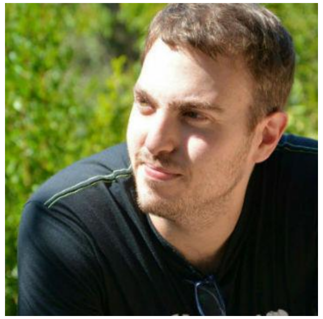
Erasure-Resilient Sublinear-Time Graph Algorithms



Amit Levi

UNIVERSITY OF WATERLOO
DAVID R. CHERITON SCHOOL OF COMPUTER SCIENCE

Ramesh Krishnan S. Pallavoor

BOSTON UNIVERSITY

Sofya Raskhodnikova

BOSTON UNIVERSITY

<u>Nithin Varma</u>

אוניברסיטת חיפה
University of Haifa
جامعة حيفا

Goal

Model and investigate sublinear-time algorithms that run on graphs with incomplete information

Sublinear-time algorithms for graphs

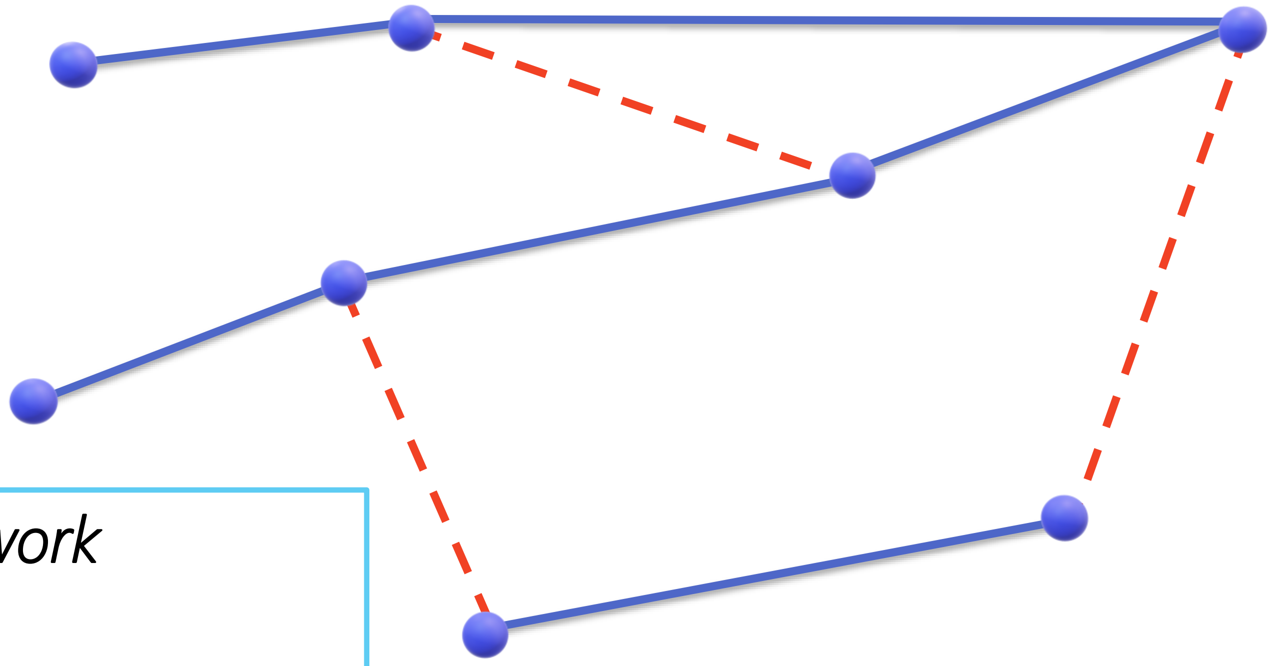
- Typical assumption: Query or sample access to graphs

Sublinear-time algorithms for graphs

- Typical assumption: Query or sample access to graphs
- Does not accurately reflect reality in some situations

Sublinear-time algorithms for graphs

- Typical assumption: Query or sample access to graphs
- Does not accurately reflect reality in some situations

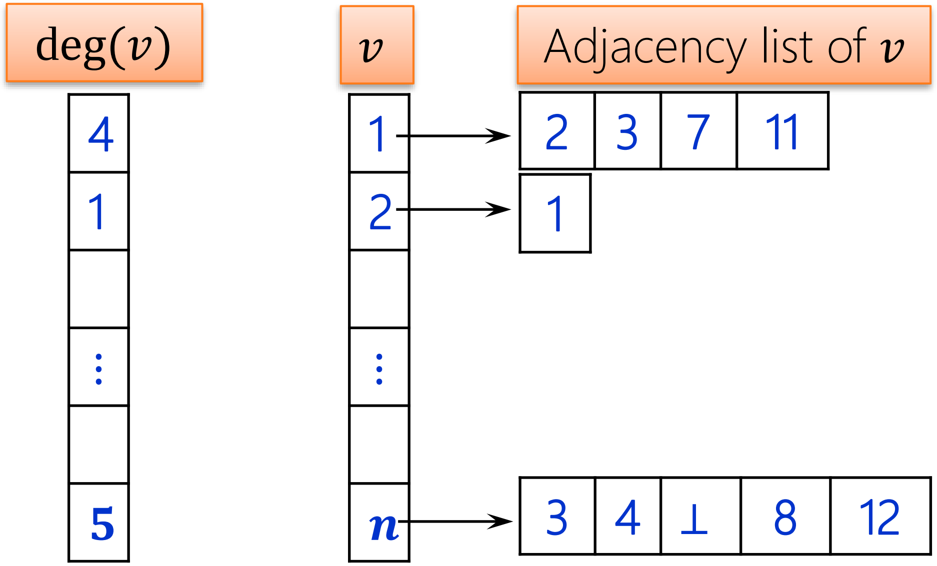


Social Network

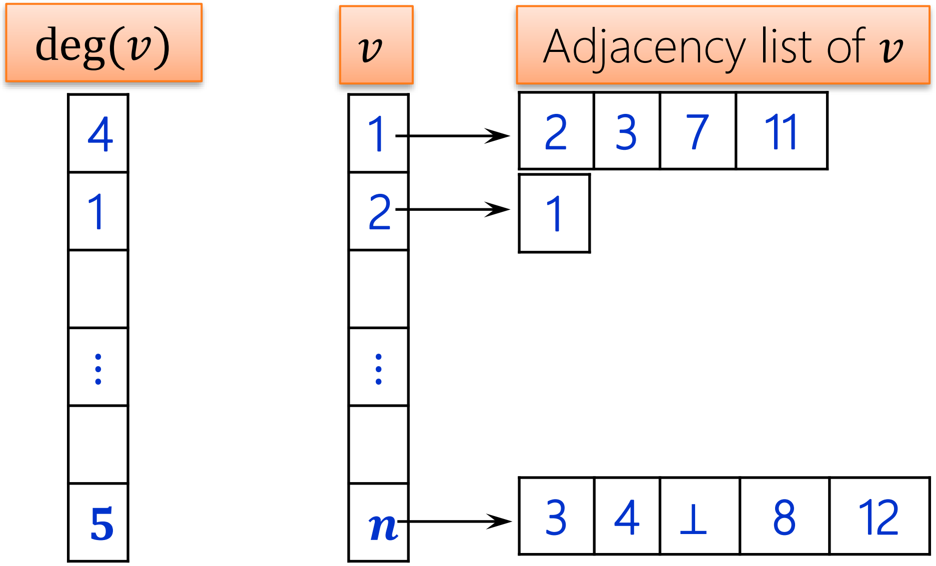
Vertices are users, and edges are friendships

Friendships hidden for privacy

Partially erased graphs: Representation

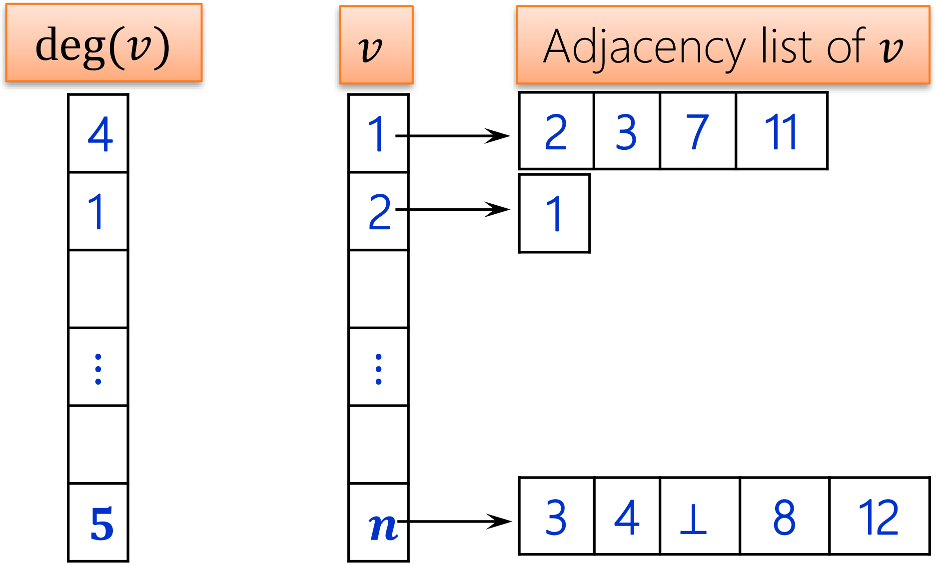


Partially erased graphs: Representation



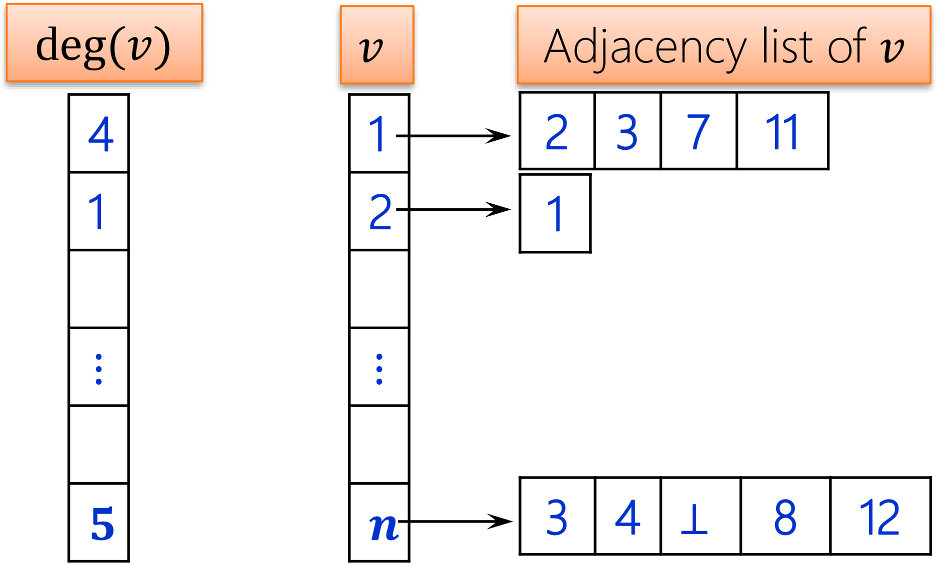
- α-erased graph: At most an $\alpha \in (0,1)$ fraction of entries in the adjacency lists is adversarially erased

Partially erased graphs: Representation



- α-erased graph: At most an $\alpha \in (0,1)$ fraction of entries in the adjacency lists is adversarially erased
- A completion is a valid graph obtained by filling in the erased entries

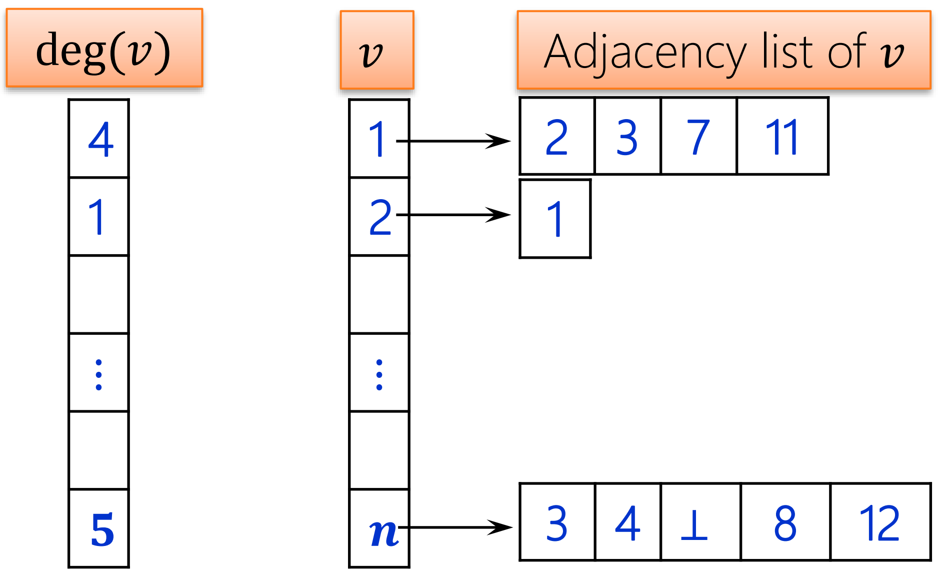
Partially erased graphs: Representation



Adaptation of erasure-resilient model for testing properties of functions by [Dixit Raskhodnikova Thakurta Varma 18] to the case of graphs

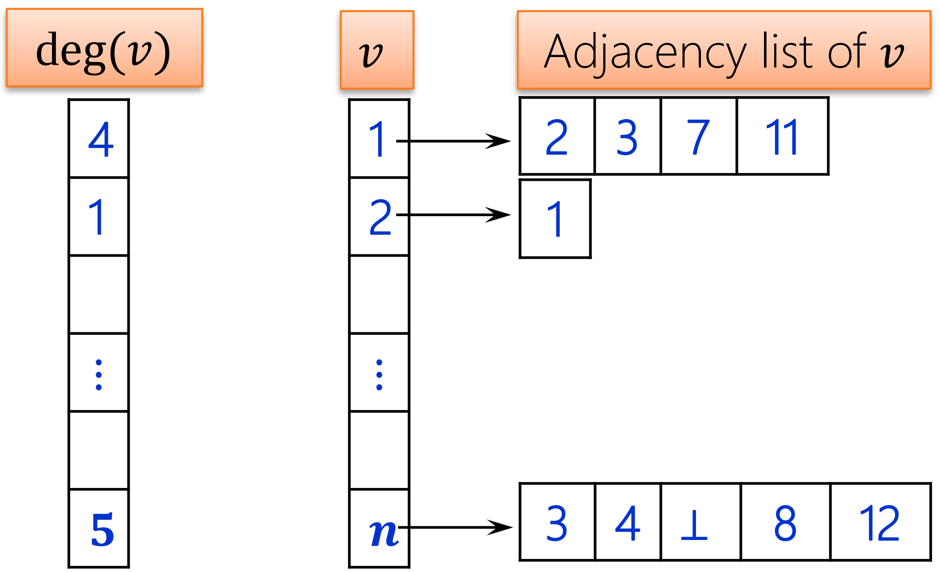
- α-erased graph: At most an $\alpha \in (0,1)$ fraction of entries in the adjacency lists is adversarially erased
- A completion is a valid graph obtained by filling in the erased entries

Erasure-resilient graph algorithms



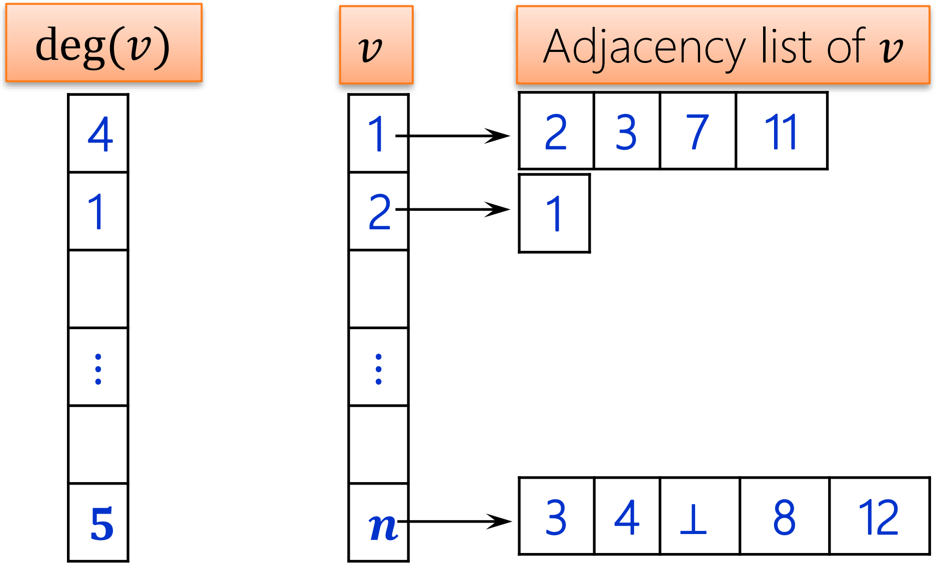
- Algorithm gets parameter $\alpha \in (0,1)$ and query access to α-erased graph

Erasure-resilient graph algorithms



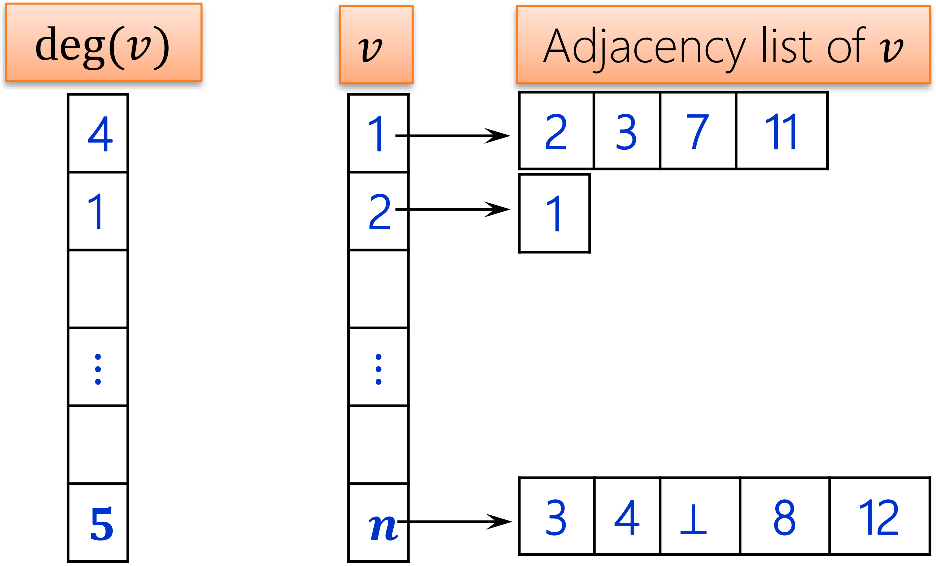
- Algorithm gets parameter $\alpha \in (0,1)$ and query access to α-erased graph
- Degree query v is answered with $\mathbf{deg}(v)$

Erasure-resilient graph algorithms



- Algorithm gets parameter $\alpha \in (0,1)$ and query access to α-erased graph
- Degree query v is answered with $\mathbf{deg}(v)$
- Neighbor query (v, i) is answered with i-th entry in adjacency list of v

Erasure-resilient graph algorithms



Performance of algorithms analyzed in the worst-case over all α-erased graphs

- Algorithm gets parameter $\alpha \in (0,1)$ and query access to α-erased graph
- Degree query v is answered with $\mathbf{deg}(v)$
- Neighbor query (v, i) is answered with i-th entry in adjacency list of v

Computational tasks that we study

- Graph property testing

Computational tasks that we study

- Graph property testing
  - Initiated by [Goldreich Goldwasser Ron 98]

Computational tasks that we study

- Graph property testing
  - Initiated by [Goldreich Goldwasser Ron 98]
  - Our model is a generalization of general graph model of [Parnas Ron 02] which is in turn a generalization of the bounded degree graph model of [Goldreich Ron 02]

Computational tasks that we study

- Graph property testing
  - Initiated by [Goldreich Goldwasser Ron 98]
  - Our model is a generalization of general graph model of [Parnas Ron 02] which is in turn a generalization of the bounded degree graph model of [Goldreich Ron 02]
  - Testers in general graph model for many properties [PR02, Alon Kaufman Krivelevich Ron 08, Kaufman Krivelevich Ron 12, Kusumoto Yoshida 14, Babu Khoury Newman 16,…]

Computational tasks that we study

- Graph property testing
  - Initiated by [Goldreich Goldwasser Ron 98]
  - Our model is a generalization of general graph model of [Parnas Ron 02] which is in turn a generalization of the bounded degree graph model of [Goldreich Ron 02]
  - Testers in general graph model for many properties [PR02, Alon Kaufman Krivelevich Ron 08, Kaufman Krivelevich Ron 12, Kusumoto Yoshida 14, Babu Khoury Newman 16,…]

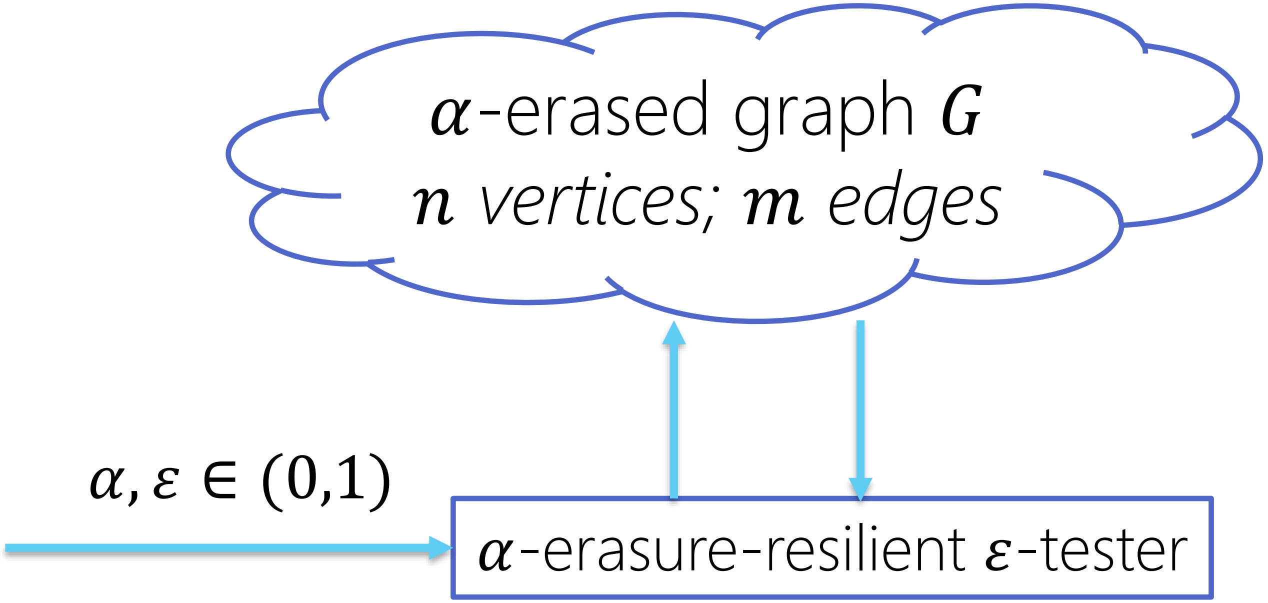
- Estimating graph parameters

Computational tasks that we study

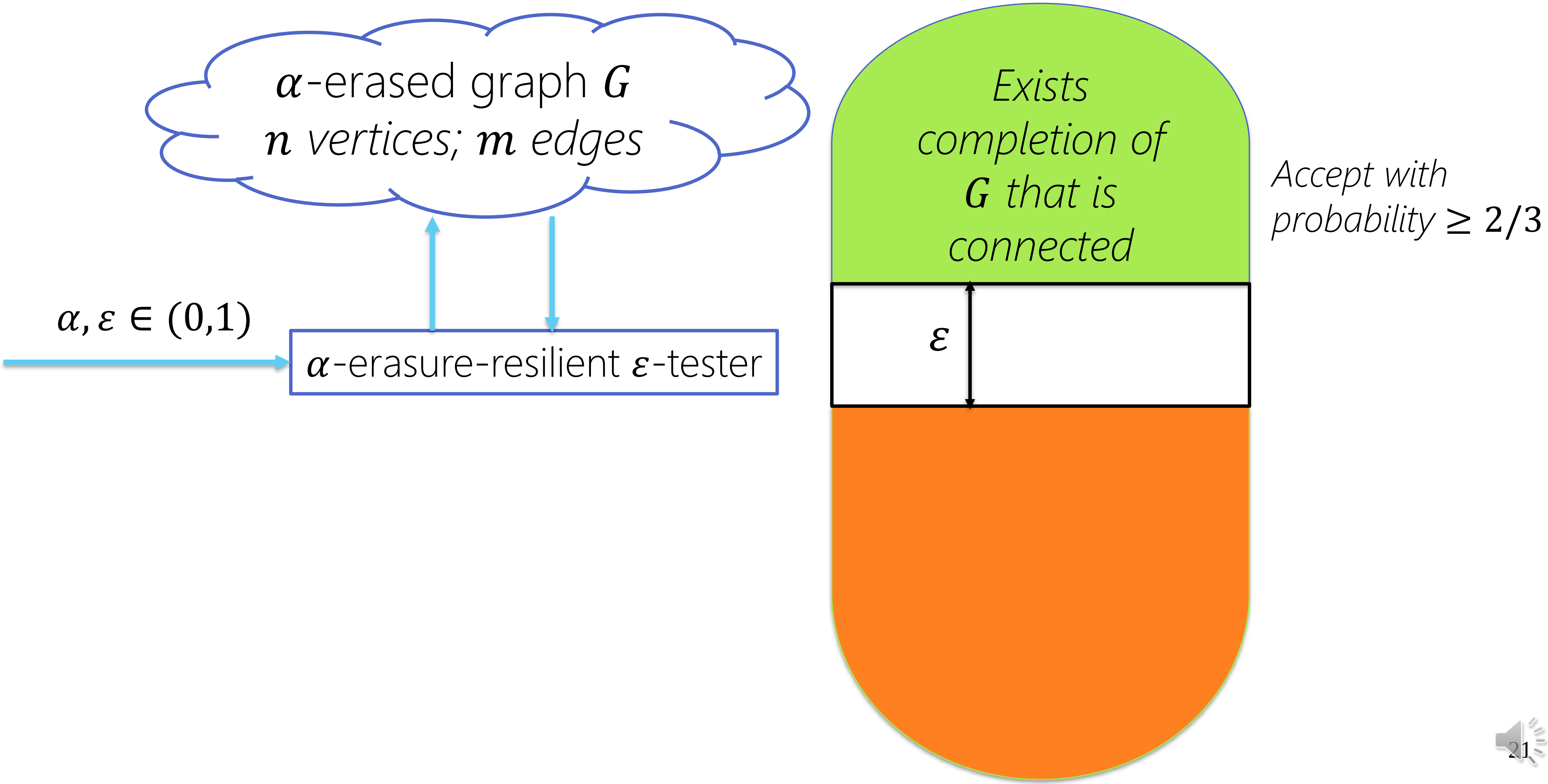
- Graph property testing
  - Initiated by [Goldreich Goldwasser Ron 98]
  - Our model is a generalization of general graph model of [Parnas Ron 02] which is in turn a generalization of the bounded degree graph model of [Goldreich Ron 02]
  - Testers in general graph model for many properties [PR02, Alon Kaufman Krivelevich Ron 08, Kaufman Krivelevich Ron 12, Kusumoto Yoshida 14, Babu Khoury Newman 16,...]

- Estimating graph parameters
  - Sublinear-time algorithms for estimating:
    - Weight of min. spanning tree [Chazelle Rubinfeld Trevisan 05]
    - Number of connected components [CRT05, Berenbrink Krayenhoff Mallmann-Trenn 14]
    - Average degree [Feige 06, Goldreich Ron 08]
    - Moments of degree distribution [Gonen Ron Shavitt 11, Eden Ron Seshadhri 17]
    - and more...

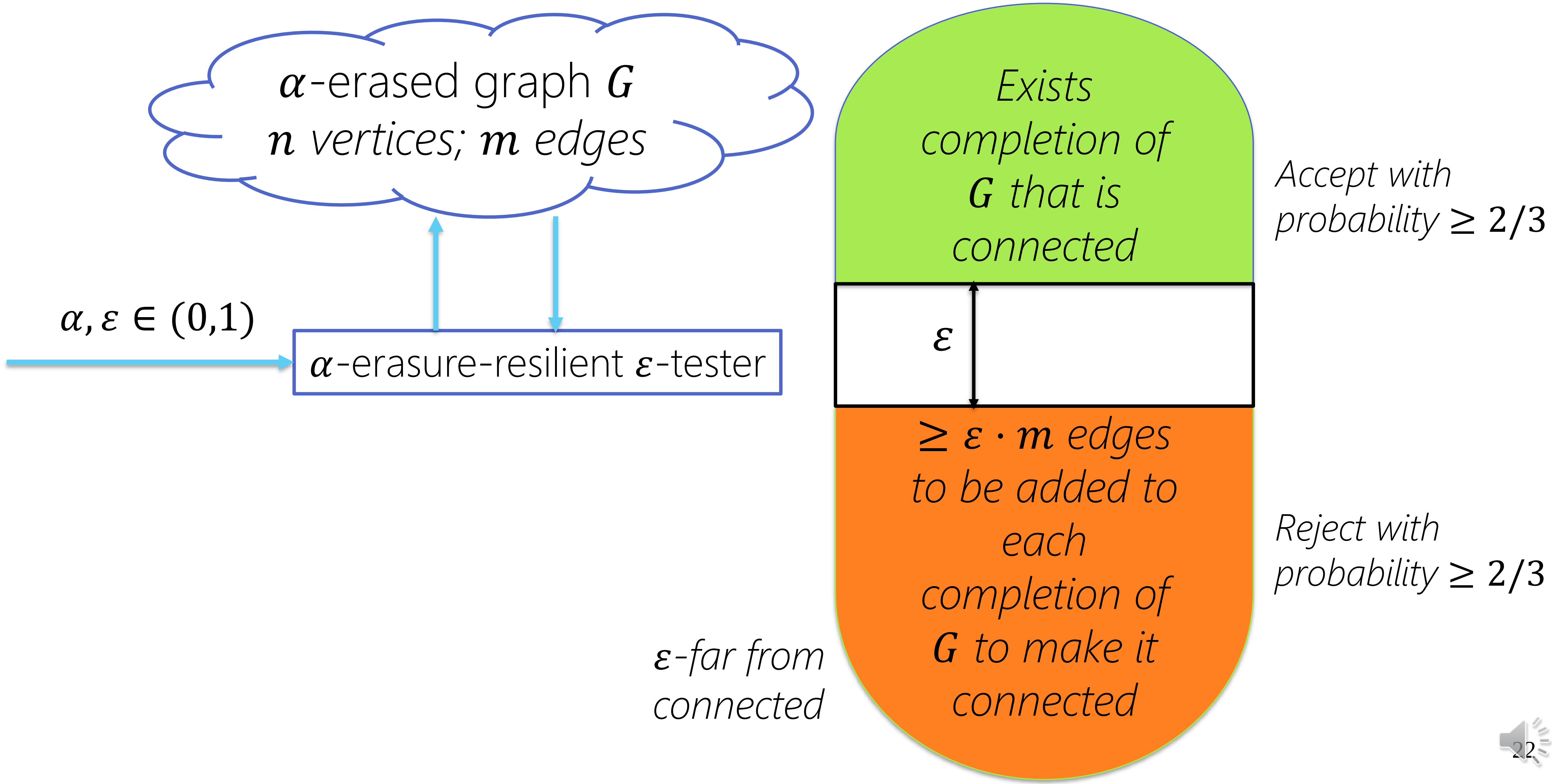
Erasure-resilient testing connectedness of graphs



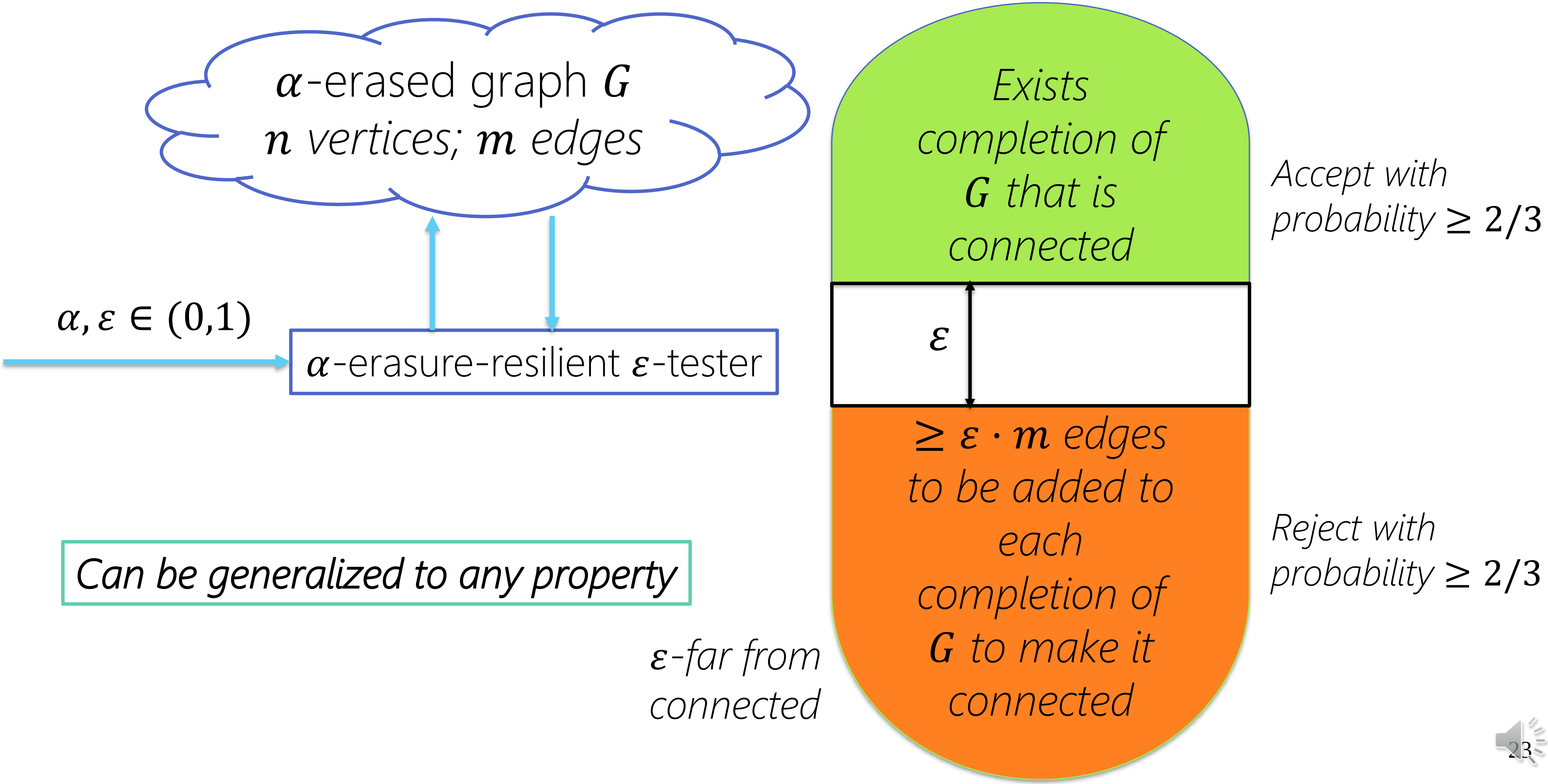
Erasure-resilient testing connectedness of graphs



Erasure-resilient testing connectedness of graphs



Erasure-resilient testing connectedness of graphs



Testing connectedness of graphs

- In the special case of no erasures:

Testing connectedness of graphs

- In the special case of no erasures:
  - Studied by [Goldreich Ron 02, Parnas Ron 02], and [Berman Raskhodnikova Yaroslavtsev 14]
  - Prior best ε-tester [BRY14] has query complexity $O\left(\left(\frac{1}{\varepsilon \bar{d}}\right)^2\right)$, where $\bar{d}$ is the average degree

$\boldsymbol{\alpha}$-erasure-resilient $\boldsymbol{\varepsilon}$-testing connectedness: Our results

Algorithms and lower bounds for $\boldsymbol{\alpha}$-erasure-resilient $\boldsymbol{\varepsilon}$-testing connectedness for graphs of average degree $\bar{d}$

α vs. ε	Query complexity
$\alpha \geq \varepsilon$	$\Omega(n)$

$\boldsymbol{\alpha}$-erasure-resilient $\boldsymbol{\varepsilon}$-testing connectedness: Our results

Algorithms and lower bounds for $\boldsymbol{\alpha}$-erasure-resilient $\boldsymbol{\varepsilon}$-testing connectedness for graphs of average degree $\bar{d}$

α vs. ε	Query complexity
$\alpha \geq \varepsilon$	$\Omega(n)$
$\alpha < \varepsilon$ $\varepsilon' = \varepsilon - \alpha$	$O\left(\min\left\{\frac{1}{(\varepsilon'\bar{d})^3}, \frac{1}{(\varepsilon')^2\bar{d}}\right\}\right)$

$\boldsymbol{\alpha}$-erasure-resilient $\boldsymbol{\varepsilon}$-testing connectedness: Our results

Algorithms and lower bounds for $\boldsymbol{\alpha}$-erasure-resilient $\boldsymbol{\varepsilon}$-testing connectedness for graphs of average degree $\bar{d}$

α vs. ε	Query complexity
$\alpha \geq \varepsilon$	$\Omega(n)$
$\alpha < \varepsilon$, $\varepsilon' = \varepsilon - \alpha$	$O\left(\min\left\{\frac{1}{(\varepsilon'\bar{d})^3}, \frac{1}{(\varepsilon')^2\bar{d}}\right\}\right)$
$\alpha < \varepsilon/2$, $\varepsilon'' = \frac{\varepsilon}{2} - \alpha$	$O\left(\min\left\{\frac{1}{(\varepsilon''\bar{d})^2}, \frac{1}{\varepsilon''} \cdot \log\frac{1}{\varepsilon''\bar{d}}\right\}\right)$

Erasure-resilient testing connectedness: Our results

- Phase transition:
  - If $\alpha < \varepsilon$, the problem is solvable in time independent of the size of the input graph
  - If $\alpha \geq \varepsilon$, the problem requires linear time to solve

Erasure-resilient testing connectedness: Our results

- Phase transition:
  - If $\alpha < \varepsilon$, the problem is solvable in time independent of the size of the input graph
  - If $\alpha \geq \varepsilon$, the problem requires linear time to solve

- In the **special case of no erasures**, complexity of our tester is $O\left(\min\{\frac{1}{\varepsilon} \cdot \log\left(\frac{1}{\varepsilon \bar{d}}\right), \left(\frac{1}{\varepsilon \bar{d}}\right)^2\}\right)$, which is better than the best prior upper bound $O(\left(\frac{1}{\varepsilon \bar{d}}\right)^2)$

Erasure-resilient testing connectedness: Our results

- Phase transition:
  - If $\alpha < \varepsilon$, the problem is solvable in time independent of the size of the input graph
  - If $\alpha \geq \varepsilon$, the problem requires linear time to solve

- In the **special case of no erasures**, complexity of our tester is $O\left(\min\{\frac{1}{\varepsilon} \cdot \log\left(\frac{1}{\varepsilon \bar{d}}\right), \left(\frac{1}{\varepsilon \bar{d}}\right)^2\}\right)$, which is better than the best prior upper bound $O(\left(\frac{1}{\varepsilon \bar{d}}\right)^2)$
  - Our upper bound is tight, as evidenced by a matching lower bound [Pallavoor Raskhodnikova Varma]

Estimating the average degree

- In the **special case of no erasures,** studied by [Feige 06], [Goldreich Ron 08], [Eden Ron Seshadhri 17], [Eden Ron Seshadhri 19]

Estimating the average degree

- In the **special case of no erasures,** studied by [Feige 06], [Goldreich Ron 08], [Eden Ron Seshadhri 17], [Eden Ron Seshadhri 19]

- For $\varepsilon > 0$
  - $(2+\varepsilon)$-approximation algorithm that makes $O\left(\frac{\sqrt{n}}{\varepsilon}\right)$ degree queries [F06]

Estimating the average degree

- In the **special case of no erasures,** studied by
  [Feige 06], [Goldreich Ron 08], [Eden Ron Seshadri 17], [Eden Ron Seshadri 19]

- For $\varepsilon > 0$

  - $(2+\varepsilon)$-approximation algorithm that makes
    $O\left(\frac{\sqrt{n}}{\varepsilon}\right)$ degree queries [F06]

  - Need $\Omega(n)$ queries for a $2 - o(1)$-approximation,
    if one only has degree queries [F06]

Estimating the average degree

- In the **special case of no erasures,** studied by
  [Feige 06], [Goldreich Ron 08], [Eden Ron Seshadri 17], [Eden Ron Seshadri 19]

- For $\varepsilon > 0$
  - $(2+\varepsilon)$-approximation algorithm that makes $O\left(\frac{\sqrt{n}}{\varepsilon}\right)$ degree queries [F06]
  - Need $\Omega(n)$ queries for a $2 - o(1)$-approximation, if one only has degree queries [F06]
  - $(1+\varepsilon)$-approximation algorithm that makes $\tilde{O}\left(\sqrt{n} \cdot poly\left(\frac{1}{\varepsilon}\right)\right)$ degree and neighbor queries [GR08, ERS17, ERS19]

Estimating the average degree: Our results

Estimating average degree of α-erased graphs

Estimating the average degree: Our results

Estimating average degree of $\boldsymbol{\alpha}$-erased graphs

Approximation ratio	Complexity
$1+\varepsilon+\min(2\alpha,1)$	$\tilde{O}\left(\sqrt{n}\cdot poly\left(\frac{1}{\varepsilon}\right)\right)$

Estimating the average degree: Our results

Estimating average degree of $\boldsymbol{\alpha}$-erased graphs

Approximation ratio	Complexity
$1 + \varepsilon + \min(2\alpha, 1)$	$\tilde{O}\left(\sqrt{n} \cdot poly\left(\frac{1}{\varepsilon}\right)\right)$
$1 + \gamma$ for $\gamma < \alpha$	$\Omega(n)$

Estimating the average degree: Our results

Estimating average degree of α-erased graphs

Approximation ratio	Complexity
$1 + \varepsilon + \min(2\alpha, 1)$	$\tilde{O}\left(\sqrt{n} \cdot poly\left(\frac{1}{\varepsilon}\right)\right)$
$1 + \gamma$ for $\gamma < \alpha$	$\Omega(n)$

- Interpolation between model with only degree queries, and model with both degree and neighbor queries

Estimating the average degree: Our results

Estimating average degree of α-erased graphs

Approximation ratio	Complexity
$1 + \varepsilon + \min(2\alpha, 1)$	$\tilde{O}\left(\sqrt{n} \cdot poly\left(\frac{1}{\varepsilon}\right)\right)$
$1 + \gamma$ for $\gamma < \alpha$	$\Omega(n)$

- Interpolation between model with only degree queries, and model with both degree and neighbor queries
  - When $\alpha = 0$, our result identical to [GR08, ERS17, ERS19]

Estimating the average degree: Our results

Estimating average degree of α-erased graphs

Approximation ratio	Complexity
$1 + \varepsilon + \min(2\alpha, 1)$	$\tilde{O}\left(\sqrt{n} \cdot poly\left(\frac{1}{\varepsilon}\right)\right)$
$1 + \gamma$ for $\gamma < \alpha$	$\Omega(n)$

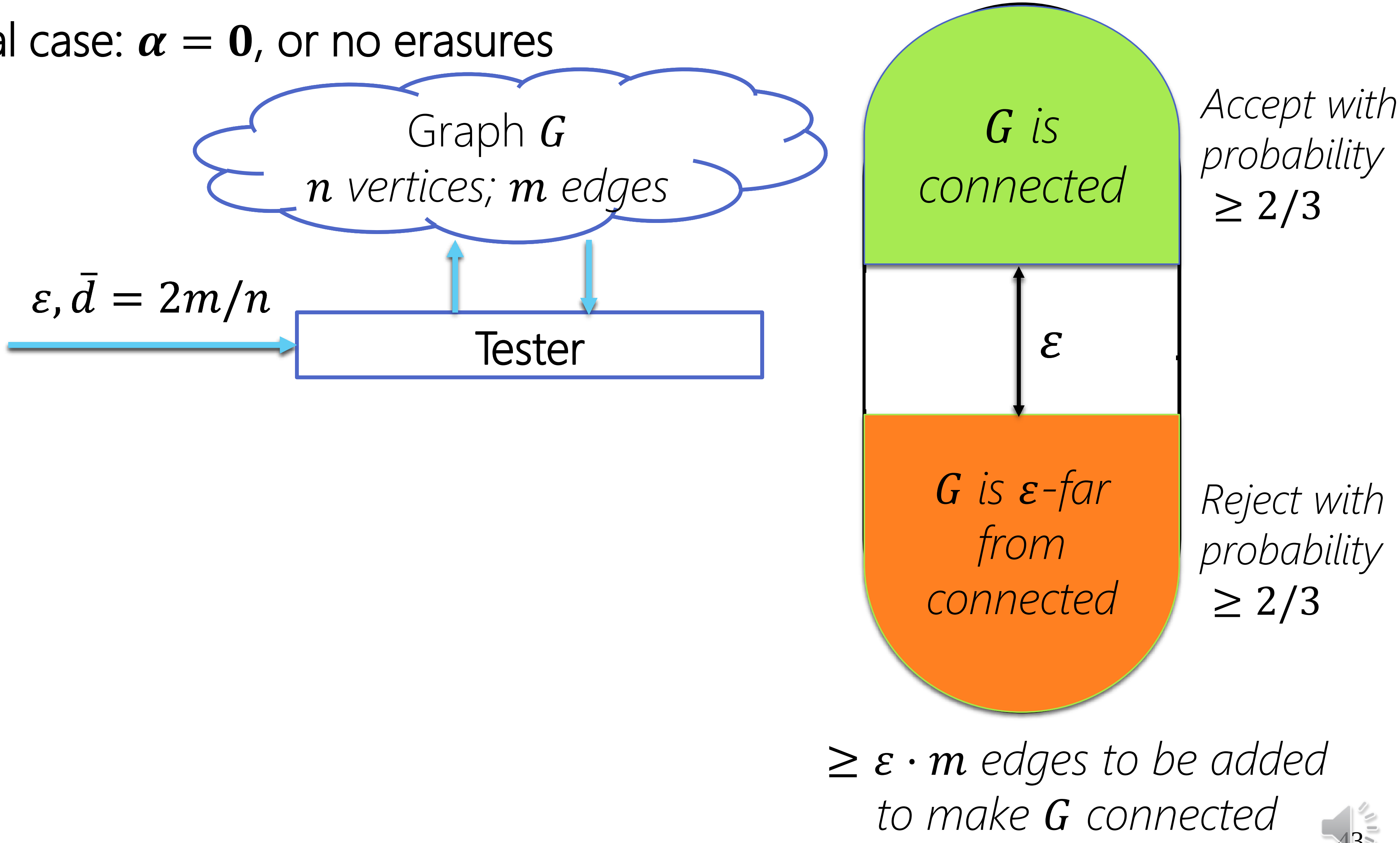
- Interpolation between model with only degree queries, and model with both degree and neighbor queries
  - When $\alpha = 0$, our result identical to [GR08, ERS17, ERS19]
  - When $\alpha = 1/2$, "have access to only degree queries" and we obtain a $2 + \varepsilon$ approximation like [F06]

Erasure-resilient connectedness tester for small $\boldsymbol{\alpha}$

- Today: Special case: $\boldsymbol{\alpha = 0}$, or no erasures

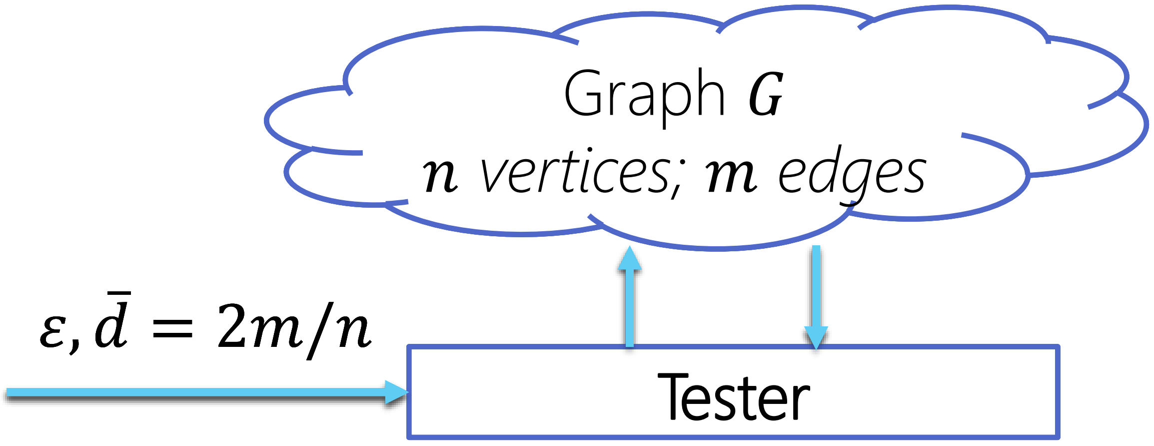
Erasure-resilient connectedness tester for small $\boldsymbol{\alpha}$

- Today: Special case: $\boldsymbol{\alpha} = \mathbf{0}$, or no erasures
- Requirement:

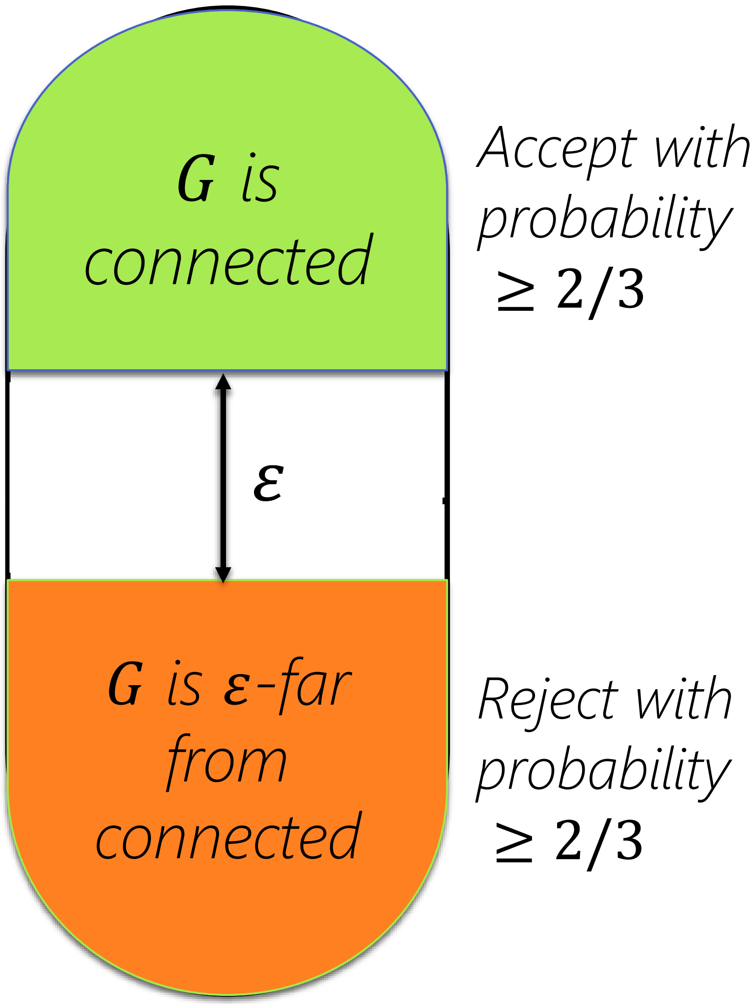


Erasure-resilient connectedness tester for small $\boldsymbol{\alpha}$

- Today: Special case: $\boldsymbol{\alpha} = \mathbf{0}$, or no erasures
- Requirement:



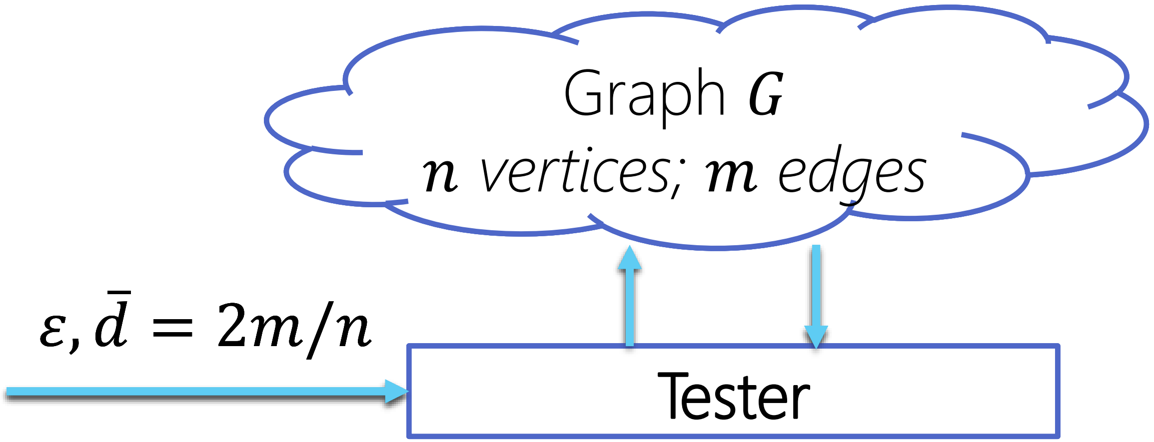
- Basic Algorithmic Idea [GR02]

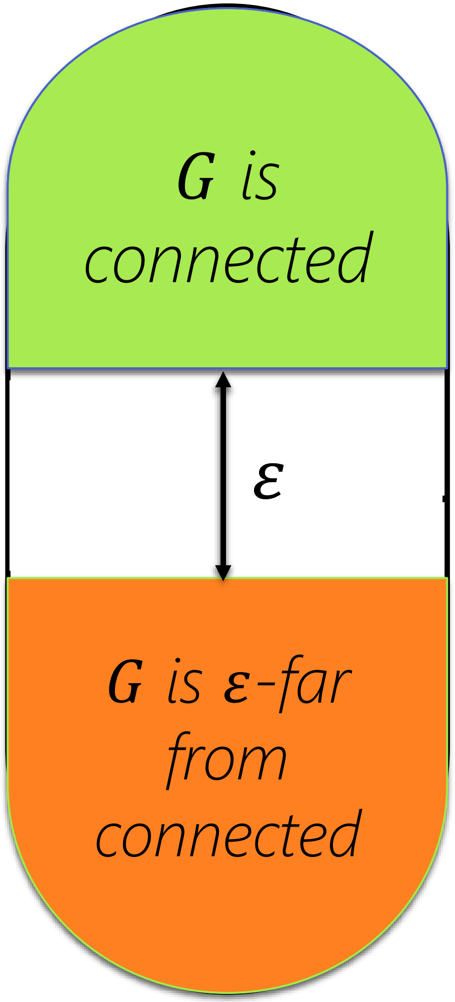


$\geq \varepsilon \cdot m$ edges to be added to make G connected

Erasure-resilient connectedness tester for small α

- Today: Special case: $\alpha = 0$, or no erasures
- Requirement:





Accept with probability $\geq 2/3$

Reject with probability $\geq 2/3$

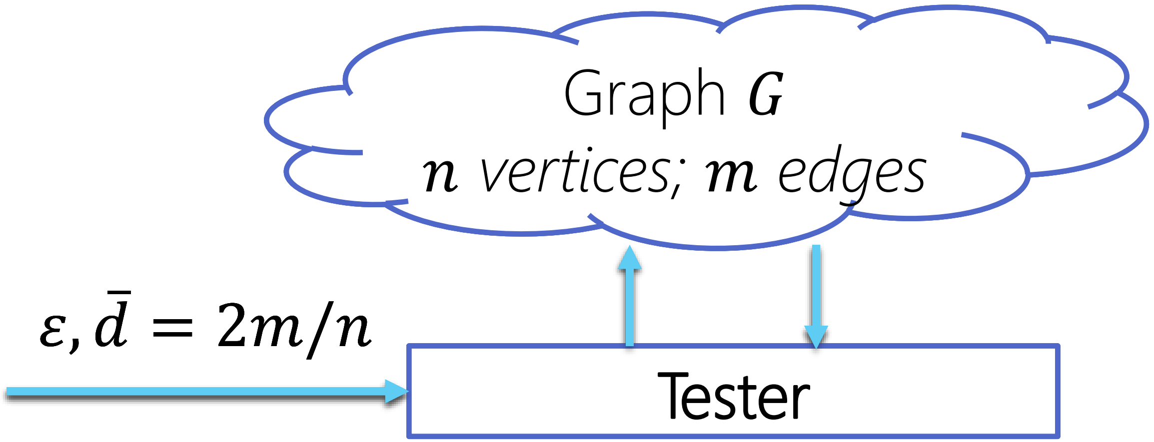
$\geq \varepsilon \cdot m$ edges to be added to make G connected

- Basic Algorithmic Idea [GR02]

Reject only if we are certain that G is not connected

Erasure-resilient connectedness tester for small α

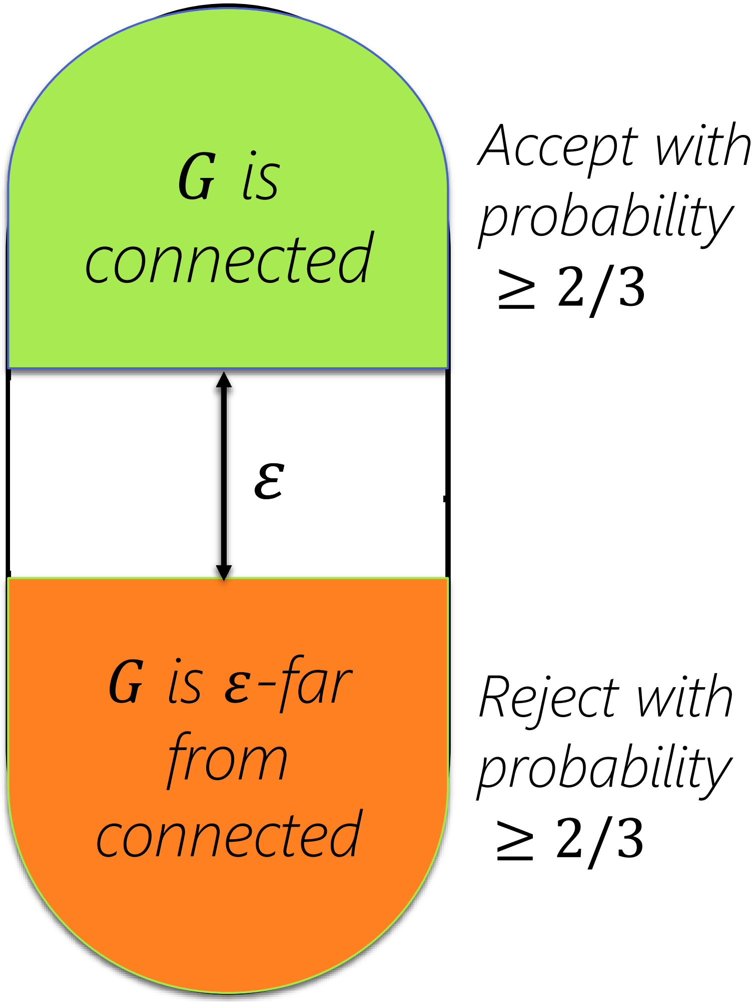
- Today: Special case: $\alpha = 0$, or no erasures
- Requirement:



- Basic Algorithmic Idea [GR02]

Reject only if we are certain that G is not connected

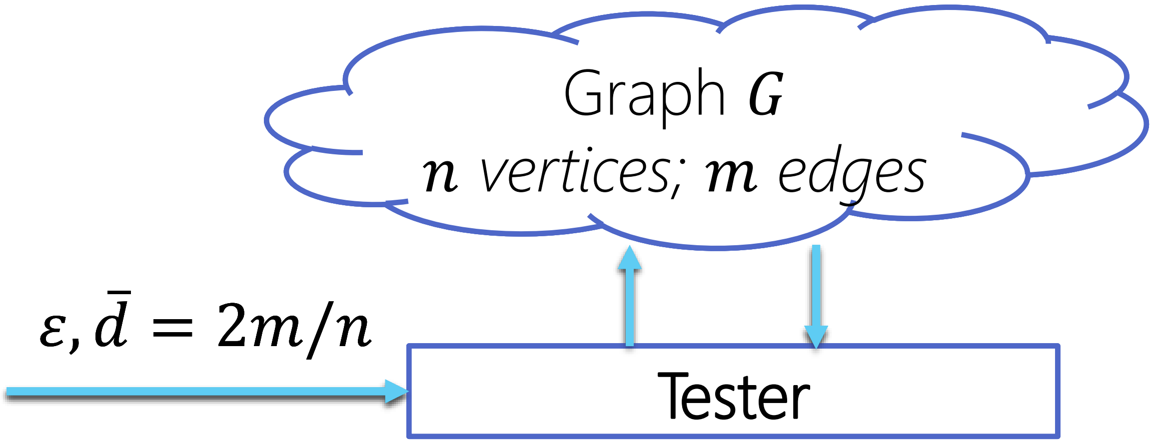
Observation: If G is ε-far from connected, then G has $\geq \varepsilon m$ connected components (CCs)



$\geq \varepsilon \cdot m$ edges to be added to make G connected

Erasure-resilient connectedness tester for small $\boldsymbol{\alpha}$

- Today: Special case: $\boldsymbol{\alpha = 0}$, or no erasures
- Requirement:

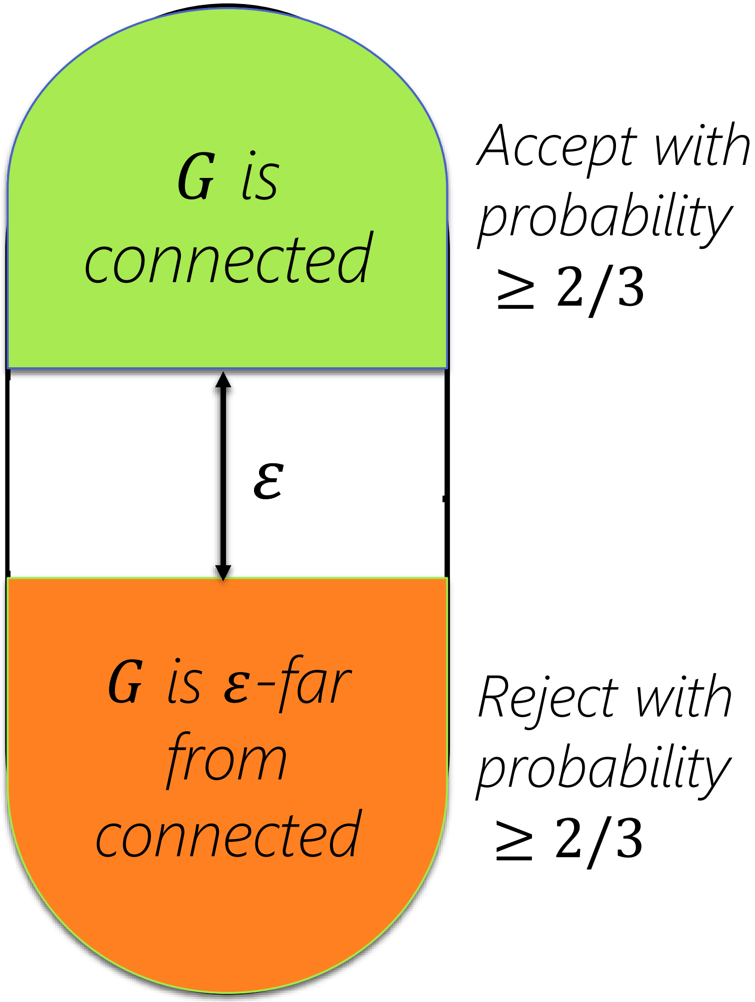


- Basic Algorithmic Idea [GR02]

Reject only if we are certain that $\boldsymbol{G}$ is not connected

Observation: If $\boldsymbol{G}$ is $\boldsymbol{\varepsilon}$-far from connected, then $\boldsymbol{G}$ has $\geq \boldsymbol{\varepsilon m}$ connected components (CCs)

Idea: Detect CCs via BFSs from random vertices



$\geq \boldsymbol{\varepsilon \cdot m}$ edges to be added to make $\boldsymbol{G}$ connected

Detecting graphs that are far from connected

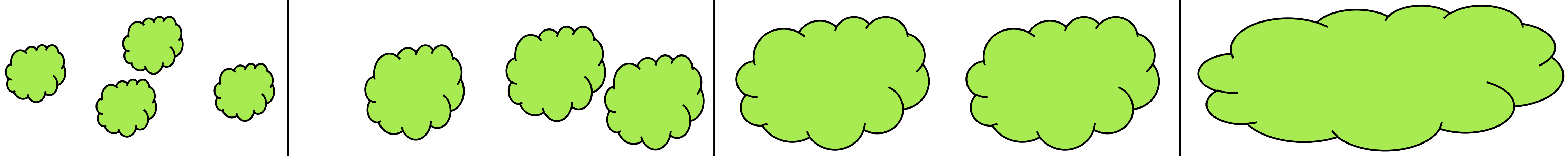
- Observation 1 [GR02]: If G is ε-far from connected, then G has $\geq \varepsilon m$ CCs

Detecting graphs that are far from connected

- Observation 1 [GR02]: If G is ε-far from connected, then G has $\geq \varepsilon m$ CCs
- Observation 2 [GR02]: Not many CCs can have "too many vertices" in them $\Longrightarrow$ At least $\varepsilon m/2$ small CCs (at most $B = 2n/\varepsilon m$ vertices)

Detecting graphs that are far from connected

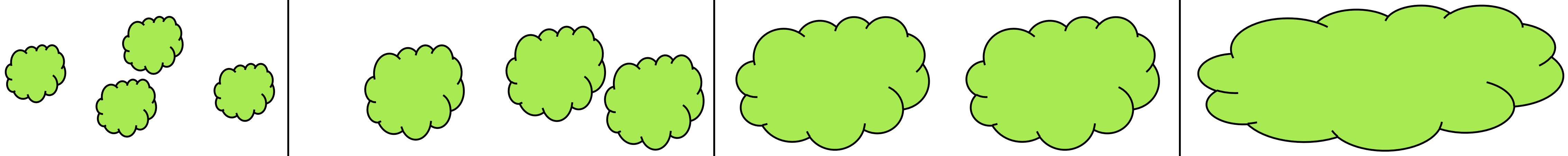
- **Observation 1** [GR02]: If G is ε-far from connected, then G has $\geq \varepsilon m$ CCs
- **Observation 2** [GR02]: Not many CCs can have "too many vertices" in them $\Longrightarrow$ At least $\varepsilon m/2$ small CCs (at most $B = 2n/\varepsilon m$ vertices)
- Classify **small CCs** into $\log B$ **buckets** [GR02]



No. of vertices ∈ $[1, 2)$	No. of vertices ∈ $[2, 4)$	No. of vertices ∈ $[2^{i-1}, 2^{i})$....	No. of vertices ∈ $[B/2, B]$

Detecting graphs that are far from connected

- Observation 1 [GR02]: If G is ε-far from connected, then G has $\geq \varepsilon m$ CCs
- Observation 2 [GR02]: Not many CCs can have "too many vertices" in them $\Longrightarrow$ At least $\varepsilon m/2$ small CCs (at most $B = 2n/\varepsilon m$ vertices)
- Classify **small CCs** into $\log B$ **buckets** [GR02]

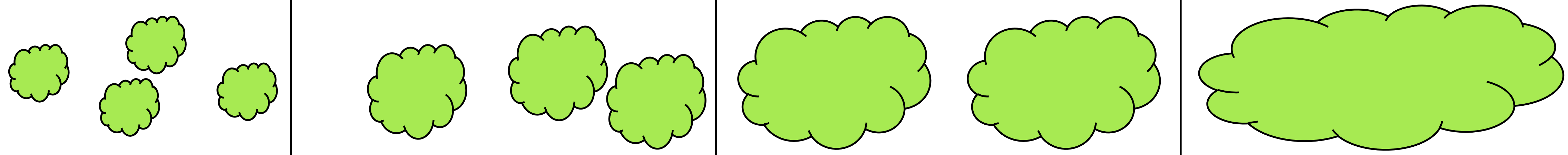


No. of vertices ∈ $[1,2)$	No. of vertices ∈ $[2,4)$	No. of vertices ∈ $[2^{i-1}, 2^{i})$....	No. of vertices ∈ $[B/2, B]$

- **Detecting a small CC** (Work investment strategy [BRY14])

Detecting graphs that are far from connected

- Observation 1 [GR02]: If G is ε-far from connected, then G has $\geq \varepsilon m$ CCs
- Observation 2 [GR02]: Not many CCs can have "too many vertices" in them $\Longrightarrow$ At least $\varepsilon m/2$ small CCs (at most $B = 2n/\varepsilon m$ vertices)
- Classify **small CCs** into $\log B$ **buckets** [GR02]



No. of vertices $\in [1, 2)$	No. of vertices $\in [2, 4)$	No. of vertices $\in [2^{i-1}, 2^i)$....	No. of vertices $\in [B/2, B]$

- **Detecting a small CC** (Work investment strategy [BRY14])
  - For $i \in [\log B]$, sample $O\left(\frac{B}{2^i}\right)$ uniformly random vertices
  - With probability $\geq 2/3$, $\exists i$ such that some vertex in ith iteration is in ith bucket

Prior best connectedness tester

- Input: $\varepsilon, \bar{d} = 2m/n$, query access to graph G of average degree $\bar{d}$

Prior best connectedness tester

- Input: $\varepsilon, \bar{d} = 2m/n$, query access to graph G of average degree $\bar{d}$
- Let $B = \frac{2n}{\varepsilon m} = \frac{4}{\varepsilon \bar{d}}$
- For $i \in [\log B]$
  - Repeat $O\left(\frac{B}{2^i}\right)$ times:
    - Sample a vertex v
    - Run BFS from v until a small CC is found (**reject**) or nbr. query budget is over
      Query budget: 2^{2i} neighbor queries

Prior best connectedness tester

- Input: $\varepsilon, \bar{d} = 2m/n$, query access to graph G of average degree $\bar{d}$
- Let $B = \frac{2n}{\varepsilon m} = \frac{4}{\varepsilon \bar{d}}$
- For $i \in [\log B]$
  - Repeat $O\left(\frac{B}{2^i}\right)$ times:
    - Sample a vertex v
    - Run BFS from v until a small CC is found (**reject**) or nbr. query budget is over
      Query budget: 2^{2i} neighbor queries

Work investment strategy [BRY14]
- With probability $\geq 2/3$, $\exists i$ such that some vertex in ith iteration is in bucket with no. of vertices $\in [2^{i-1}, 2^i)$

Our connectedness tester

- Input: $\varepsilon, \bar{d} = 2m/n$, query access to graph G of average degree $\bar{d}$
- Let $B = \frac{2n}{\varepsilon m} = \frac{4}{\varepsilon \bar{d}}$
- For $i \in [\log B]$
  - Repeat $O\left(\frac{B}{2^i}\right)$ times:
    - Sample a vertex v
    - Run BFS from v until a small CC is found (**reject**) or nbr. query budget is over
      Query budget: ~~2^{2i}~~ $\deg(v) \cdot 2^i$ neighbor queries

Work investment strategy [BRY14]

- With probability $\geq 2/3$, $\exists i$ such that some vertex in ith iteration is in bucket with no. of vertices $\in [2^{i-1}, 2^i)$

Our connectedness tester

- Input: $\varepsilon, \bar{d} = 2m/n$, query access to graph G of average degree $\bar{d}$
- Let $B = \frac{2n}{\varepsilon m} = \frac{4}{\varepsilon \bar{d}}$
- For $i \in [\log B]$
  - Repeat $O\left(\frac{B}{2^i}\right)$ times:
    - Sample a vertex v
    - Run BFS from v until a small CC is found (**reject**) or nbr. query budget is over
      Query budget: ~~2^{2i}~~ $\deg(v) \cdot 2^i$ neighbor queries

Work investment strategy [BRY14]

- With probability $\geq 2/3$, $\exists i$ such that some vertex in ith iteration is in bucket with no. of vertices $\in [2^{i-1}, 2^i)$

Correctness

- BFS query budget is "sufficient" to see CC

Our connectedness tester

- Input: $\varepsilon, \bar{d} = 2m/n$, query access to graph G of average degree $\bar{d}$
- Let $B = \frac{2n}{\varepsilon m} = \frac{4}{\varepsilon \bar{d}}$
- For $i \in [\log B]$
  - Repeat $O\left(\frac{B}{2^i}\right)$ times:
    - Sample a vertex v
    - Run BFS from v until a small CC is found (**reject**) or nbr. query budget is over
      Query budget: ~~2^{2i}~~ $\deg(v) \cdot 2^i$ neighbor queries

Work investment strategy [BRY14]

- With probability $\geq 2/3$, $\exists i$ such that some vertex in ith iteration is in bucket with no. of vertices $\in [2^{i-1}, 2^i)$

Correctness

- BFS query budget is "sufficient" to see CC
- For CC with vertex set C and edge set E_C

$$\frac{1}{|C|}\sum_{v \in C} \deg(v) \cdot |C| = 2|E_C|$$

Our connectedness tester

- Input: $\varepsilon, \bar{d} = 2m/n$, query access to graph G of average degree $\bar{d}$
- Let $B = \frac{2n}{\varepsilon m} = \frac{4}{\varepsilon \bar{d}}$
- For $i \in [\log B]$
  - Repeat $O\left(\frac{B}{2^i}\right)$ times:
    - Sample a vertex v
    - Run BFS from v until a small CC is found (**reject**) or nbr. query budget is over
      Query budget: ~~2^{2i}~~ $\deg(v) \cdot 2^i$ neighbor queries

Work investment strategy [BRY14]

- With probability $\geq 2/3$, $\exists i$ such that some vertex in ith iteration is in bucket with no. of vertices $\in [2^{i-1}, 2^i)$

Correctness

- BFS query budget is "sufficient" to see CC
- For CC with vertex set C and edge set E_C

$$\frac{1}{|C|} \sum_{v \in C} \deg(v) \cdot |C| = 2|E_C|$$

Expected query complexity

$$\sum_{i \in [\log B]} O\left(\frac{B}{2^i}\right) \cdot 2^i \cdot \mathrm{E}_{v \in V}[\deg(v)]$$

$$= O(B\bar{d} \log B) = O\left(\frac{1}{\varepsilon} \cdot \log\left(\frac{1}{\varepsilon \bar{d}}\right)\right)$$

Connectedness testing without erasures: What we get

Can ε*-test connectedness with* $O\left(\min\left\{\frac{1}{\varepsilon}\cdot\log\left(\frac{1}{\varepsilon\bar{d}}\right), \frac{1}{(\varepsilon\bar{d})^2}\right\}\right)$ *queries*

Connectedness testing without erasures: What we get

Can ε-test connectedness with $O\left(\min\left\{\frac{1}{\varepsilon} \cdot \log\left(\frac{1}{\varepsilon \bar{d}}\right), \frac{1}{(\varepsilon \bar{d})^2}\right\}\right)$ *queries*

- Improvement in complexity when $\bar{d} \lesssim \sqrt{\frac{1}{\varepsilon}}$, i.e., when average degree is small

Connectedness testing without erasures: What we get

Can $\boldsymbol{\varepsilon}$*-test connectedness with* $\boldsymbol{O}\left(\min\left\{\frac{1}{\varepsilon}\cdot\log\left(\frac{1}{\varepsilon\bar{d}}\right), \frac{1}{(\varepsilon\bar{d})^2}\right\}\right)$ *queries*

- Improvement in complexity when $\bar{d} \lesssim \sqrt{\frac{1}{\varepsilon}}$, i.e., when average degree is small
- Several large graphs of interest are sparse and have low average degree

Erasure-resilient connectedness tester

- For $\alpha < \varepsilon/2$
  - Several erasure-free small CCs in a graph that is far from connected
  - Generalization of the strategy in the case without erasures

Erasure-resilient connectedness tester

- For $\alpha < \varepsilon/2$
  - Several erasure-free small CCs in a graph that is far from connected
  - Generalization of the strategy in the case without erasures

- For $\alpha \in \left[\frac{\varepsilon}{2}, \varepsilon\right)$
  - In a graph that is far from connected, there need not be any CC without erasures

Erasure-resilient connectedness tester

- For $\alpha < \varepsilon/2$
  - Several erasure-free small CCs in a graph that is far from connected
  - Generalization of the strategy in the case without erasures

- For $\alpha \in \left[\frac{\varepsilon}{2}, \varepsilon\right)$
  - In a graph that is far from connected, there need not be any CC without erasures
  - Find a special <u>subgraph that forms a CC in every completion</u> of the input

Erasure-resilient connectedness tester

- For $\alpha < \varepsilon/2$
  - Several erasure-free small CCs in a graph that is far from connected
  - Generalization of the strategy in the case without erasures

- For $\alpha \in \left[\frac{\varepsilon}{2}, \varepsilon\right)$
  - In a graph that is far from connected, there need not be any CC without erasures
  - Find a special <u>subgraph that forms a CC in every completion</u> of the input
  - Larger query complexity

Open problems

- Secondary phase transition in the complexity of erasure-resilient connectedness testing
  - Is the change in complexity at $\alpha = \varepsilon/2$ inherent?

Open problems

- Secondary phase transition in the complexity of erasure-resilient connectedness testing
  - Is the change in complexity at $\alpha = \varepsilon/2$ inherent?
- Erasure-resilient testing of monotone properties
  - Property is monotone if it is preserved under deletion of edges and vertices; examples: bipartiteness, triangle-freeness

Open problems

- Secondary phase transition in the complexity of erasure-resilient connectedness testing
  - Is the change in complexity at $\alpha = \varepsilon/2$ inherent?
- Erasure-resilient testing of monotone properties
  - Property is monotone if it is preserved under deletion of edges and vertices; examples: bipartiteness, triangle-freeness
  - Observation: In the bounded degree model with max degree D, the cost of erasure-resilience is a factor of D^2 in query complexity.

Open problems

- Secondary phase transition in the complexity of erasure-resilient connectedness testing
  - Is the change in complexity at $\alpha = \varepsilon/2$ inherent?
- Erasure-resilient testing of monotone properties
  - Property is monotone if it is preserved under deletion of edges and vertices; examples: bipartiteness, triangle-freeness
  - Observation: In the bounded degree model with max degree D, the cost of erasure-resilience is a factor of D^2 in query complexity.
  - How much does erasure-resilience affect query complexity of testing monotone properties of general graphs?

Open problems

- Asymmetric vs. symmetric erasures
  - In our model, v can be erased from $Adj(u)$ but u is present in $Adj(v)$

Open problems

- Asymmetric vs. symmetric erasures
  - In our model, v can be erased from $Adj(u)$ but u is present in $Adj(v)$
  - Does testing become easier if we assume that erasures are made symmetrically?

Open problems

- Asymmetric vs. symmetric erasures
  - In our model, $\boldsymbol{v}$ can be erased from $\boldsymbol{Adj(u)}$ but $\boldsymbol{u}$ is present in $\boldsymbol{Adj(v)}$
  - Does testing become easier if we assume that erasures are made symmetrically?
- Erasure-resilient vs. tolerant testing of graphs
  - For $\delta < \varepsilon$, a (δ, ε)-tolerant tester [Parnas Ron Rubinfeld 06] for a property P must distinguish, with high probability, between inputs that are δ-close to P and inputs that are ε-far

Open problems

- Asymmetric vs. symmetric erasures
  - In our model, v can be erased from $Adj(u)$ but u is present in $Adj(v)$
  - Does testing become easier if we assume that erasures are made symmetrically?
- Erasure-resilient vs. tolerant testing of graphs
  - For $\delta < \varepsilon$, a (δ, ε)-tolerant tester [Parnas Ron Rubinfeld 06] for a property P must distinguish, with high probability, between inputs that are δ-close to P and inputs that are ε-far
  - Observation [DRTV18]: Tolerant tester for a property can be converted to an erasure-resilient tester with the same query complexity

Open problems

- Asymmetric vs. symmetric erasures
  - In our model, $\boldsymbol{v}$ can be erased from $\boldsymbol{Adj(u)}$ but $\boldsymbol{u}$ is present in $\boldsymbol{Adj(v)}$
  - Does testing become easier if we assume that erasures are made symmetrically?
- Erasure-resilient vs. tolerant testing of graphs
  - For $\delta < \varepsilon$, a (δ, ε)-tolerant tester [Parnas Ron Rubinfeld 06] for a property P must distinguish, with high probability, between inputs that are δ-close to P and inputs that are ε-far
  - Observation [DRTV18]: Tolerant tester for a property can be converted to an erasure-resilient tester with the same query complexity
    - Works by filling in queried erasures with arbitrary values
    - Does not work for graphs represented as adjacency lists containing erasures

Open problems

- Asymmetric vs. symmetric erasures
  - In our model, $\boldsymbol{v}$ can be erased from $\boldsymbol{Adj(u)}$ but $\boldsymbol{u}$ is present in $\boldsymbol{Adj(v)}$
  - Does testing become easier if we assume that erasures are made symmetrically?
- Erasure-resilient vs. tolerant testing of graphs
  - For $\delta < \varepsilon$, a (δ, ε)-tolerant tester [Parnas Ron Rubinfeld 06] for a property P must distinguish, with high probability, between inputs that are δ-close to P and inputs that are ε-far
  - Observation [DRTV18]: Tolerant tester for a property can be converted to an erasure-resilient tester with the same query complexity
    - Works by filling in queried erasures with arbitrary values
    - Does not work for graphs represented as adjacency lists containing erasures
  - What is the relationship between erasure-resilient and tolerant testing in the general graph model?

Thank you!