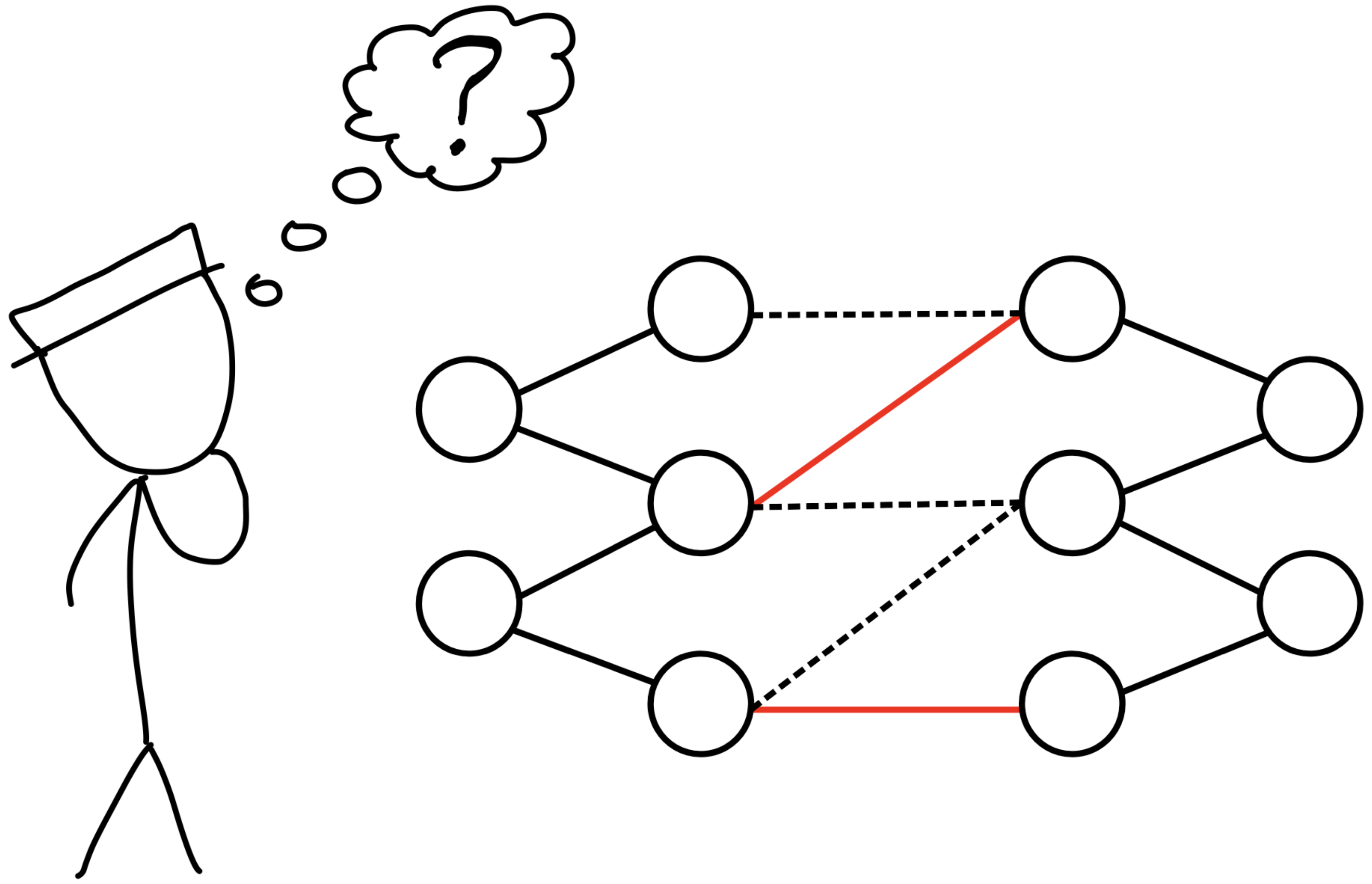
# COSTLY EXPLORATION: GRAPH CONNECTIVITY WITH NOISY QUERIES



*Evangelia Gergatsouli*

University of Wisconsin-Madison

Nov 15 2022

Based on joint work with: Dimitris Fotakis[1], Charilaos Pipis[1], Miltos Stouras[2], Christos Tzamos[3]

[1]NTUA, [2]EPFL, [3]UW Madison

# Model

*Adversary*


*Moldgraph $G$*

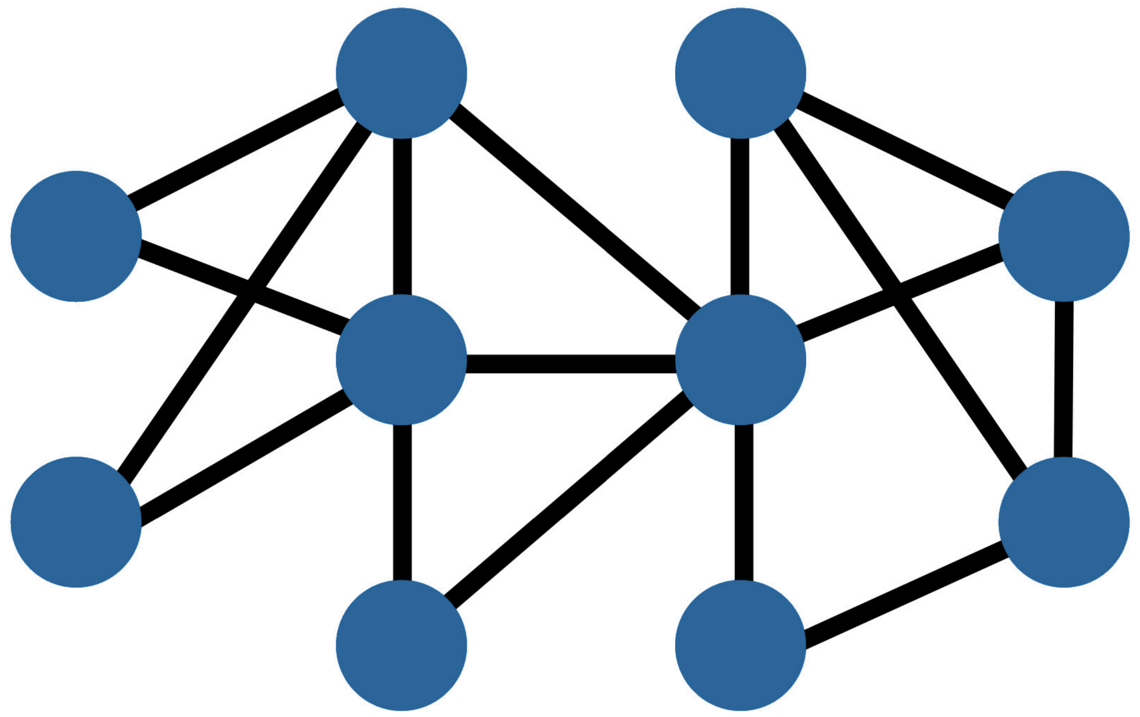
*Realized graph*

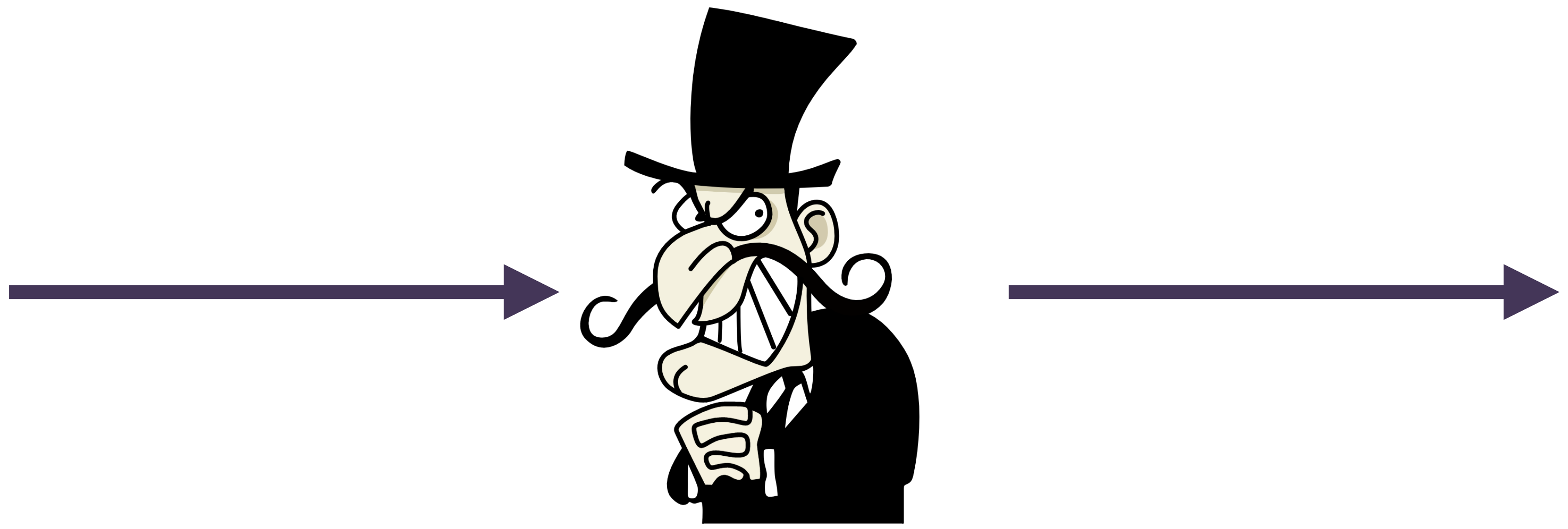
*Oracle*

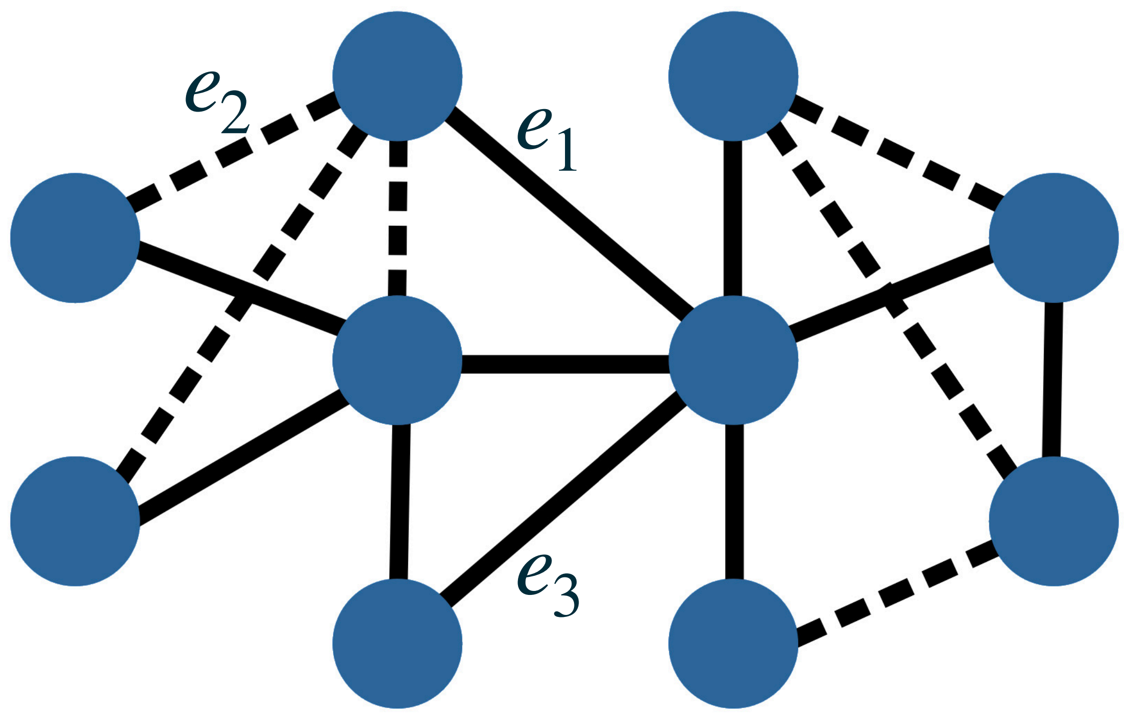
- Initial graph G
- Adversary deletes edges
- *Noisy Oracle* answers questions of the form "*Does edge $e$ exist*"?

# Model

*Adversary*

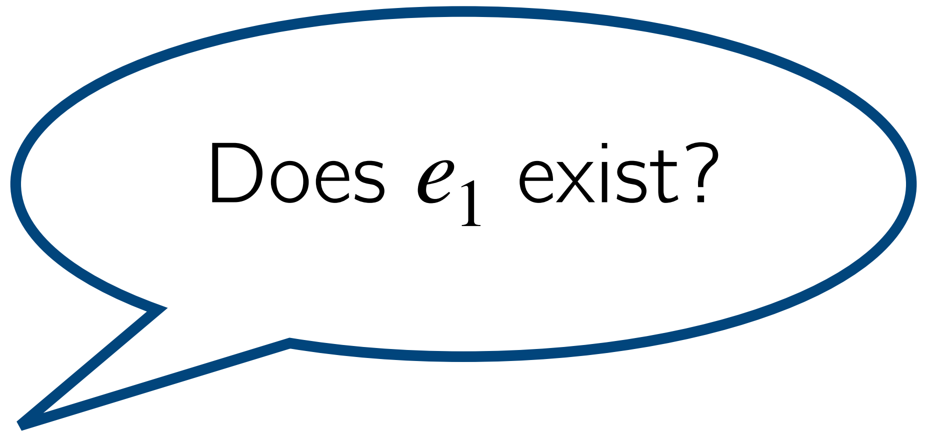







*Moldgraph $G$*

*Realized graph*

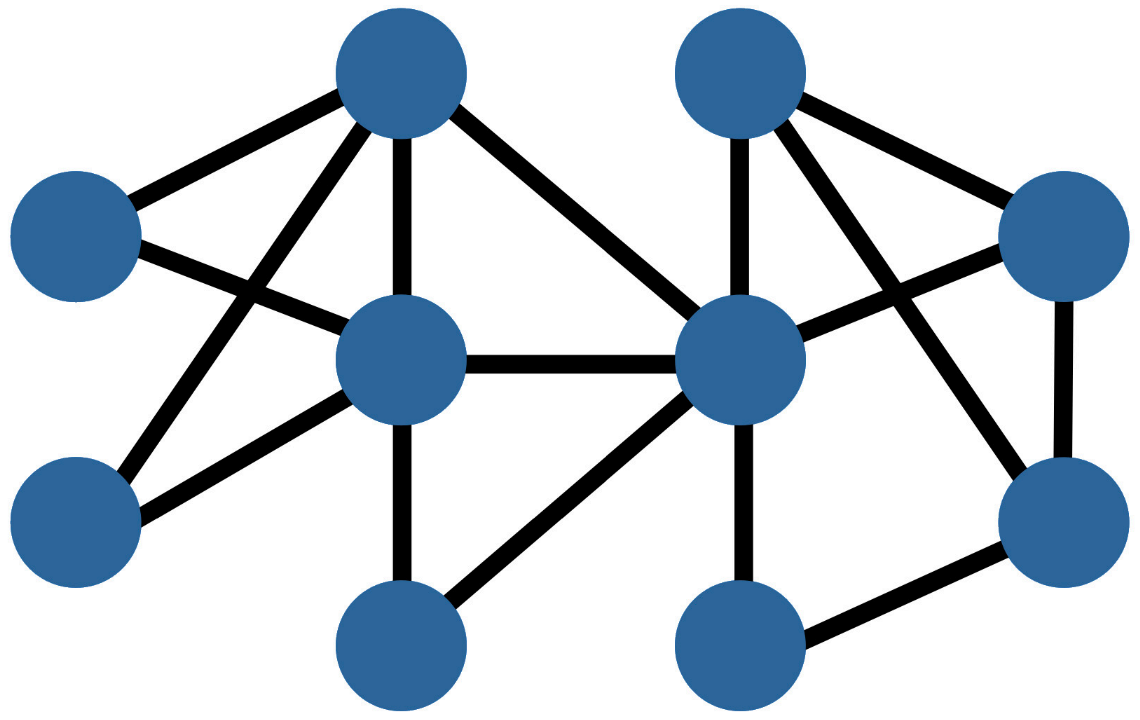



*Oracle*

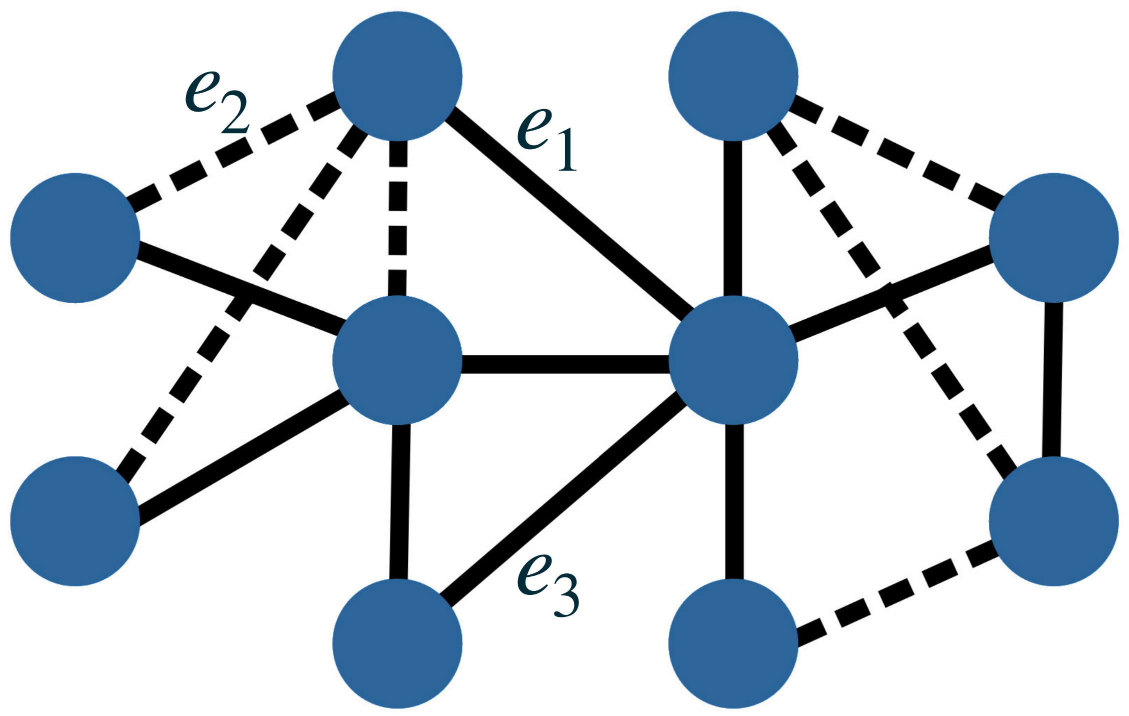
- Initial graph G
- Adversary deletes edges
- *Noisy Oracle* answers questions of the form "*Does edge $e$ exist*"?

# Model

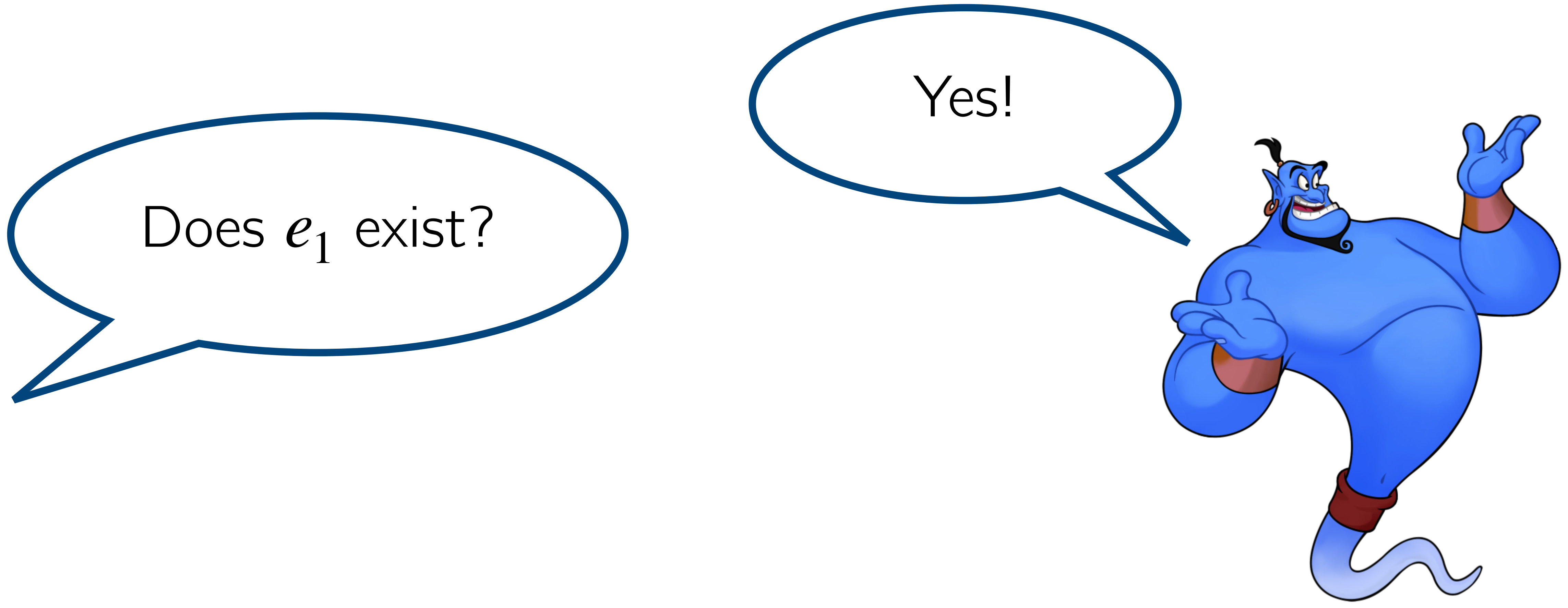
*Adversary*



*Moldgraph $G$*



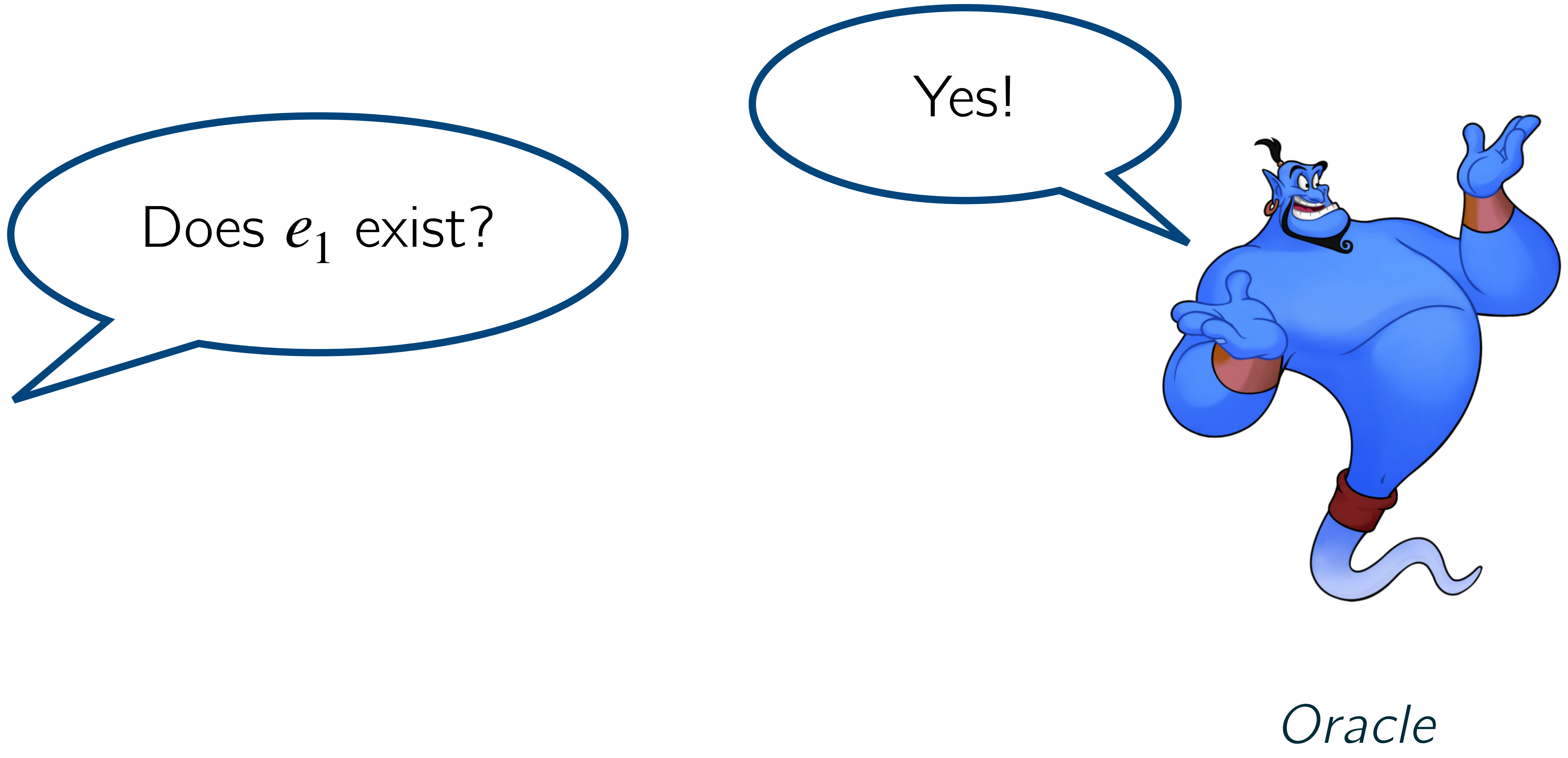

*Realized graph*




*Oracle*

- Initial graph G
- Adversary deletes edges
- *Noisy Oracle* answers questions of the form “*Does edge $e$ exist*”?

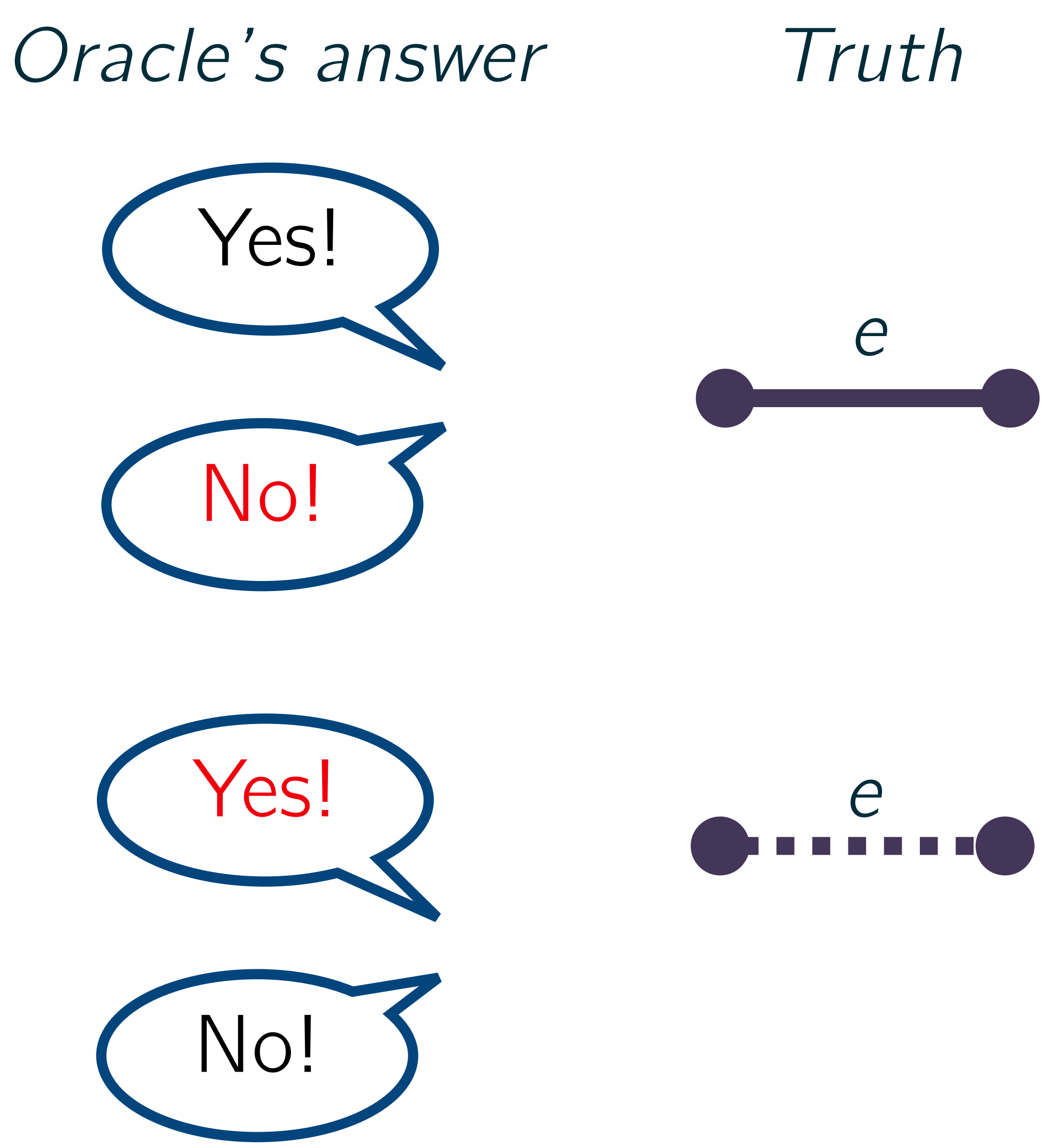
# Model


**Goal:** find a spanning tree with minimal number of queries to $\mathcal{O}$

# Error Regimes

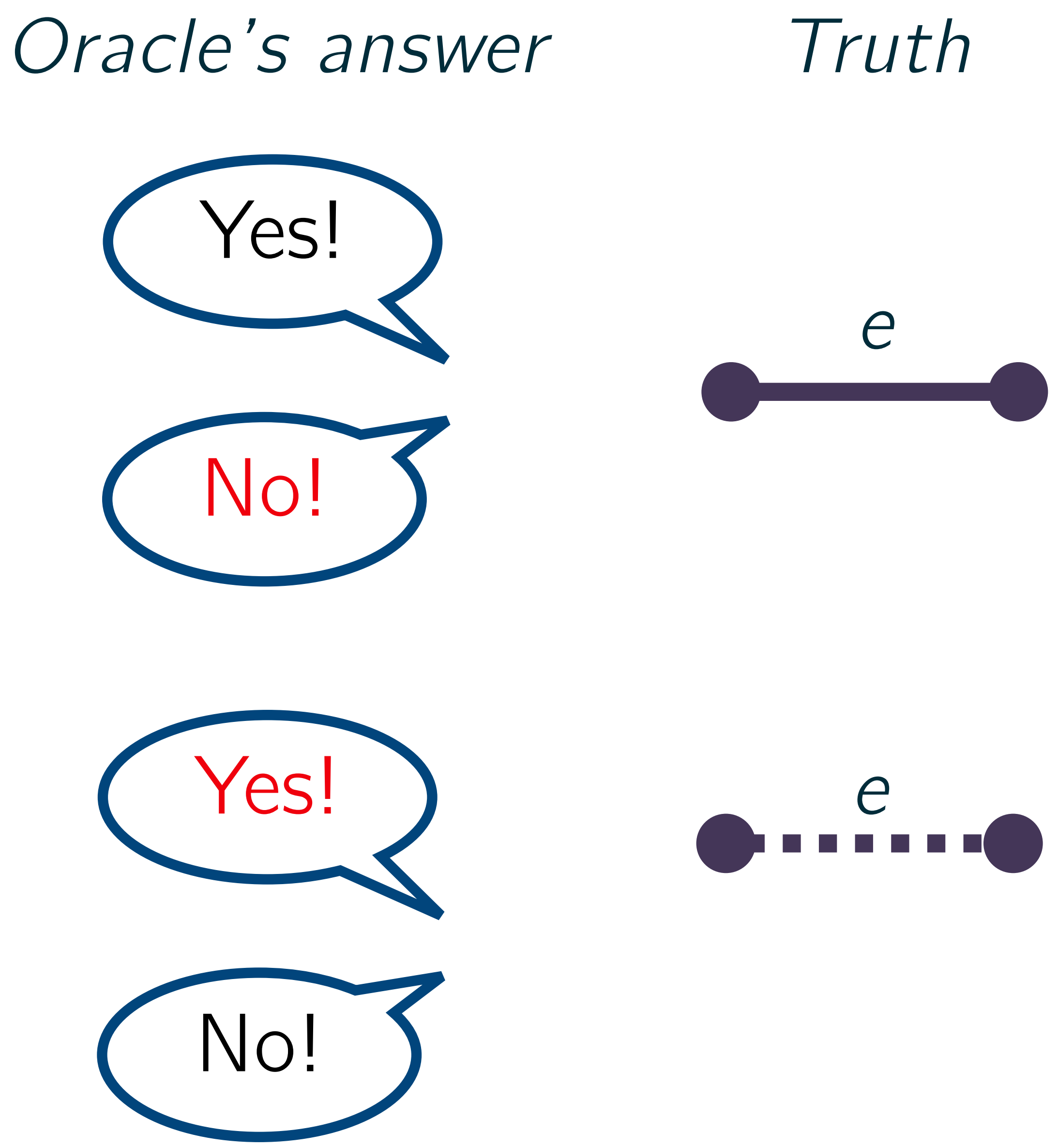
## 2-sided error



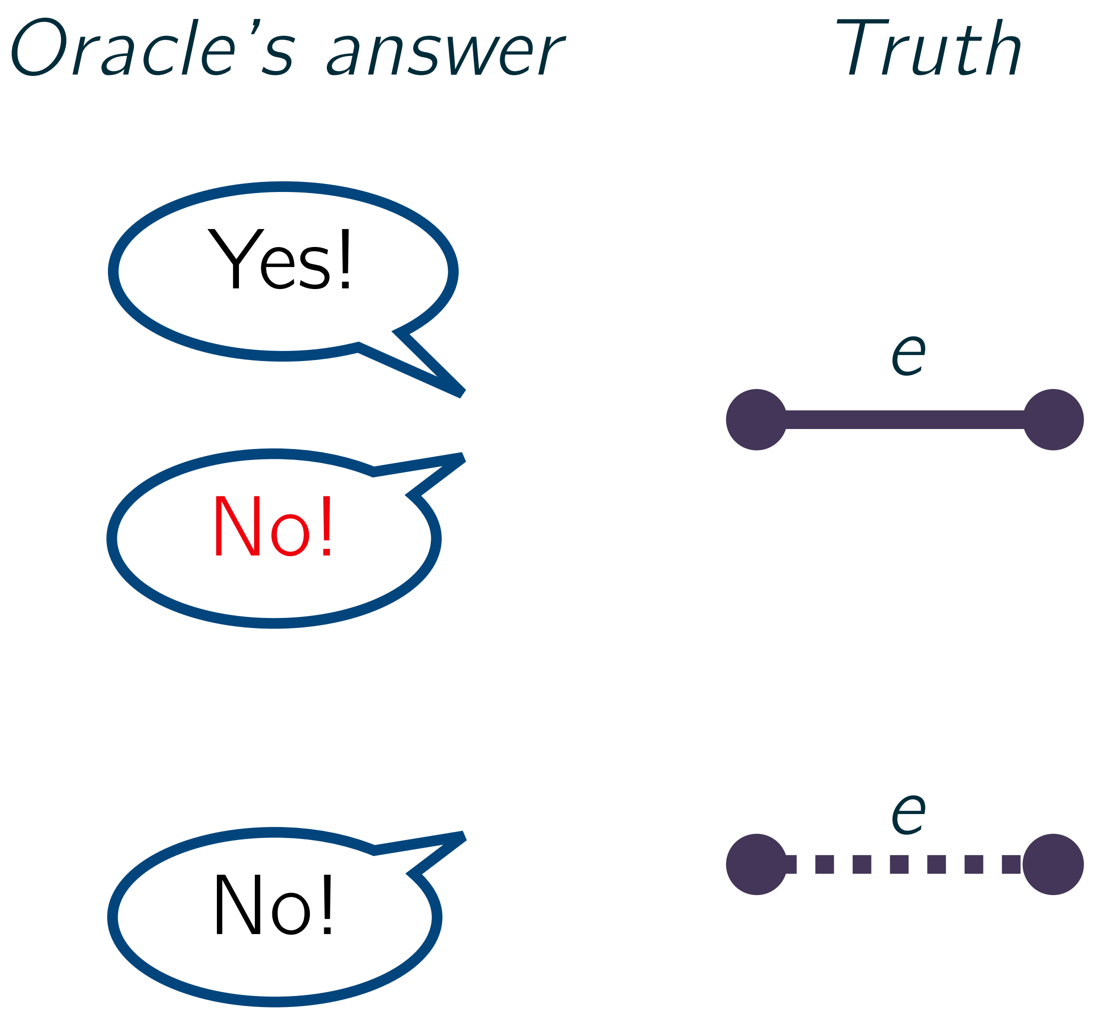
# Error Regimes

## 1-sided errors

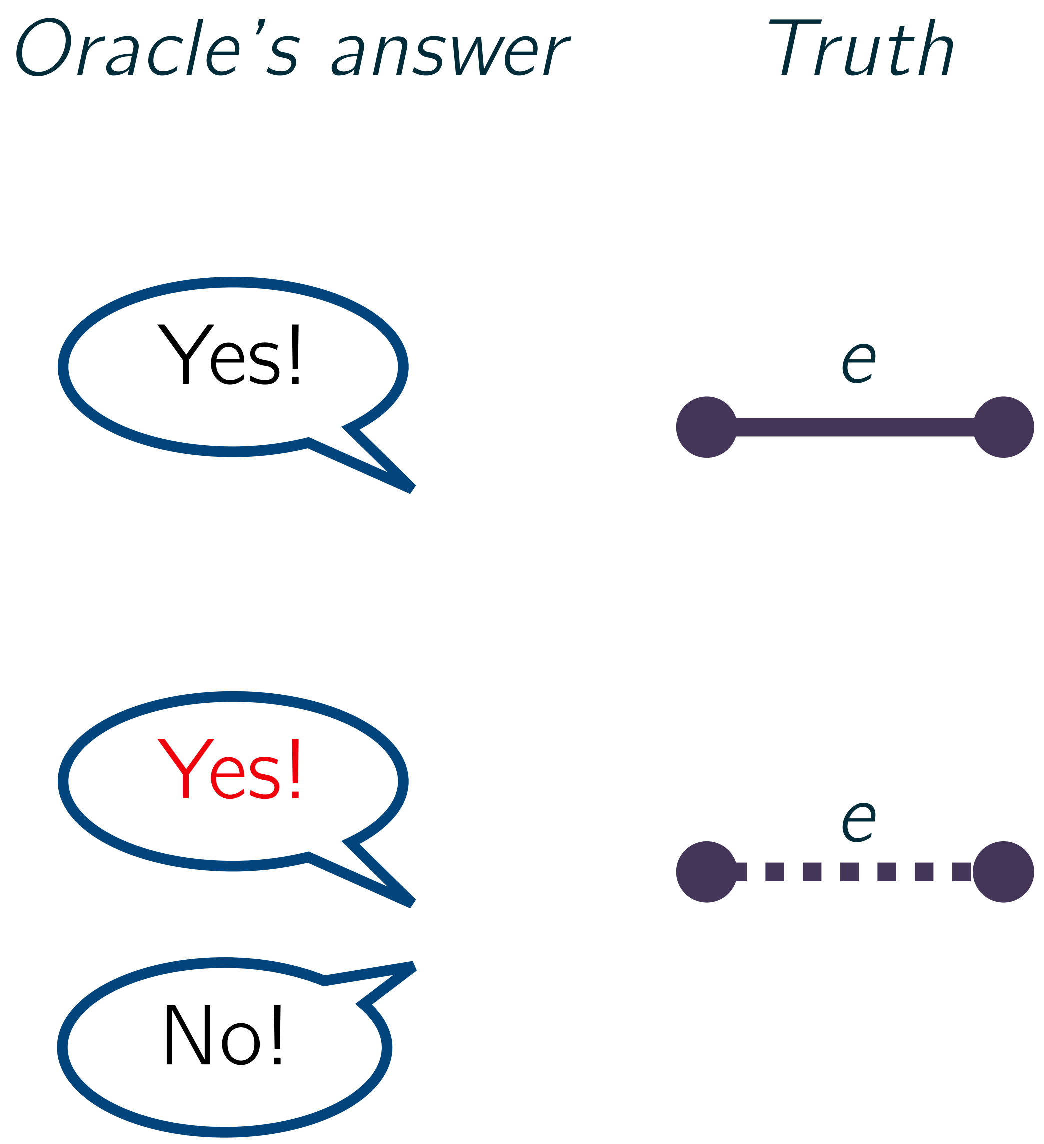
### 2-sided error




### False Negatives (FN)

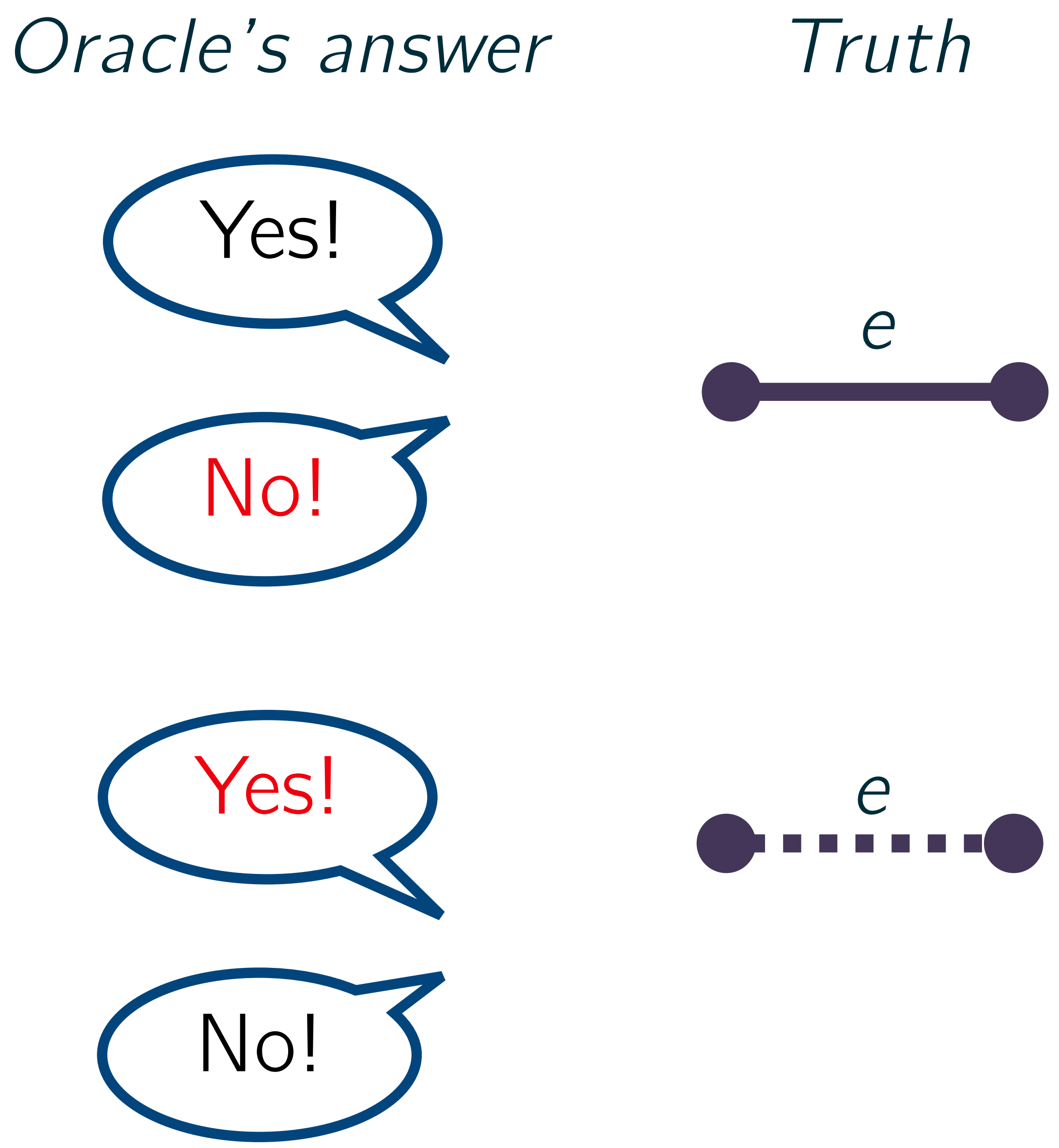



### False Positives (FP)



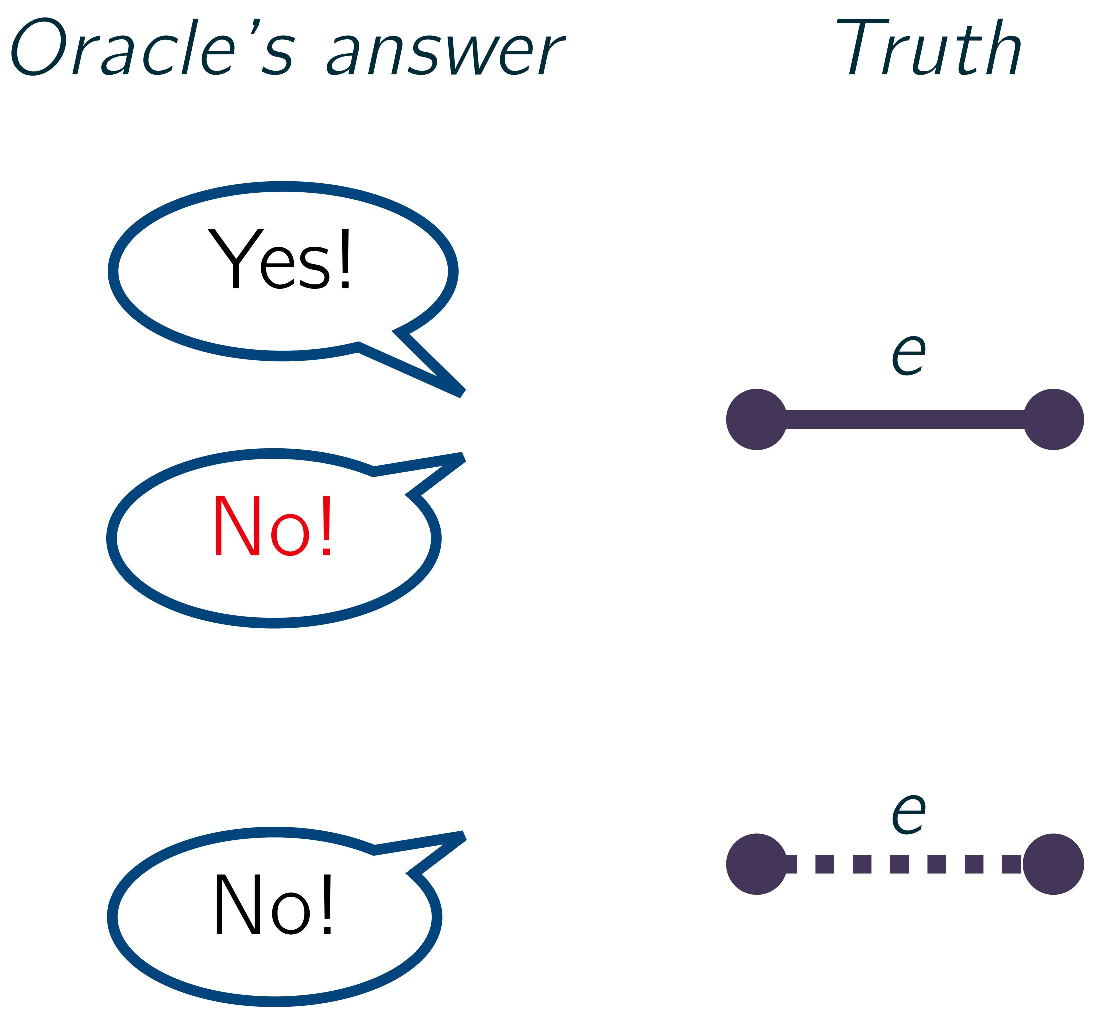
# Error Regimes

## 2-sided error

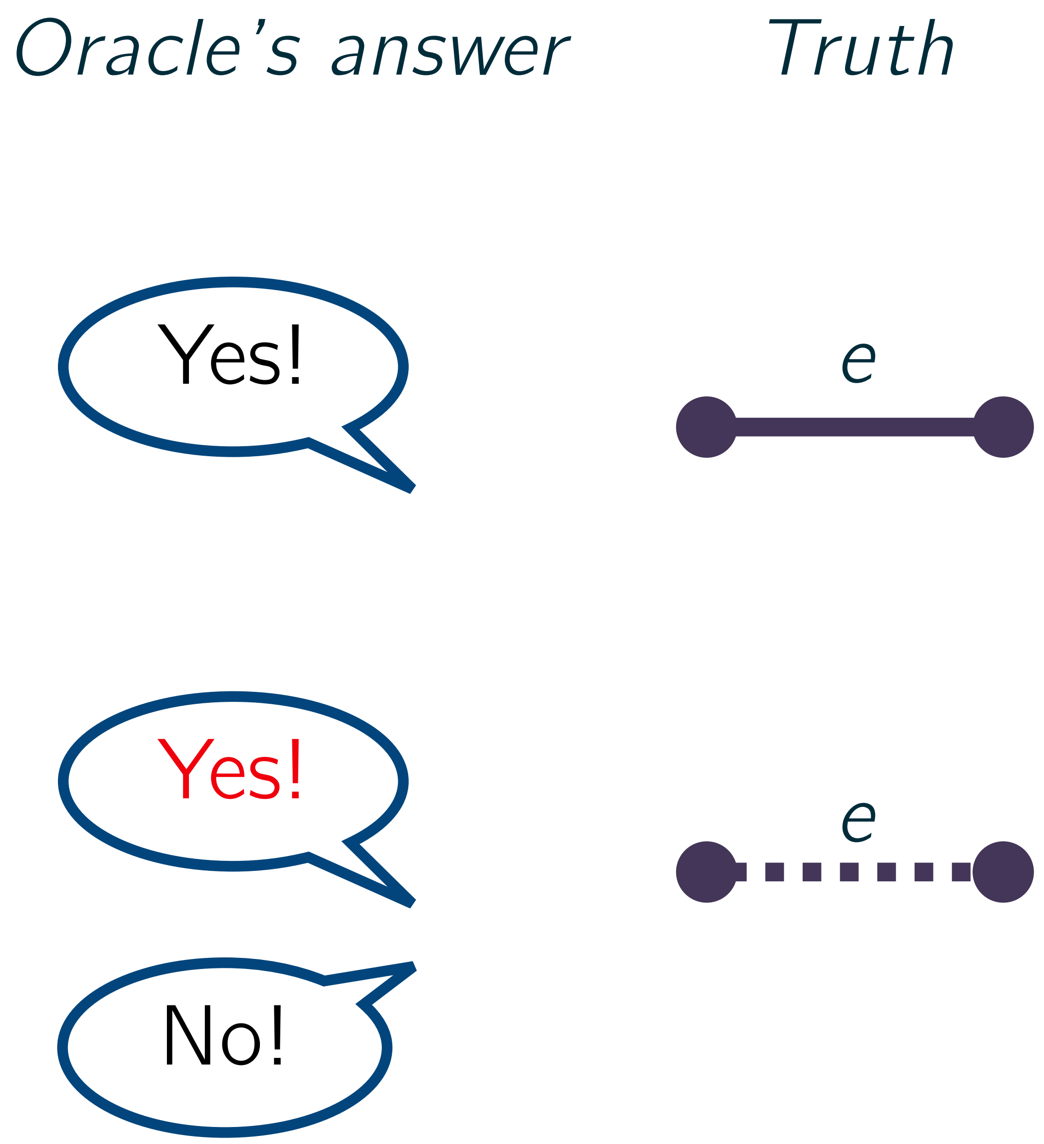



## 1-sided errors

### False Negatives (FN)




### False Positives (FP)




**Probability of error is** $p < 1/2$

# Previous Work

Assumes independence on edges' existence {

Clustering with Noisy queries
*[Mazumdar, Saha '17]*

Connectivity in Erdos Renyi graphs
*[Fu et al.'17, Fu et al. '14]*

Computing with noisy information
*[Feige et al.'94]*

MST verification, under uncertainty
*[Hoffman er al. '08, Erlebach et al. '14]*

} Only handle consistent answers to queries

# Roadmap

| | General Graphs |
|---|---|
| 2-sided error | $\Theta(m \log m)$ |
| 1-sided error (False Positives) | $O(m \log m)$ |
| 1-sided error (False Negatives) | $\Theta(m \log m)$ |

# Roadmap

| | General Graphs |
|---|---|
| **2-sided error** | $\Theta(m \log m)$ |
| **1-sided error (False Positives)** | $O(m \log m)$ |
| **1-sided error (False Negatives)** | $\Theta(m \log m)$ |

*Lower bound even for planar graphs!*

$O(m)$ *if realized graph is acyclic*

$O(\rho m)$ *if* $G$ *is* $\rho$*-sparse*

# Roadmap

| | General Graphs |
|---|---|
| **2-sided error** | $\Theta(m \log m)$ |
| **1-sided error (False Positives)** | $O(m \log m)$ |
| **1-sided error (False Negatives)** | $\Theta(m \log m)$ |

*Lower bound even for planar graphs!*

*$O(m)$ if realized graph is acyclic*

*$O(\rho m)$ if $G$ is $\rho$-sparse*

***Special cases also tight***

# 2-sided Error - Algorithm

**Algorithm:**

- Query every edge of moldgraph $\log m$ times
- Every $e \in E$ with more “Yes” is treated as realized
- **If** graph connected
  - Output graph
- **Else**
  - Output any spanning tree of moldgraph

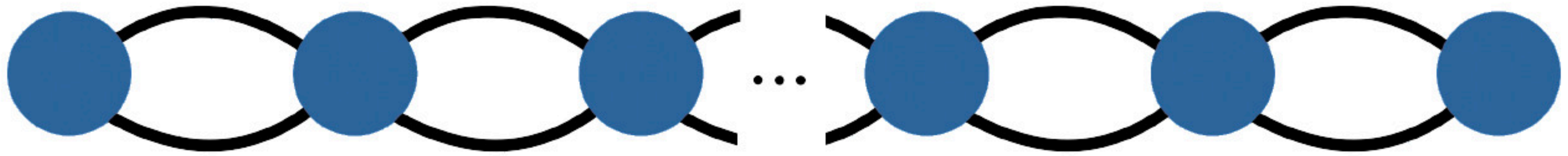
# 2-sided Error - Algorithm

**Algorithm:**

- Query every edge of moldgraph $\log m$ times
- Every $e \in E$ with more "Yes" is treated as realized
- **If** graph connected
  - Output graph
- **Else**
  - Output any spanning tree of moldgraph

Algorithm finds a spanning tree w.h.p. using $O(m \log m)$ queries

# 2-sided Error - Lower Bound

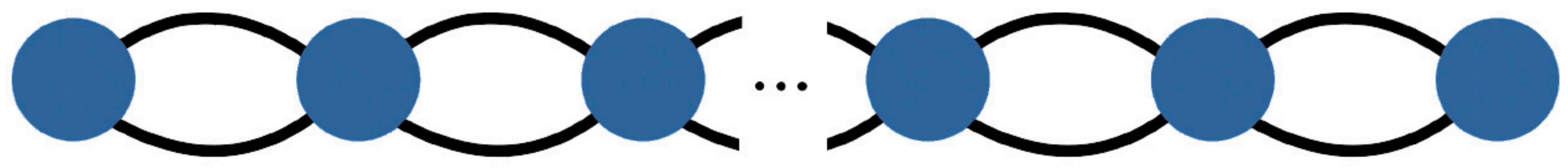



*Moldgraph* $G$

**Observations:**

- Each pair is independent of others (each pair is a cut)
- If less than $\log m$ queries per pair → error w.h.p.

# 2-sided Error - Lower Bound



*Moldgraph $G$*

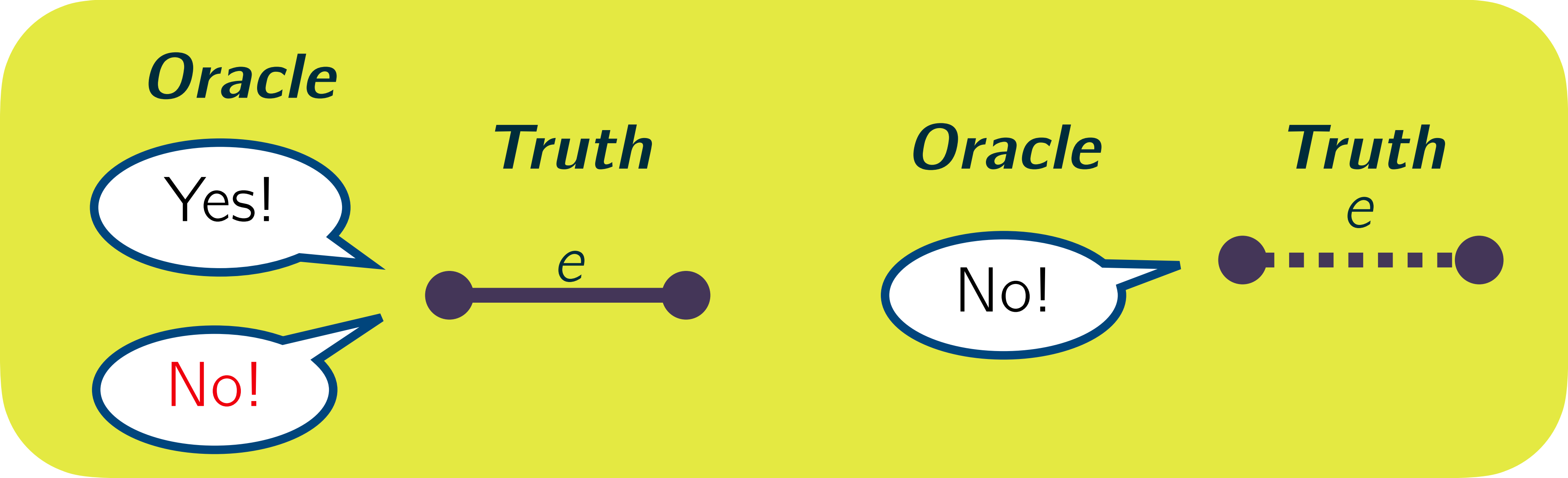
No algorithm can do better than $\Omega(m \log m)$ queries

**Observations:**

- Each pair is independent of others (each pair is a cut)
- If less than $\log m$ queries per pair $\rightarrow$ error w.h.p.

# 1-sided, False Negatives

***Recall:***



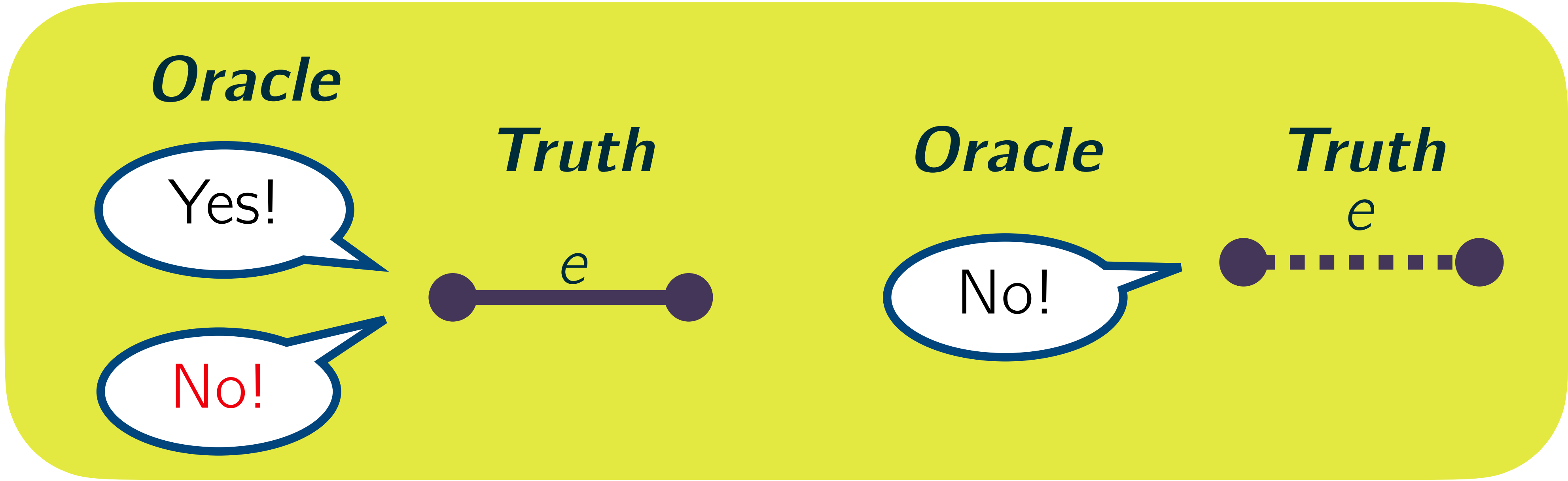

**Observations:**

- Every realized edge, will eventually give a "Yes" answer
- Every cut of $G$ contains at least one realized edge
  - Cuts are small in sparse graphs

$\rho$-sparse:
edges $\leq \rho \cdot$ nodes

# 1-sided, False Negatives

***Recall:***




**Observations:**

- Every realized edge, will eventually give a "Yes" answer
- Every cut of $G$ contains at least one realized edge
  - Cuts are small in sparse graphs

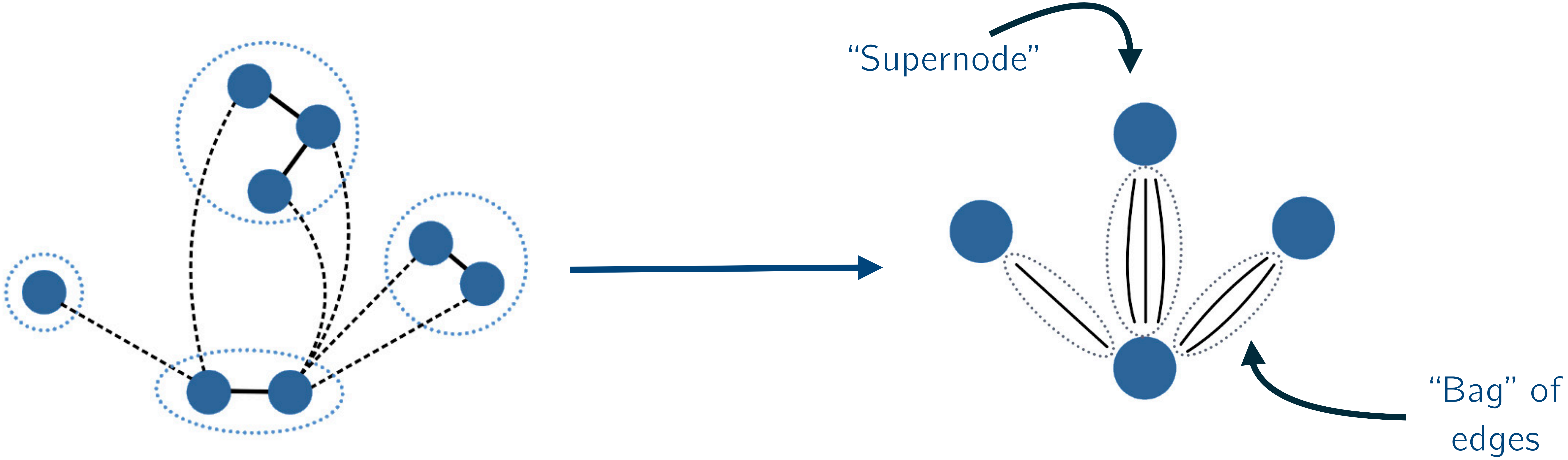
$\rho$-sparse:
edges $\leq \rho \cdot$ nodes

***Idea 1***: *can naively ask about every edge*

***Idea 2***: *only need to find one edge per cut*

# 1-sided, False Negatives

***Idea 2**: only need to find one edge per cut*



# 1-sided, False Negatives

**Naive algorithm:**

- Round-robin query all edges
  until there is spanning tree

+

**Sparse graphs algorithm:**

- **While** ( graph has more than 1 nodes )
  - Find min degree node $u \in V$
  - **For** every "Bag of edges" neighboring $u$
    - Select random edge to query
  - Find realized edge $e$ and contract it

# 1-sided, False Negatives

**Naive algorithm:**

- Round-robin query all edges
  until there is spanning tree

+

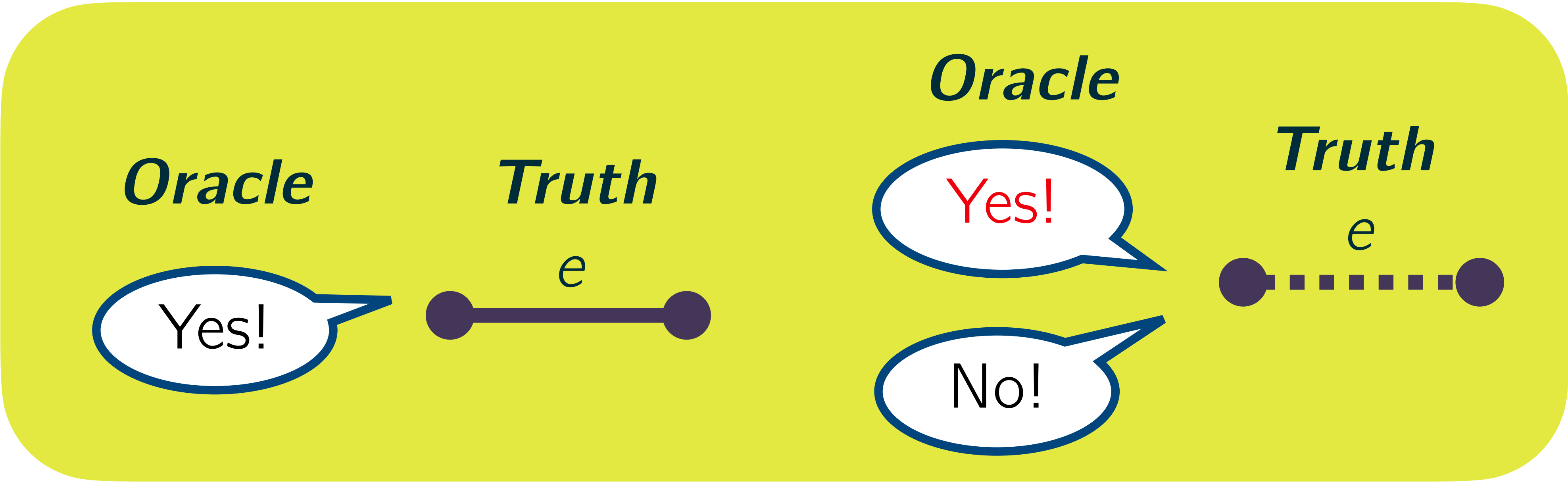
**Sparse graphs algorithm:**

- **While** ( graph has more than 1 nodes )
  - Find min degree node $u \in V$
  - **For** every "Bag of edges" neighboring $u$
    - Select random edge to query
  - Find realized edge $e$ and contract it

Algorithm finds a sp. tree using $O(m \log m)$ queries.
If graph is $\rho$-sparse then uses $O(\rho m)$ queries

# 1-sided, False Positives

***Recall:***




**Observations:**

- Every non-realized edge, will eventually give a “No” answer → ***Idea 1***: *can naively ask about every edge*

Dual graph:

- Node per face of $G$
- Edge if faces in $G$ are separated by edge

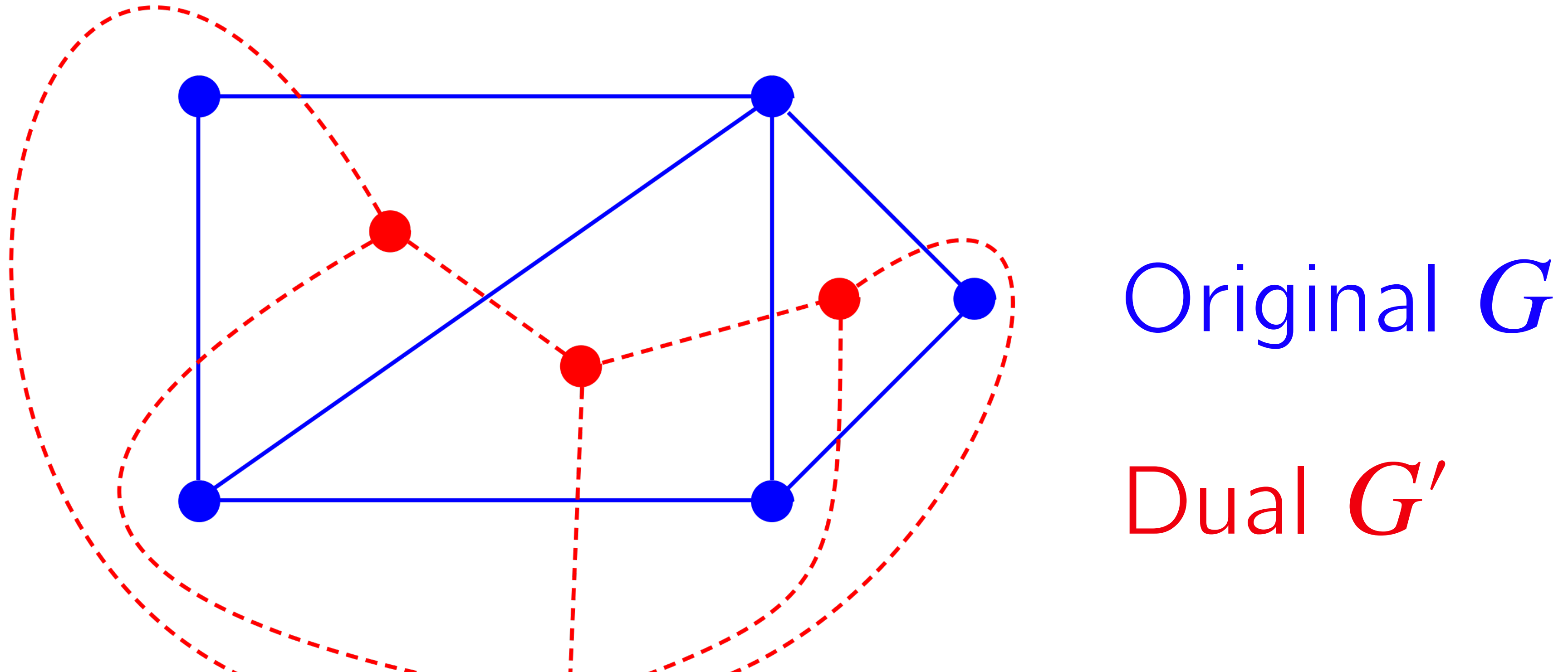
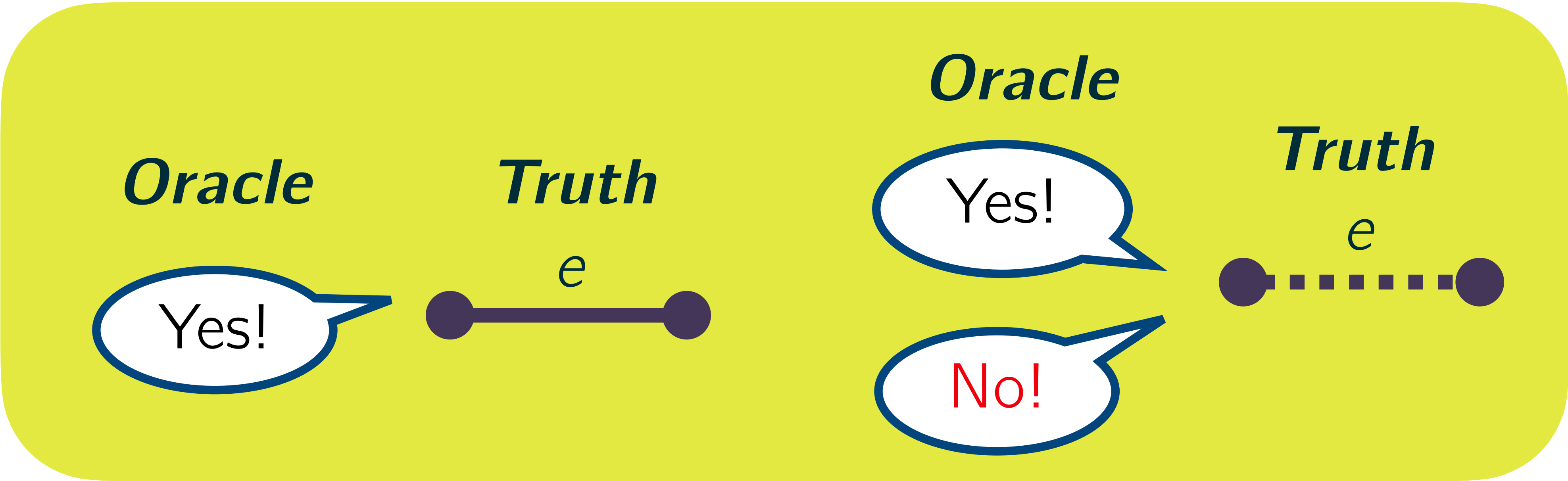



Image source: wikipedia

# 1-sided, False Positives

***Recall:***




**Observations:**

- Every non-realized edge, will eventually give a "No" answer → ***Idea 1***: *can naively ask about every edge*

- **If** realized graph is tree & $G$ is planar
  - Every cycle of $G$ contains at least one **non**-realized edge → ***Idea 2***: *can use FN algorithm in dual graph*
  - Cycles in initial graph $\equiv$ Cuts in the dual graph

# 1-sided, False Positives

**Naive algorithm:**

- **While** there's no tree:
  - Ask about next edge $e$
  - **If** answer is "*No*" mark as non-realized
  - **Else** mark as realized

Round robin

+

**Acyclic graph algorithm:**

- Construct the dual $G'$ of initial graph
- Run FN algorithm on dual $G'$
- Return complement of edges

# 1-sided, False Positives

**Naive algorithm:**

- **While** there's no tree:
  - Ask about next edge $e$ (Round robin)
  - **If** answer is "*No*" mark as non-realized
  - **Else** mark as realized

+

**Acyclic graph algorithm:**

- Construct the dual $G'$ of initial graph
- Run FN algorithm on dual $G'$
- Return complement of edges

Algorithm finds a sp. tree w.h.p. with $O(m \log m)$ queries.
If $G$ planar and realized graph is tree then $O(m)$ queries

# Summary

| | General Graphs |
|---|---|
| **2-sided error** | $\Theta(m \log m)$ |
| **1-sided error (False Positives)** | $O(m \log m)$ |
| **1-sided error (False Negatives)** | $\Theta(m \log m)$ |

# Summary

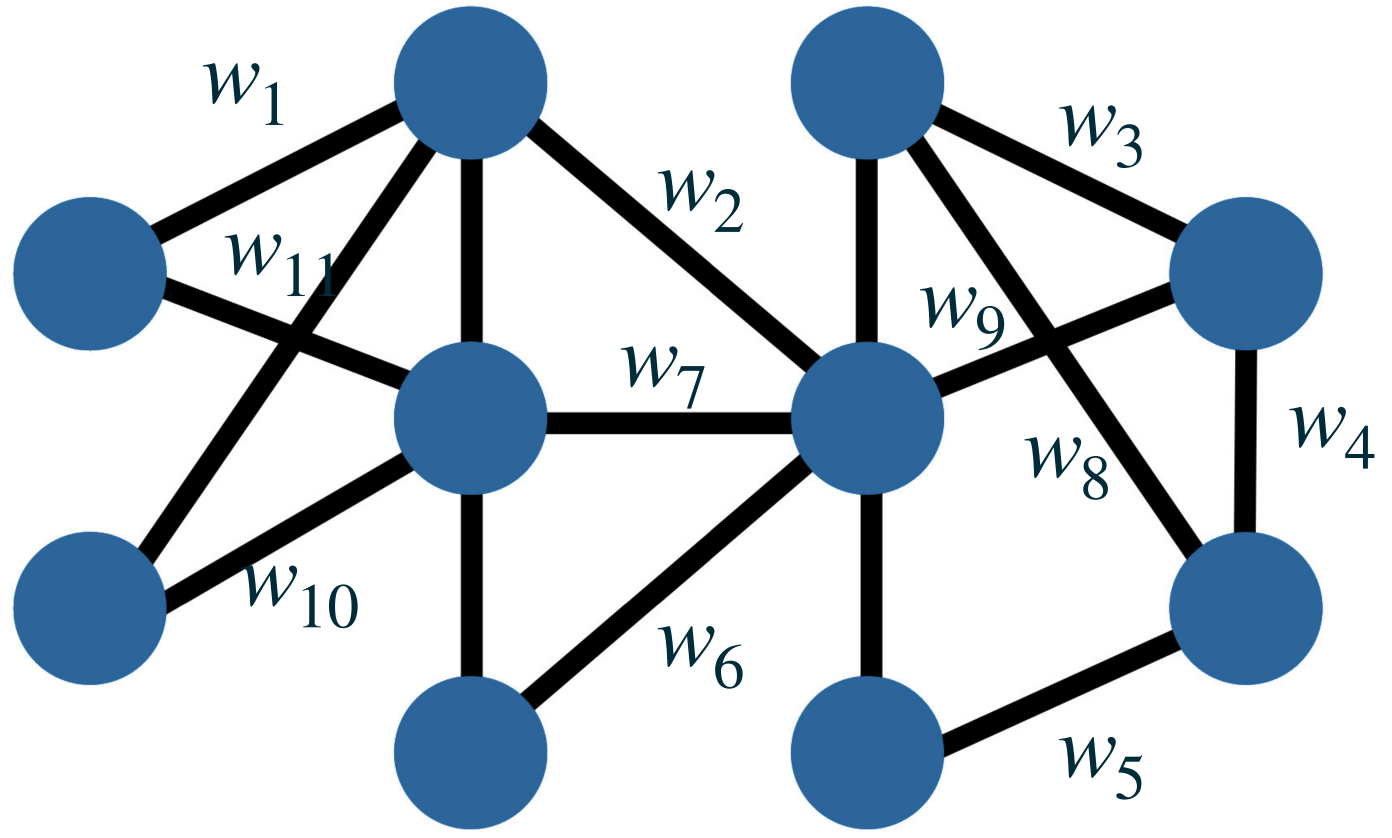
| | General Graphs |
|---|---|
| 2-sided error | $\Theta(m \log m)$ |
| 1-sided error (False Positives) | $O(m \log m)$ |
| 1-sided error (False Negatives) | $\Theta(m \log m)$ |

*Lower bound even for planar graphs!* (2-sided error)

*$O(m)$ if realized graph is acyclic* (1-sided error, False Positives)

*$O(\rho m)$ if $G$ is $\rho$-sparse* (1-sided error, False Negatives)

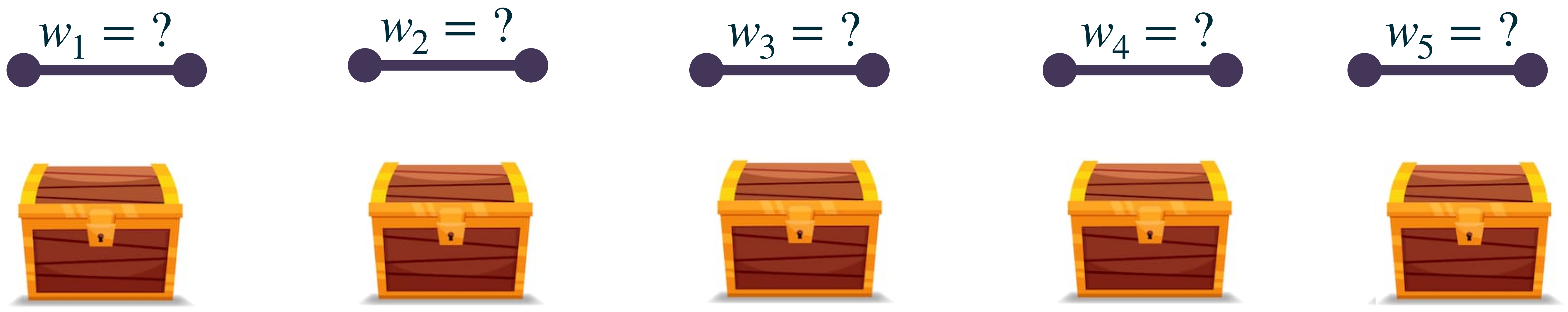
# A More General Problem

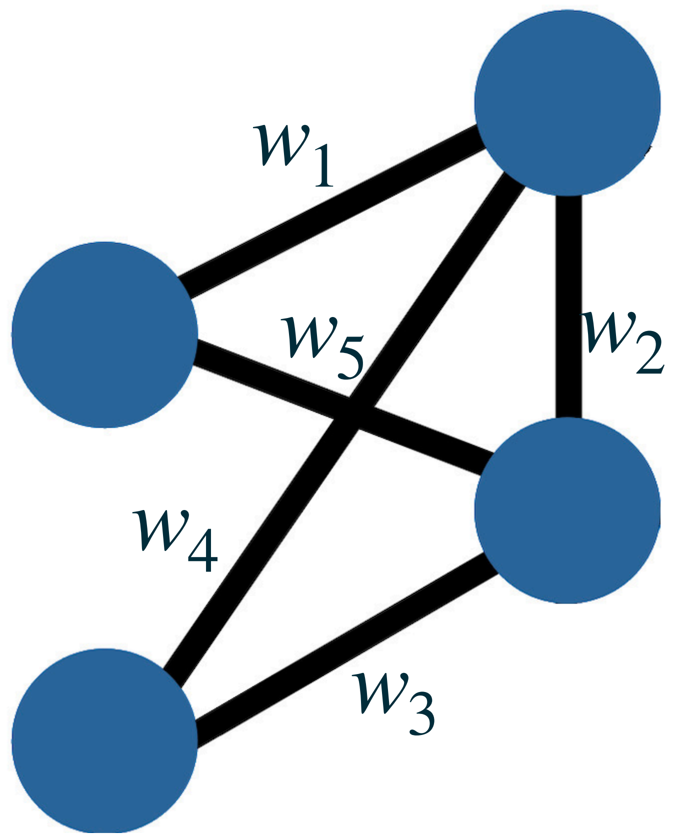



- Find MST on weighted graph
- Pay to learn weights (explore alternatives)
- Stop anytime and take best so far

# A More General Problem

Find the best alternative, with costly information!

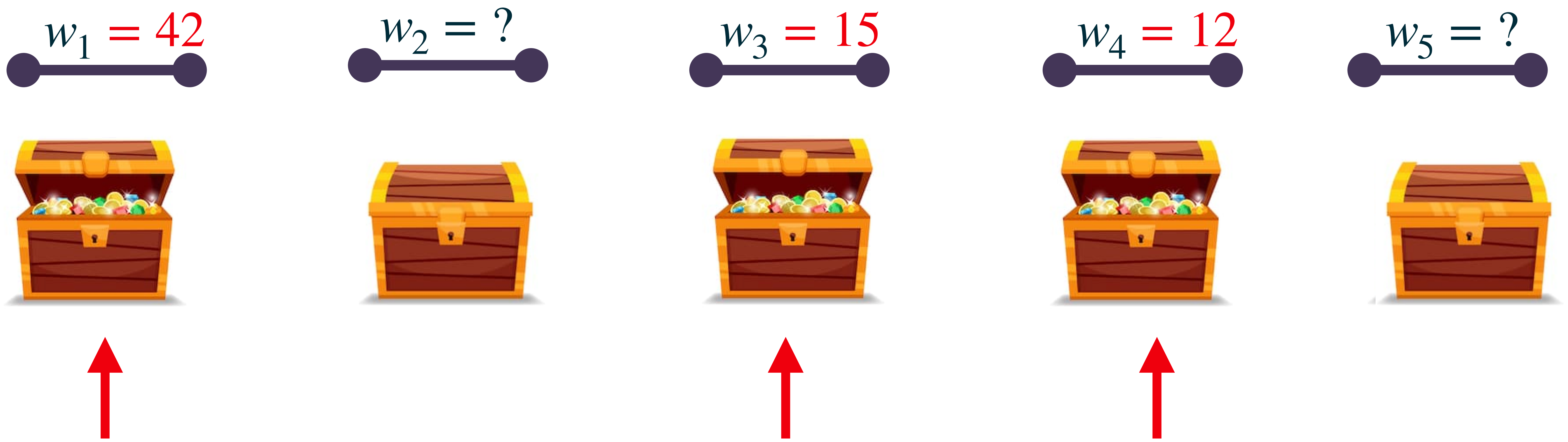


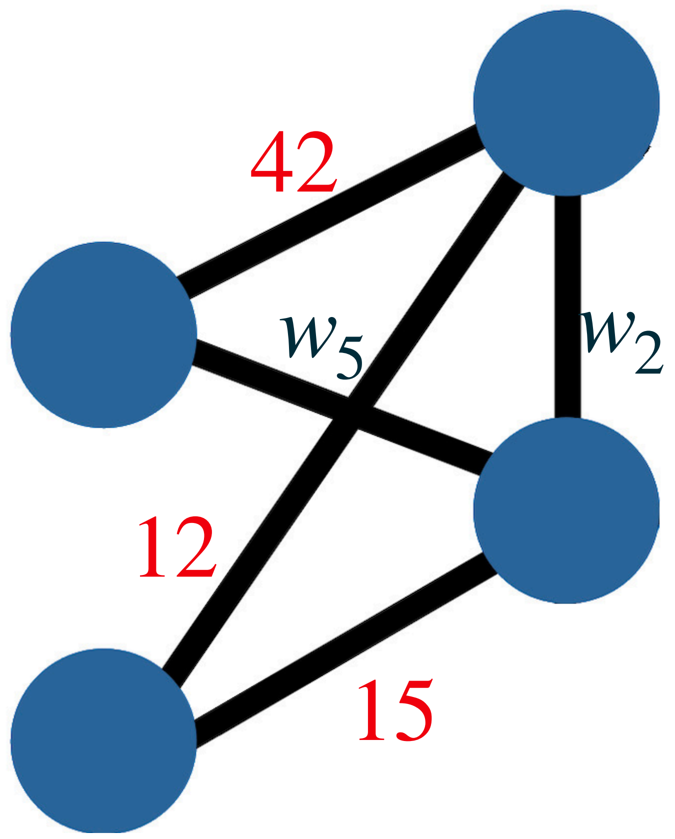




- Pay to learn weights (explore alternatives)
- **Constraint**: edges should be a tree

# A More General Problem

Find the best alternative, with costly information!

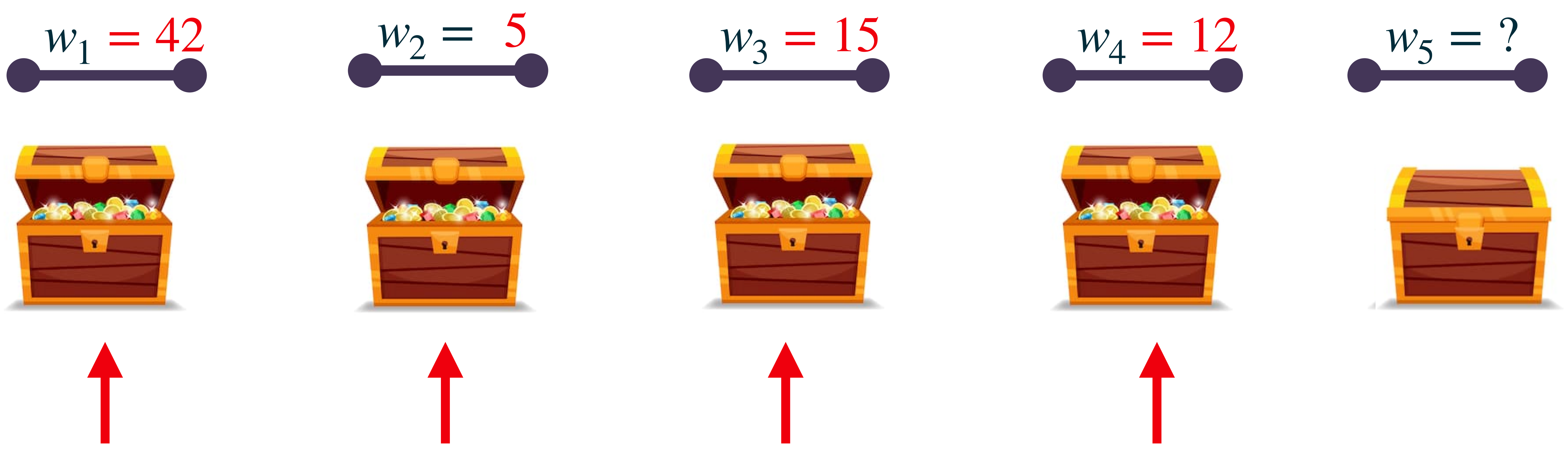


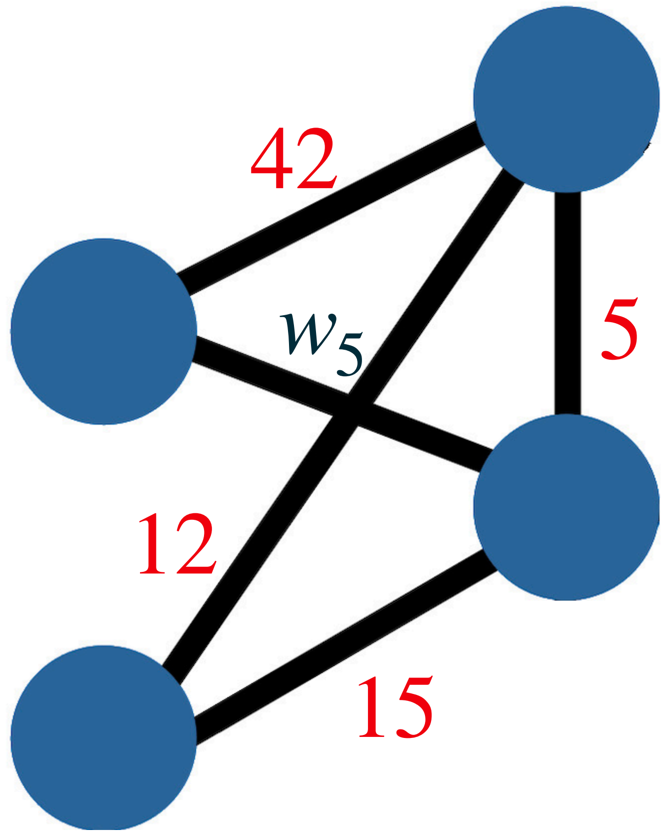




- Pay to learn weights (explore alternatives)
- **Constraint**: edges should be a tree

# A More General Problem

Find the best alternative, with costly information!




- Pay to learn weights (explore alternatives)
- **Constraint**: edges should be a tree

# Pandora's Box

Find the best alternative, with costly information!


- Information is not free
- Explore alternatives (open boxes)
- Stop anytime and take best so far

} **Strategy**

# Pandora's Box

Find the best alternative, with costly information!

$42 ?? $15 $12 ??

Opening cost: 3
Final option: box 4
**Total cost**: 12+3

- Information is not free
- Explore alternatives (open boxes)
- Stop anytime and take best so far

} **Strategy**

# Pandora's Box

Find the best alternative, with costly information!


Opening cost: 3
Final option: box 4
**Total cost**: 12+3

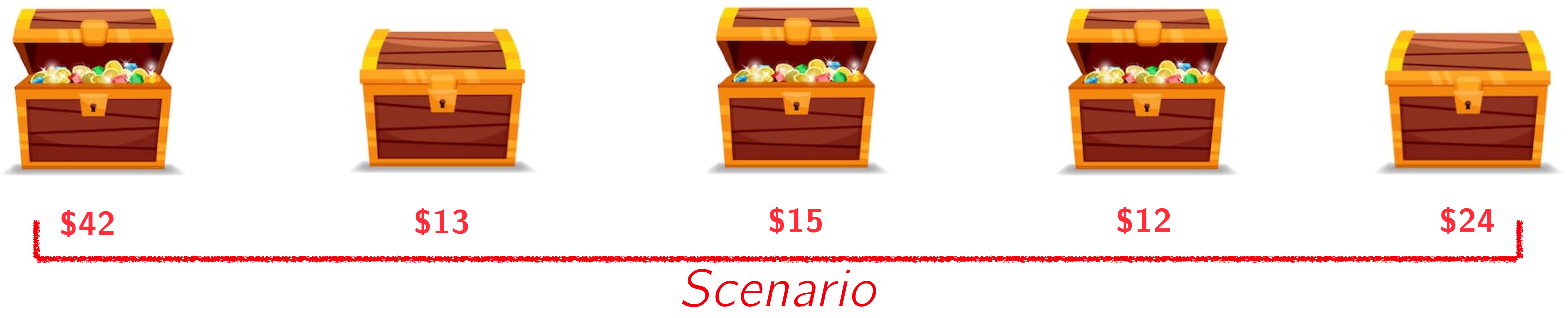
- Information is not free
- Explore alternatives (open boxes)
- Stop anytime and take best so far

} **Strategy**

**Goal:**
**Find minimum cost strategy**

# Pandora's Box

Find the best alternative, with costly information!




Opening cost: 3
Final option: box 4
**Total cost**: 12+3

- Information is not free
- Explore alternatives (open boxes)
- Stop anytime and take best so far

} **Strategy**

**Goal:**
**Find minimum cost strategy**

# Previous work

Weitzman's algorithm gives the optimal! [Weitz 1979]

**Algorithm:**

- Assign an *expected gain* index[1] to every box
- Search boxes in order of index until:
  - Current price better than index of next box

[1]Gittins index/reservation value

# Previous work

Weitzman's algorithm gives the optimal! [Weitz 1979]

**Algorithm:**

- Assign an *expected gain* index[1] to every box
- Search boxes in order of index until:
  - Current price better than index of next box

Crucial assumption: distributions are **independent**!

What about **correlation**?

[1]Gittins index/reservation value

# Our work

**Arbitrarily correlated distributions $\mathscr{D}$**

sample access to $\mathscr{D}$
Can sample scenarios

explicitly given $\mathscr{D}$
Scenarios given explicitly

solve problem over $T$ rounds
Scenarios arrive online

solve over $T$ rounds, with context
Scenarios arrive online

# Our work

**Arbitrarily correlated distributions $\mathscr{D}$**

- sample access to $\mathscr{D}$
  Can sample scenarios
  → Constant competitive choosing k, $\log$(rank) for matroids
- explicitly given $\mathscr{D}$
  Scenarios given explicitly
  → Reduction to *Optimal Decision Tree* types of problems
- solve problem over $T$ rounds
  Scenarios arrive online
- solve over $T$ rounds, with context
  Scenarios arrive online

→ Obtain **no-regret** algorithms

# Our work

**Arbitrarily correlated distributions $\mathscr{D}$**

- sample access to $\mathscr{D}$
  Can sample scenarios
  [Chawla, G, Teng, Tzamos, Zhang FOCS '20]
- explicitly given $\mathscr{D}$
  Scenarios given explicitly
  [Chawla, G, McMahan, Tzamos ArXiv '21]
- solve problem over $T$ rounds
  Scenarios arrive online
  [G, Tzamos ICML '22]
- solve over $T$ rounds, with context
  Scenarios arrive online
  [Atsidakou, Caramanis, G, Papadigenopoulos, Tzamos ArXiv '22]

# Conclusion

**Thank you!**