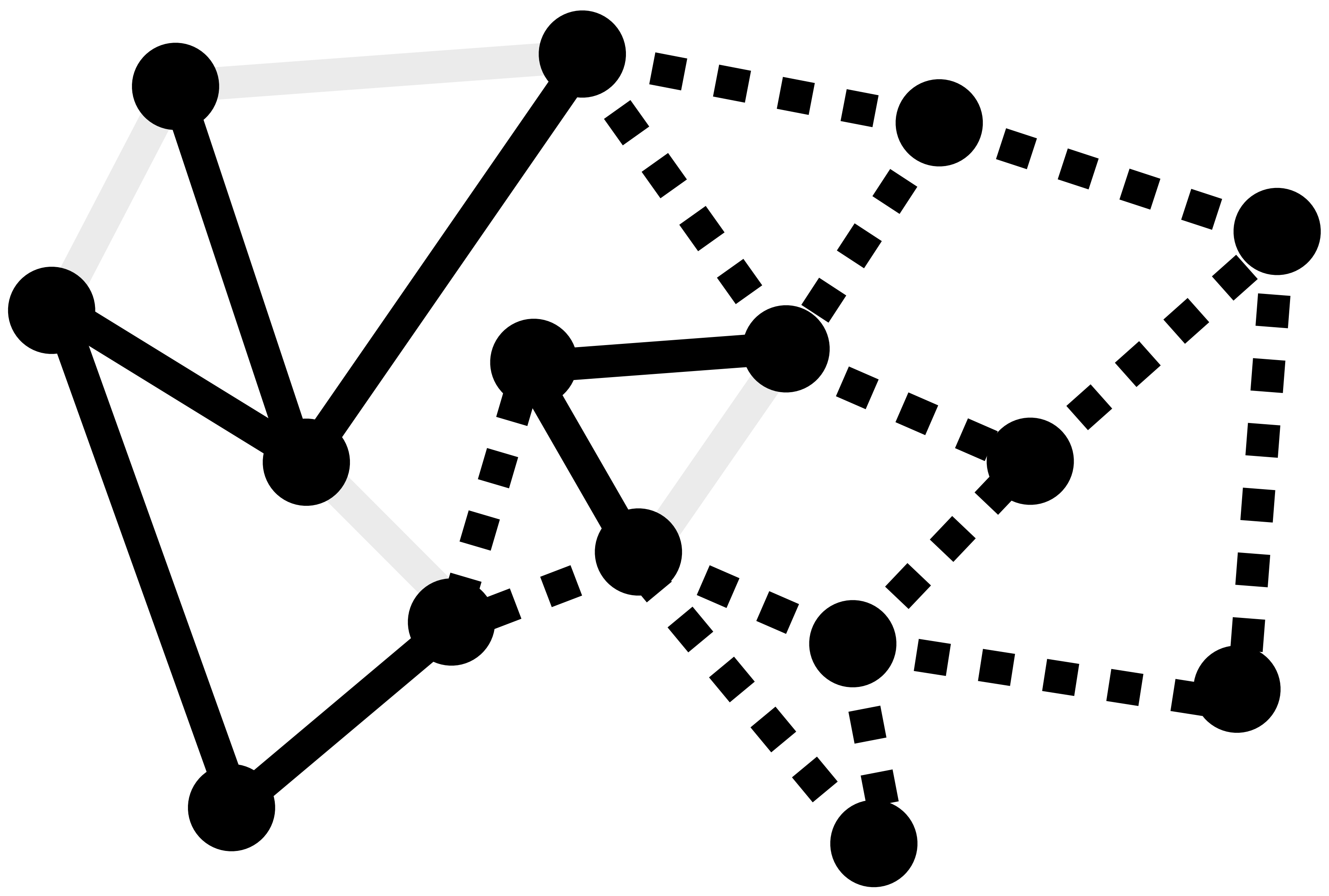


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \varnothing$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



$t = 3$

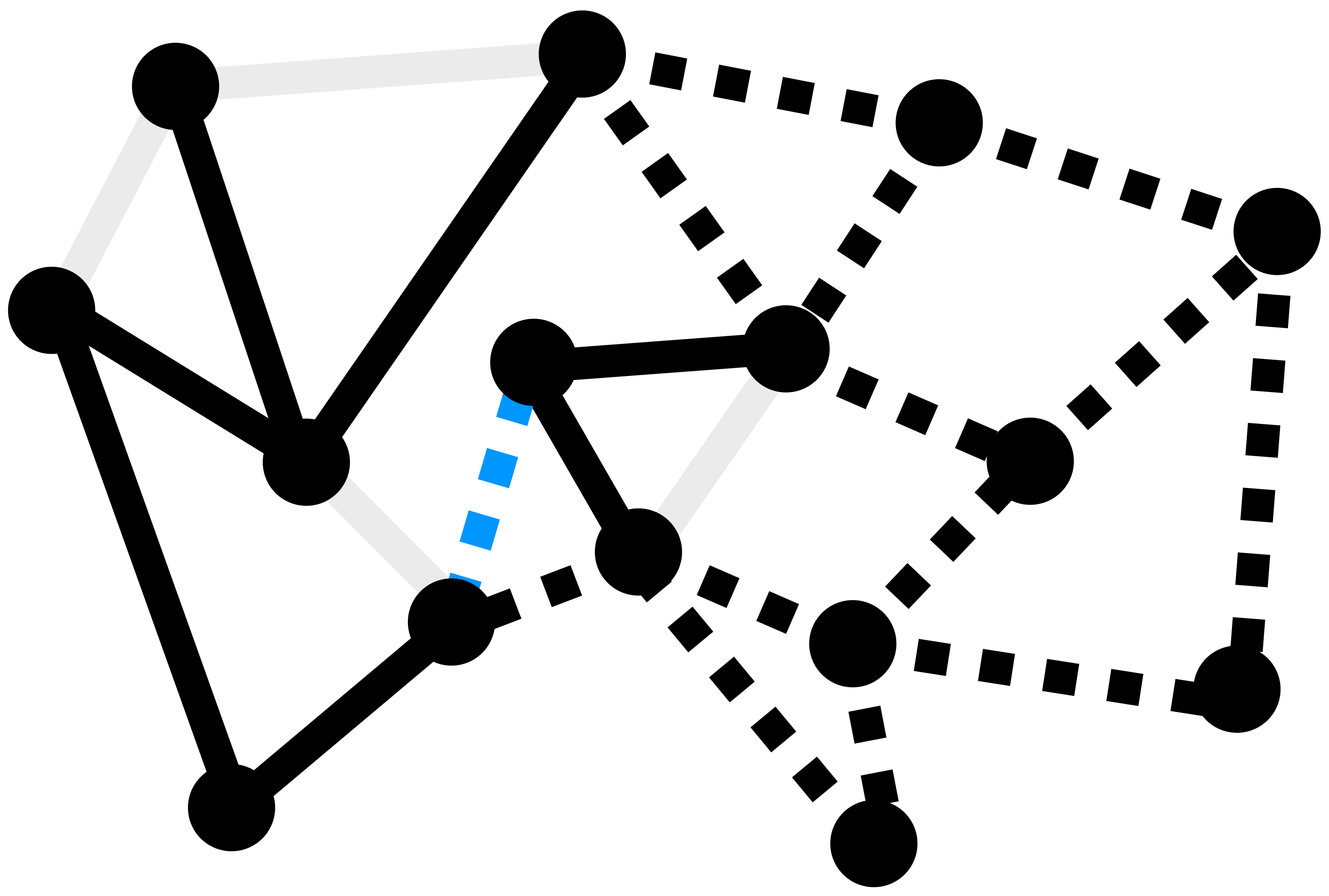
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



$t = 3$

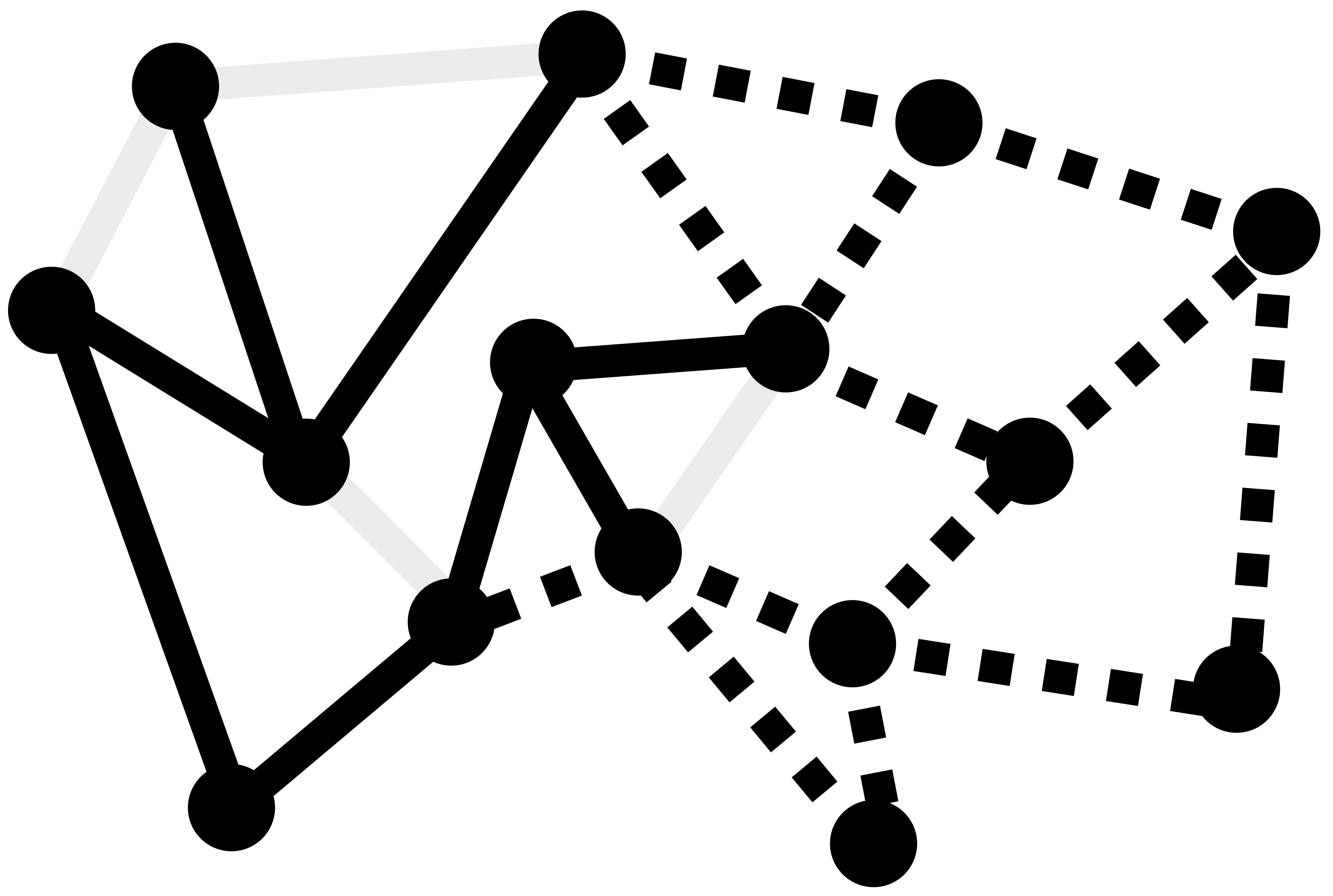
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



$t = 3$

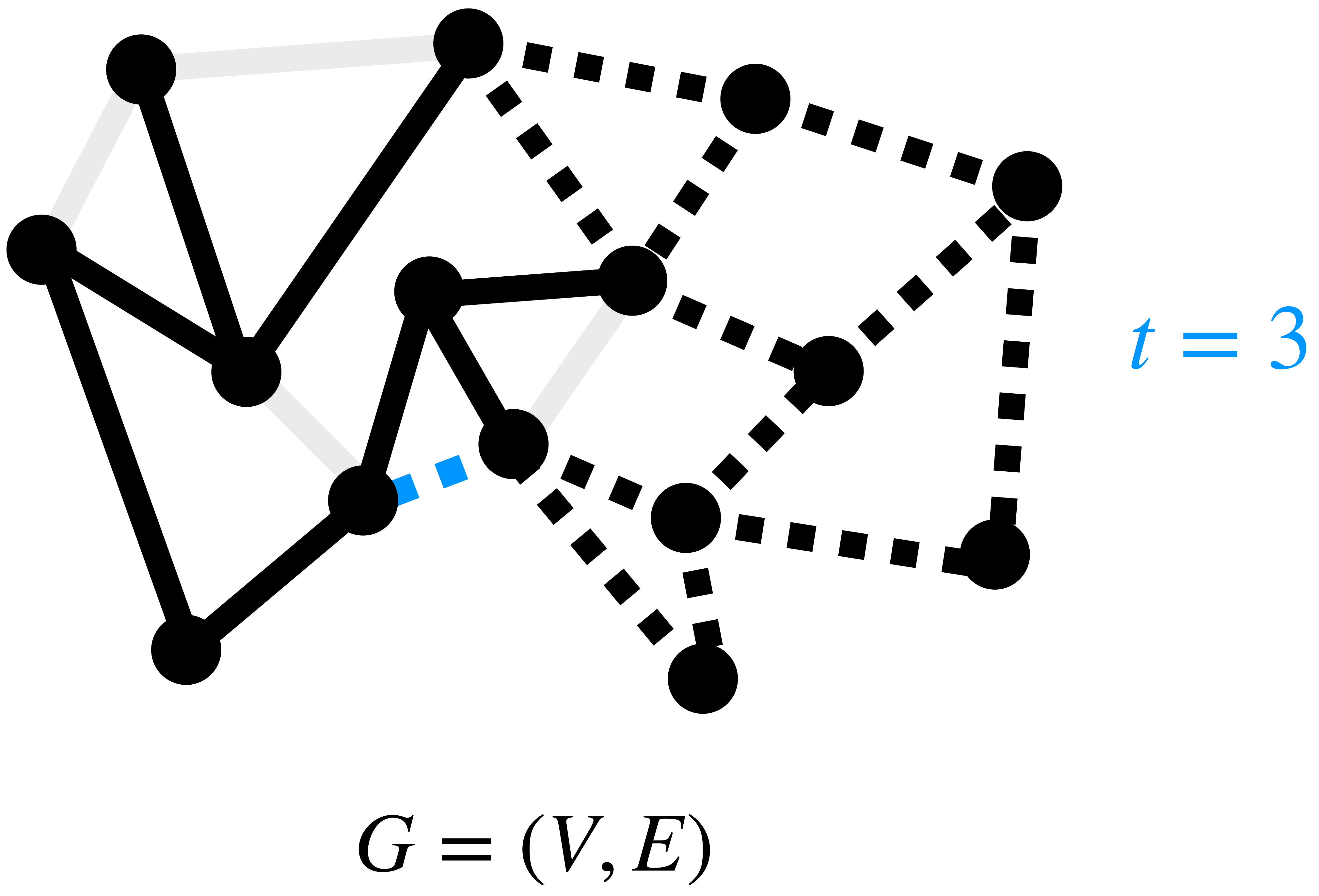
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$

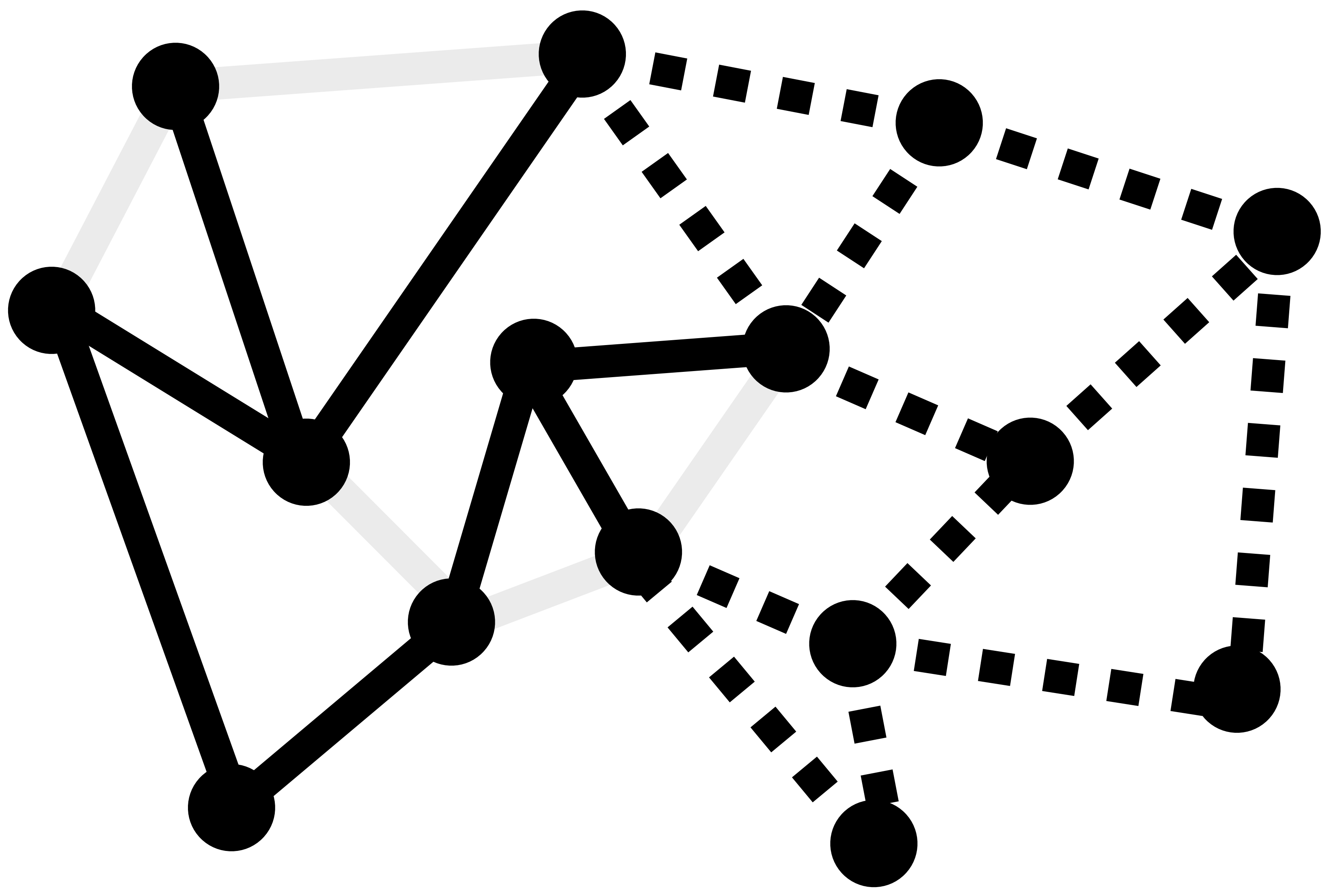


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



$t = 3$

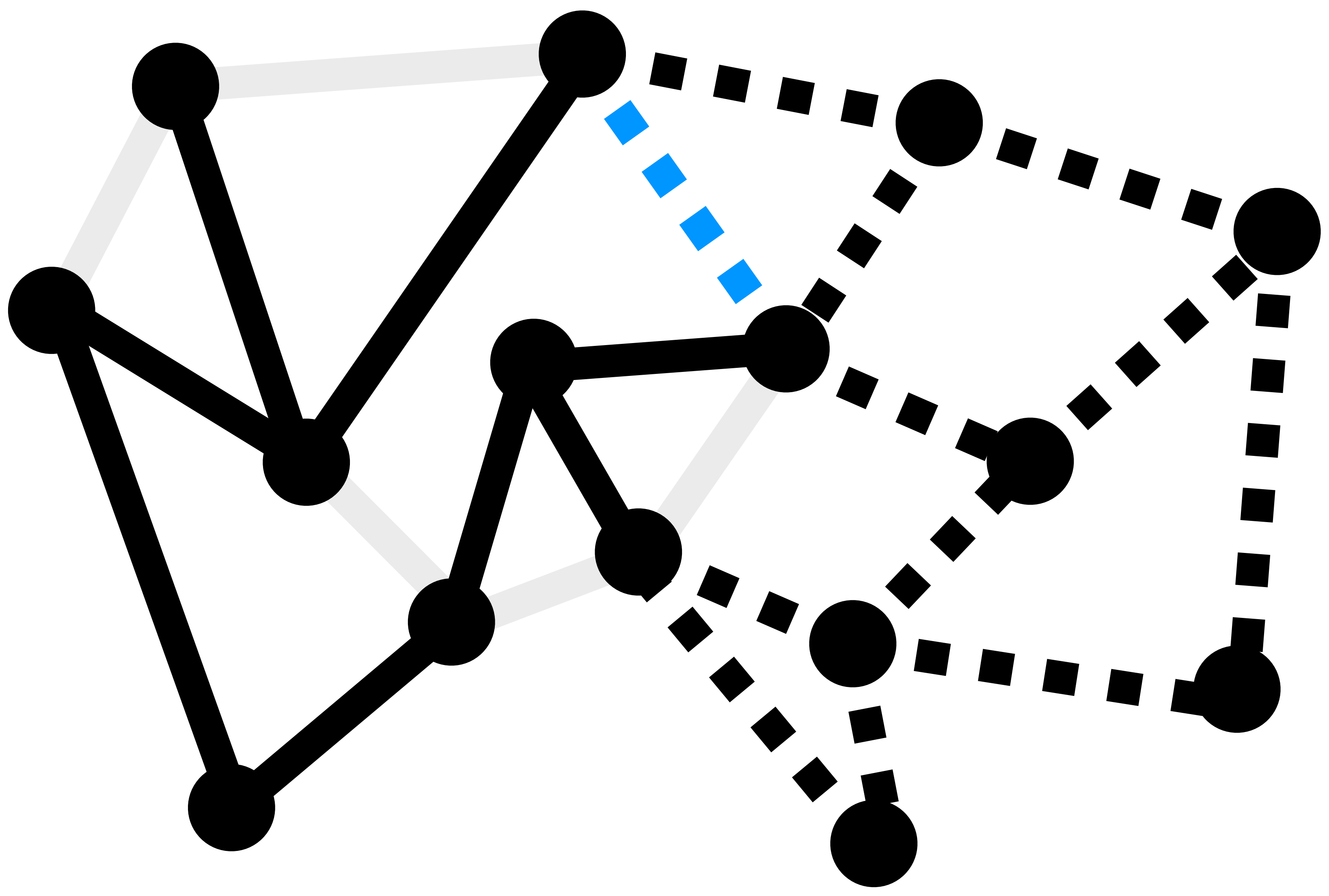
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



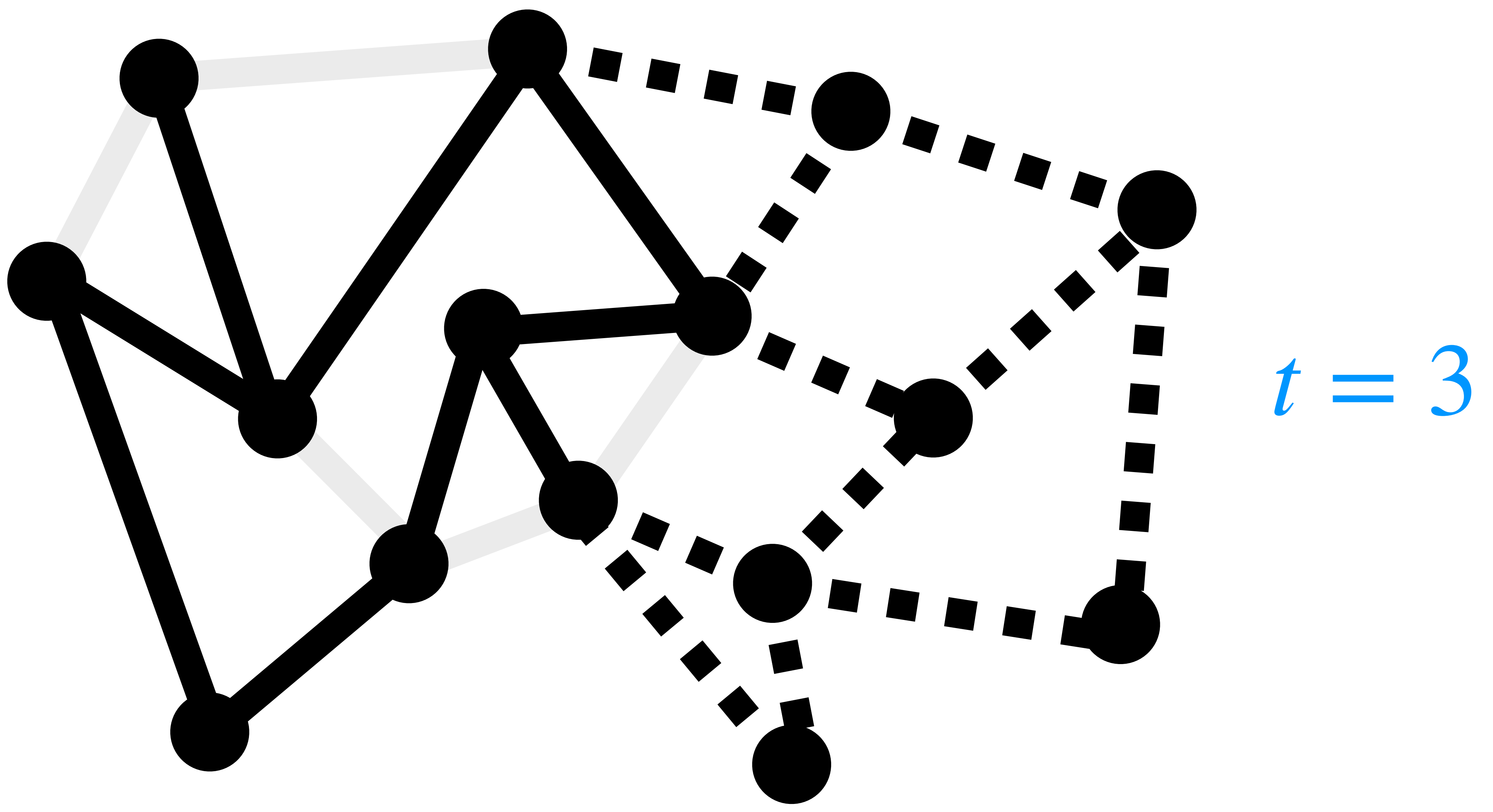
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



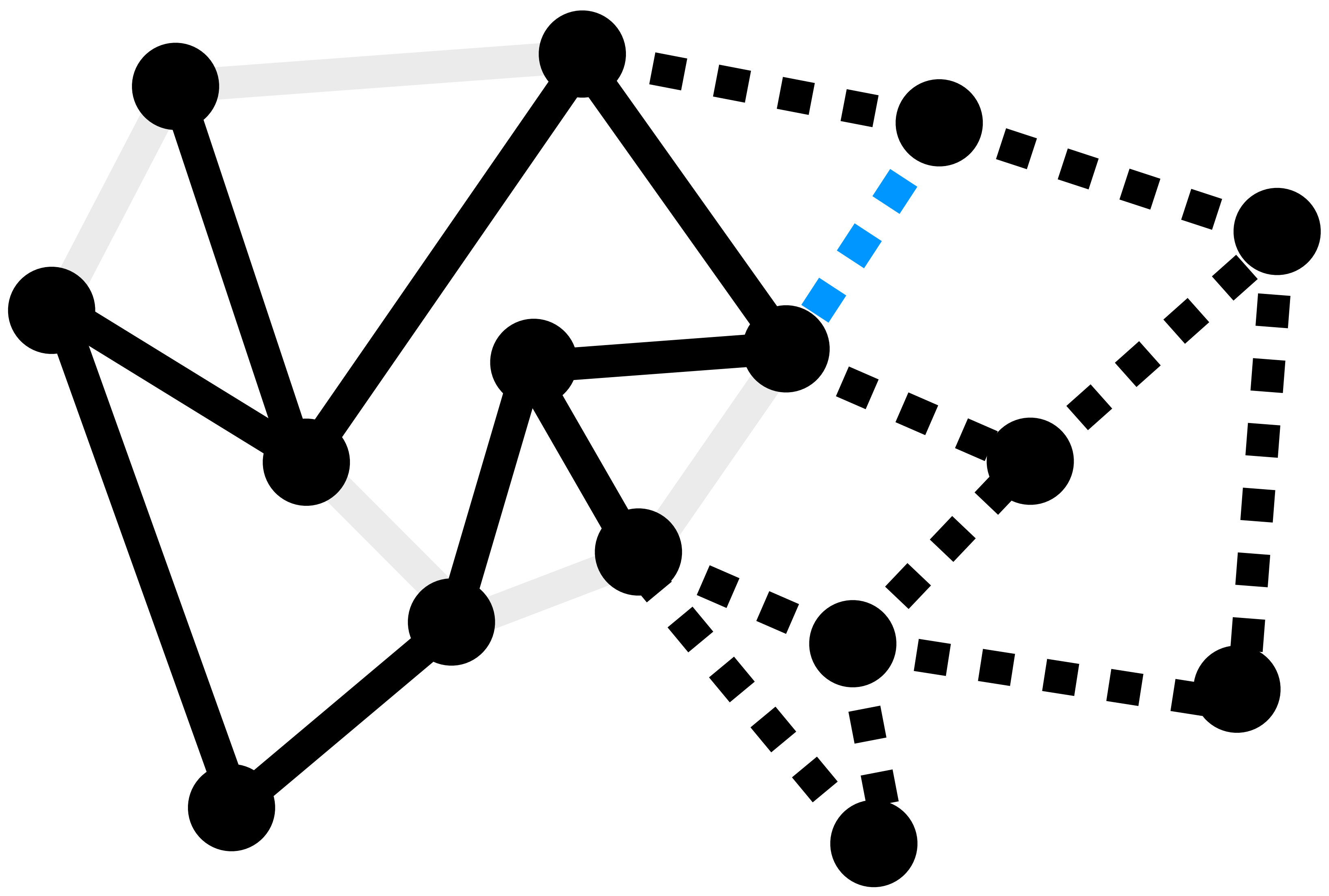
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \varnothing$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$

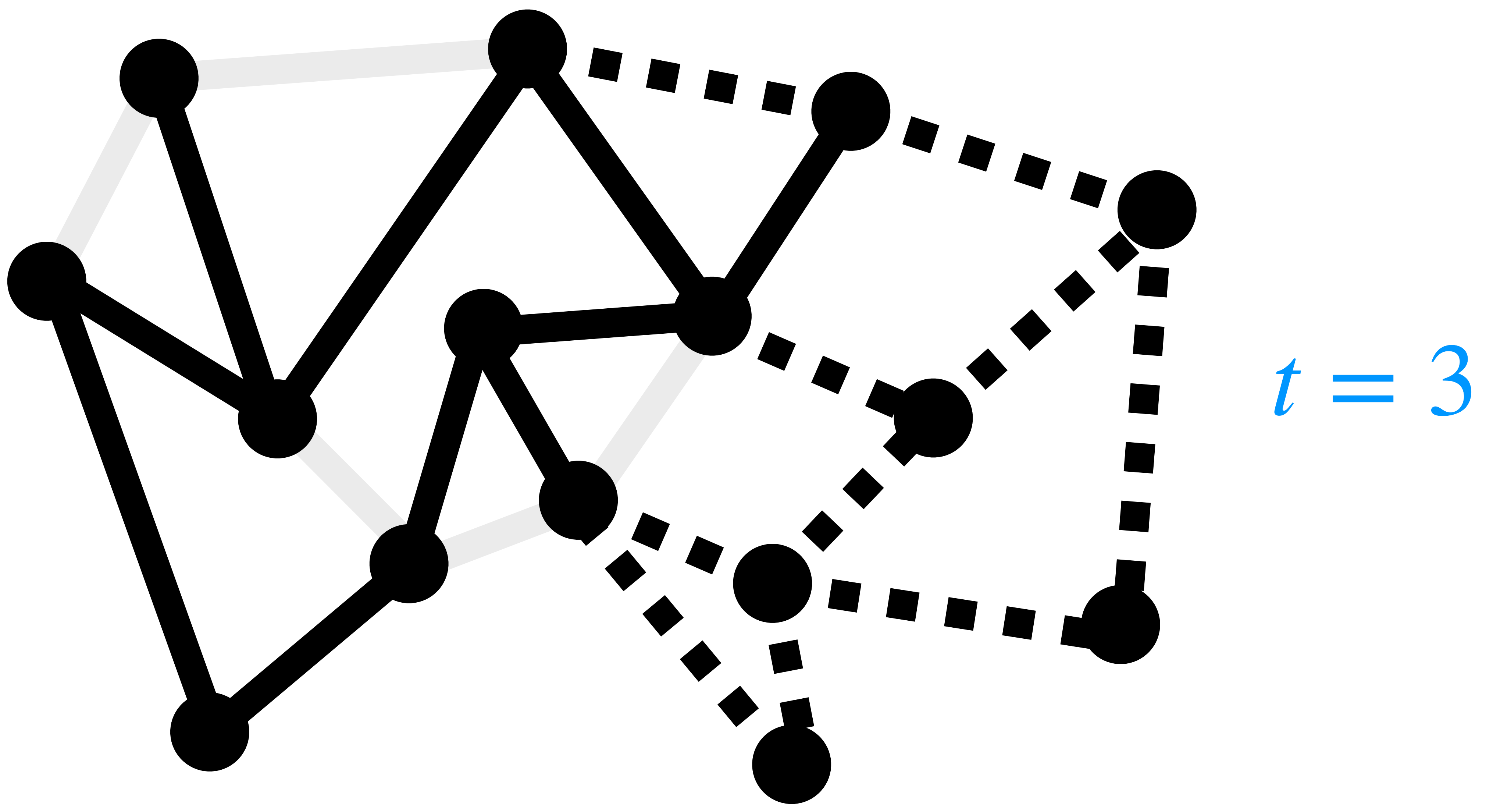


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \varnothing$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



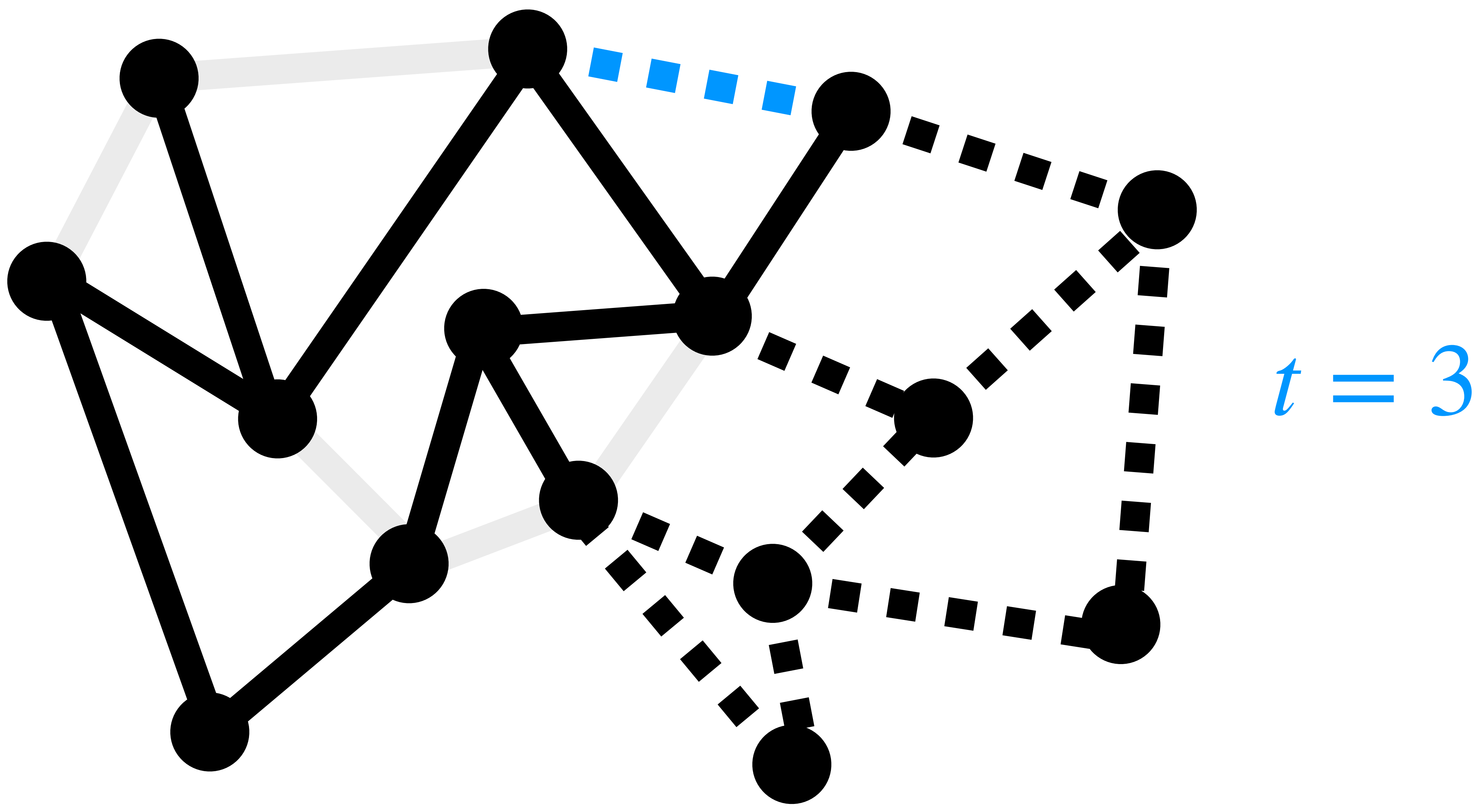
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



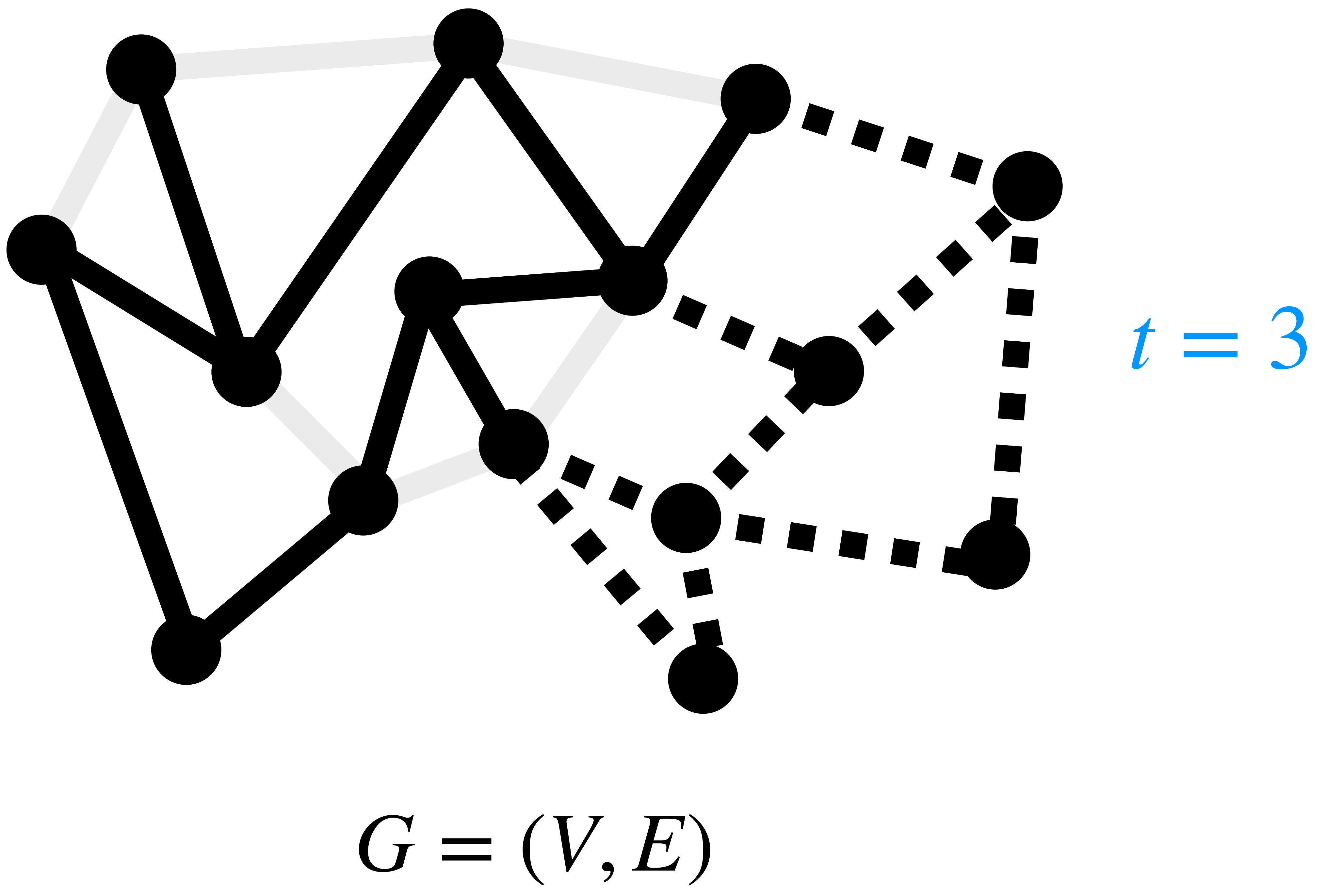
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \varnothing$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$

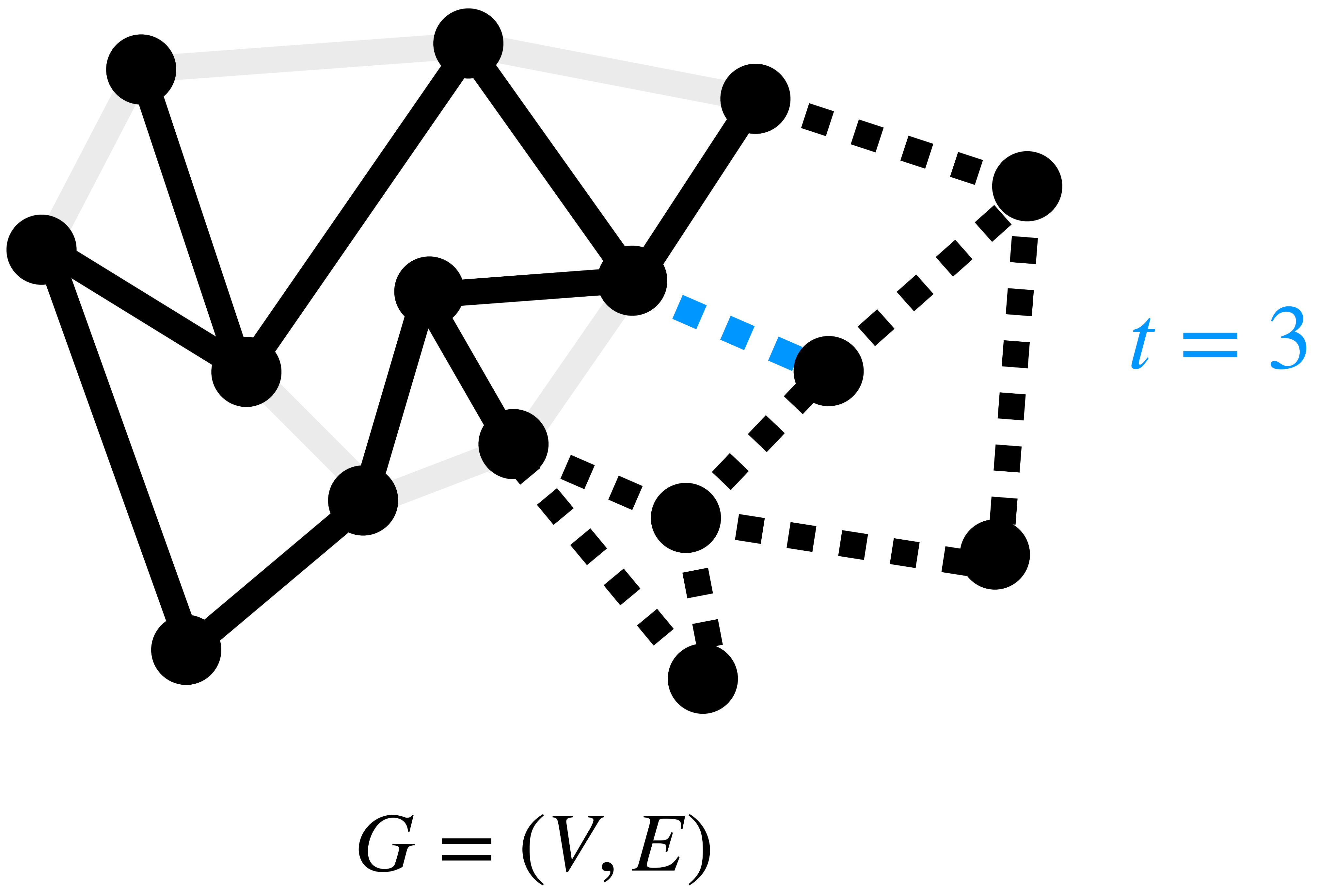


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$

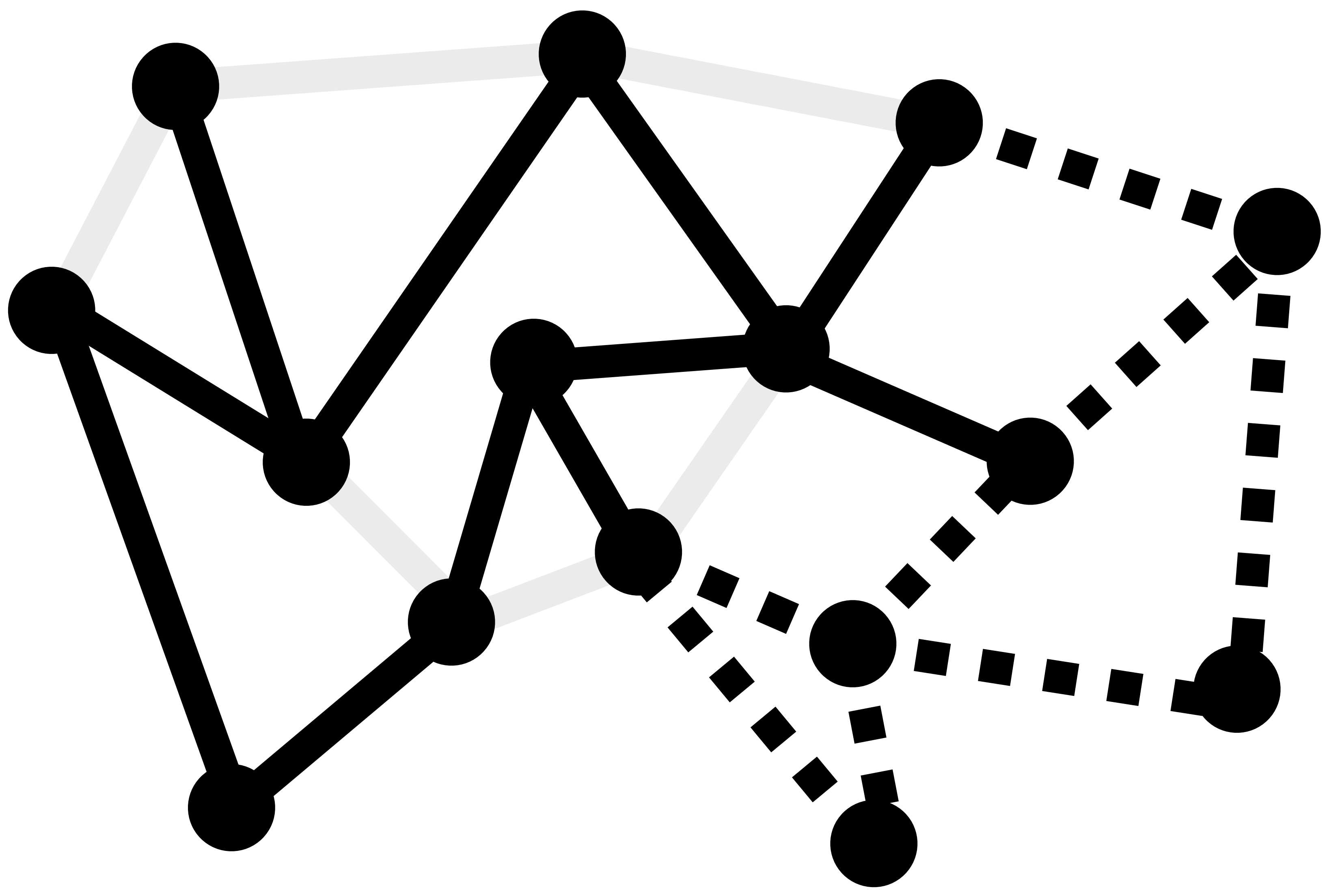


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \varnothing$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



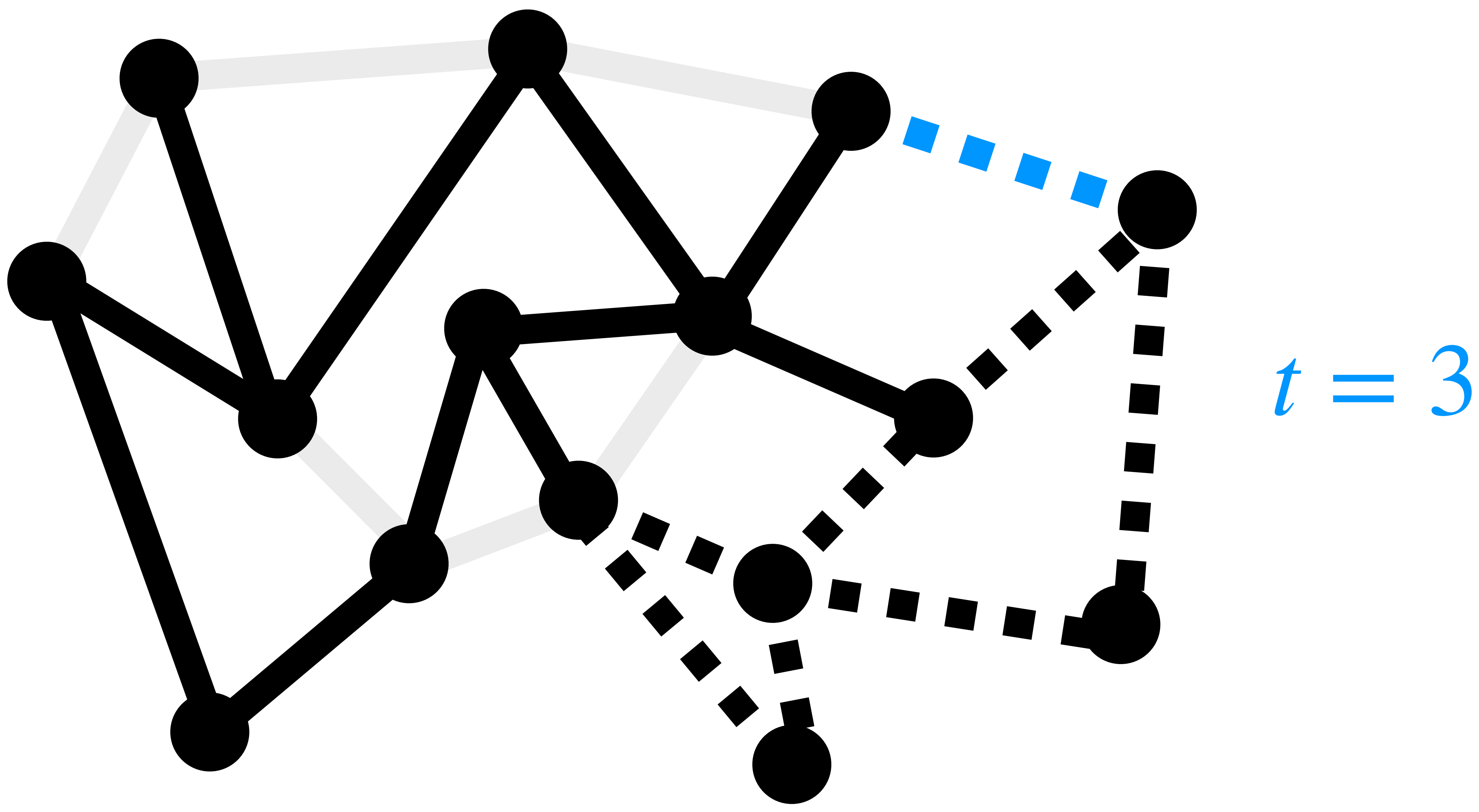
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



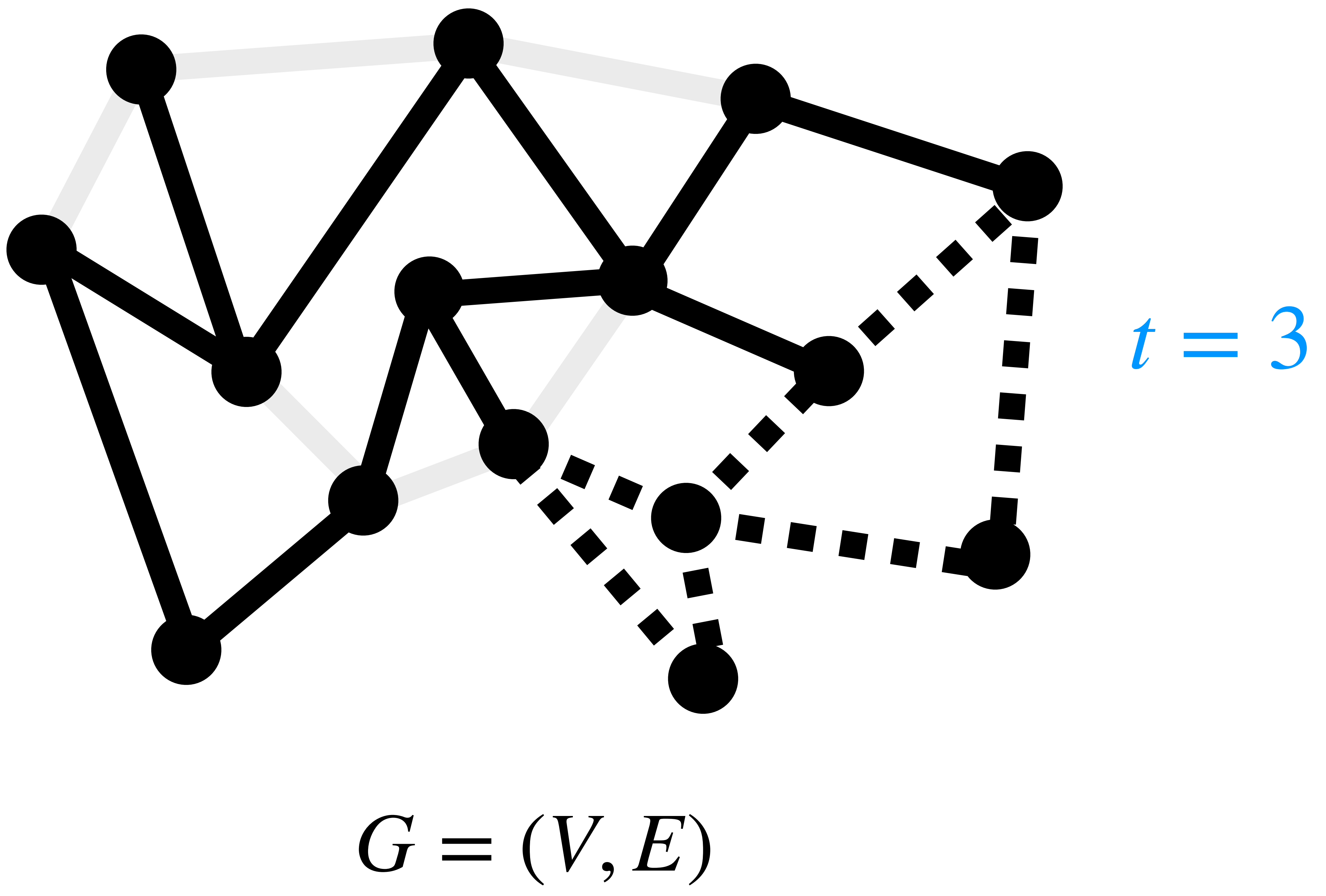
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
    - If $d_H(u, v) > t$ then

      $H \leftarrow H + \{u, v\}$

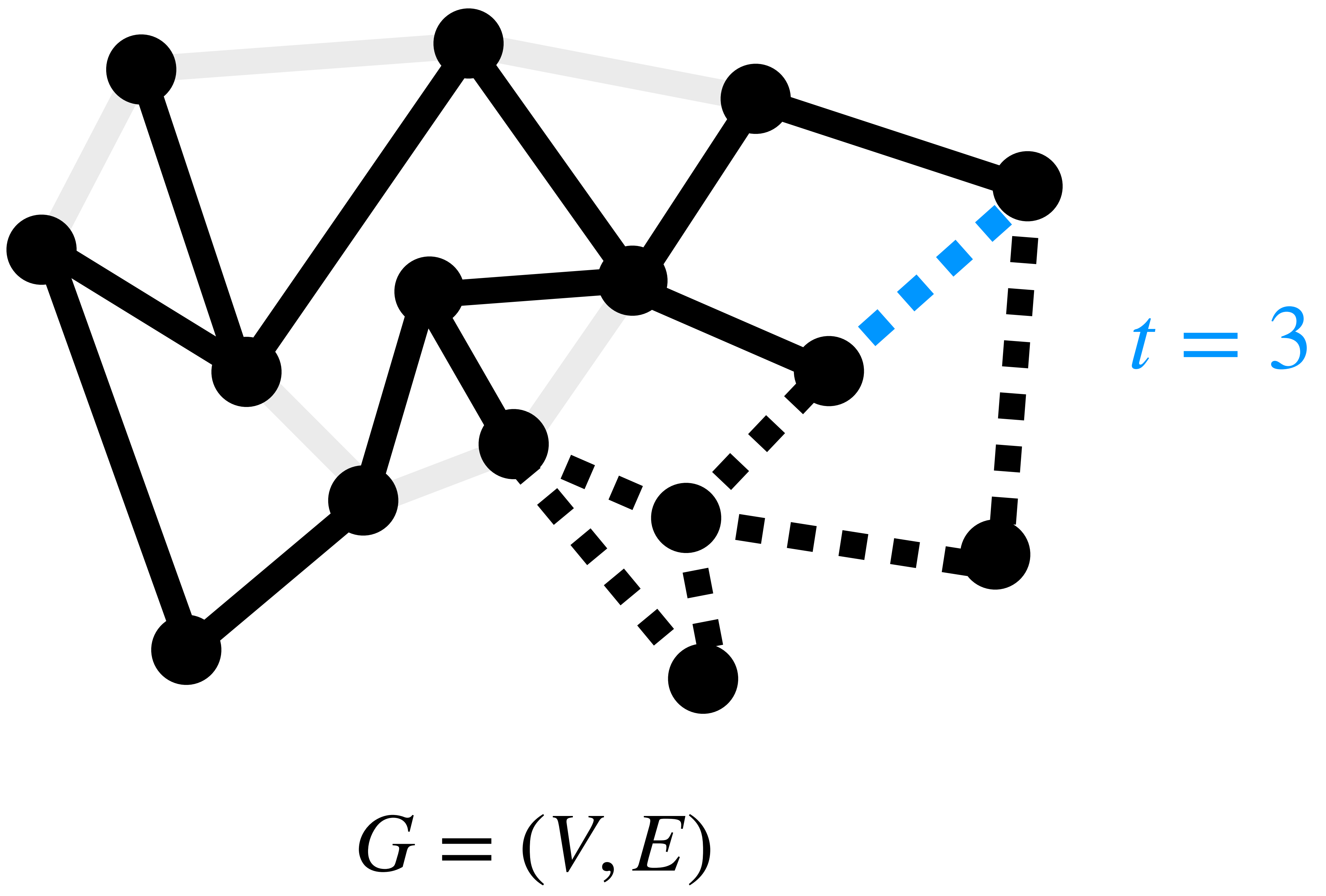


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \varnothing$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$

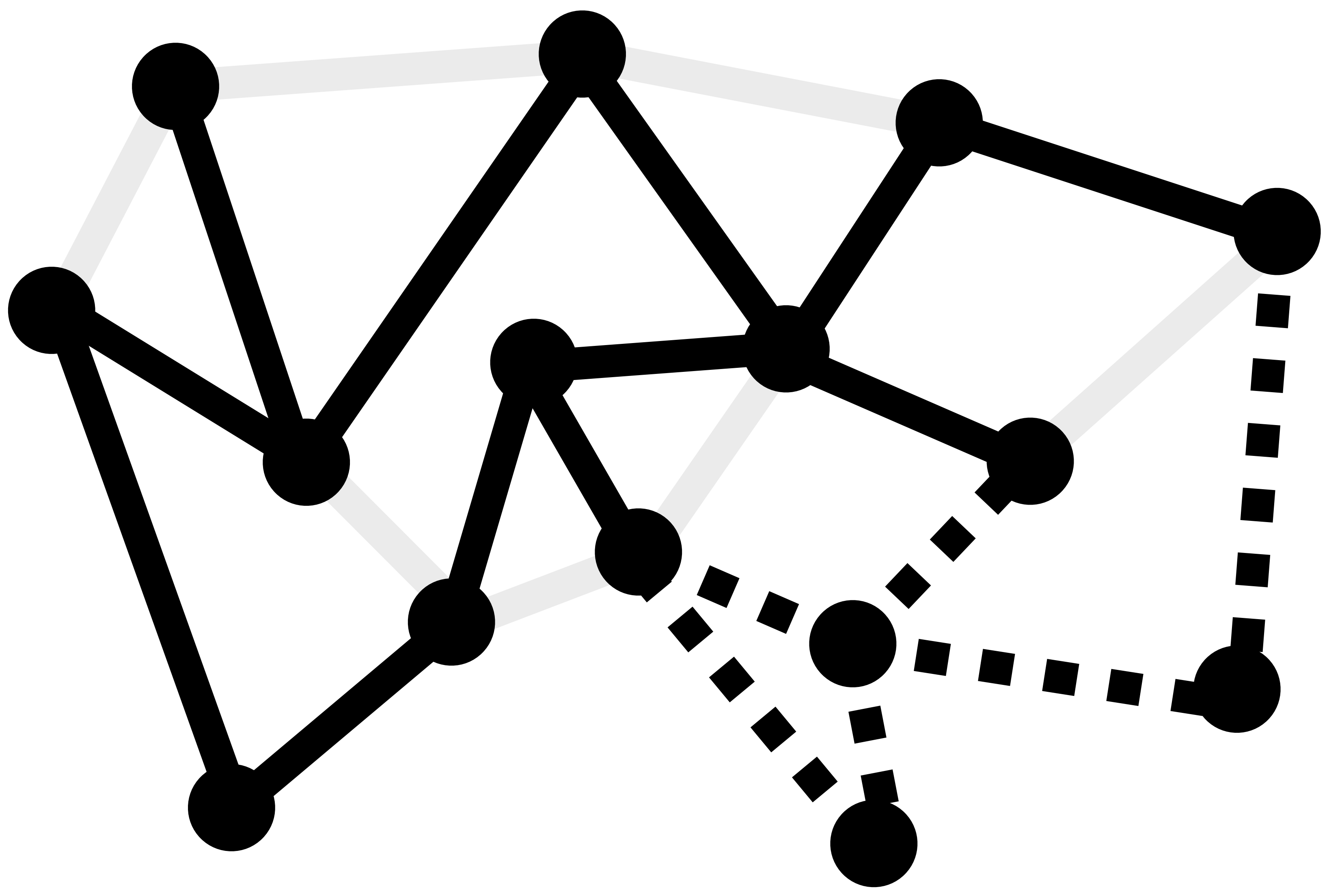


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



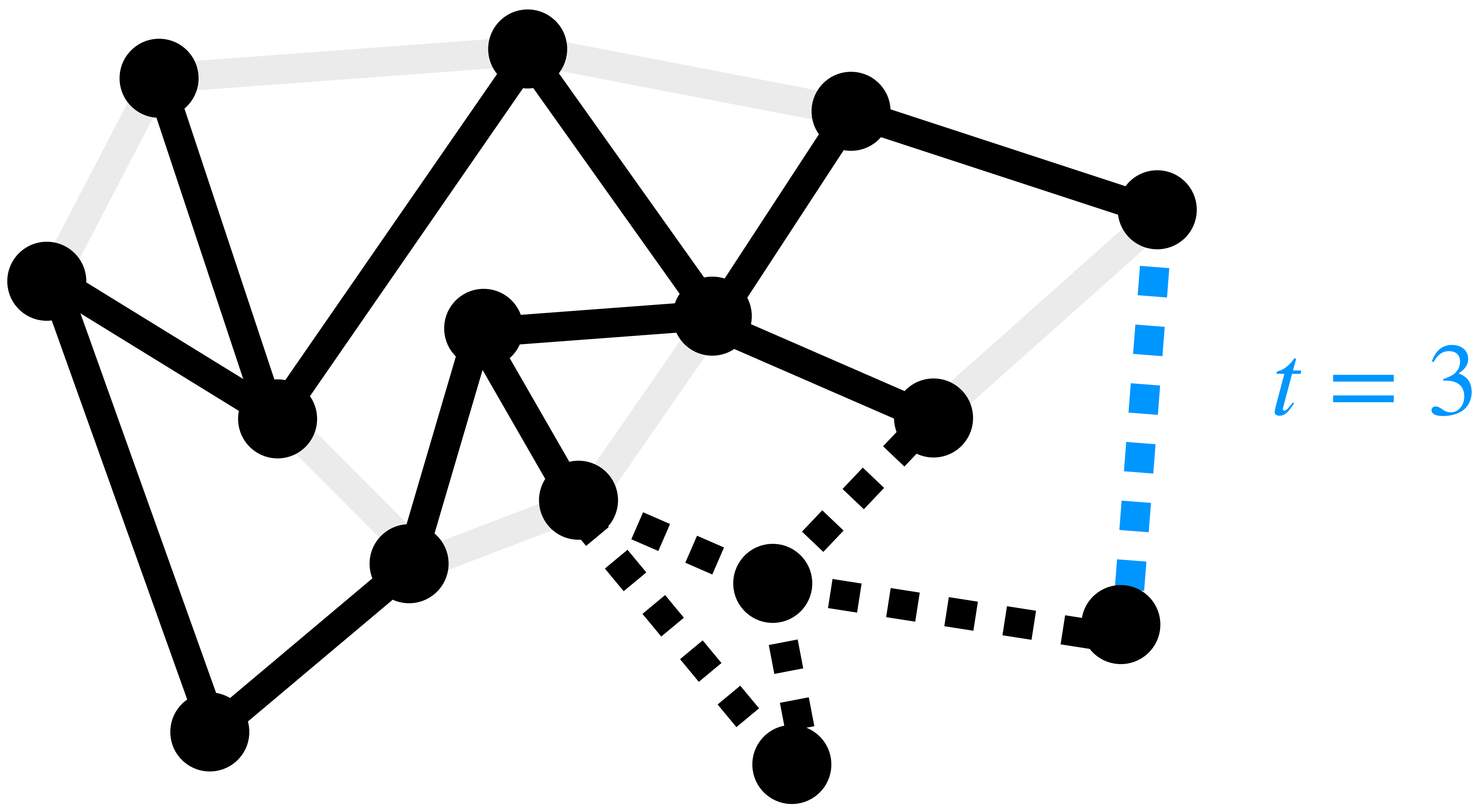
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \varnothing$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



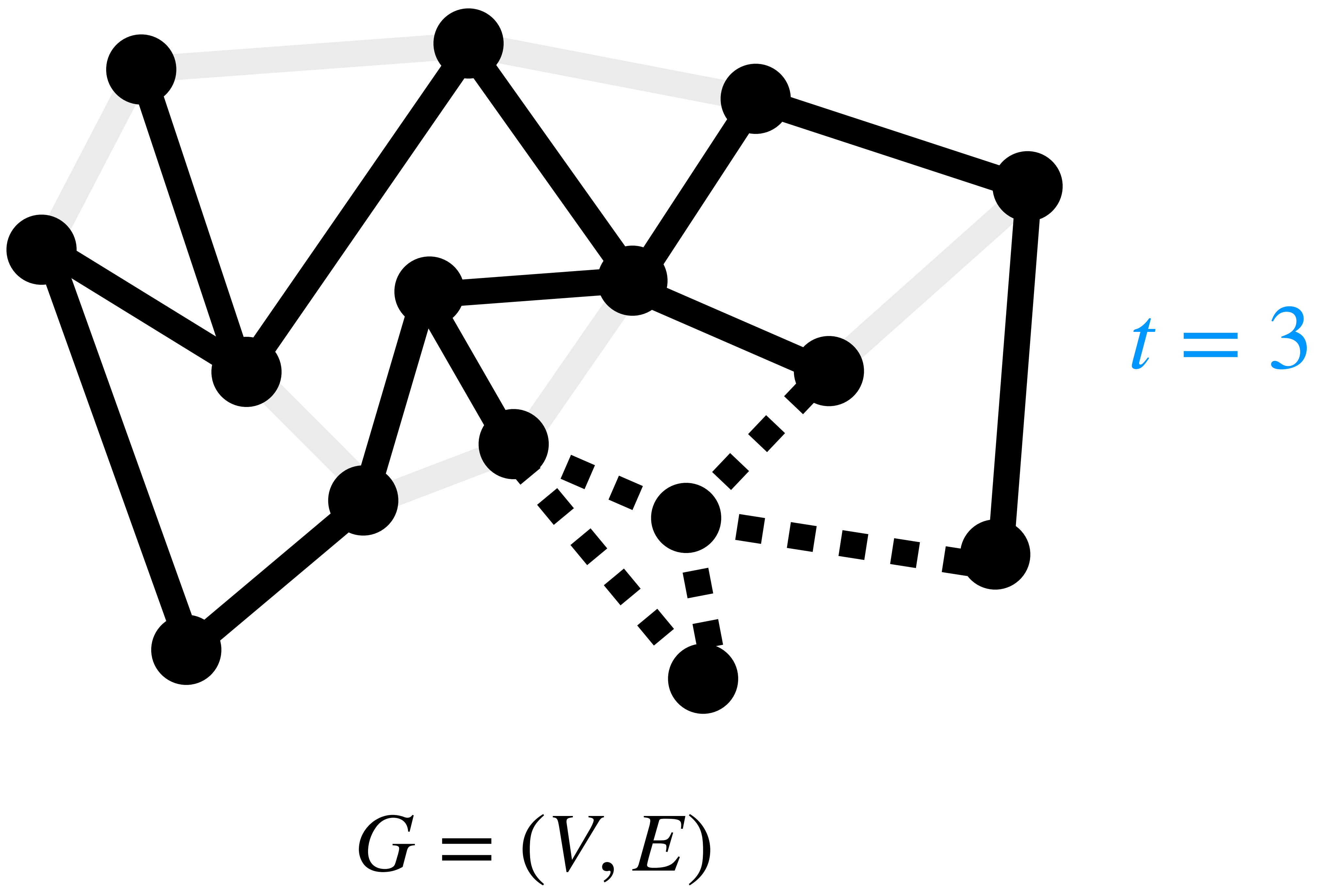
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$

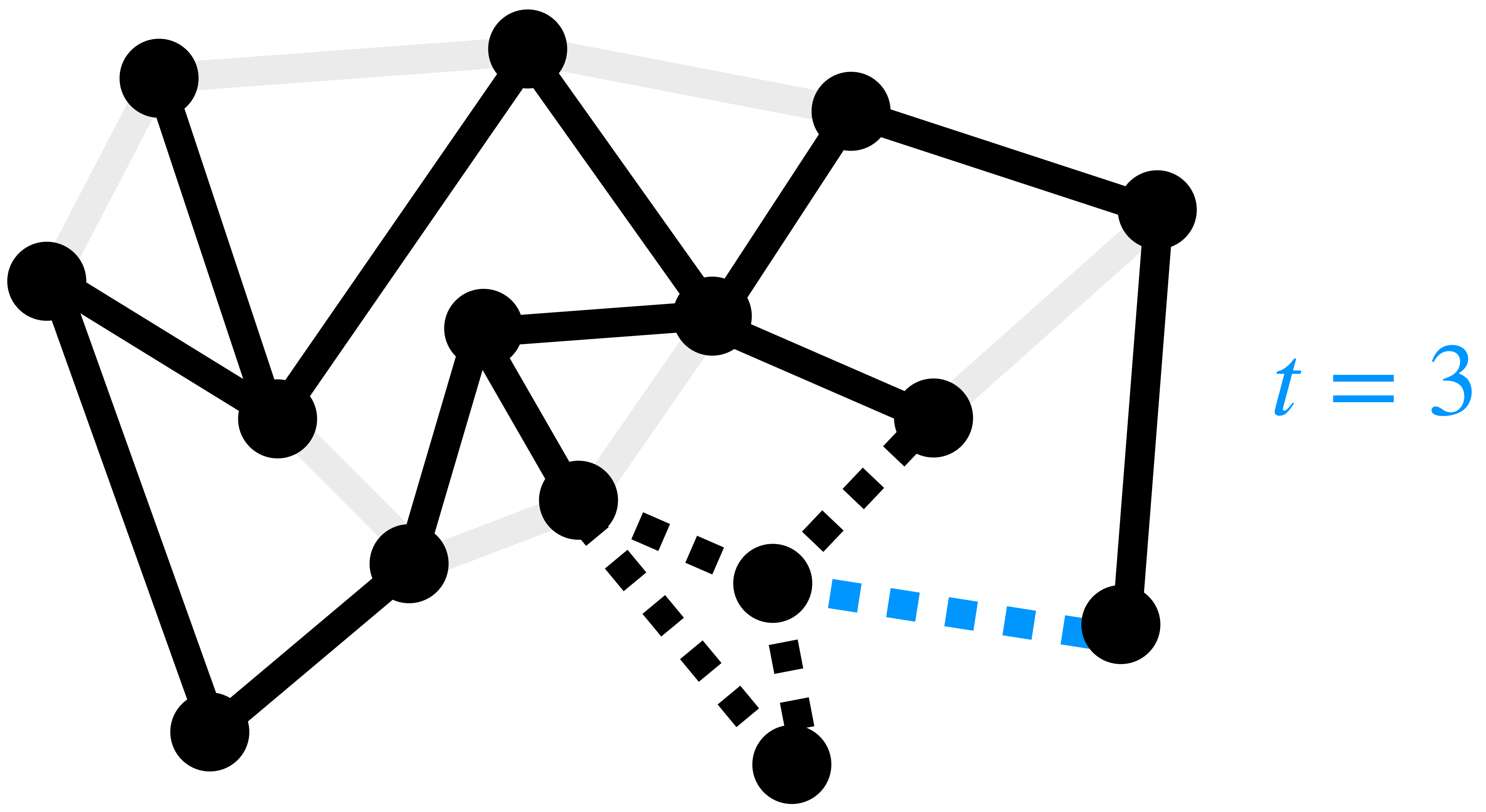


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



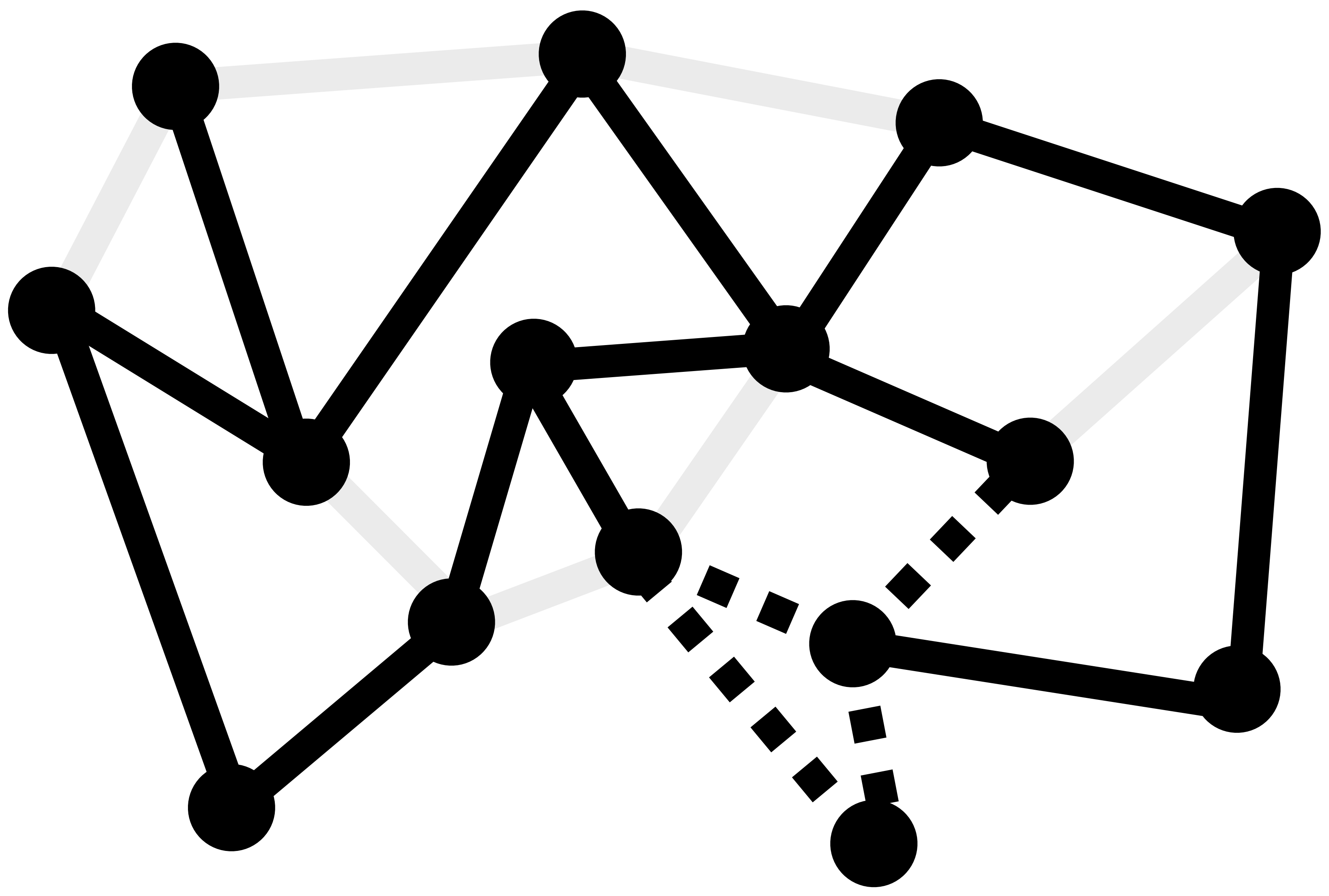
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
    - If $d_H(u, v) > t$ then

      $H \leftarrow H + \{u, v\}$



$t = 3$

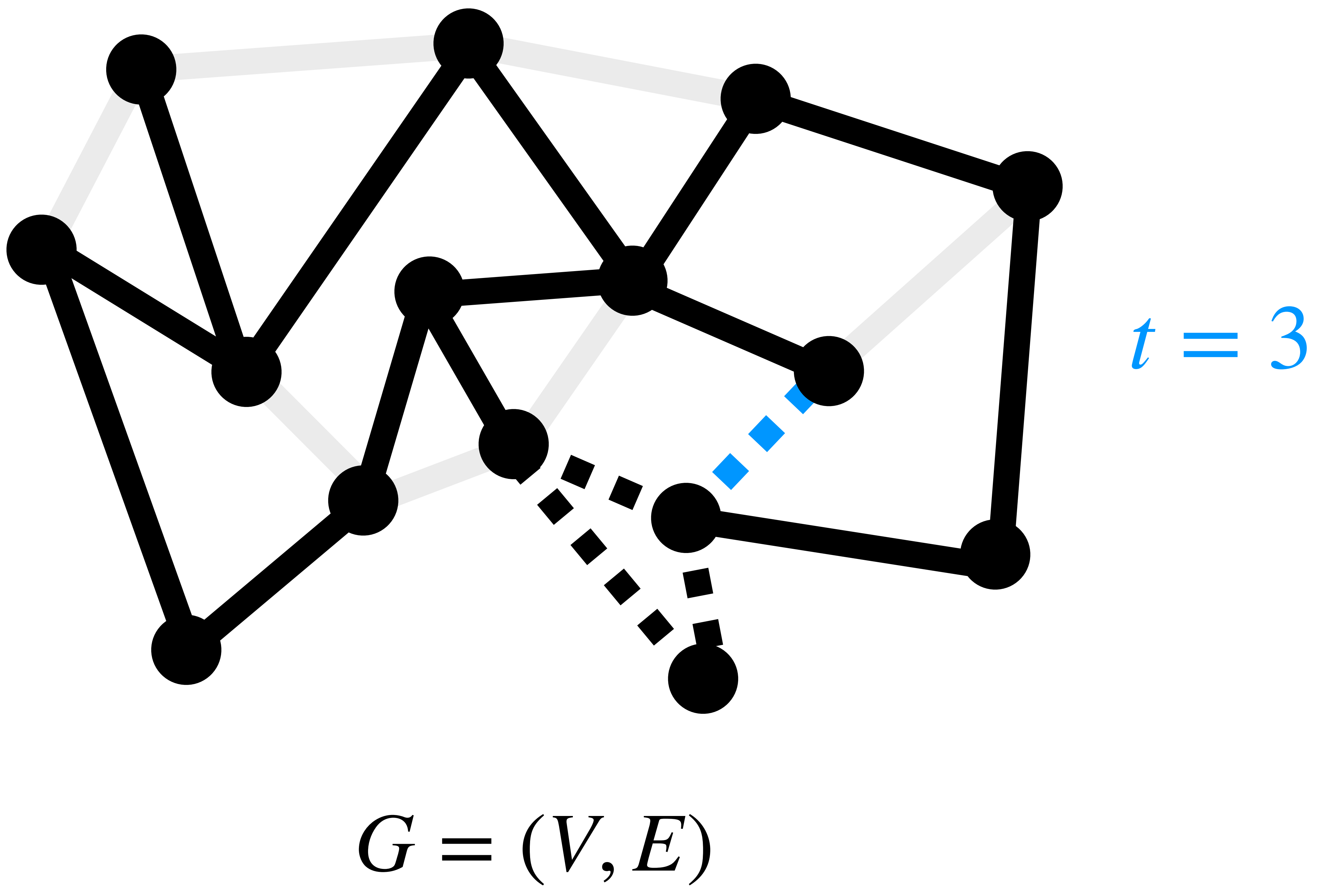
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$

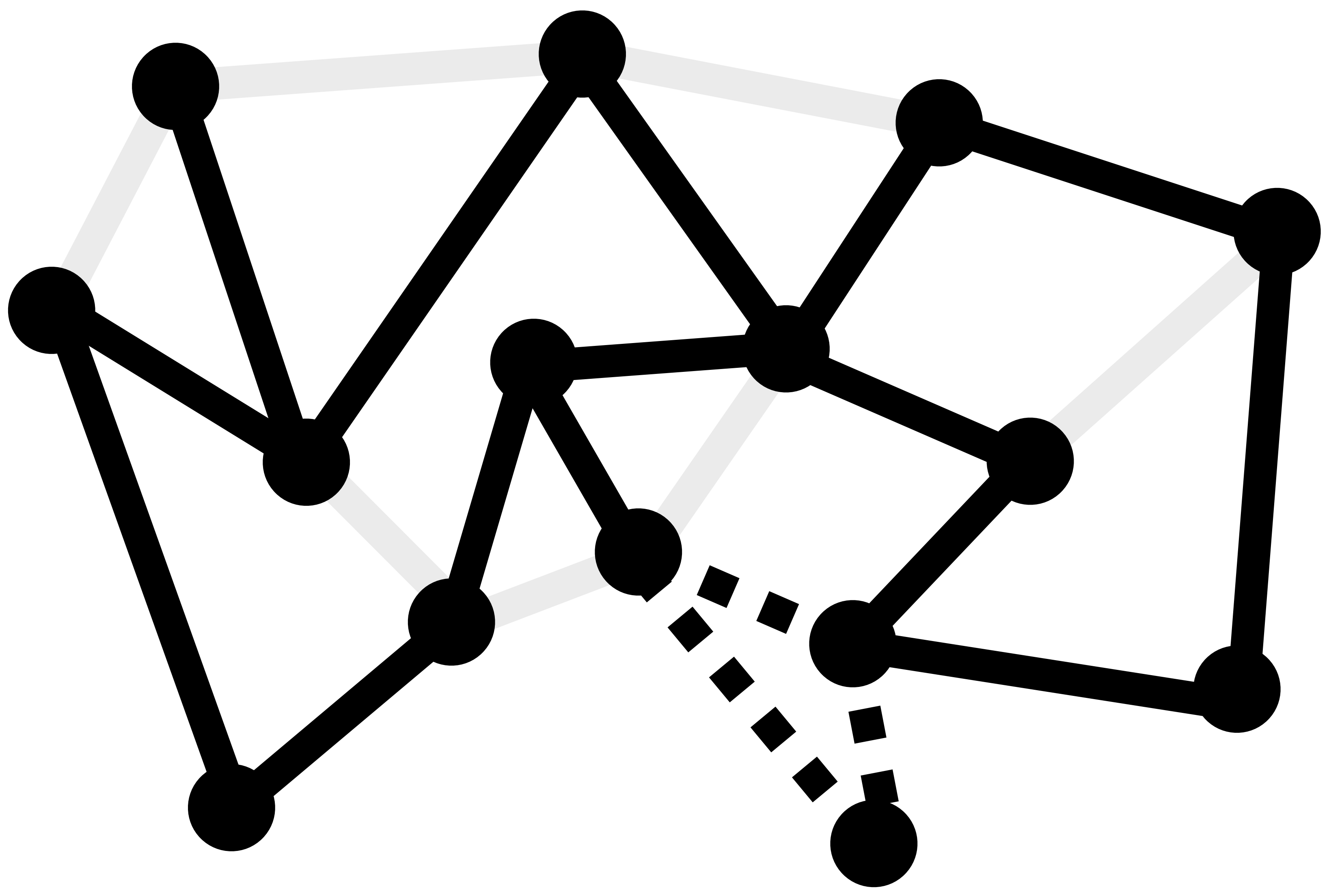


Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



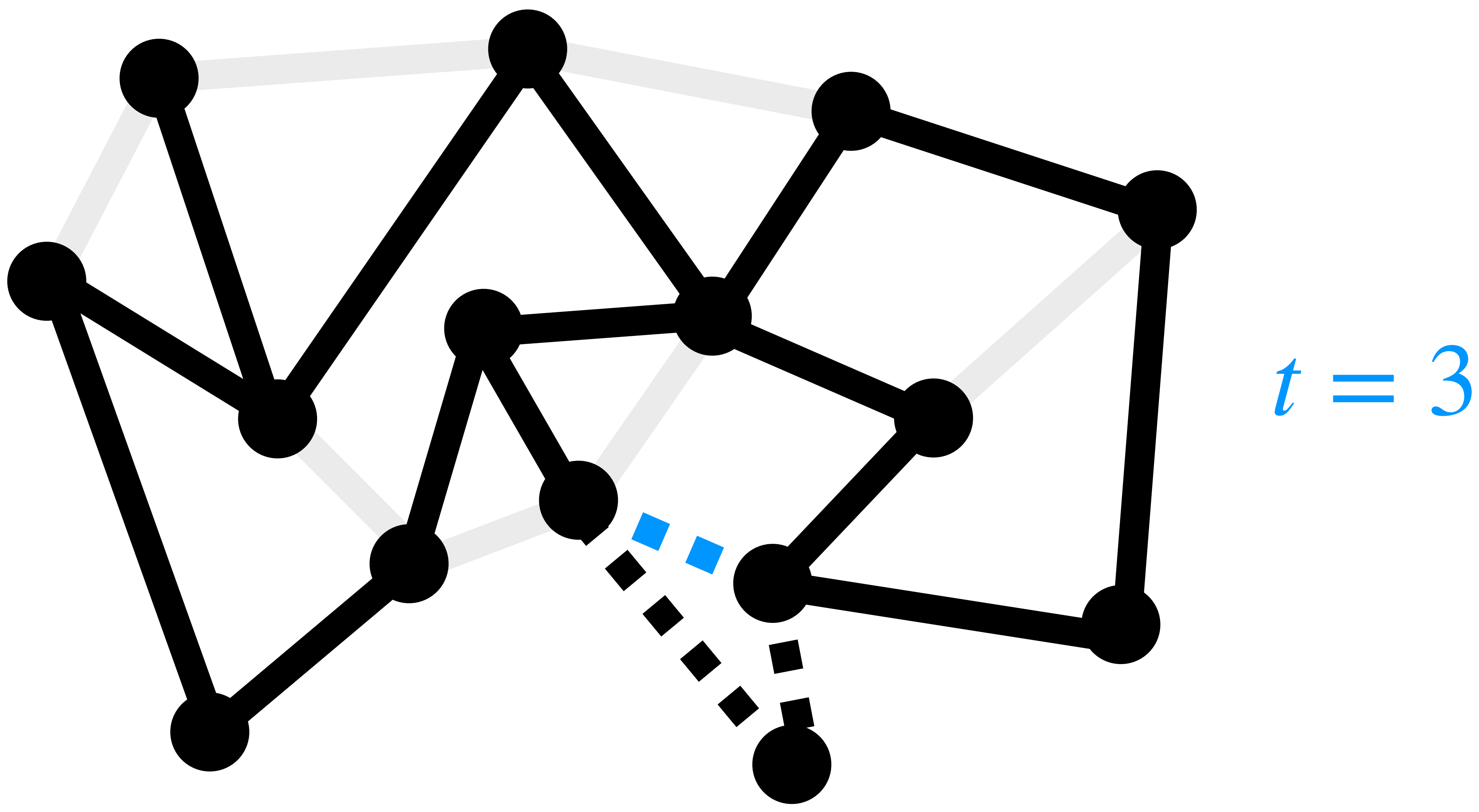
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



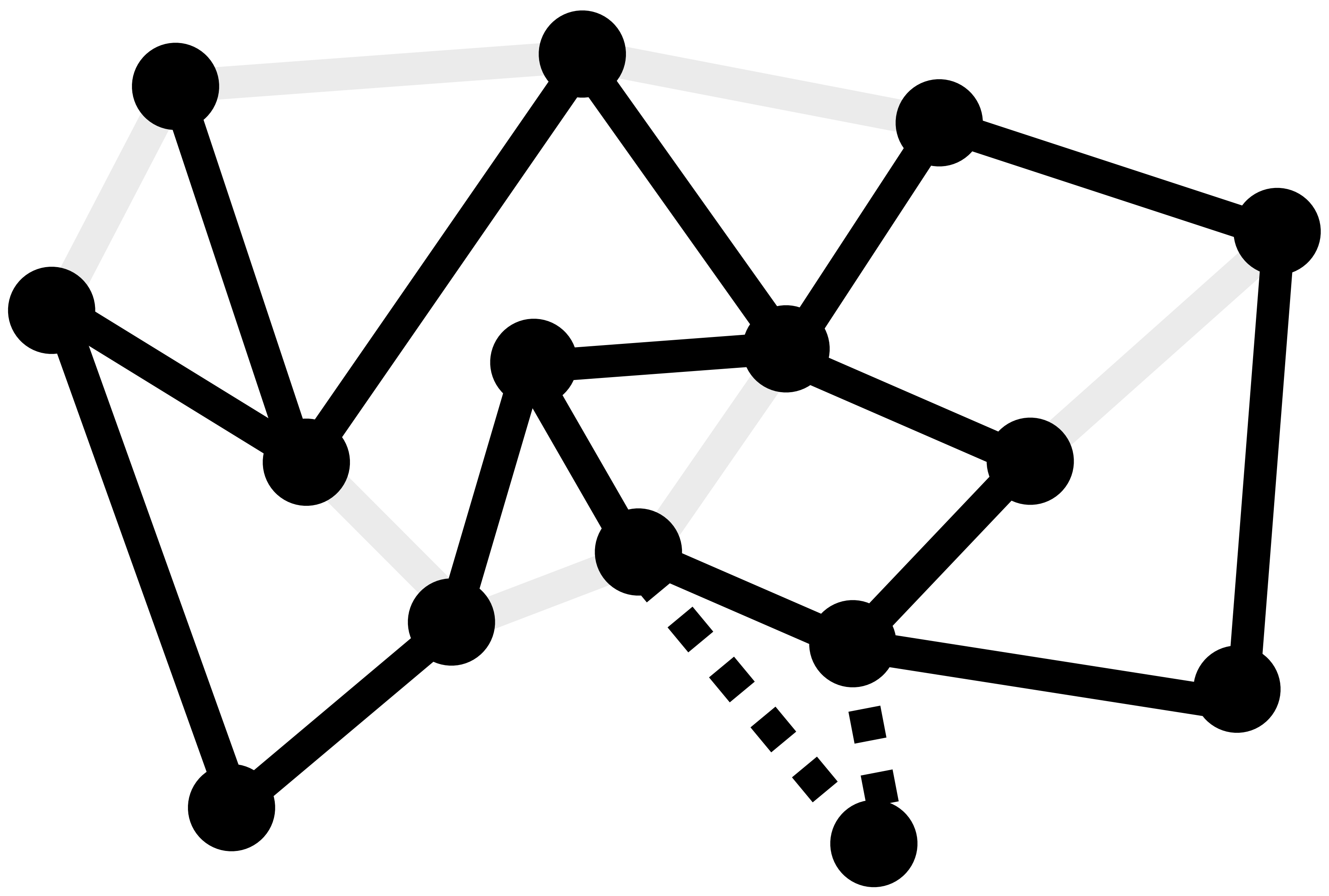
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



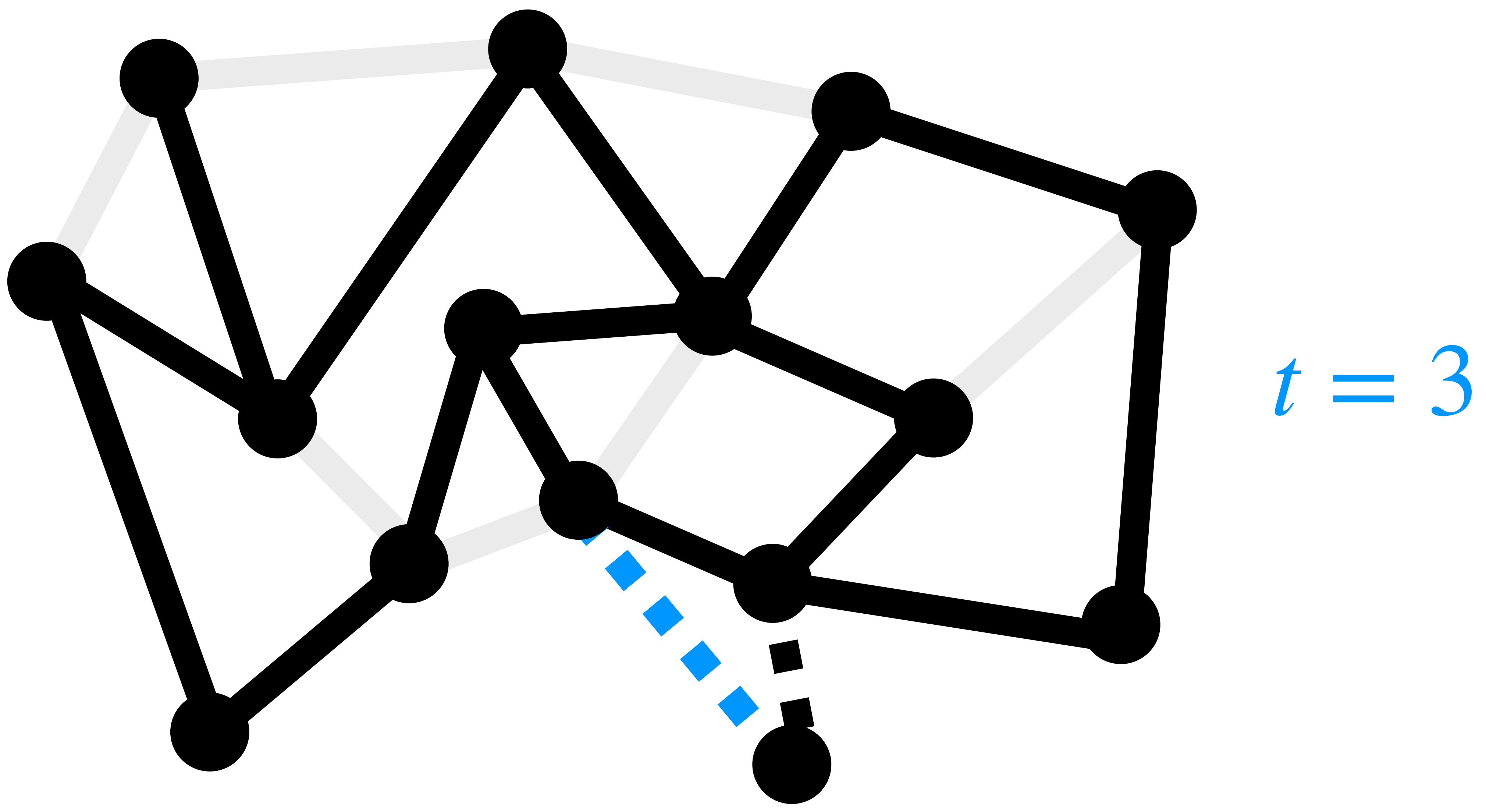
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



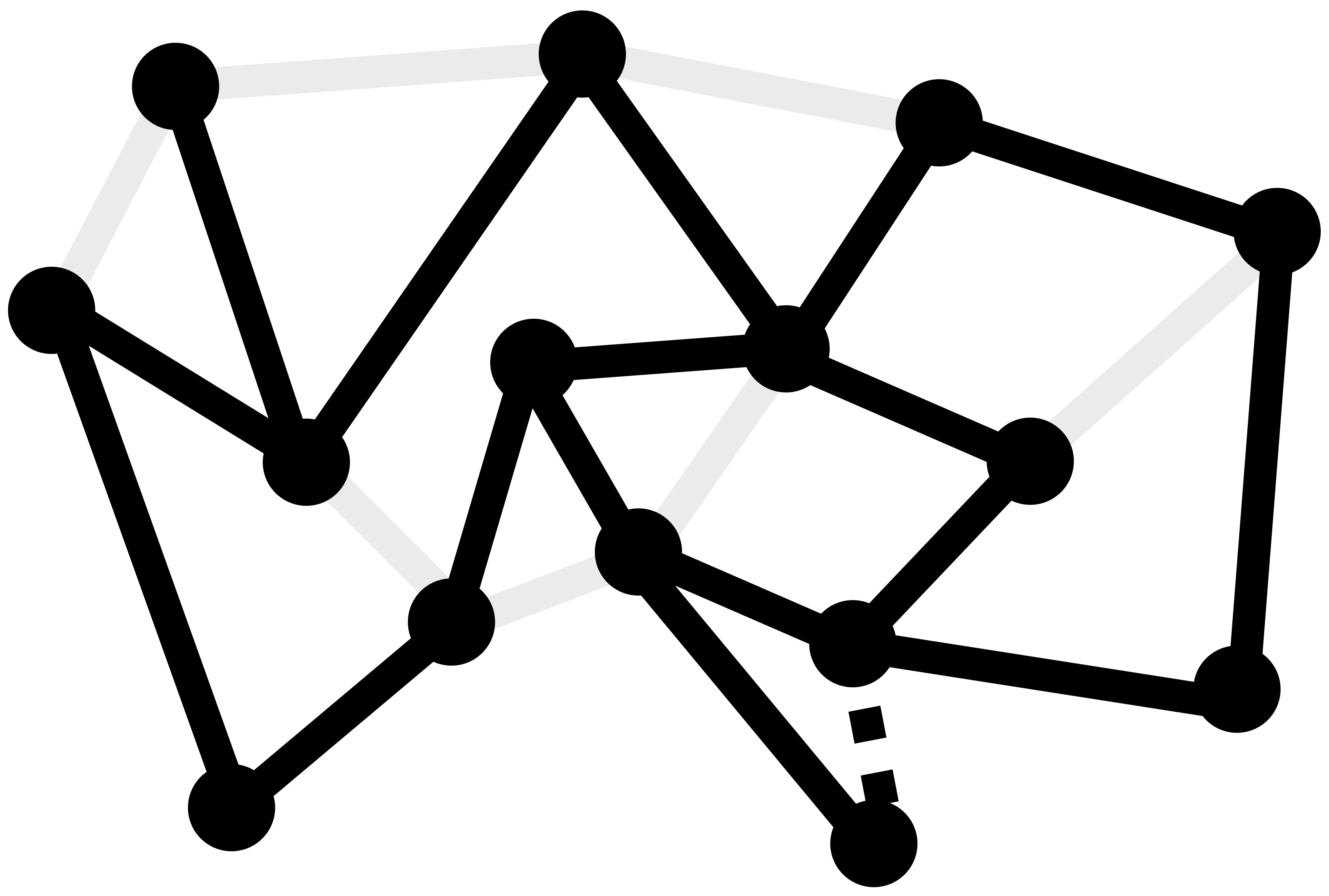
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



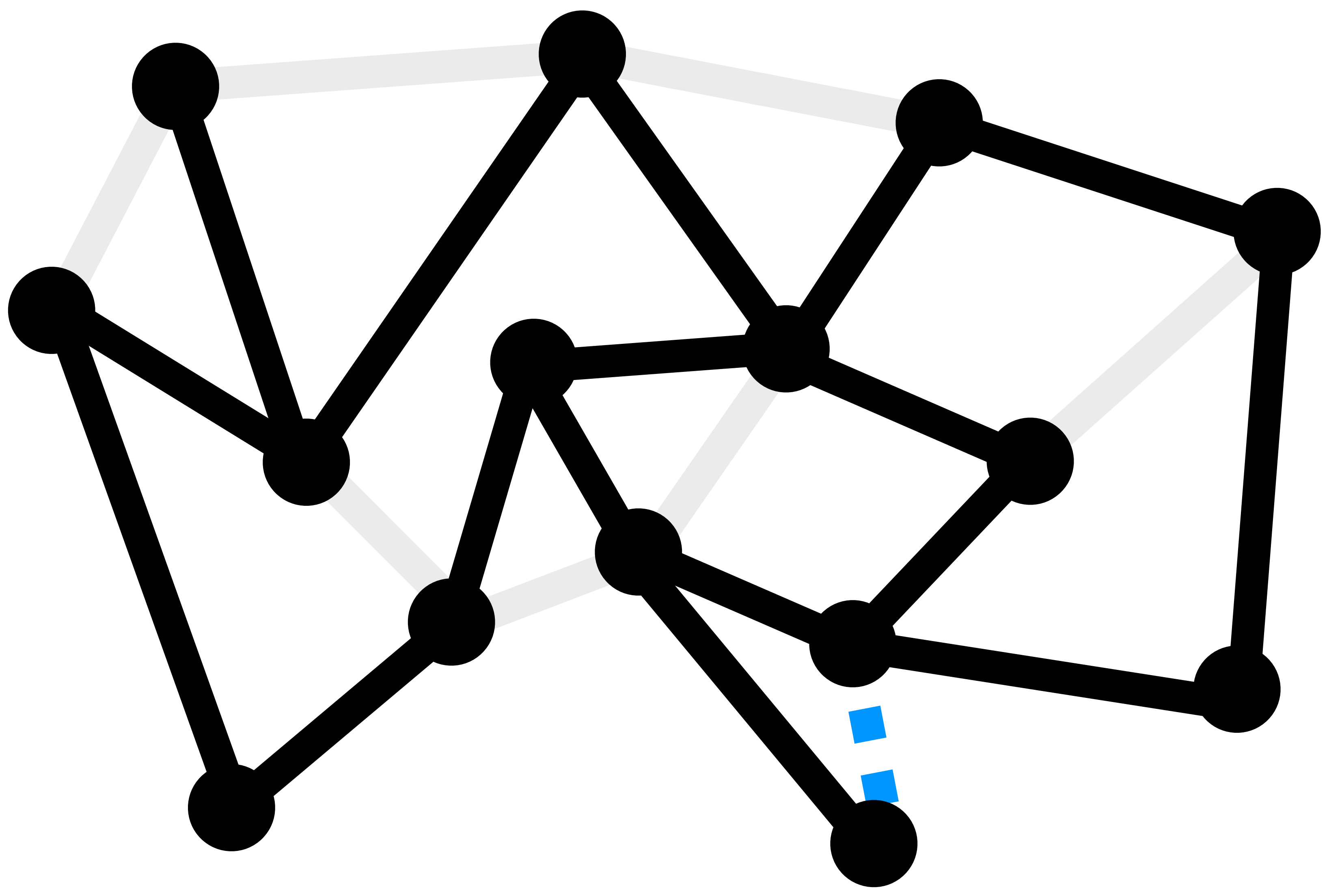
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \varnothing$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



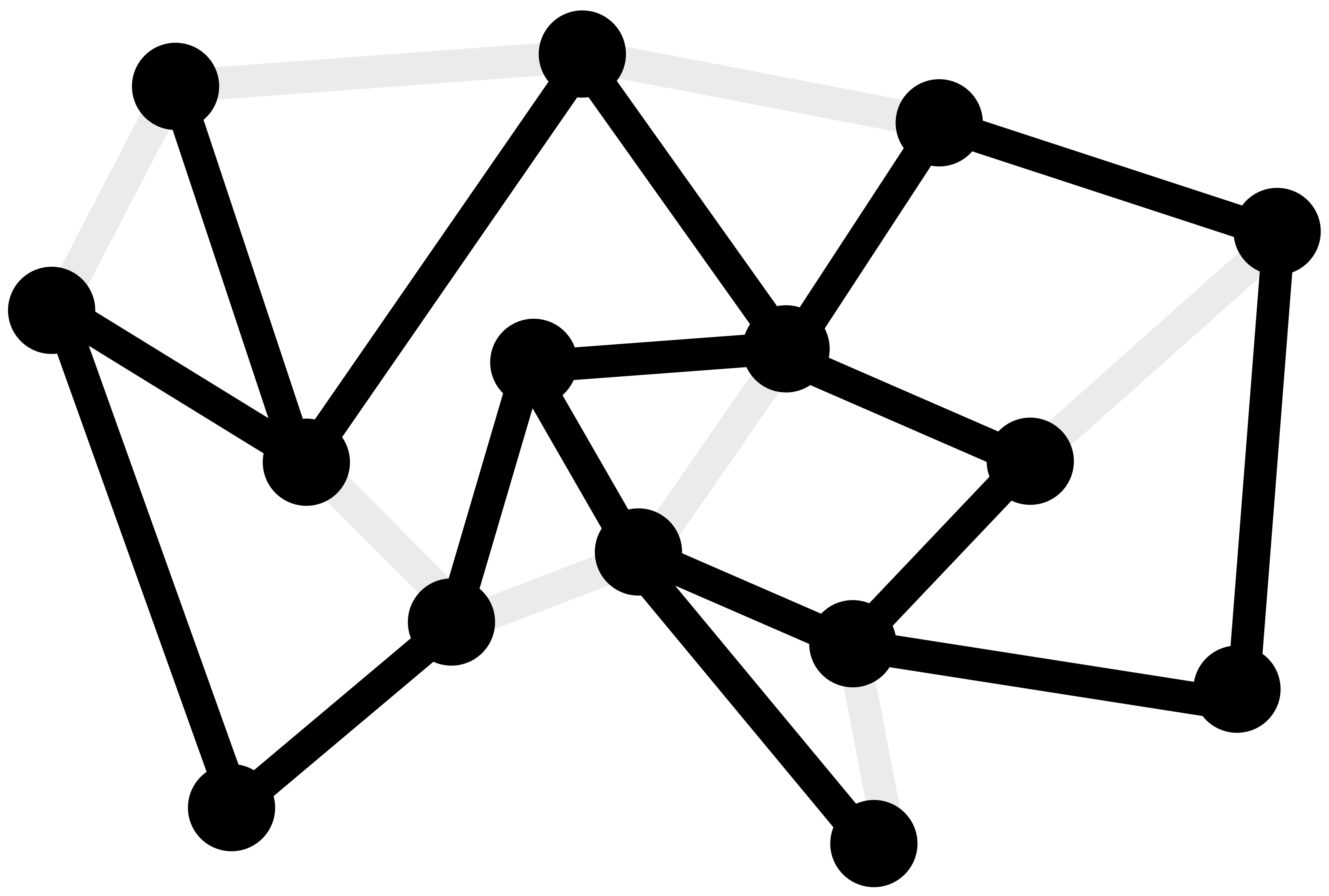
$G = (V, E)$

Idea: be greedy wrt edges

Greedy Algorithm

- $H \leftarrow \emptyset$
- For $\{u, v\} \in E$:
  - If $d_H(u, v) > t$ then

    $H \leftarrow H + \{u, v\}$



$G = (V, E)$

Idea: be greedy wrt edges

Roadmap of Proof

1. **Simple Observation**

   edge spanners suffice

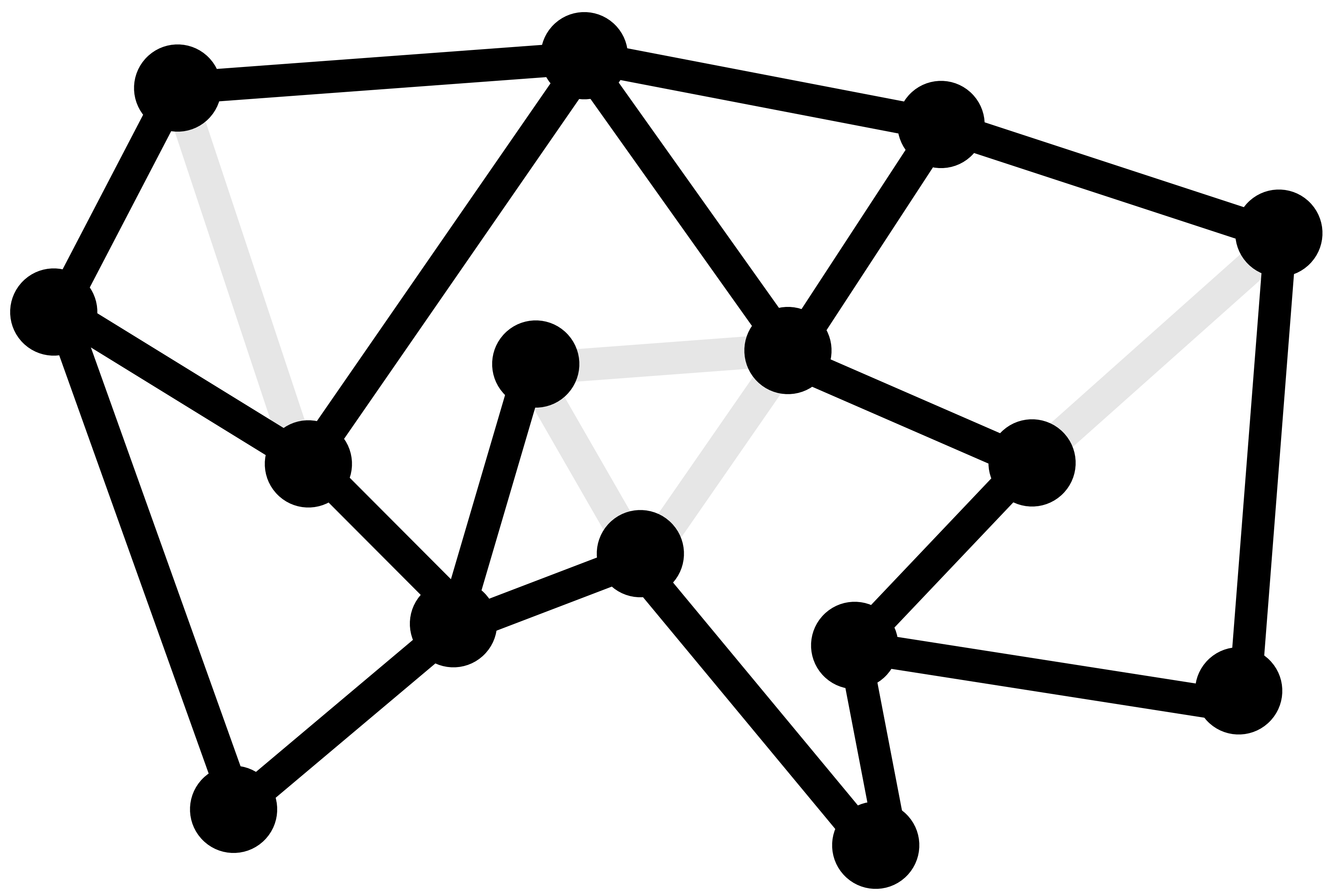
2. **Greedy Algorithm**

   suggested by observation

3. **Distortion Analysis**

4. **Size Analysis**

   by "Moore Bounds"



Theorem: every graph G has a t-spanner H w/

- **Distortion:** $t = O(\log n)$
- **Size:** $|H| = O(n)$

Roadmap of Proof

1. **Simple Observation**

   edge spanners suffice

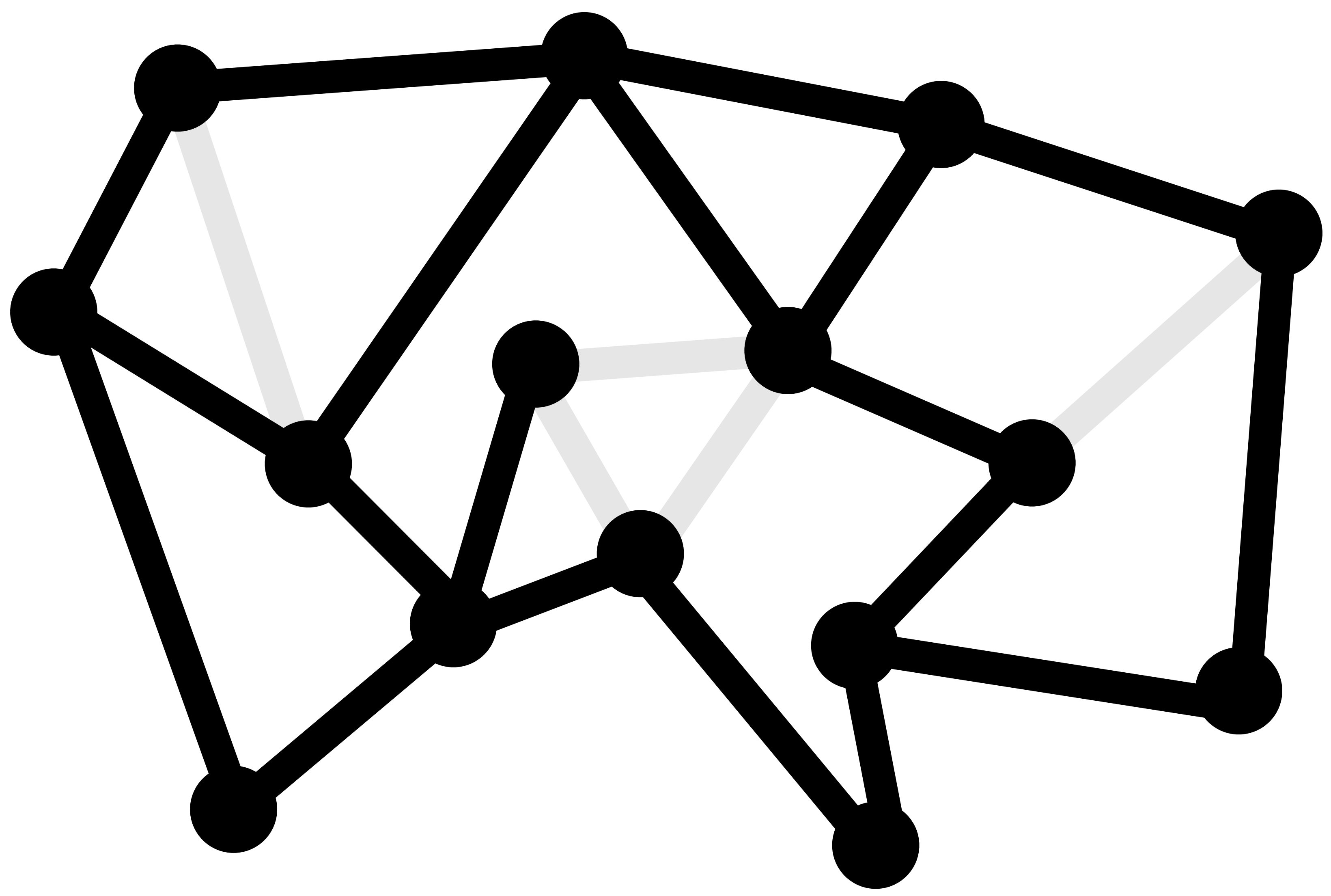
2. **Greedy Algorithm**

   suggested by observation

3. **Distortion Analysis**

4. **Size Analysis**

   by "Moore Bounds"



Theorem: every graph G has a t-spanner H w/

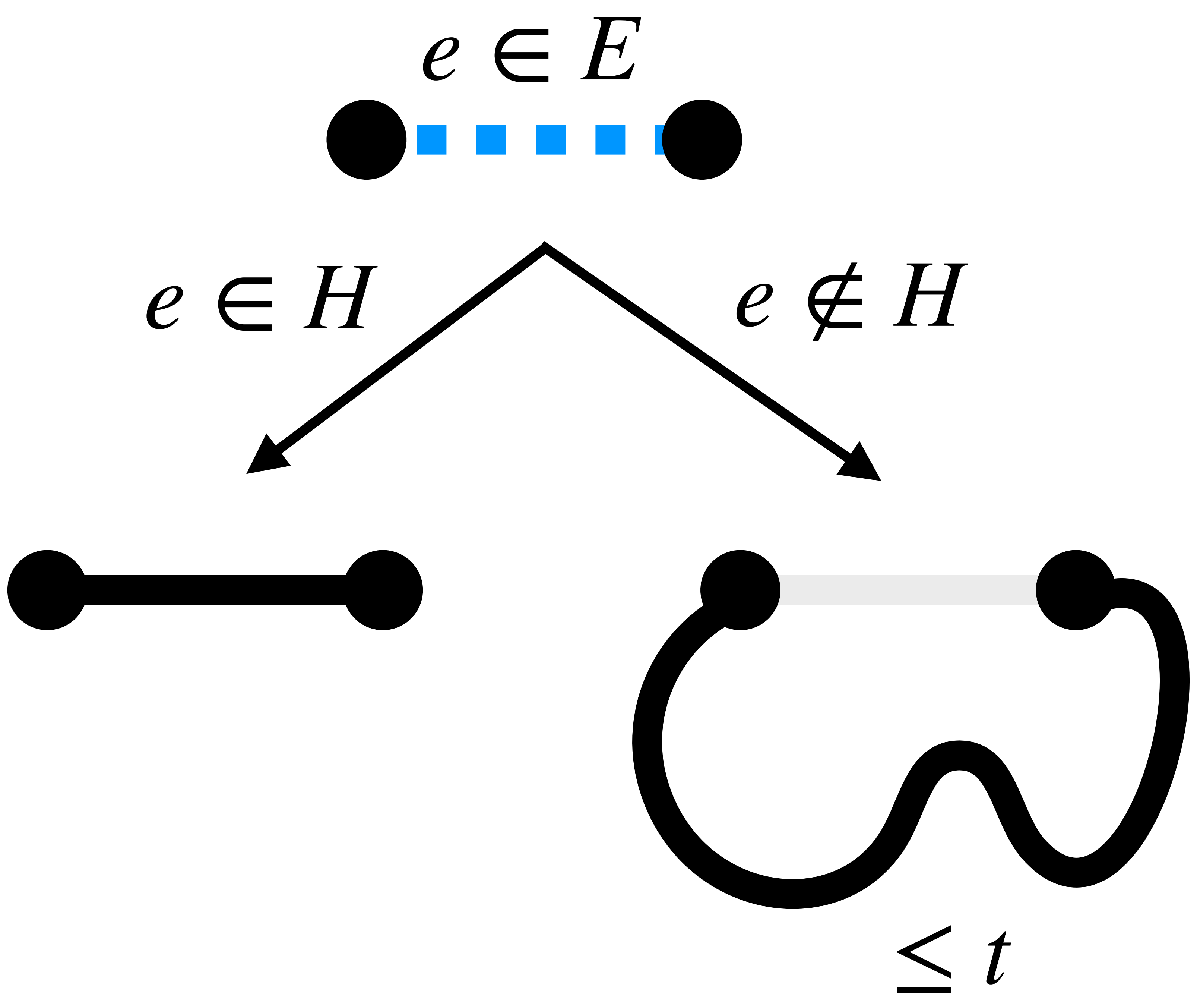
- **Distortion:** $t = O(\log n)$
- **Size:** $|H| = O(n)$

Distortion Analysis

Edge Spanners

all edges distorted $\leq t$

Claim: output H of greedy is a t-edge-spanner



Distortion Analysis

Edge Spanners

all edges distorted $\leq t$

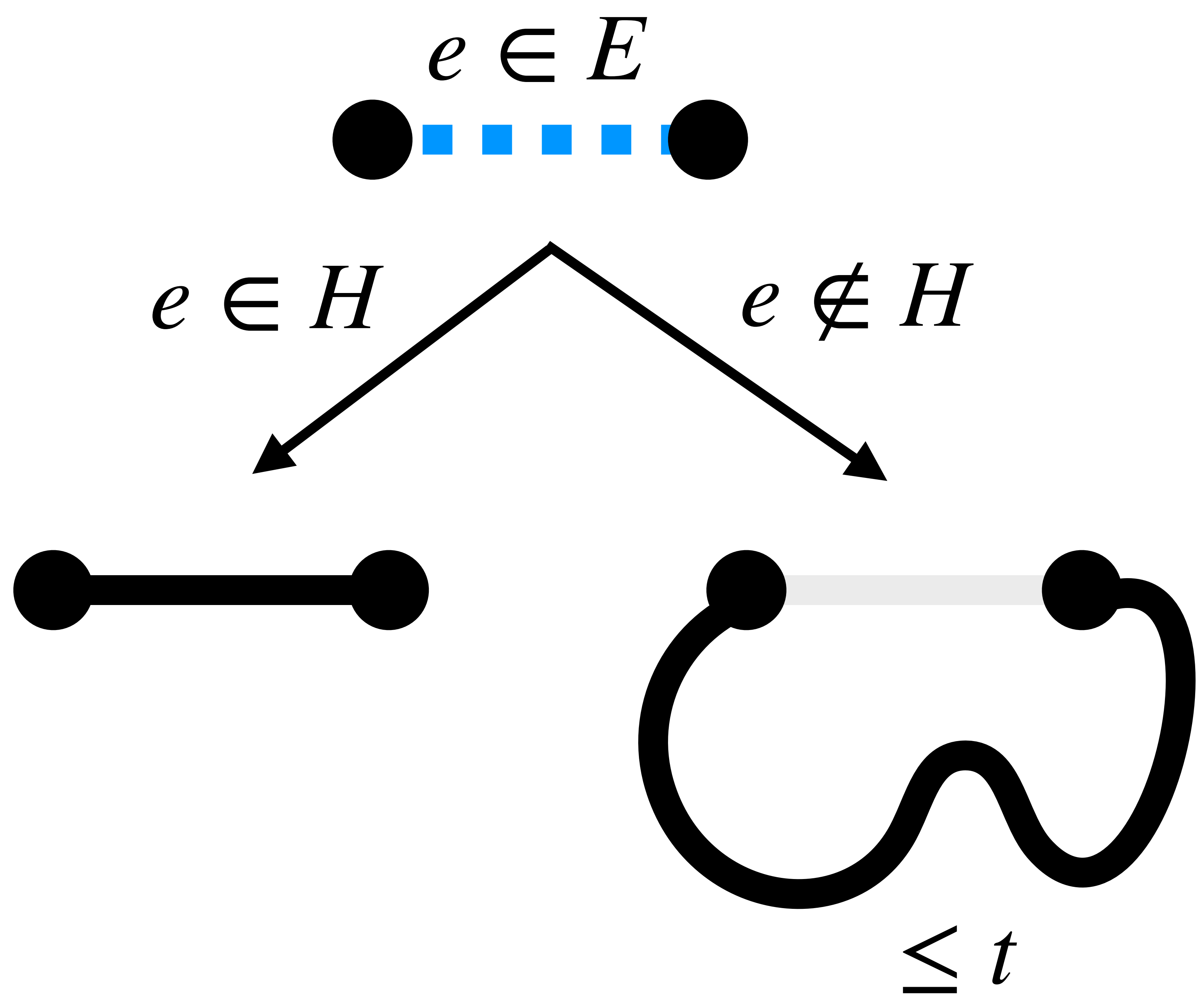
Claim: output H of greedy is a t-edge-spanner

↓

Claim: H is a t-spanner iff it is a t-edge-spanner

↓

Claim: output of greedy is a t-spanner



Roadmap of Proof

1. **Simple Observation**

   edge spanners suffice

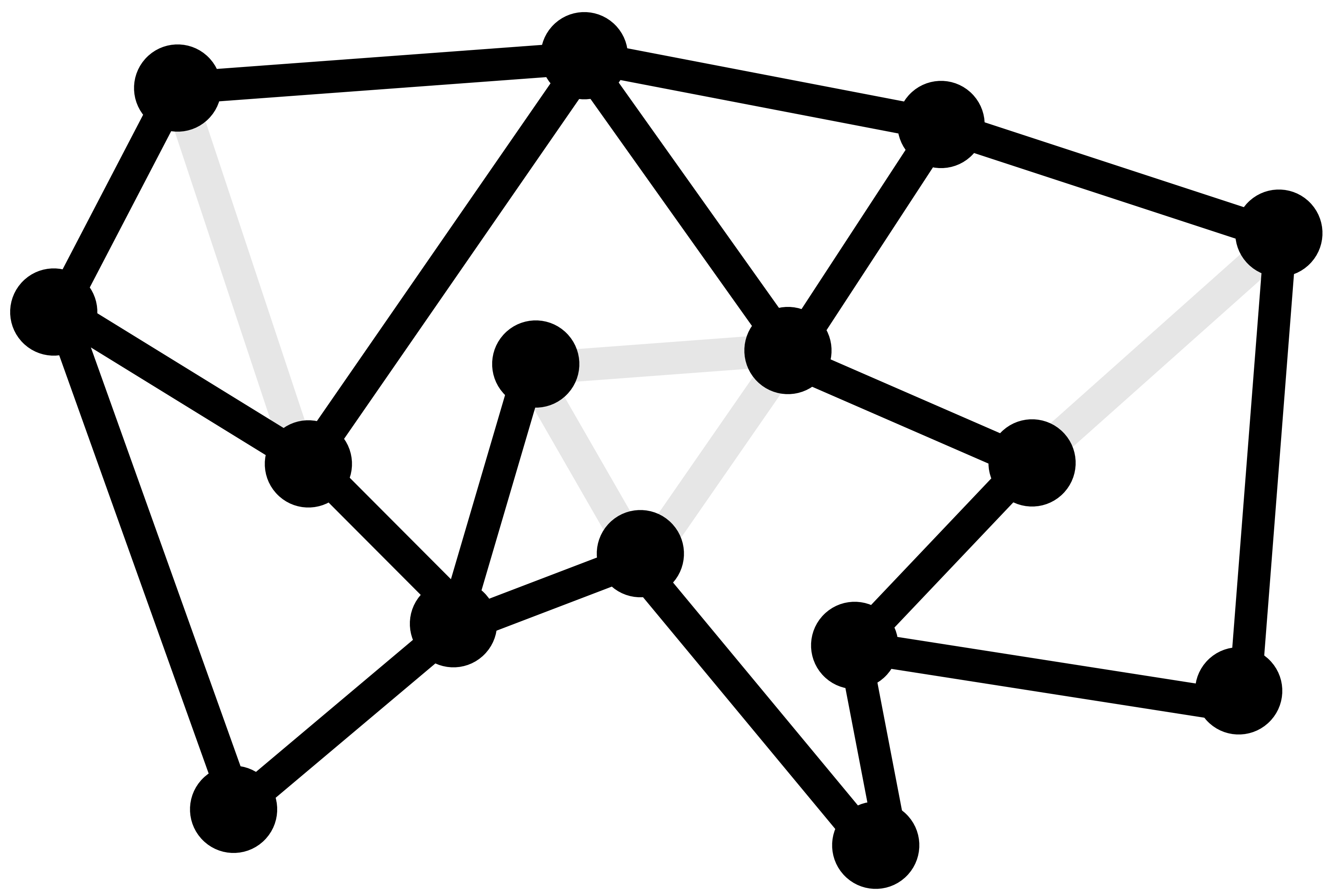
2. **Greedy Algorithm**

   suggested by observation

3. **Distortion Analysis**

4. **Size Analysis**

   by "Moore Bounds"



Theorem: every graph G has a t-spanner H w/

- **Distortion:** $t = O(\log n)$
- **Size:** $|H| = O(n)$

Roadmap of Proof

1. **Simple Observation**

   edge spanners suffice

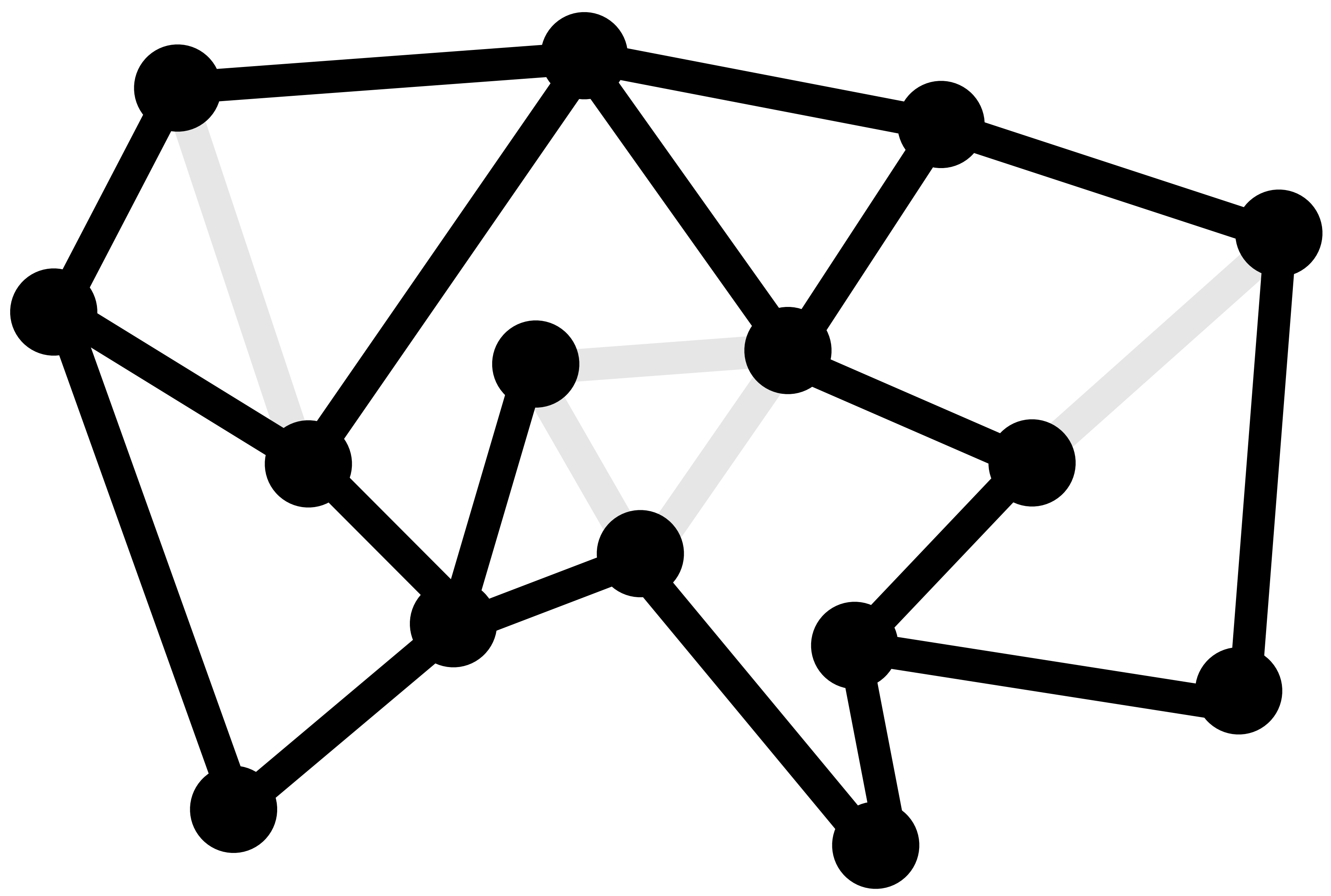
2. **Greedy Algorithm**

   suggested by observation

3. **Distortion Analysis**

4. **Size Analysis**

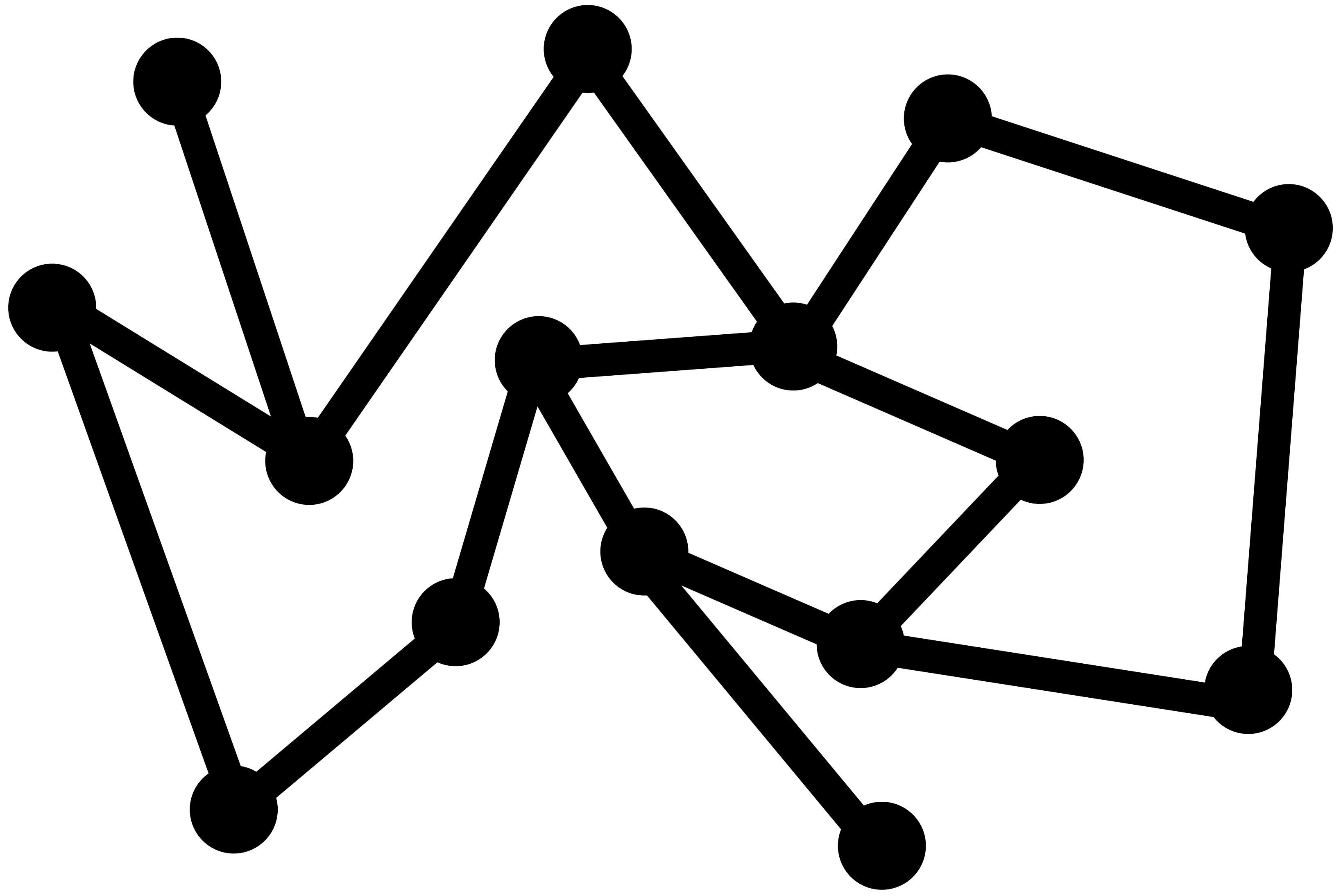
   by "Moore Bounds"



Theorem: every graph G has a t-spanner H w/

- **Distortion:** $t = O(\log n)$
- **Size:** $|H| = O(n)$

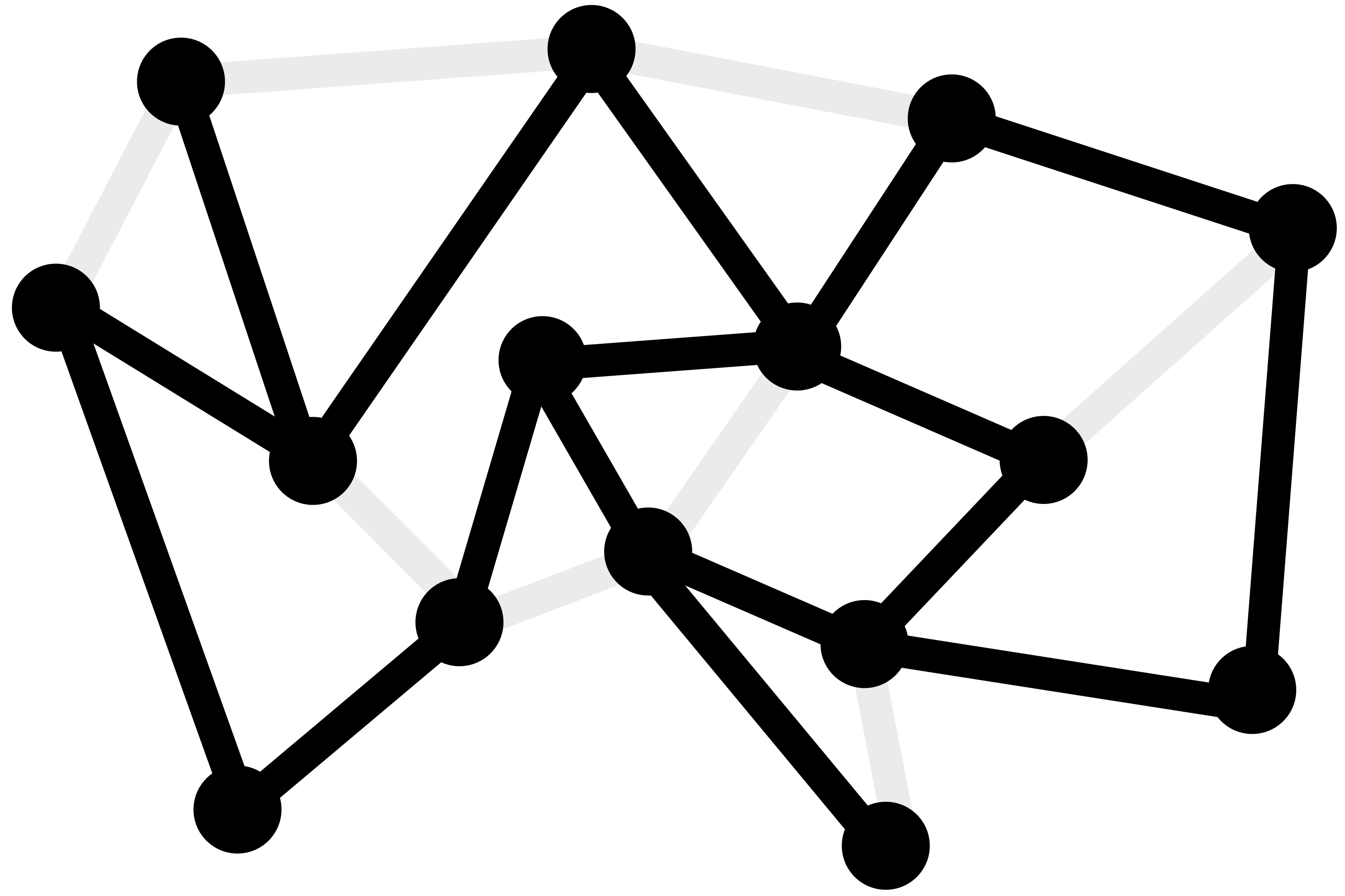
Size Analysis



girth 5

Definition (girth)**:** the girth g of graph H is the length of its shortest cycle

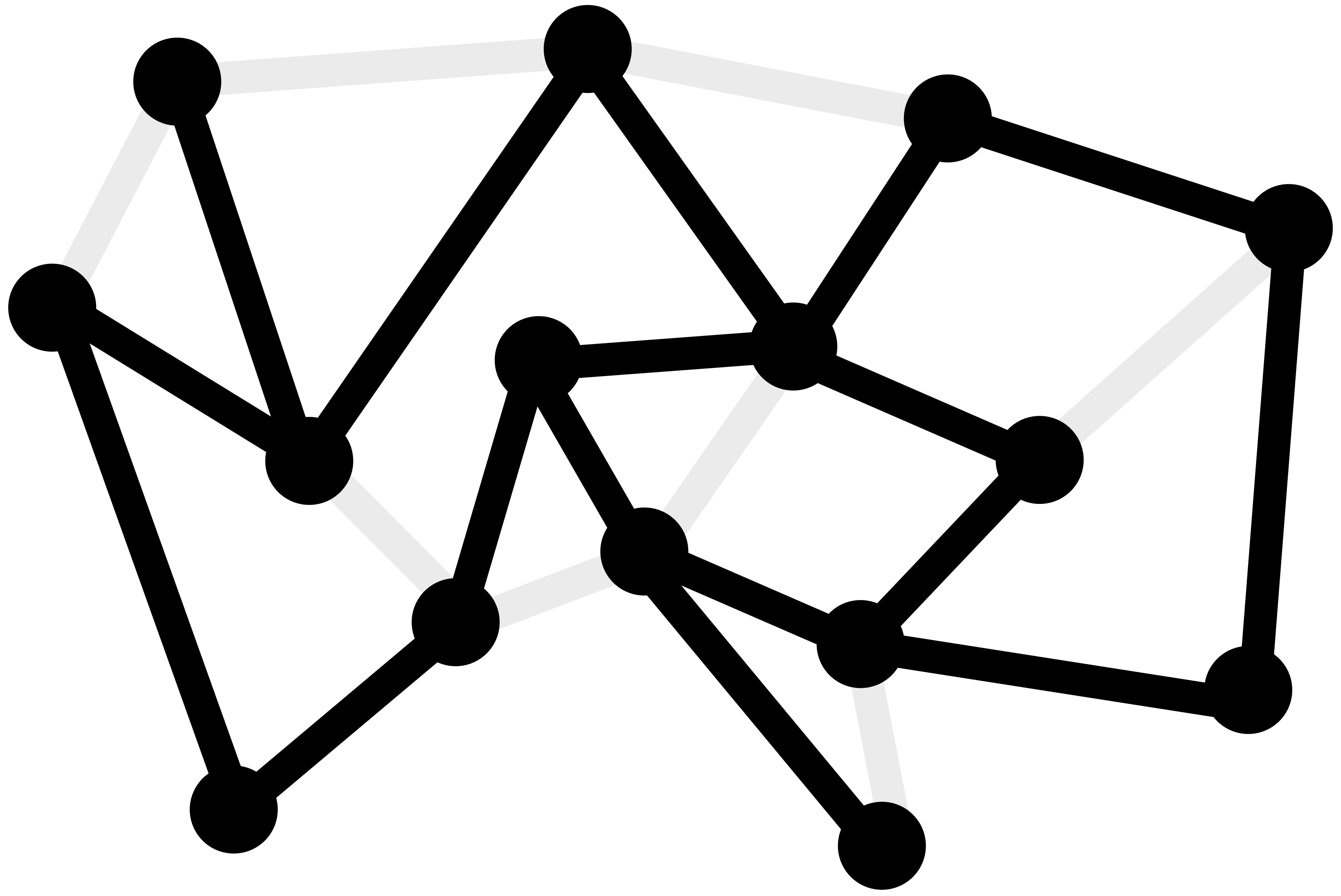
Size Analysis



output of greedy algorithm with $t = 3$

Definition (girth)**:** the girth g of graph H is the length of its shortest cycle

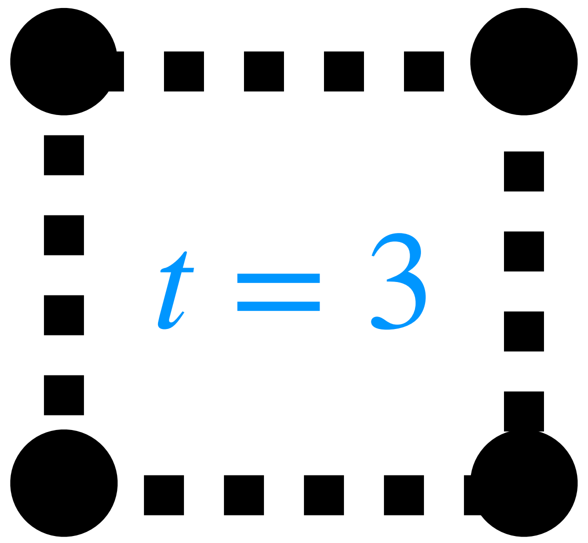
Size Analysis



output of greedy algorithm with $t = 3$

Claim: output of greedy algorithm has girth $\geq t + 2$

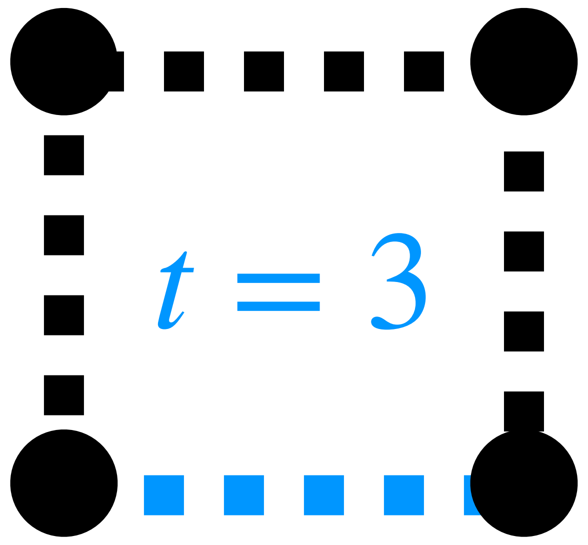
Size Analysis



AFSOC a $\leq t + 1$*-Cycle*

Claim: output of greedy algorithm has girth $\geq t + 2$

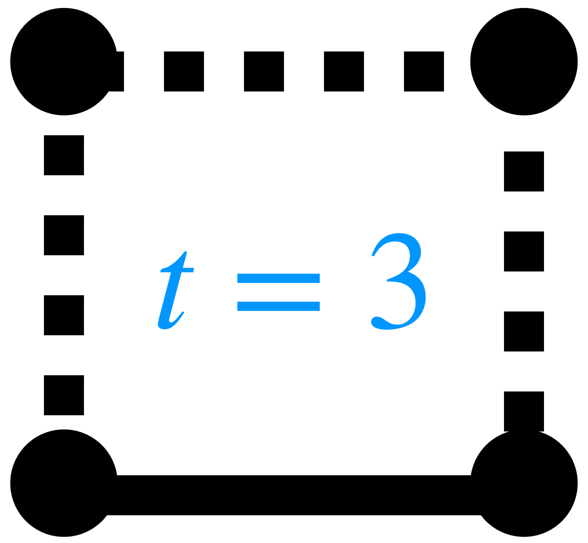
Size Analysis



AFSOC a $\leq t+1$*-Cycle*

Claim: output of greedy algorithm has girth $\geq t+2$

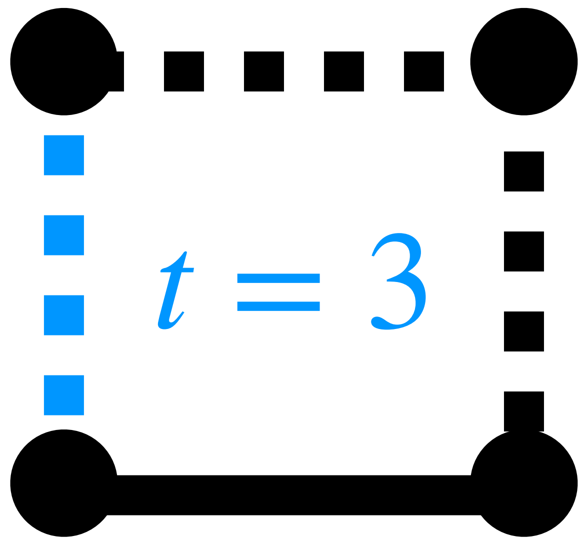
Size Analysis



AFSOC a $\leq t + 1$*-Cycle*

Claim: output of greedy algorithm has girth $\geq t + 2$

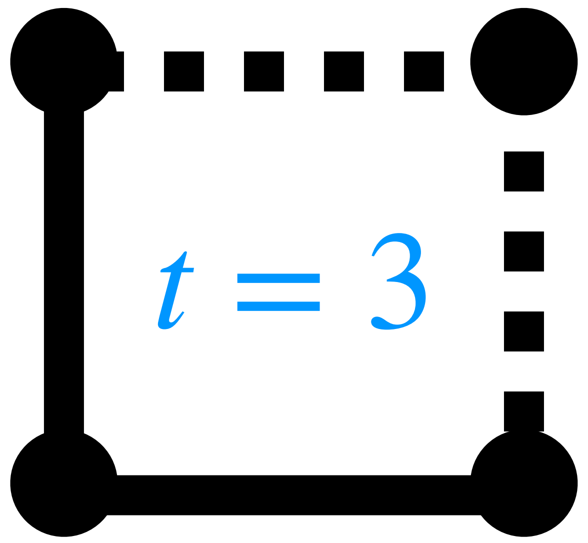
Size Analysis



AFSOC a $\leq t+1$*-Cycle*

Claim: output of greedy algorithm has girth $\geq t+2$

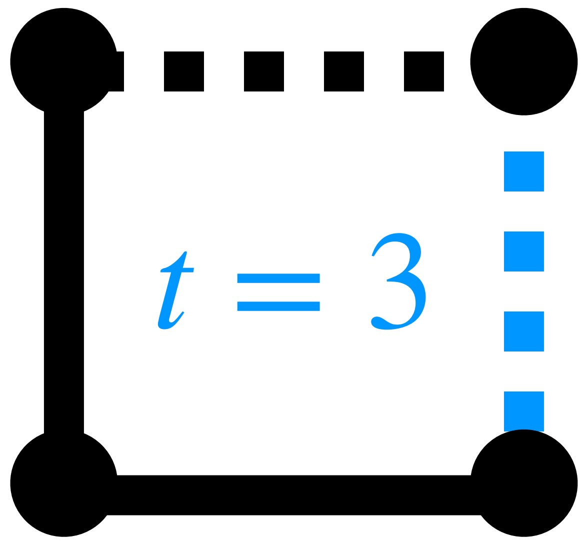
Size Analysis



AFSOC a $\leq t + 1$*-Cycle*

Claim: output of greedy algorithm has girth $\geq t + 2$

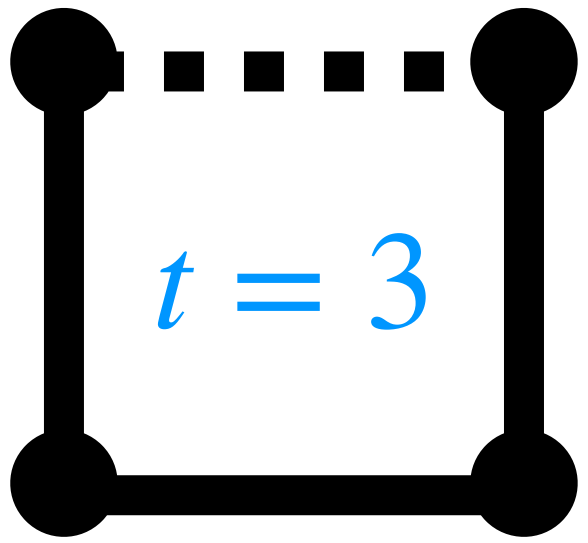
Size Analysis



AFSOC a $\leq t + 1$*-Cycle*

Claim: output of greedy algorithm has girth $\geq t + 2$

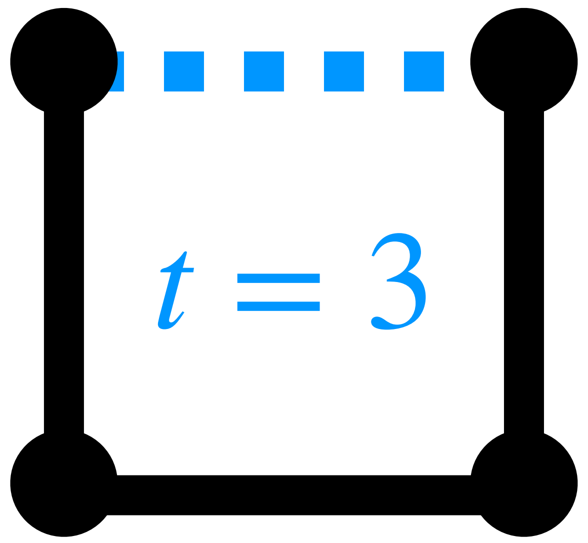
Size Analysis



AFSOC a $\leq t+1$*-Cycle*

Claim: output of greedy algorithm has girth $\geq t+2$

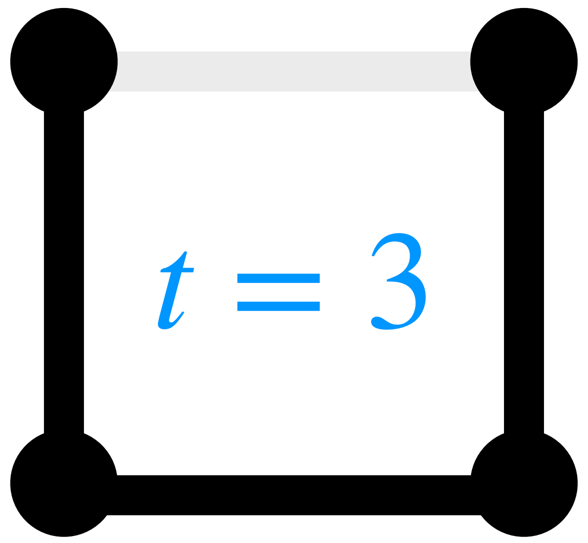
Size Analysis



AFSOC a $\leq t + 1$*-Cycle*

Claim: output of greedy algorithm has girth $\geq t + 2$

Size Analysis

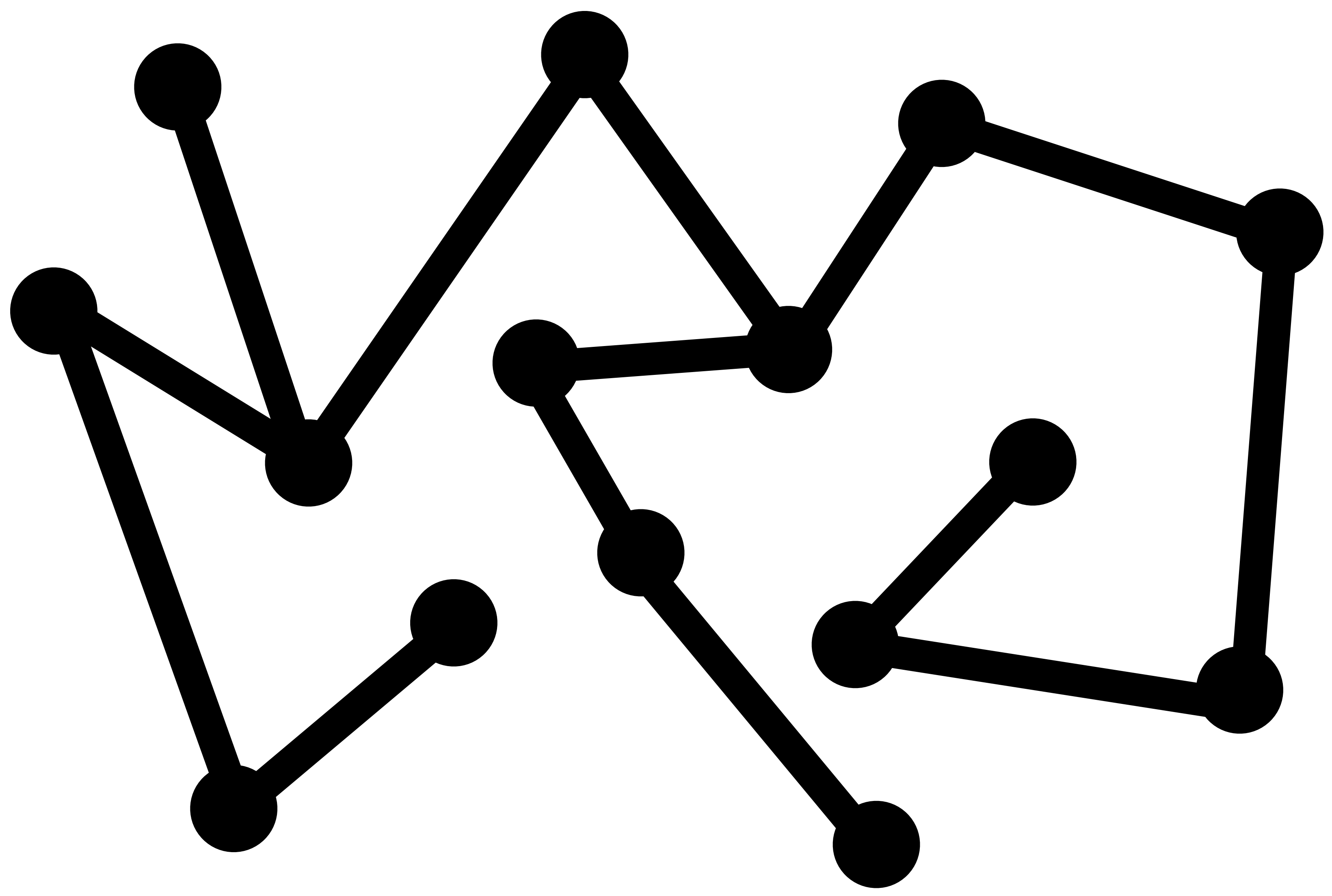


AFSOC a $\leq t+1$*-Cycle*

Claim: output of greedy algorithm has girth $\geq t+2$

Size Analysis

What's Girth Have to Do with Size?



$\leq n - 1$ *Edges*

Intuition: trees are sparse, high girth = locally tree-like

Size Analysis

What's Girth Have to Do with Size?

0

do depth $\left\lfloor \frac{g}{2} \right\rceil - 1$ BFS

girth g graph

Intuition: trees are sparse, high girth = locally tree-like

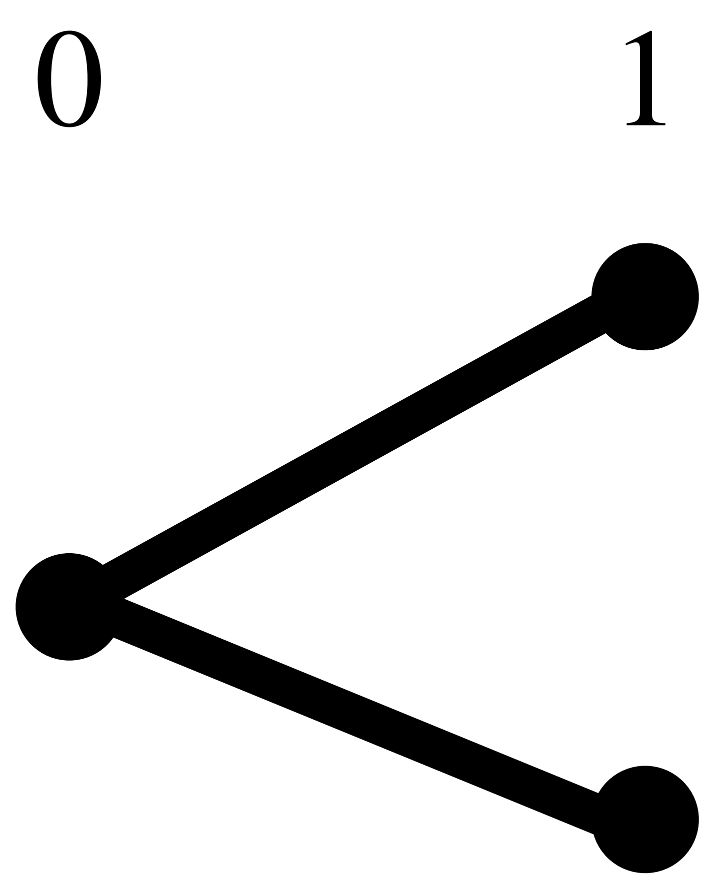
Size Analysis

What's Girth Have to Do with Size?

do

depth $\left\lceil \frac{g}{2} \right\rceil - 1$

BFS



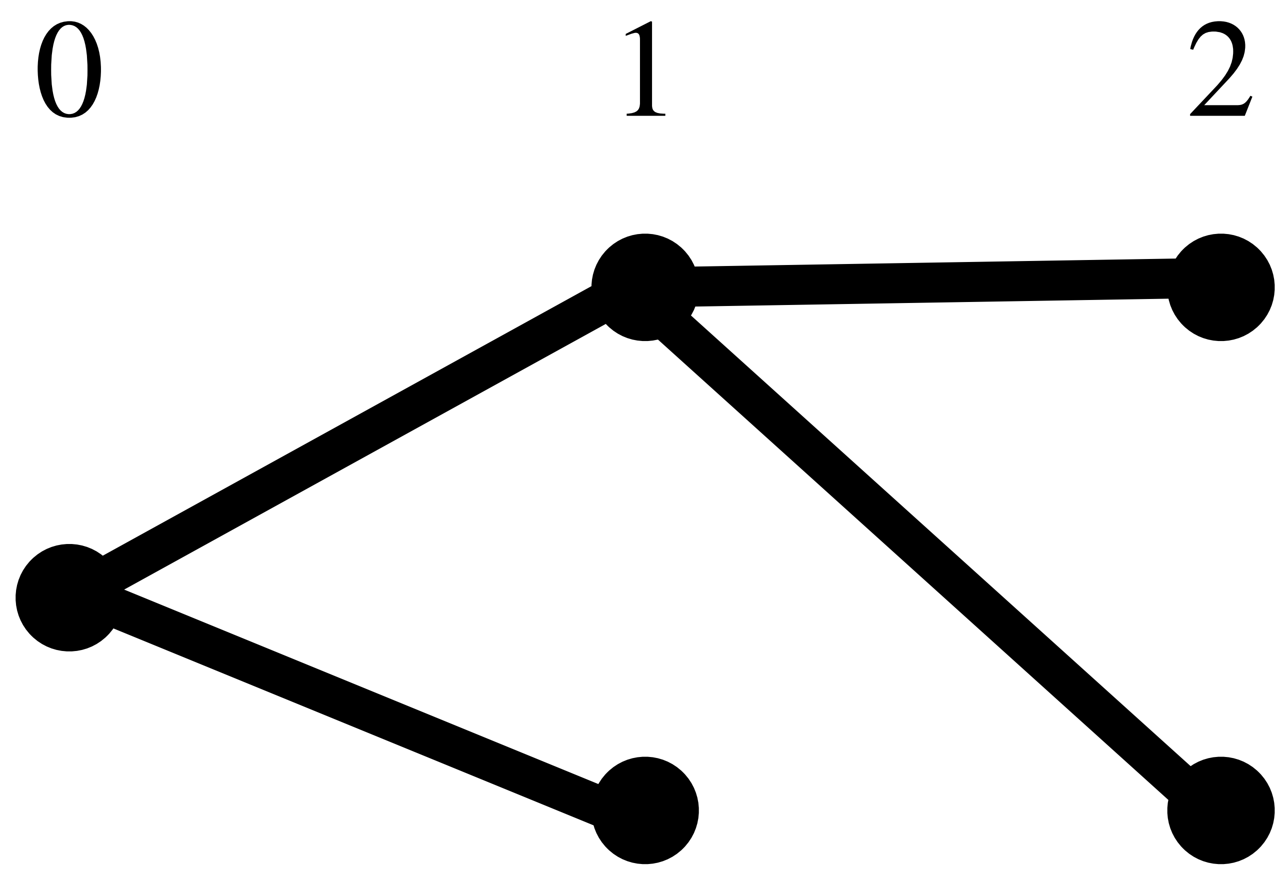
girth g graph

Intuition: trees are sparse, high girth = locally tree-like

Size Analysis

What's Girth Have to Do with Size?

do depth $\left\lfloor \frac{g}{2} \right\rfloor - 1$ BFS



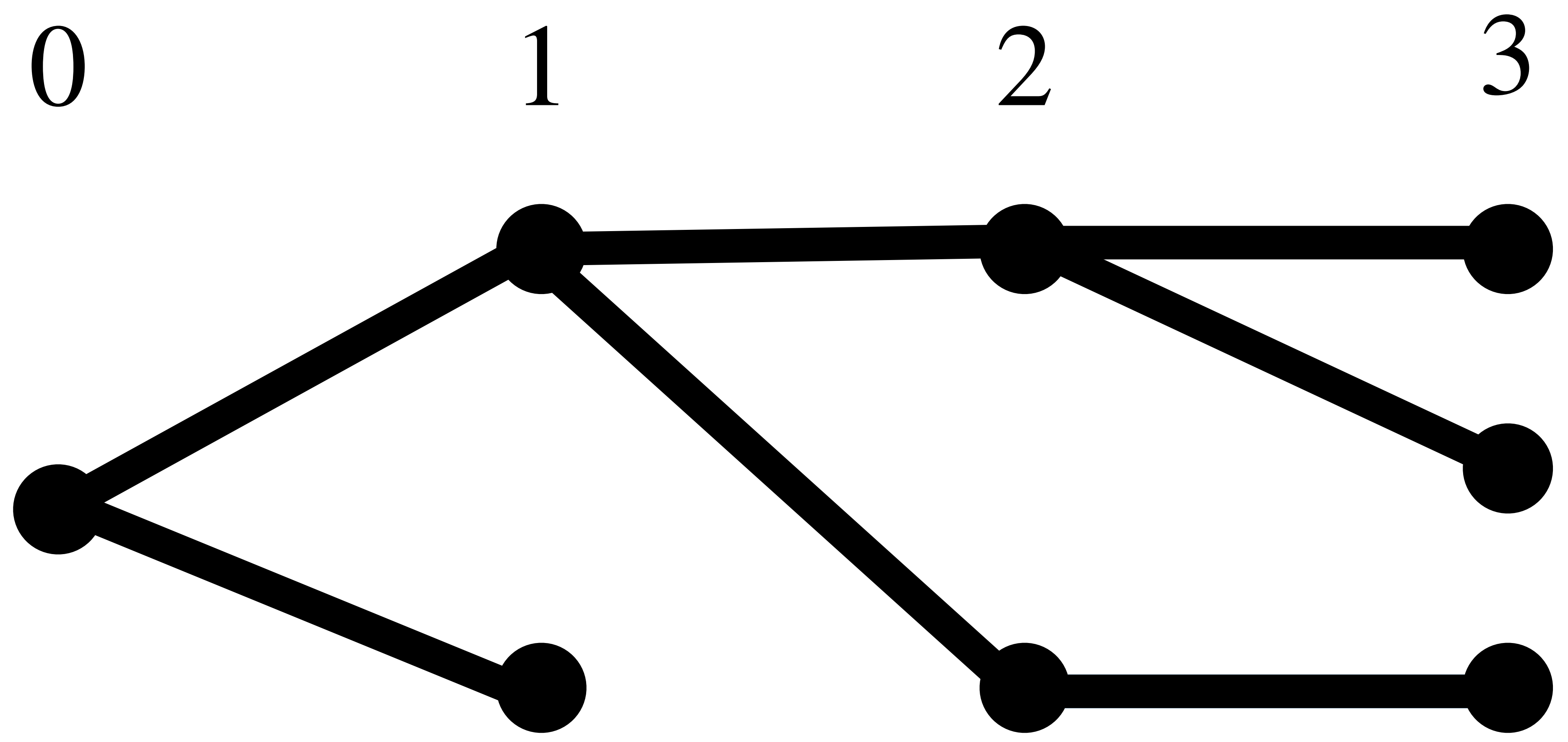
girth g graph

Intuition: trees are sparse, high girth = locally tree-like

Size Analysis

What's Girth Have to Do with Size?

do
depth $\left\lfloor \frac{g}{2} \right\rceil - 1$
BFS



girth g graph

Intuition: trees are sparse, high girth = locally tree-like

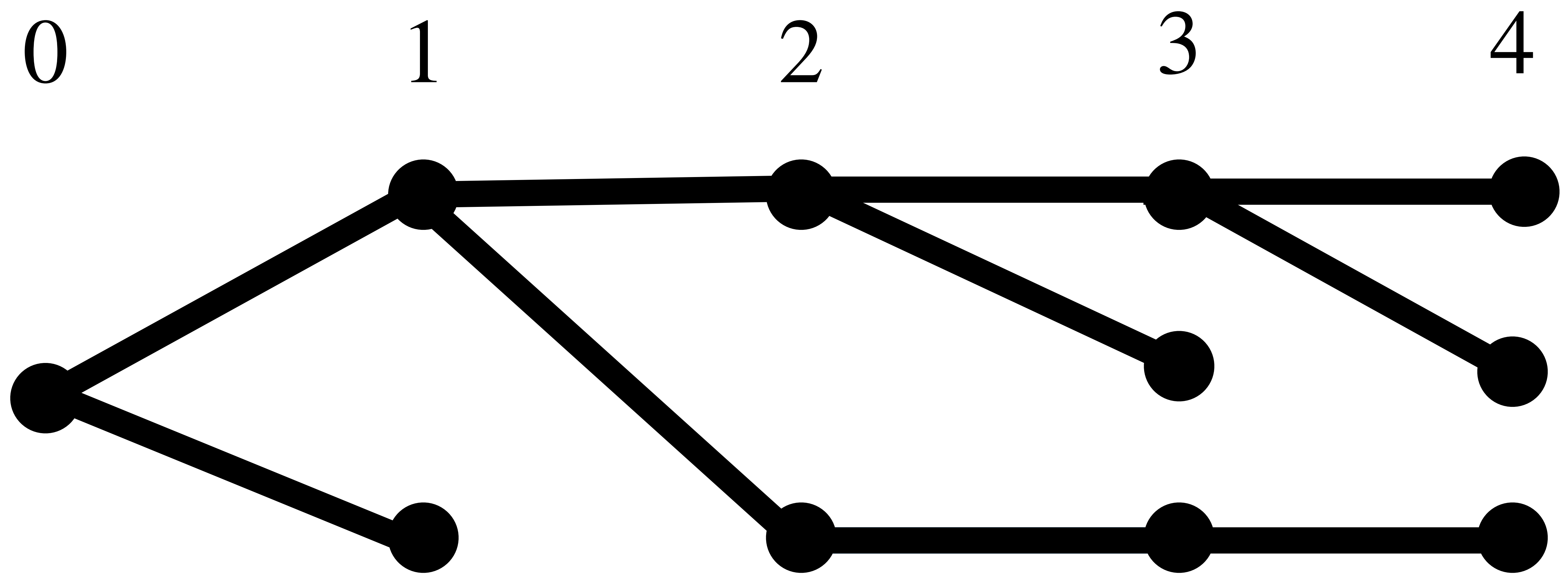
Size Analysis

What's Girth Have to Do with Size?

do

depth $\left\lfloor \frac{g}{2} \right\rceil - 1$

BFS



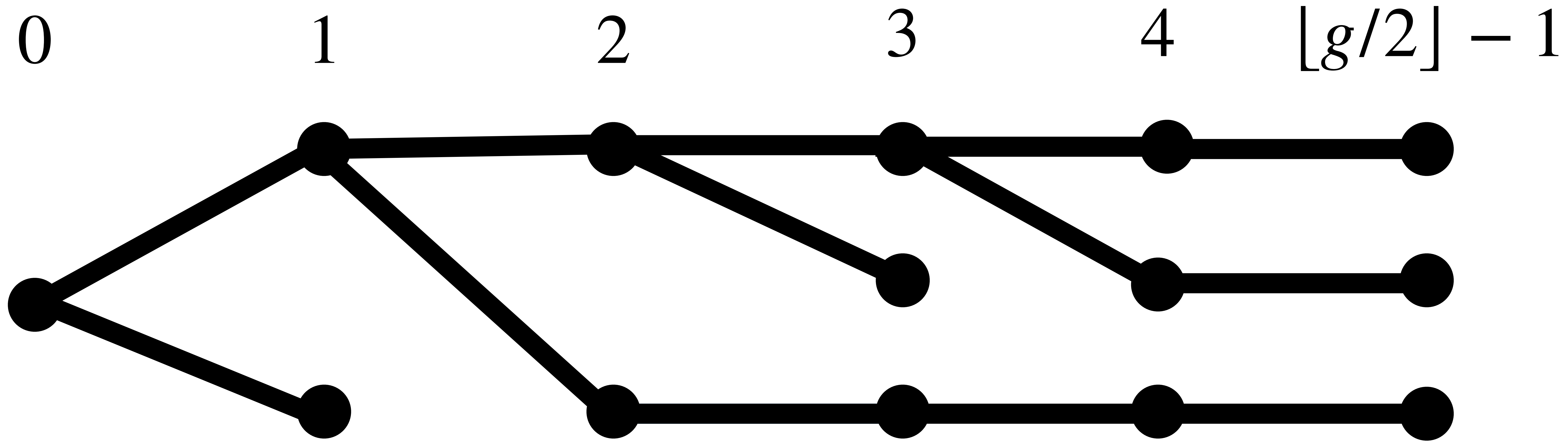
girth g graph

Intuition: trees are sparse, high girth = locally tree-like

Size Analysis

What's Girth Have to Do with Size?

do depth $\left\lfloor \frac{g}{2} \right\rfloor - 1$ BFS



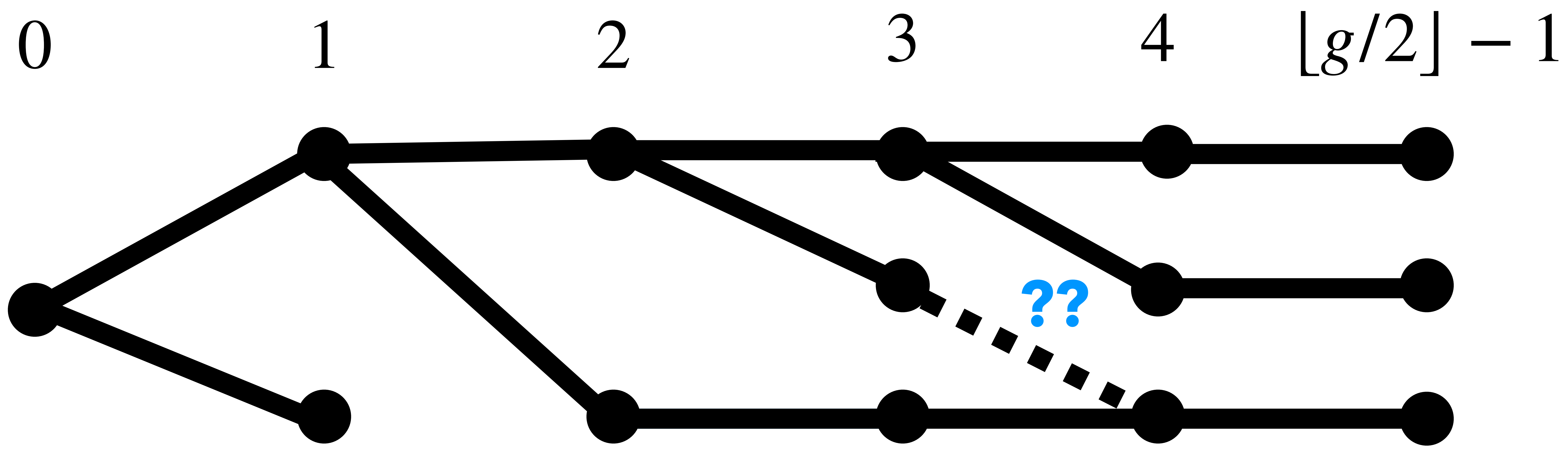
girth g graph

Intuition: trees are sparse, high girth = locally tree-like

Size Analysis

What's Girth Have to Do with Size?

do depth $\left\lfloor \frac{g}{2} \right\rfloor - 1$ BFS



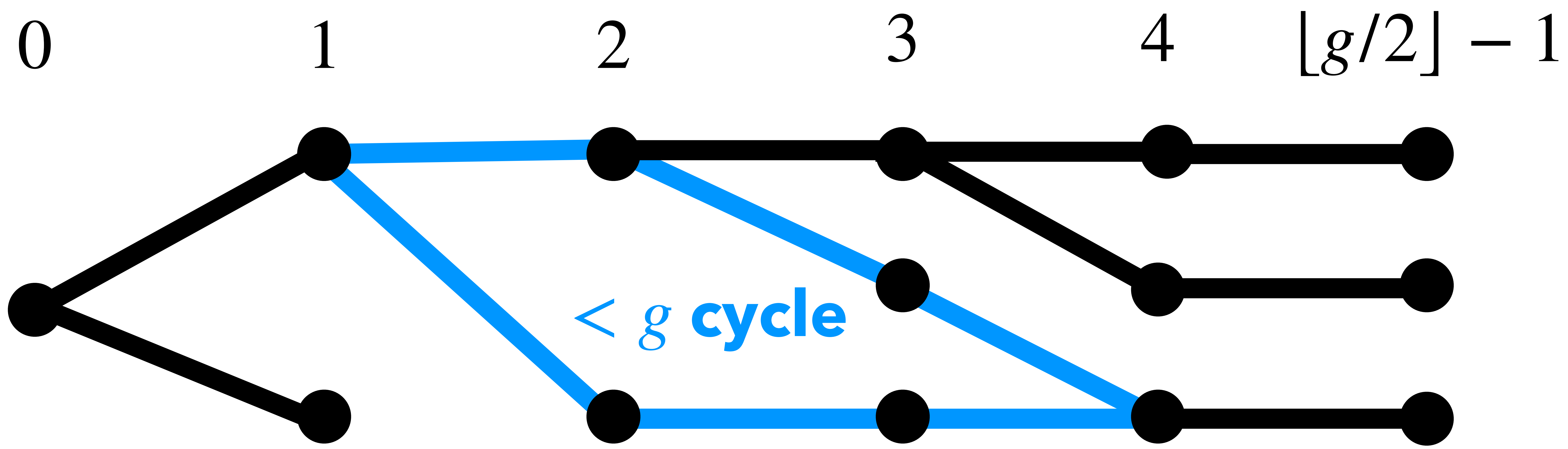
girth g graph

Intuition: trees are sparse, high girth = locally tree-like

Size Analysis

What's Girth Have to Do with Size?

do depth $\left\lfloor \frac{g}{2} \right\rfloor - 1$ BFS

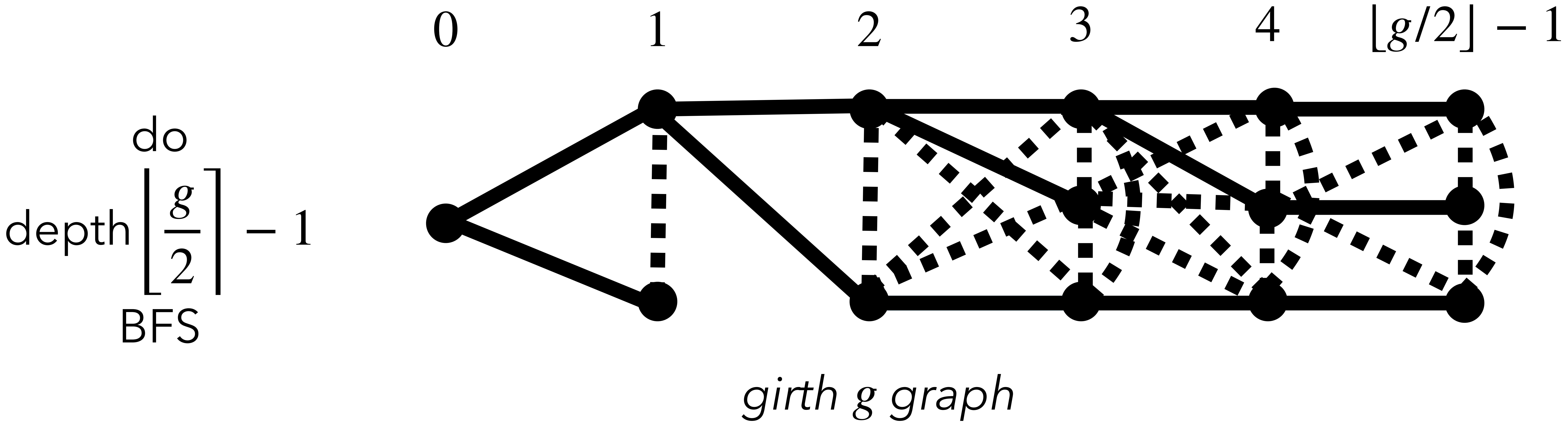


girth g graph

Intuition: trees are sparse, high girth = locally tree-like

Size Analysis

What's Girth Have to Do with Size?

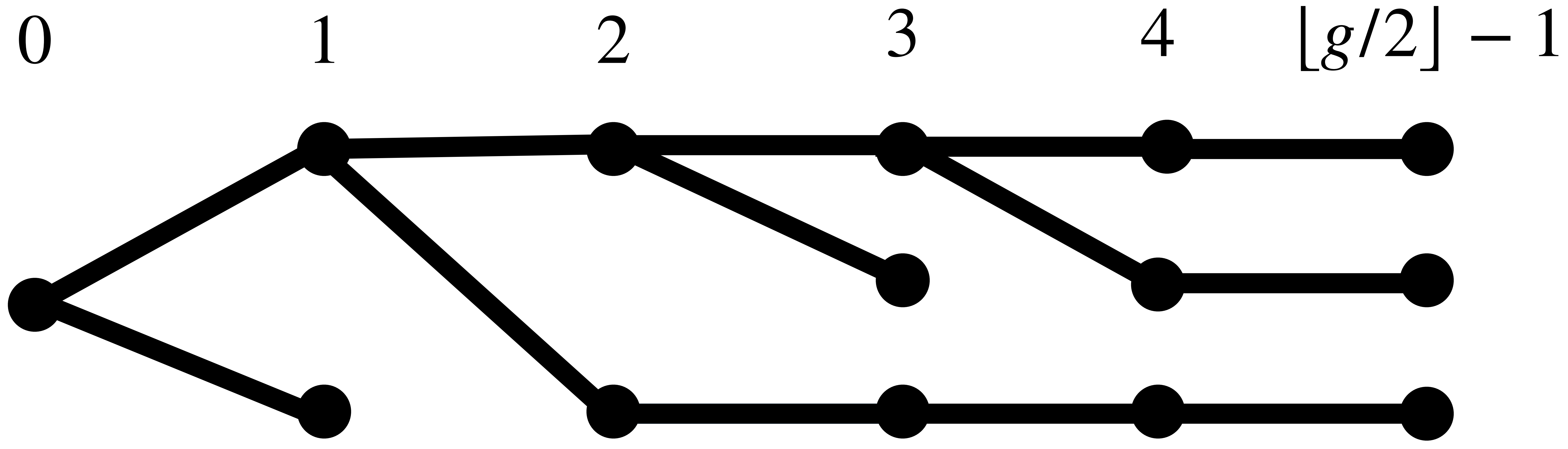


Intuition: trees are sparse, high girth = locally tree-like

Size Analysis

What's Girth Have to Do with Size?

do BFS depth $\left\lfloor \frac{g}{2} \right\rfloor - 1$

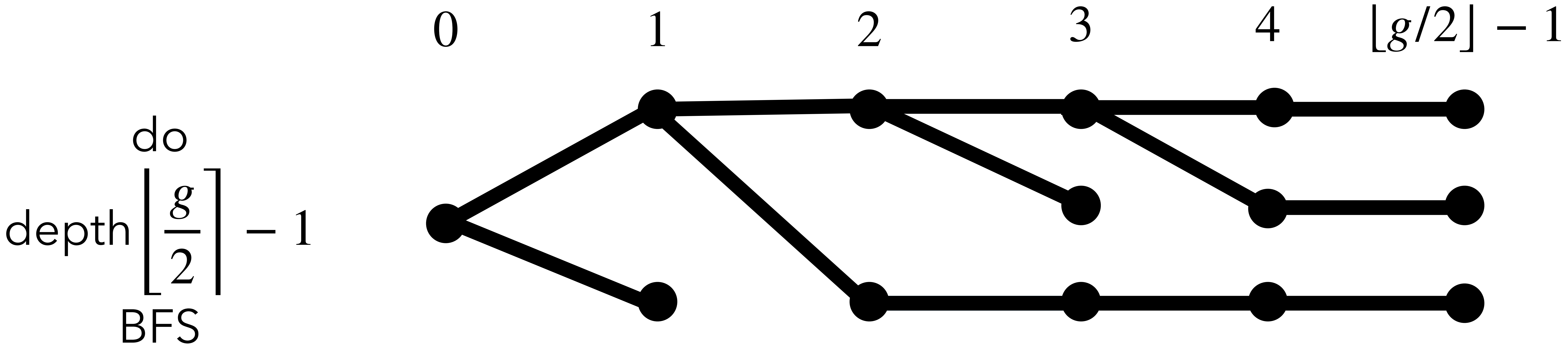


girth g graph

Intuition: trees are sparse, high girth = locally tree-like

Size Analysis

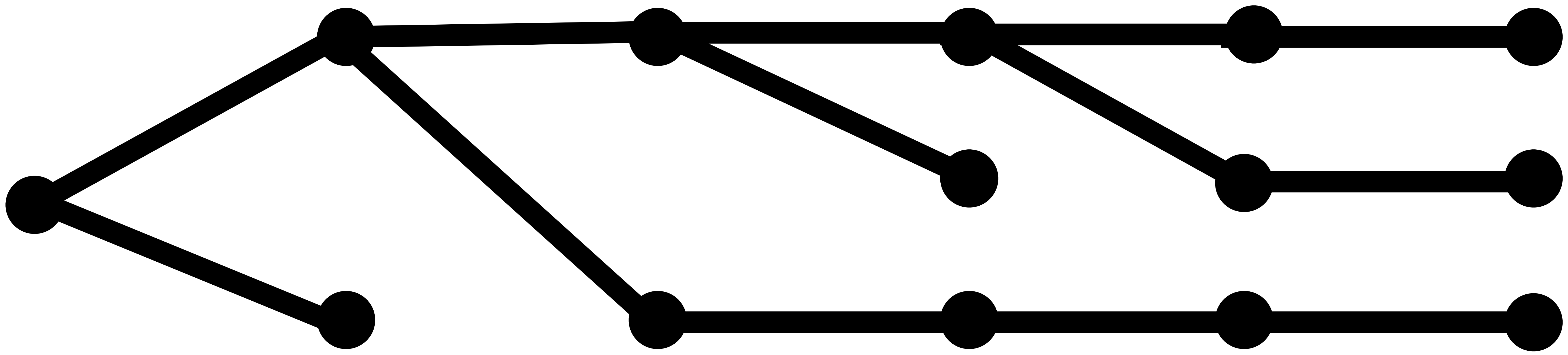
What's Girth Have to Do with Size?



girth g graph

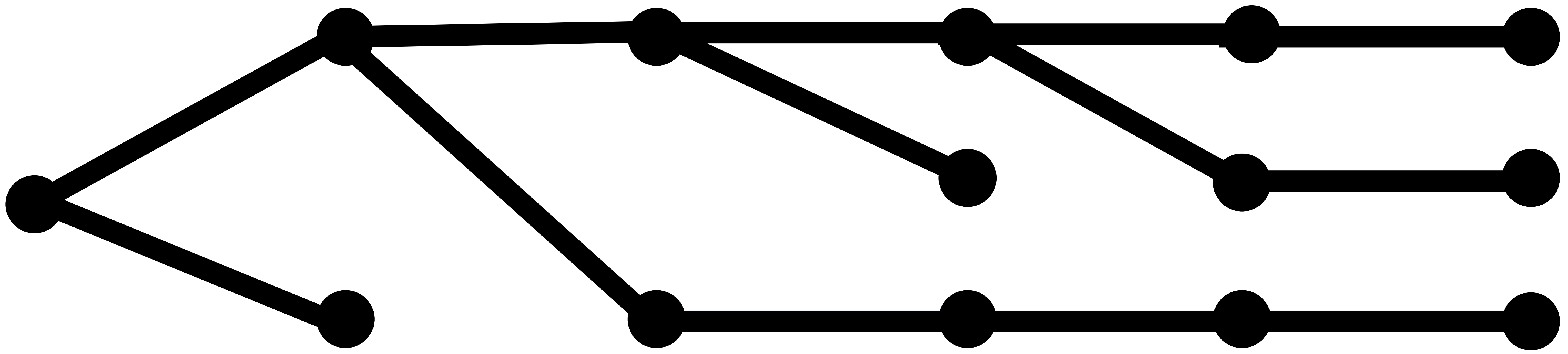
Theorem: a girth $\Omega(\log n)$ graph has at most $O(n)$ edges

Size Analysis



Theorem: a girth $\Omega(\log n)$ graph has at most $O(n)$ edges

Size Analysis



Theorem: a girth $\Omega(\log n)$ graph has at most $O(n)$ edges

↓

Claim: output of greedy algorithm has girth $\geq t + 2$

↓

Theorem: output of greedy algorithm with $t = \Omega(\log n)$ has at most $O(n)$ edges

Roadmap of Proof

1. **Simple Observation**

   edge spanners suffice

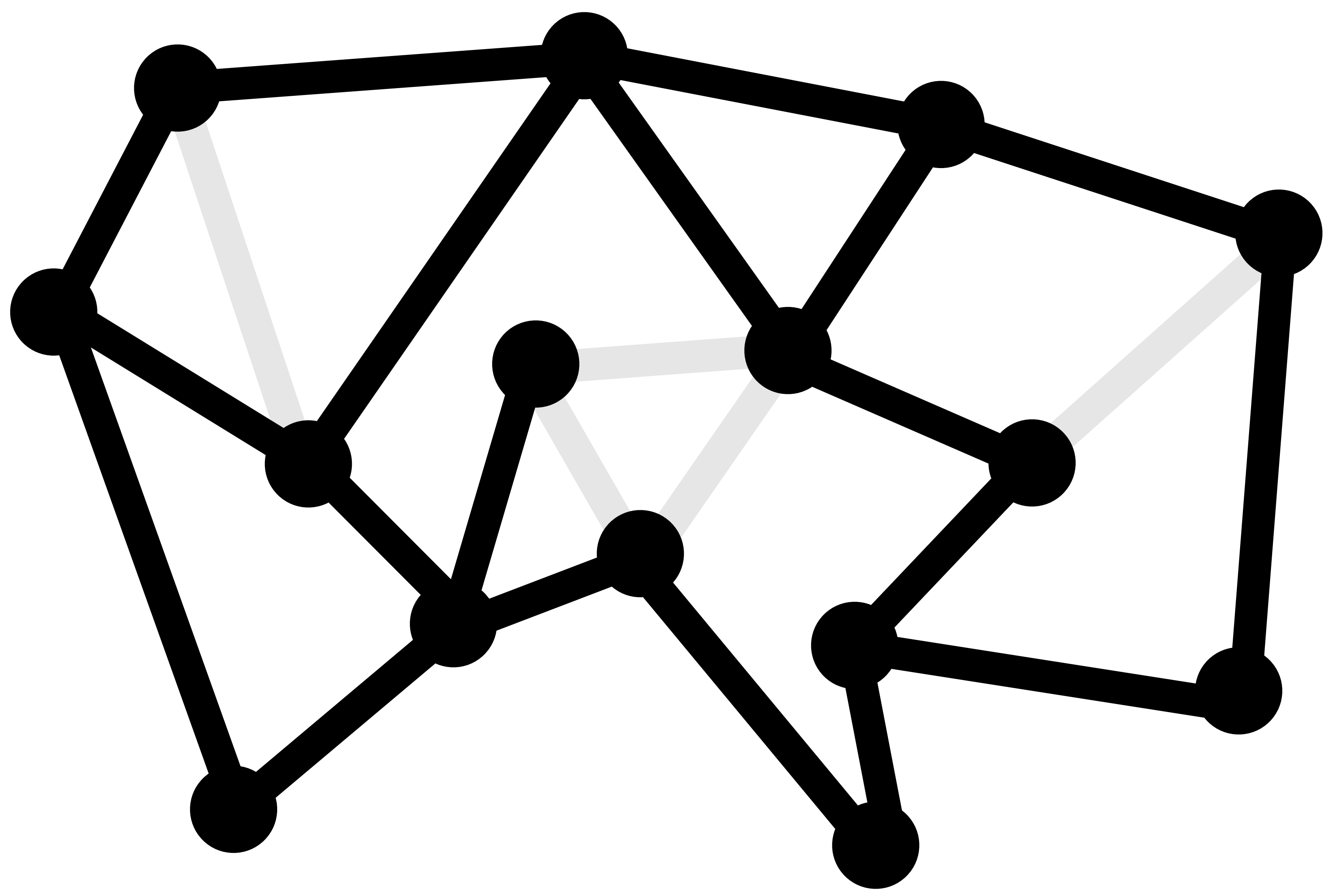
2. **Greedy Algorithm**

   suggested by observation

3. **Distortion Analysis**

4. **Size Analysis**

   by "Moore Bounds"



Theorem: every graph G has a t-spanner H w/

- **Distortion:** $t = O(\log n)$
- **Size:** $|H| = O(n)$

Roadmap of Proof

1. **Simple Observation**

   edge spanners suffice

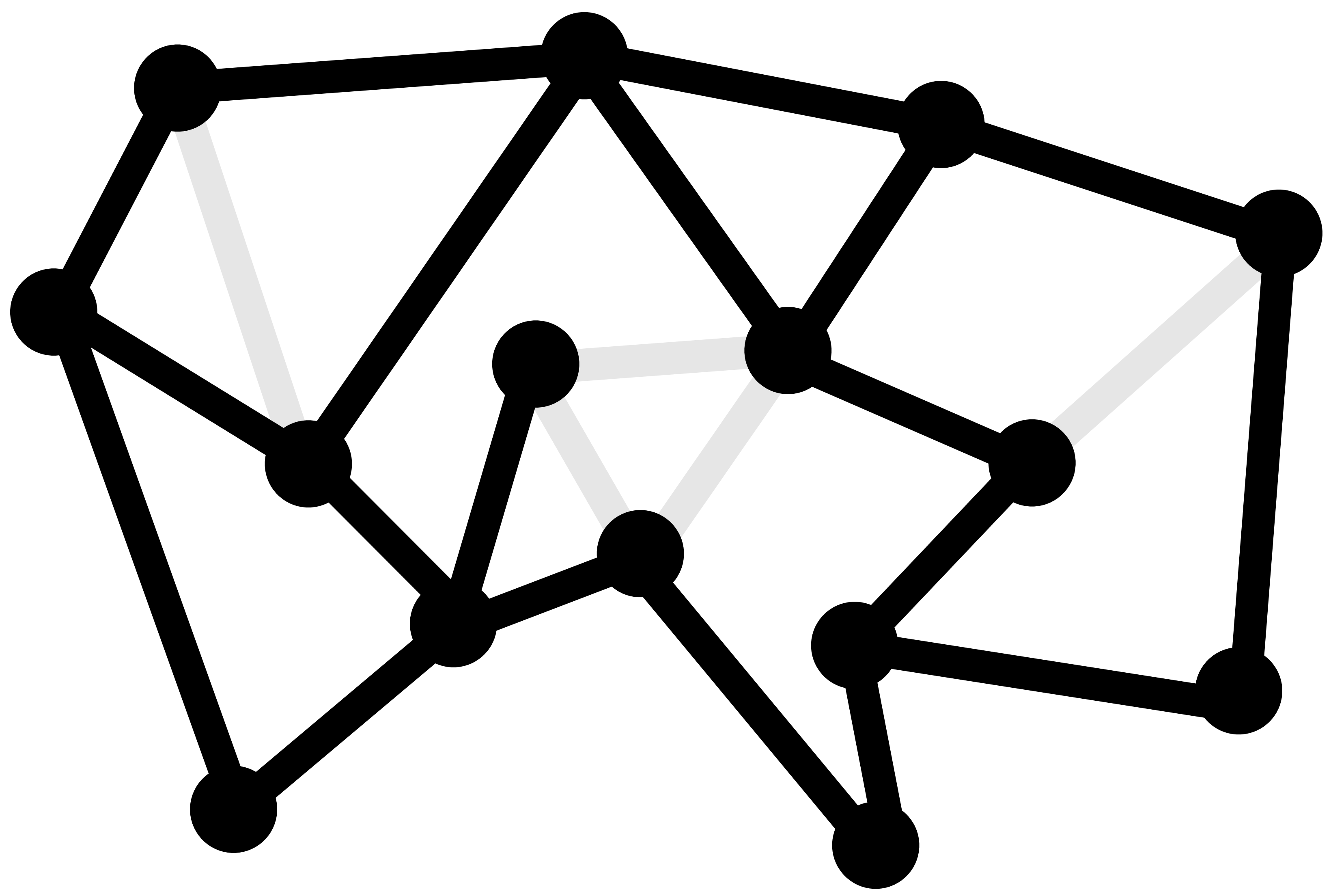
2. **Greedy Algorithm**

   suggested by observation

3. **Distortion Analysis**

4. **Size Analysis**

   by "Moore Bounds"



Theorem: every graph G has a t-spanner H w/

- **Distortion:** $t = O(\log n)$
- **Size:** $|H| = O(n)$

Observe: poly-time computable

Notable Generalizations

- Edge-weighted graphs

  Run greedy algorithm in increasing order of edge lengths

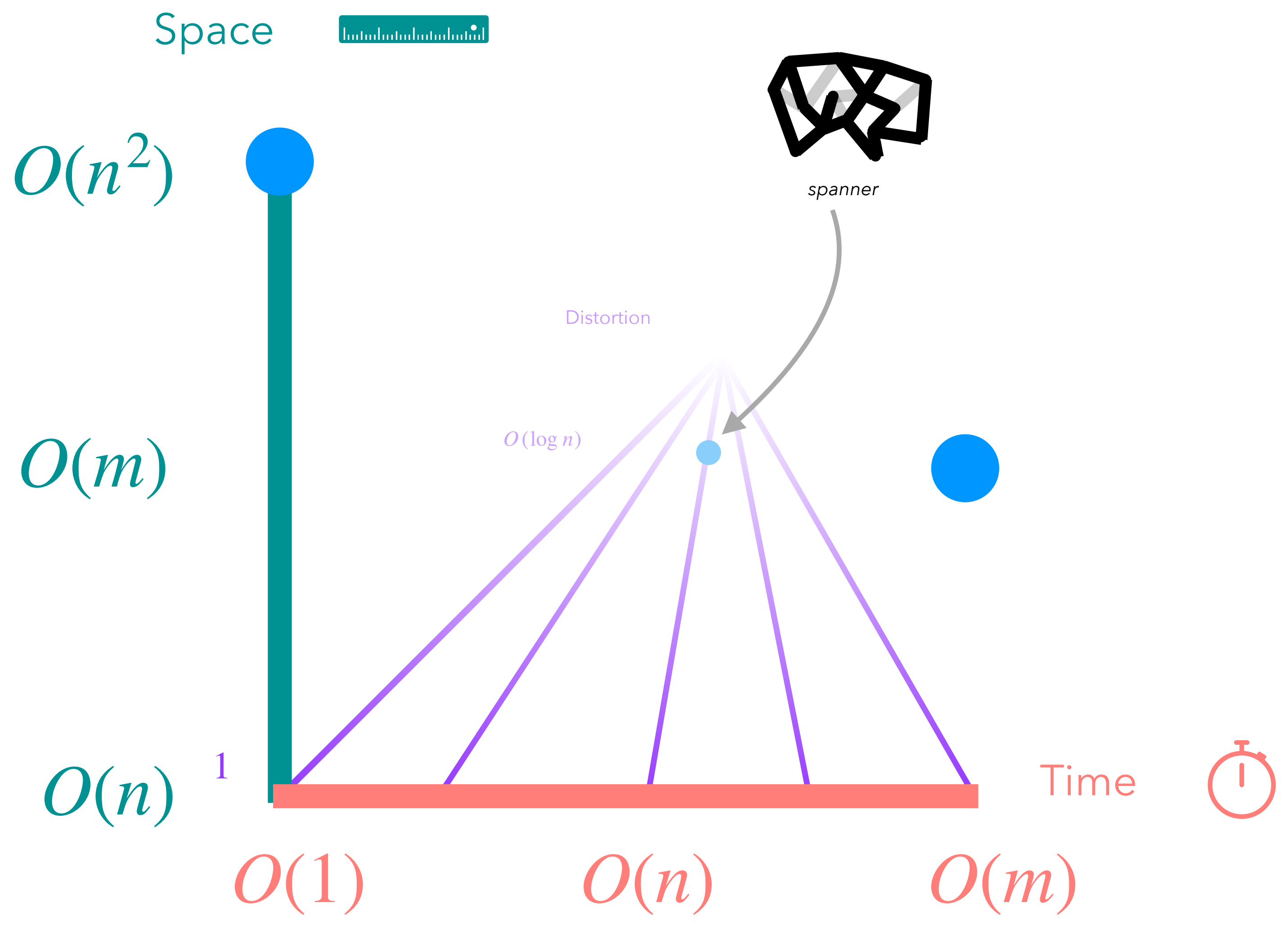
- Size-distortion tradeoff

  Just run greedy; optimal assuming "girth conjecture" of Erdős

Theorem: every graph G has a t-spanner H w/
- **Distortion:** $t = O(\log n)$
- **Size:** $|H| = O(n)$

Theorem: every graph G for every t has a t-spanner H w/
- **Distortion:** t
- **Size:** $|H| = n^{1+O\left(\frac{1}{t}\right)}$

Summary



Motivation: distance oracles

Theorem: every graph G has a t-spanner H w/

- **Distortion:** $t = O(\log n)$
- **Size:** $|H| = O(n)$

Main Result

1. **Simple Observation**

   edge spanners suffice

2. **Greedy Algorithm**

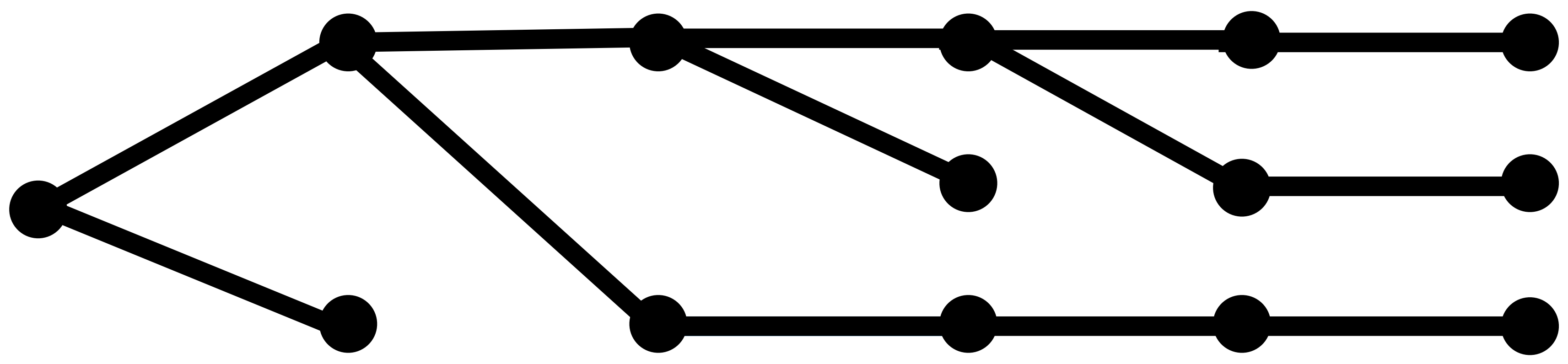
   suggested by observation

3. **Distortion Analysis**

4. **Size Analysis**

   by "Moore Bounds"

Roadmap



Key Idea: greedy output is high girth, high girth is sparse