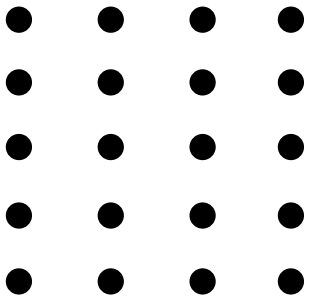
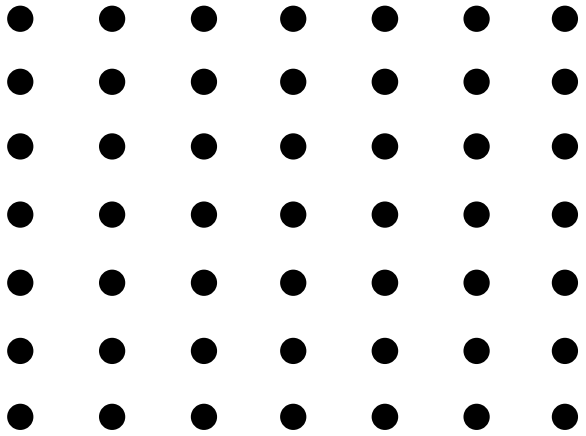
Algorithms at Scale

(Week 1)



Puzzle of the Day:

Connect the dots.
Use ?? straight lines.
Don't lift your pen from the paper.



5 x 4: Use 7 lines.

Can you end at the same place you began?

7 x 7: Use 12 lines

Sublinear Time / Sampling Algorithms

Basic question:

What can we do when we look at only a small part of our input data?

Examples:

- Given a graph... only look at a *constant* number of nodes/edges.
- Is the graph connected?
- How many connected components does it have?
- What is the weight of the MST?
- What is the average degree of the graph?
- What is the diameter of the graph?
- What is the best matching on the graph?

Comparison

	Classical	Sublinear Approximation
Is the graph connected?		
# connected components?		
Weight of MST?		

n : number of nodes
m : number of edges
d : degree of graph

W : weight of MST
ϵ : error / approximation parameter

Comparison

	Classical	Sublinear Approximation
Is the graph connected?	$O(n+m)$	
# connected components?	$O(n+m)$	
Weight of MST?	$O\left(m \log n\right)$	

n : number of nodes
m : number of edges
d : degree of graph

W : weight of MST
ϵ : error / approximation parameter

Comparison

	Classical	Sublinear Approximation
Is the graph connected?	$O(n+m)$	$O\left(\frac{1}{\epsilon^2 d}\right)$
# connected components?	$O(n+m)$	$O\left(\frac{d}{\epsilon^3}\right)$
Weight of MST?	$O\left(m \log n\right)$	$O\left(\frac{dW^4 \log W}{\epsilon^3}\right)$

n : number of nodes
m : number of edges
d : degree of graph

W : weight of MST
ϵ : error / approximation parameter

Comparison

does not matter how big the graph is!

	Classical	Sublinear Approximation
Is the graph connected?	$O(n+m)$	$O\left(\frac{1}{\epsilon^2 d}\right)$
# connected components?	$O(n+m)$	$O\left(\frac{d}{\epsilon^3}\right)$
Weight of MST?	$O\left(m \log n\right)$	$O\left(\frac{dW^4 \log W}{\epsilon^3}\right)$

n : number of nodes
m : number of edges
d : degree of graph

W : weight of MST
ϵ : error / approximation parameter

Comparison

	Classical	Sublinear Approximation
Average degree?	$O(n+m)$	$O\left(\frac{\sqrt{n}}{\epsilon^{9/2}}\right)$
Diameter?	$O\left(n(m+n)\right)$	$O\left(\frac{1}{\epsilon^3}\right)$
Maximum matching?	$O\left(mn^2\right)$	$O\left(\frac{d^4}{\epsilon^2}\right)$

n : number of nodes
m : number of edges
d : degree of graph

W : weight of MST
ϵ : error / approximation parameter

Algorithms

Today

Toy example 1: array all 0's?

- Gap-style question:
  All 0's or far from all 0's?

Toy example 2: Faction of 1's?

- Additive ± ε approximation
- Hoeffding Bound

Is the graph connected?

- Gap-style question.
- O(1) time algorithm.
- Correct with probability 2/3.

Next week:

Number of connected components in a graph.

- *Additive* approximation algorithm.

Weight of MST

- *Multiplicative* approximation algorithm.

Trade-off: speed vs. accuracy

MST(G) = weight of MST

ALG(G) = weight of spanning tree returned by algorithm

Approximate solutions:

Example: relative error

$$MST(G)(1-\epsilon) \leq ALG(G) \leq MST(G)(1+\epsilon)$$

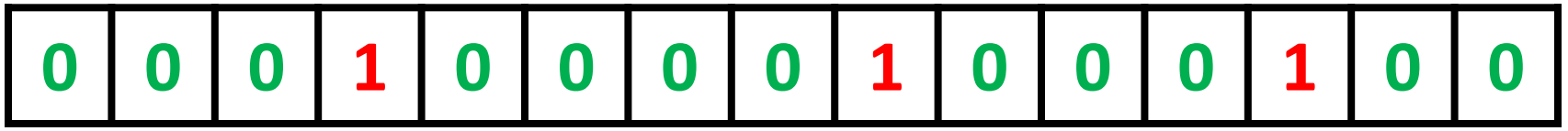
Example: absolute error

$$MST(G) - \epsilon \leq ALG(G) \leq MST(G) + \epsilon$$

Example: gap error

- If G is connected, then return TRUE.
- If G is **ϵ-far** from connected, then return FALSE.
- Otherwise, don't care.

Warm-Up Problem: All Zeros



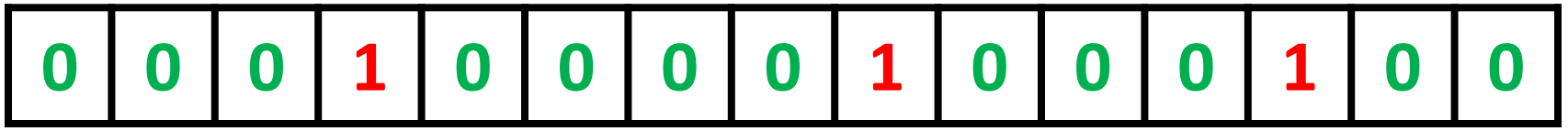
Assumptions:

Given n element array

- Each element is 0 or 1.
- 0 = good test.
- 1 = failed test.

Output: Is the array all 0?

Warm-Up Problem: All Zeros

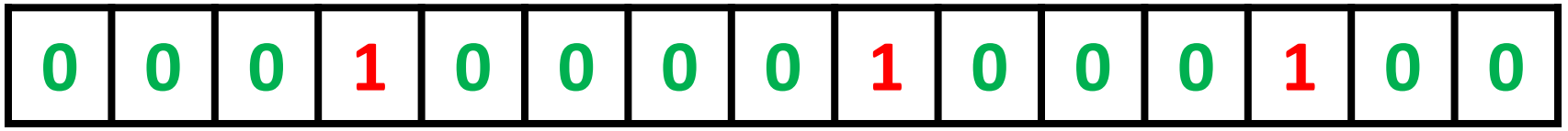


Algorithm: Check every cell.

Running time: $O(n)$

Can we do any better?

Warm-Up Problem: All Zeros



Algorithm: Check every cell.

Running time: $O(n)$

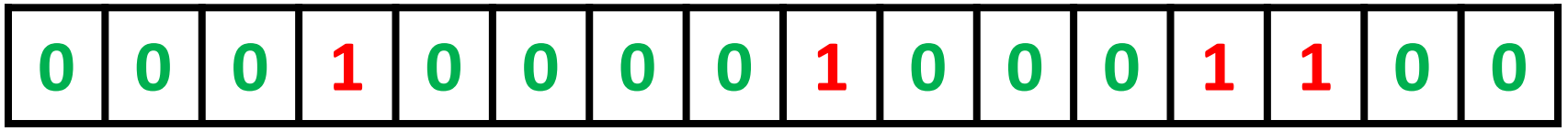
Can we do any better? No!

Lower bound: $\Omega(n)$

Challenge #1:

Prove it.

Warm-Up Problem: All Zeros



Relaxed (approximate) version:

Output: Is the array *mostly* 0?

if all 0's	:	**return** true
if $\geq \epsilon n$ 1's	:	**return** false
otherwise	:	**return** true **or** false

Warm-Up Problem: All Zeros

0	0	0	1	0	0	0	0	1	0	0	0	1	1	0	0

n = 16
ϵ = 1/4

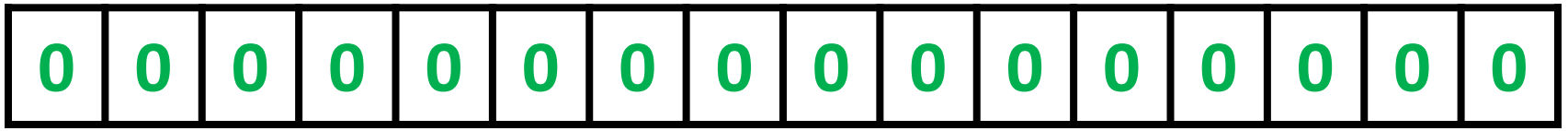
FALSE

Relaxed (approximate) version:

Output: Is the array *mostly* 0?

if all 0's	:	**return** true
if $\geq \epsilon n$ 1's	:	**return** false
otherwise	:	**return** true **or** false

Warm-Up Problem: All Zeros



n = 16
ϵ = 1/4

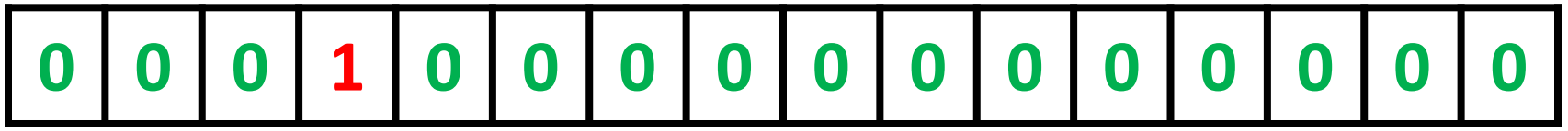
TRUE

Relaxed (approximate) version:

Output: Is the array *mostly* 0?

if all 0's	:	**return** true
if $\geq \epsilon n$ 1's	:	**return** false
otherwise	:	**return** true **or** false

Warm-Up Problem: All Zeros



n = 16
$\epsilon = 1/4$

TRUE OR FALSE

Relaxed (approximate) version:

Output: Is the array *mostly* 0?

if all 0's	:	**return** true
if $\geq \epsilon n$ 1's	:	**return** false
otherwise	:	**return** true **or** false

Warm-Up Problem: All Zeros

0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

All-Zeros(A, ∈)

Repeat s times:

Choose random i in [1, n] .

if A[i] = 1 then return FALSE.

Return TRUE.

Fix $s = \frac{2}{\epsilon}$

Warm-Up Problem: All Zeros

0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

All-Zeros(A, ∈)

Repeat s times:

Choose random i in [1, n] .

if A[i] = 1 then return FALSE.

Return TRUE.

Fix $s = \frac{2}{\epsilon}$

Claim 1: If array is all 0's, then always returns TRUE.

Warm-Up Problem: All Zeros

0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

All-Zeros(A, ∈)

Repeat s times:

Choose random i in [1, n] .

if A[i] = 1 then return FALSE.

Return TRUE.

Fix $s = \frac{2}{\epsilon}$

Claim 2: If array has ≥ εn 1's, then returns FALSE with probability at least 2/3.

Warm-Up Problem: All Zeros

Fix $s = \frac{2}{\epsilon}$

0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0

Claim 2: If array has ≥ εn 1's, then returns FALSE with probability at least 2/3.

Proof:

Assume ≥ εn 1's.

For sample i : Pr[A[i] = 1] ≥ εn/n ≥ ε.

$$\begin{aligned} \Pr\{\text{all samples are } 0\} &\leq (1-\epsilon)^s \\ &\leq (1-\epsilon)^{2/\epsilon} \\ &\leq e^{-2} \\ &\leq \frac{1}{3} \end{aligned}$$

Warm Up Problem: All Zeros

Fix $s = \frac{2}{\epsilon}$

0	1	0	0	0	0

eturns FALSE with

Death Bed Fact: (0 < x < 1)

$$e^{-x} = 1 - x + x^2/2 - \dots$$

$$\frac{1}{e^2} \leq \left(1 - \frac{1}{x}\right)^x \leq \frac{1}{e}$$

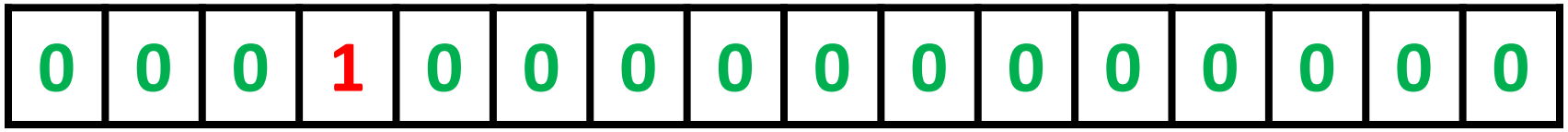
Proof:

Assume ≥ εn 1's.

For sample i : Pr[A[i] = 1] ≥ εn/n ≥ ε.

$$\begin{aligned} \Pr\{\text{all samples are } 0\} &\leq (1-\epsilon)^s \\ &\leq (1-\epsilon)^{2/\epsilon} \\ &\leq e^{-2} \\ &\leq \frac{1}{3} \end{aligned}$$

Warm-Up Problem: All Zeros



All-Zeros(A, ∈)

Repeat s times:

Choose random i in [1, n] .

if A[i] = 1 then return FALSE.

Return TRUE.

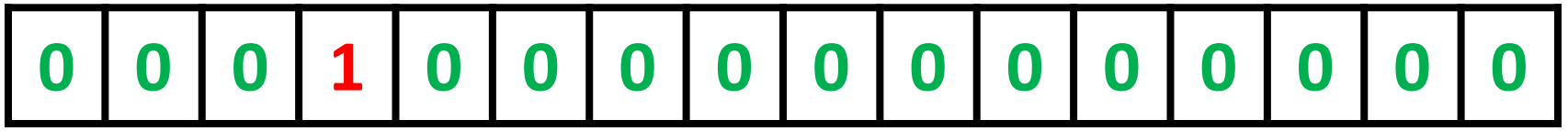
Fix $s = \frac{2}{\epsilon}$

$\text{time} = \frac{2}{\epsilon}$

Claim 1: If array is all 0's, then always returns TRUE.

Claim 2: If array has ≥ εn 1's, then returns FALSE with probability at least 2/3.

Warm-Up Problem: All Zeros



All-Zeros(A, ∈)

Repeat s times:

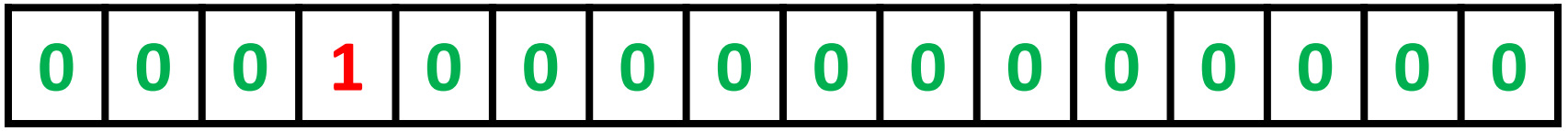
Choose random i in [1, n] .

if A[i] = 1 then return FALSE.

Return TRUE.

What if we want the algorithm to be correct with probability $\geq 1 - \delta$?

Warm-Up Problem: All Zeros



All-Zeros(A, ∈)

Repeat s times:

Choose random i in [1, n] .

if A[i] = 1 then return FALSE.

Return TRUE.

Test:

Find s.

What if we want the algorithm to be correct with probability ≥ 1 – δ?

Fix $s = ?$

Warm-Up Problem: How Many Zeros

0	0	0	1	0	0	0	0	1	0	0	0	1	0	0

Input: n element array

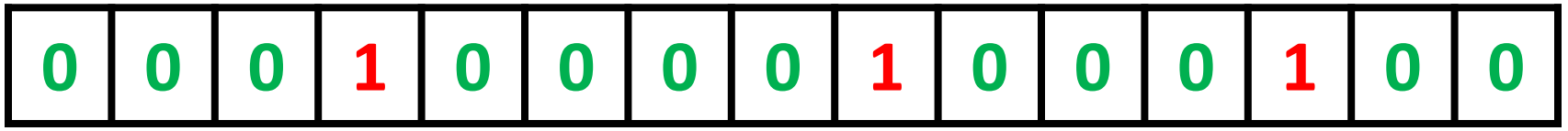
- Each element is 0 or 1.
- 0 = good test, 1 = failed test.

Output: What fraction of the array is 1's?

Approximation: ± ε

Probability correct: 2/3

Warm-Up Problem: How Many Zeros

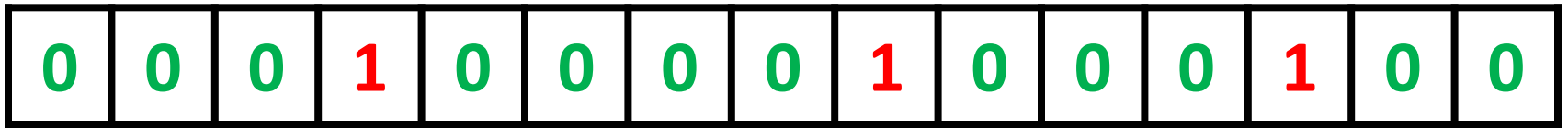


n = 16
ϵ = 0 → answer = 3/16

n = 16
ϵ = 1/16 → answer: [1/8, 1/4]

n = 16
ϵ = 1/100 → answer: [0.1775, 0.1975]

Warm-Up Problem: How Many Zeros



Fraction-Zeros(A, ε)

sum = 0

Repeat s times:

Choose random i in [1, n]

sum = sum + A[i]

Return sum/s

$$s = \frac{1}{\epsilon^2}$$

$$\text{time} = O\left(\frac{1}{\epsilon^2}\right)$$

Key tool: Hoeffding Bound

Requirements:

1) Random variables:

$Y_1, Y_2, \ldots, Y_s$ are random variables.

$$Z = \sum_{j=1}^{s} Y_j$$

2) Independent:

$Y_1, Y_2, \ldots, Y_s$ are *independent*.

3) Bounded:

Each Y_j is in the range [0,1].

Key tool: Hoeffding Bound

Requirements:

1) Random variables:

$Y_1, Y_2, \ldots, Y_s$ are random variables.

$$Z = \sum_{j=1}^{s} Y_j$$

2) Independent:

$Y_1, Y_2, \ldots, Y_s$ are *independent*.

3) Bounded:

Each Y_j is in the range [0,1].

Example:

1) Random variables:

Y_j = value of array sampled in j^{th} iteration of the loop.

2) Independent:

Each sample is independent.

3) Bounded:

Each array entry is in [0,1].

Key tool: Hoeffding Bound

If $Y_1, Y_2, \ldots, Y_s$ are independent random variables in the range [0,1] and if $Z = \sum_{j=1}^{s} Y_j$ then:

$$\Pr[|Z - \mathrm{E}[Z]| \geq \delta] \leq 2e^{-2\delta^2/s}$$

Conclusion:

The value of Z is within δ of its expected value with "good" probability.

Example: Flipping Coins

Claim:

Imagine you flip a coin n times.

Then you will see:

- at least $\frac{n}{2} - \sqrt{n}$ heads
- at most $\frac{n}{2} + \sqrt{n}$ heads

with probability at least 2/3.

Example: Flipping Coins

Claim:

Imagine you flip a coin n times.

Then you will see:

- at least $\frac{n}{2} - \sqrt{n}$ heads
- at most $\frac{n}{2} + \sqrt{n}$ heads

with probability at least 2/3.

Random variables:

1) Random variables:

$$Y_j = 1 \text{ if flip } j \text{ is heads.}$$
$$Y_j = 0 \text{ if flip } j \text{ is tails.}$$

$$Z = \sum_{j=1}^{n} Y_j$$

$$\mathrm{E}[Z] = n/2$$

2) Independent: YES

3) Bounded: YES

Example: Flipping Coins

Hoeffding Bound:

$Y_j = 1$ if flip j is heads.
$Y_j = 0$ if flip j is tails.

$Z = \sum_{j=1}^{n} Y_j$

$\mathrm{E}[Z] = n/2$

$$\begin{aligned}
\Pr[|Z - n/2| \geq \sqrt{n}] &\leq \Pr[|Z - \mathrm{E}[Z]| \geq \sqrt{n}] \\
&\leq 2e^{-2(\sqrt{n})^2/n} \\
&\leq 2e^{-2} \\
&\leq 1/3
\end{aligned}$$

Example: Flipping Coins

Claim:

Imagine you flip a coin n times.

Then you will see:

- at least $\frac{n}{2} - \sqrt{n}$ heads
- at most $\frac{n}{2} + \sqrt{n}$ heads

with probability at least 2/3.

Key tool: Hoeffding Bound

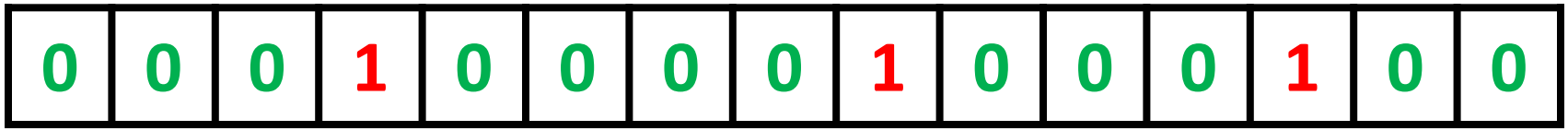
If $Y_1, Y_2, \ldots, Y_s$ are independent random variables in the range [0,1] and if $Z = \sum_{j=1}^{s} Y_j$ then:

$$\Pr[|Z - \mathrm{E}[Z]| \geq \delta] \leq 2e^{-2\delta^2/s}$$

Conclusion:

The value of Z is within δ of its expected value with "good" probability.

Warm-Up Problem: How Many Zeros



Fraction-Zeros(A, ε)

sum = 0

Repeat s times:

Choose random i in [1, n]

sum = sum + A[i]

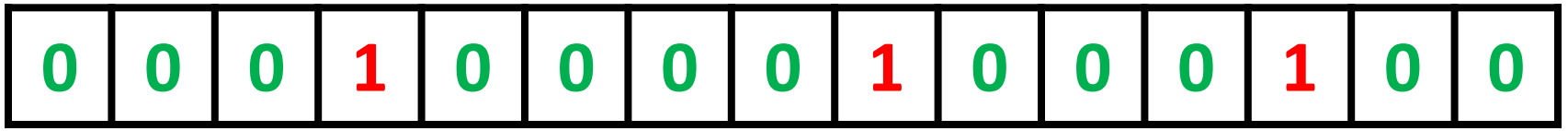
Return sum/s

$Y_j = 1$ if sample j is 1.
$Y_j = 0$ if sample j is 0.

$sum = \sum_{j=1}^{n} Y_j$

Value returned: sum/s

Warm-Up Problem: How Many Zeros



$$\begin{aligned} \mathrm{E}[Y_j] &= 1 \cdot \Pr[Y_j = 1] + 0 \cdot \Pr[Y_j = 0] \\ &= \Pr[Y_j = 1] \\ &= f \end{aligned}$$

Let f = fraction of 1's in the array.

(The "right" answer for the algorithm is f.)

Warm-Up Problem: How Many Zeros

0	0	0	1	0	0	0	0	1	0	0	0	1	0	0

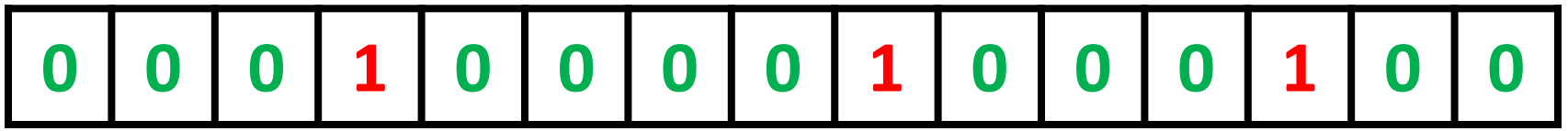
$$\begin{aligned} \mathrm{E}[Y_j] &= 1 \cdot \Pr[Y_j = 1] + 0 \cdot \Pr[Y_j = 0] \\ &= \Pr[Y_j = 1] \\ &= f \end{aligned}$$

$$\begin{aligned} \mathrm{E}[sum] &= \mathrm{E}\left[\sum_{j=1}^{s} Y_j\right] \\ &= \sum_{j=1}^{s} \mathrm{E}[Y_j] \\ &= sf \end{aligned}$$

Let f = fraction of 1's in the array.

(The "right" answer for the algorithm is f.)

Warm-Up Problem: How Many Zeros



Fraction-Zeros(A, ε)

sum = 0

Repeat s times:

Choose random i in [1, n]

sum = sum + A[i]

Return sum/s

$Y_j = 1$ if sample j is 1.
$Y_j = 0$ if sample j is 0.

$sum = \sum_{j=1}^{n} Y_j$
$\mathrm{E}[sum] = s \cdot sum$

value returned: sum/s
$\mathrm{E}[\text{value returned}] = f$

Unbiased estimator! That's good…

Use a Hoeffding Bound!

V = value returned

$Y_j = 1$ if sample j is 1.
$Y_j = 0$ if sample j is 0.

$sum = \sum_{j=1}^{n} Y_j$
$\mathrm{E}[sum] = s \cdot sum$

value returned: sum/s
$\mathrm{E}[\text{value returned}] = f$

$$\Pr[|V - f| \leq \epsilon] \quad = \quad \Pr[|V - \mathrm{E}[V] \leq \epsilon]$$

Can we use a Hoeffding Bound here?

Use a Hoeffding Bound!

V = value returned

$Y_j = 1$ if sample j is 1.
$Y_j = 0$ if sample j is 0.

$sum = \sum_{j=1}^{n} Y_j$
$\mathrm{E}[sum] = s \cdot sum$

value returned: sum/s
$\mathrm{E}[\text{value returned}] = f$

$$\Pr[|V - f| \leq \epsilon] \quad = \quad \Pr[|V - \mathrm{E}[V] \leq \epsilon]$$

Can we use a Hoeffding Bound here?

No, not yet.
V is not a sum of random variables.

Use a Hoeffding Bound!

V = value returned

$Y_j = 1$ if sample j is 1.
$Y_j = 0$ if sample j is 0.

$sum = \sum_{j=1}^{n} Y_j$
$\mathrm{E}[sum] = s \cdot sum$

value returned: sum/s
$\mathrm{E}[\text{value returned}] = f$

$$\begin{aligned} \Pr[|V - f| \leq \epsilon] &= \Pr[|V - \mathrm{E}[V] \leq \epsilon] \\ &= \Pr[|sV - sf| \leq s \cdot \epsilon] \end{aligned}$$

Multiple everything by s.

Use a Hoeffding Bound!

$V =$ value returned

$Y_j = 1$ if sample j is 1.
$Y_j = 0$ if sample j is 0.

$sum = \sum_{j=1}^{n} Y_j$
$\mathrm{E}[sum] = s \cdot sum$

value returned: sum/s
$\mathrm{E}[\text{value returned}] = f$

$$\begin{aligned} \Pr[|V - f| \leq \epsilon] &= \Pr[|V - \mathrm{E}[V] \leq \epsilon] \\ &= \Pr[|sV - sf| \leq s \cdot \epsilon] \\ &= \Pr[|sum - \mathrm{E}[sum]| \leq s \cdot \epsilon] \end{aligned}$$

Can we use a Hoeffding Bound here?

Yes.
The random variable *sum* is the sum of independent 0/1 random variables.

Use a Hoeffding Bound!

V = value returned

$Y_j = 1$ if sample j is 1.
$Y_j = 0$ if sample j is 0.

$sum = \sum_{j=1}^{n} Y_j$
$\mathrm{E}[sum] = s \cdot sum$

value returned: sum/s
$\mathrm{E}[\text{value returned}] = f$

$$\begin{aligned}
\Pr[|V - f| \leq \epsilon] &= \Pr[|V - \mathrm{E}[V] \leq \epsilon] \\
&= \Pr[|sV - sf| \leq s \cdot \epsilon] \\
&= \Pr[|sum - \mathrm{E}[sum]| \leq s \cdot \epsilon] \\
&\leq 2e^{-2s^2\epsilon^2/s}
\end{aligned}$$

Use a Hoeffding Bound!

V = value returned

$Y_j = 1$ if sample j is 1.
$Y_j = 0$ if sample j is 0.

$sum = \sum_{j=1}^{n} Y_j$
$\mathrm{E}[sum] = s \cdot sum$

value returned: sum/s
$\mathrm{E}[\text{value returned}] = f$

$$
\begin{aligned}
\Pr[|V - f| \le \epsilon] &= \Pr[|V - \mathrm{E}[V] \le \epsilon] \\
&= \Pr[|sV - sf| \le s \cdot \epsilon] \\
&= \Pr[|sum - \mathrm{E}[sum]| \le s \cdot \epsilon] \\
&\le 2e^{-2s^2\epsilon^2/s} \\
&\le 2e^{-2s\epsilon^2}
\end{aligned}
$$

Use a Hoeffding Bound!

V = value returned

$Y_j = 1$ if sample j is 1.
$Y_j = 0$ if sample j is 0.

$sum = \sum_{j=1}^{n} Y_j$
$\mathrm{E}[sum] = s \cdot sum$

value returned: sum/s
$\mathrm{E}[\text{value returned}] = f$

$$
\begin{aligned}
\Pr[|V - f| \leq \epsilon] &= \Pr[|V - \mathrm{E}[V] \leq \epsilon] \\
&= \Pr[|sV - sf| \leq s \cdot \epsilon] \\
&= \Pr[|sum - \mathrm{E}[sum]| \leq s \cdot \epsilon] \\
&\leq 2e^{-2s^2\epsilon^2/s} \\
&\leq 2e^{-2s\epsilon^2} \\
&\leq 2e^{-2\epsilon^2/\epsilon^2}
\end{aligned}
$$

$$\mathrm{s} = \frac{1}{\epsilon^2}$$

Use a Hoeffding Bound!

V = value returned

$Y_j = 1$ if sample j is 1.
$Y_j = 0$ if sample j is 0.

$sum = \sum_{j=1}^{n} Y_j$
$\mathrm{E}[sum] = s \cdot sum$

value returned: sum/s
$\mathrm{E}[\text{value returned}] = f$

$$
\begin{aligned}
\Pr[|V - f| \leq \epsilon] &= \Pr[|V - \mathrm{E}[V] \leq \epsilon] \\
&= \Pr[|sV - sf| \leq s \cdot \epsilon] \\
&= \Pr[|sum - \mathrm{E}[sum]| \leq s \cdot \epsilon] \\
&\leq 2e^{-2s^2\epsilon^2/s} \\
&\leq 2e^{-2s\epsilon^2} \\
&\leq 2e^{-2\epsilon^2/\epsilon^2} \\
&\leq 2e^{-2} \\
&\leq 1/3
\end{aligned}
$$

$$s = \frac{1}{\epsilon^2}$$

Conclusion: value returned V is equal to f ± ε w.p. ≥ 2/3.

Warm-Up Problem: How Many Zeros

0	0	0	1	0	0	0	0	1	0	0	0	1	0	0

Fraction-Zeros(A, ε)

sum = 0

Repeat s times:

Choose random i in [1, n]

sum = sum + A[i]

Return sum/s

$$s = \frac{1}{\epsilon^2}$$

$$\text{time} = O\left(\frac{1}{\epsilon^2}\right)$$

Conclusion: sum/s is equal to f ± ε w.p. ≥ 2/3.

Algorithms

Today

Toy example 1: array all 0's?

- Gap-style question:
  All 0's or far from all 0's?

Toy example 2: Faction of 1's?

- Additive ± ε approximation
- Hoeffding Bound

Is the graph connected?

- Gap-style question.
- O(1) time algorithm.
- Correct with probability 2/3.

Next week:

Number of connected components in a graph.

- *Additive* approximation algorithm.

Weight of MST

- *Multiplicative* approximation algorithm.

Connectivity

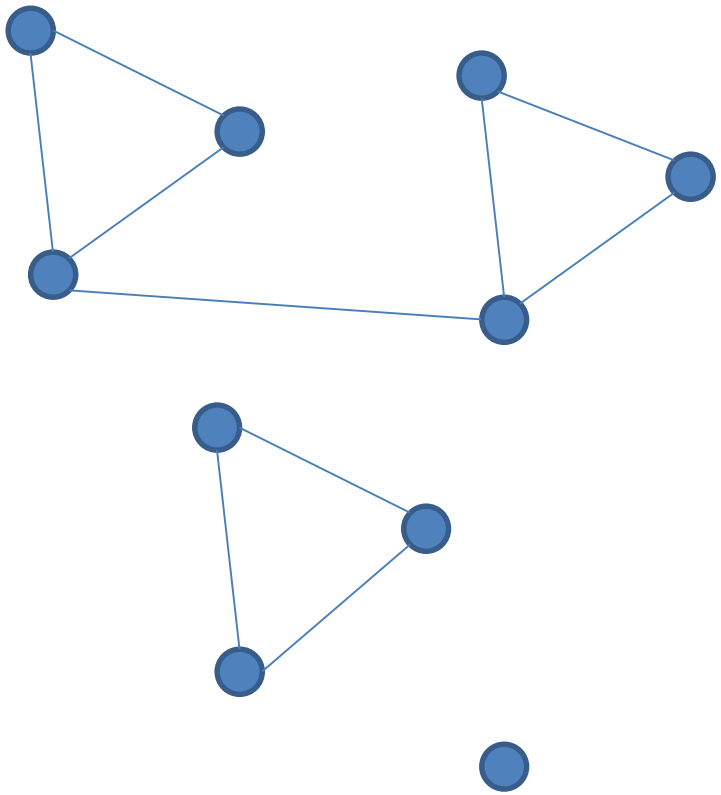
Assumptions:

Graph G = (V,E)

- Undirected
- n nodes
- m edges
- maximum degree d

Output:

Is the graph connected?



Example: output NO

Connectivity

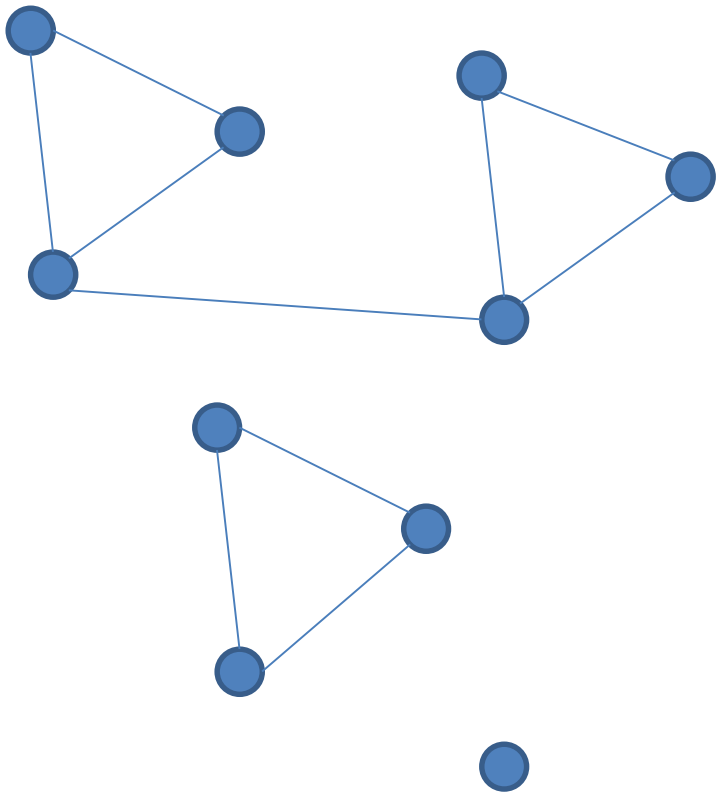
Assumptions:

Graph G = (V,E)

- Undirected
- n nodes
- m edges
- maximum degree d

Format:

- Graph is given as an adjacency list.
- Query: nbr(u, i) returns i^{th} neighbor of node u.



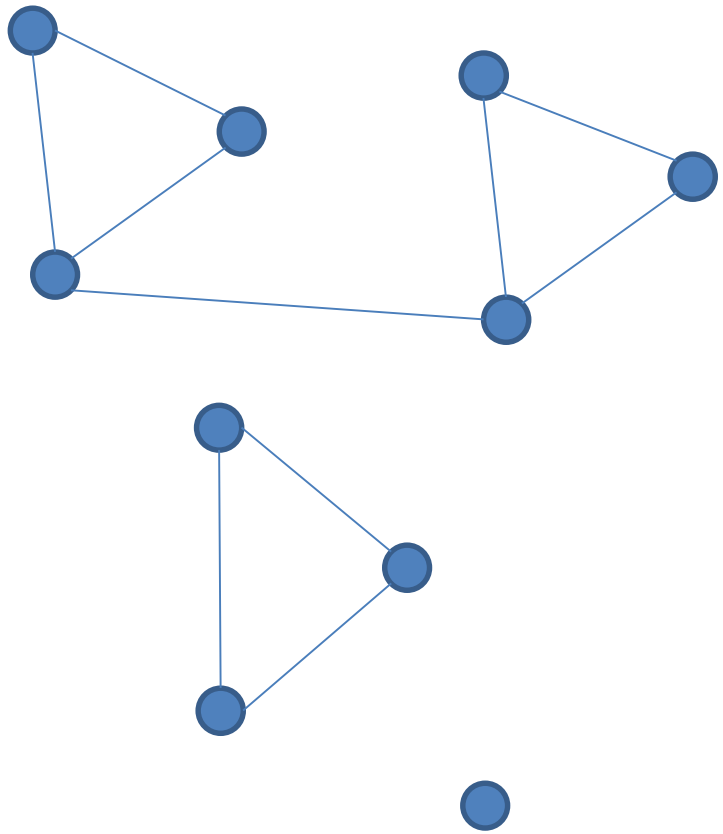
Example: output NO

Connectivity

Exact answer:

BFS solves in: $O(m+n)$

Cannot do faster than: $\Omega(m+n)$



Challenge #2:

Prove it.

Connectivity

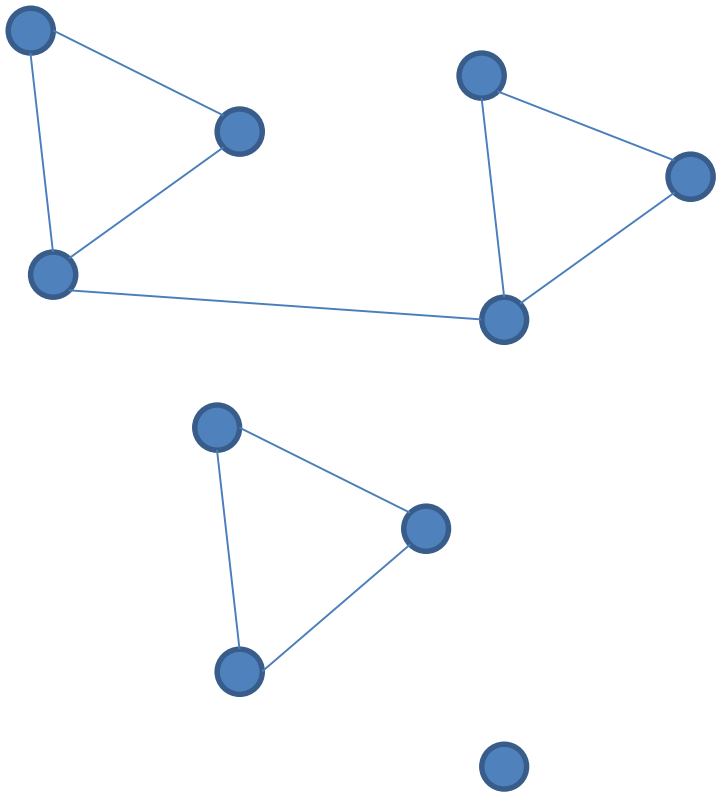
Assumptions:

Graph G = (V,E)

- Undirected
- n nodes
- m edges
- maximum degree d

Output:

Is the graph *close to*" connected or "*far from*" connected?



Example: output NO

Connectivity

Gap Approximation:

If G is connected:

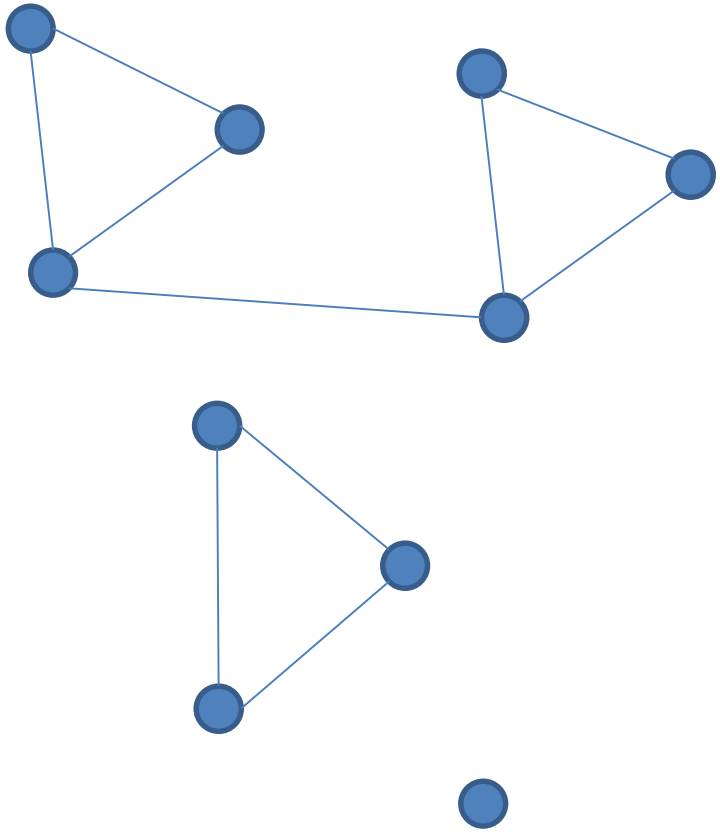
Return TRUE

If G is ε-far from connected:

Return FALSE

Otherwise:

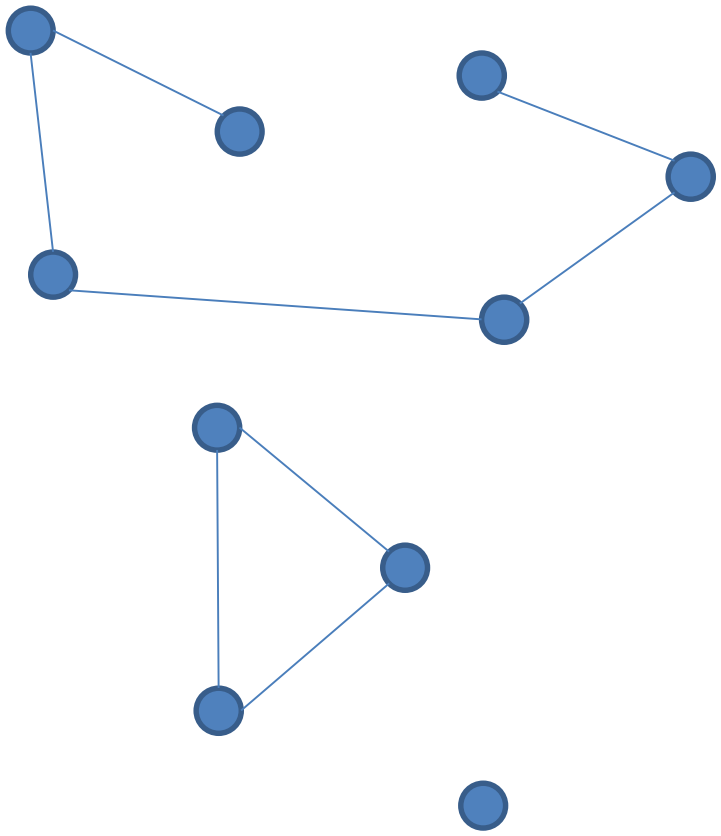
Don't care.



Connectivity

Definition:

G is ε-close to connected if you can add/modify at most εnd entries in the adjacency list to create a new graph G' that is connected.

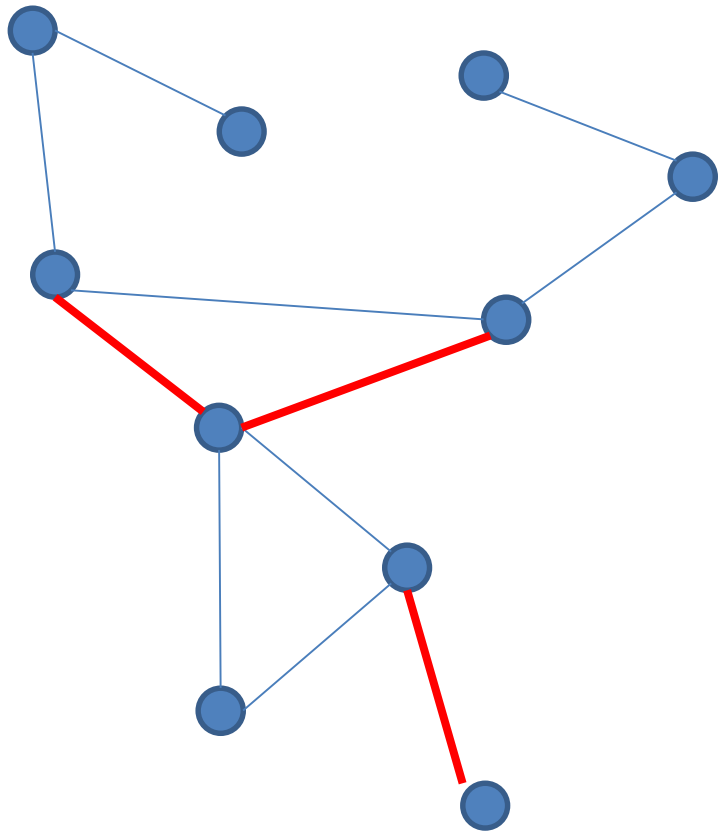


Note: adding an edge requires modifying two entries in the adjacency list.

Connectivity

Definition:

G is ε-close to connected if you can add/modify at most εnd entries in the adjacency list to create a new graph G' that is connected.

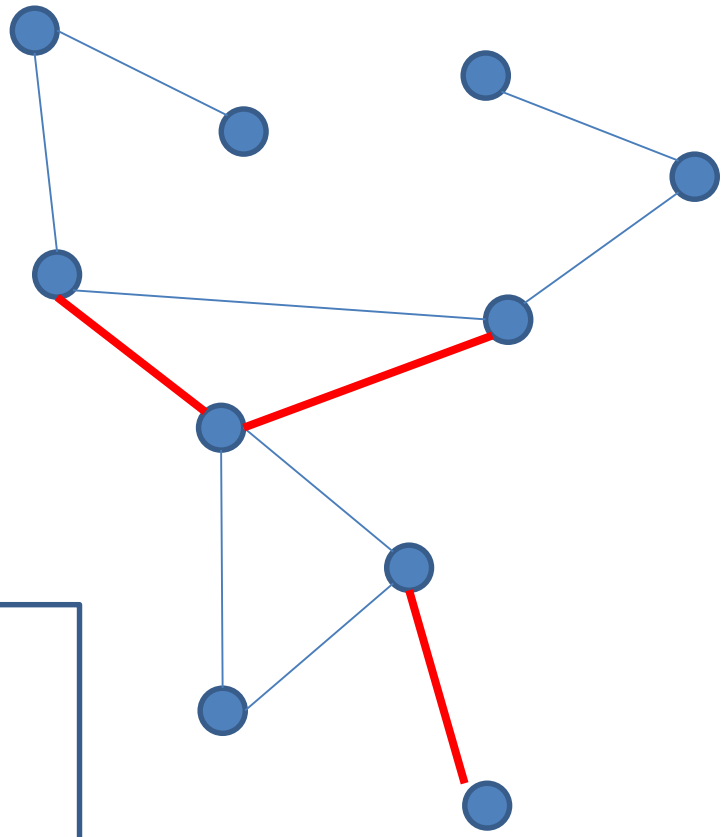


Note: adding an edge requires modifying two entries in the adjacency list.

Connectivity

Definition:

G is ε-close to connected if you can add/modify at most εnd entries in the adjacency list to create a new graph G' that is connected.



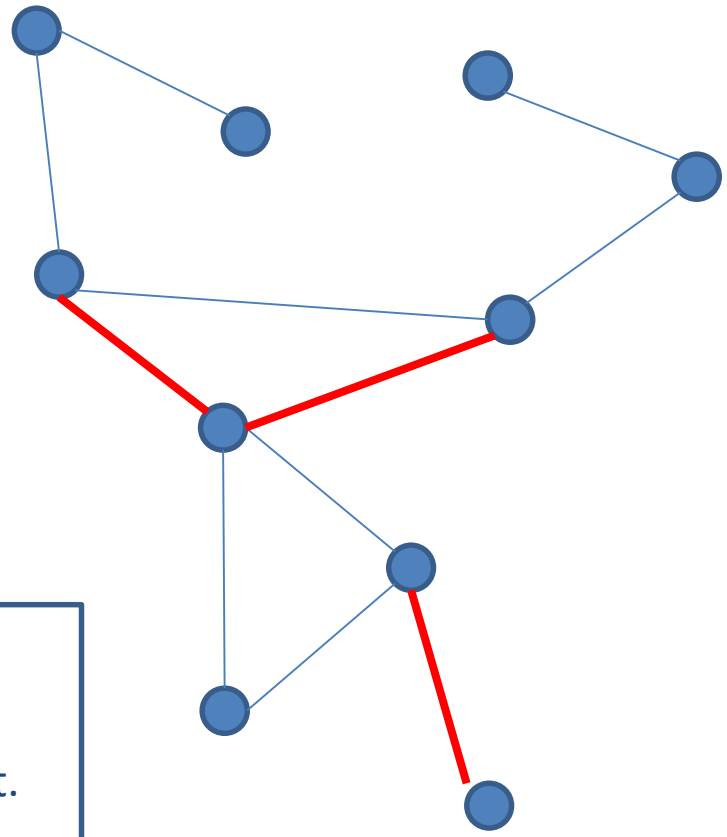
Example: n = 10, d = 3

Note: adding an edge requires modifying two entries in the adjacency list.

Connectivity

Definition:

G is ε-close to connected if you can add/modify at most εnd entries in the adjacency list to create a new graph G' that is connected.



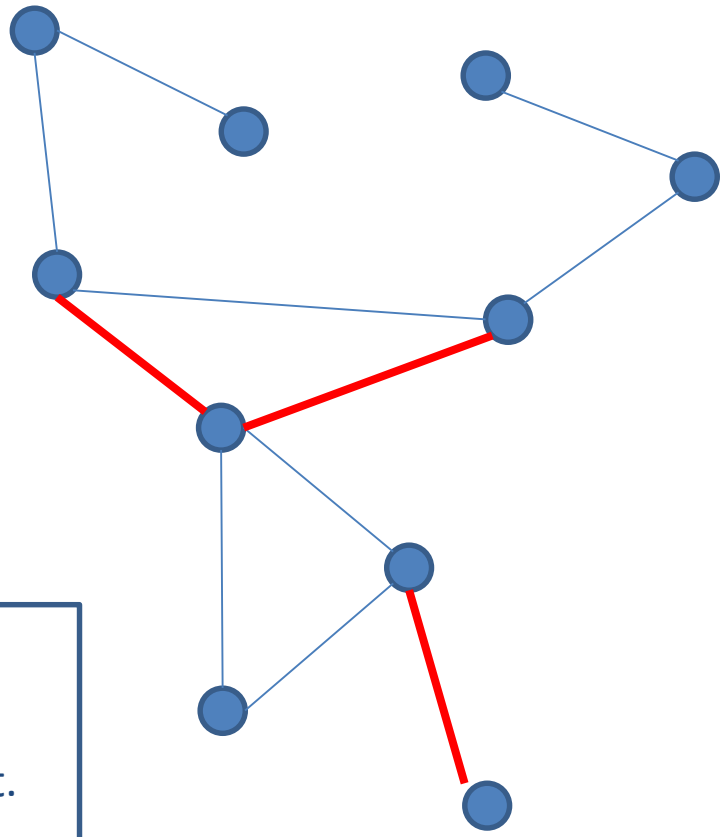
Example: n = 10, d = 3
Add 3 edges to connect graph.
Modify 6 entries in adjacency list.

Note: adding an edge requires modifying two entries in the adjacency list.

Connectivity

Definition:

G is ε-close to connected if you can add/modify at most εnd entries in the adjacency list to create a new graph G' that is connected.



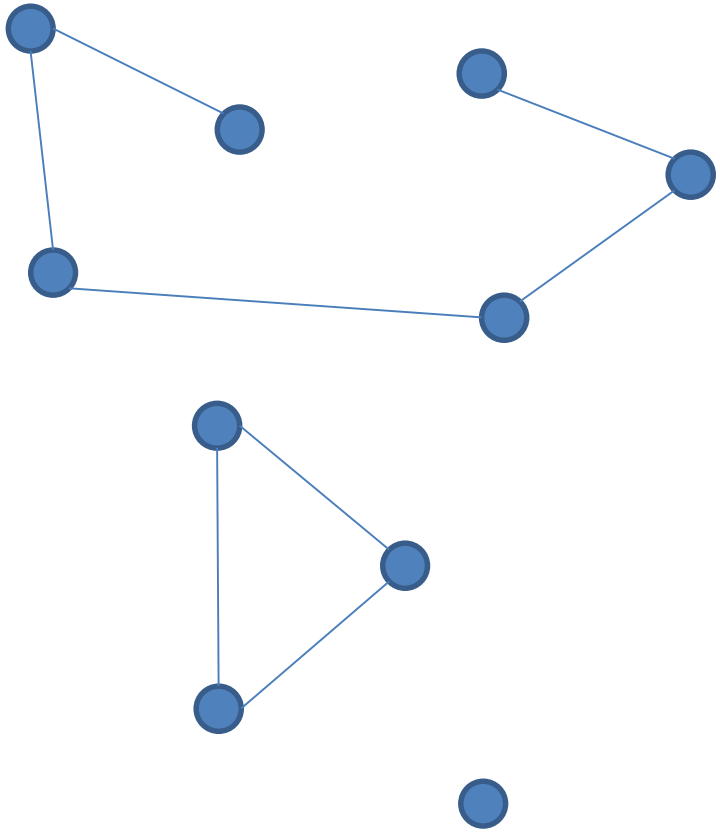
Example: n = 10, d = 3
Add 3 edges to connect graph.
Modify 6 entries in adjacency list.
nd = 30
G is 1/5-close to connected.

Note: adding an edge requires modifying two entries in the adjacency list.

Connectivity

Definition:

G is ε-close to connected if you can add/modify at most εnd entries in the adjacency list to create a new graph G' that is connected.



G is ε-far from connected if it is not ε-close to connected.

Note: adding an edge requires modifying two entries in the adjacency list.

Connectivity

Assumptions:

Graph G = (V,E)

- Undirected
- n nodes
- m edges
- maximum degree d

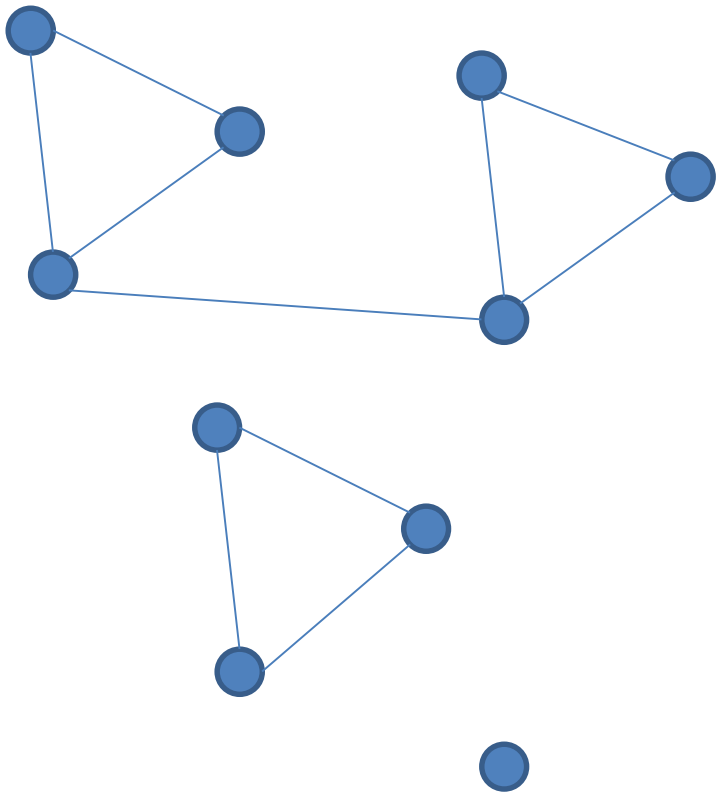
Output:

If G is connected: return TRUE.

If G is ε-far from connected: return FALSE.

Else: don't care.

Correct: with probability ≥ 2/3.

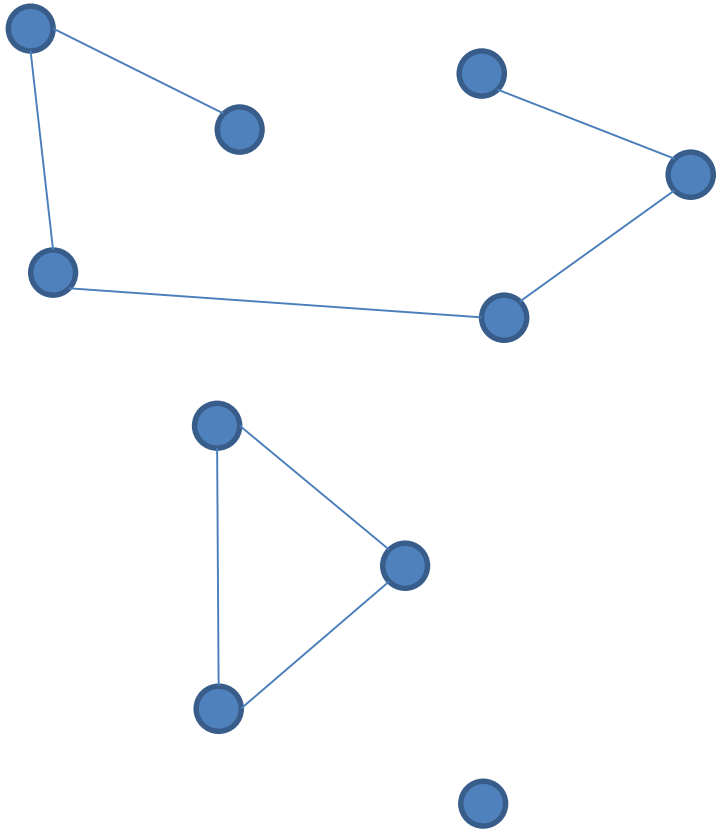


Example: output NO

Key Claim

Lemma:

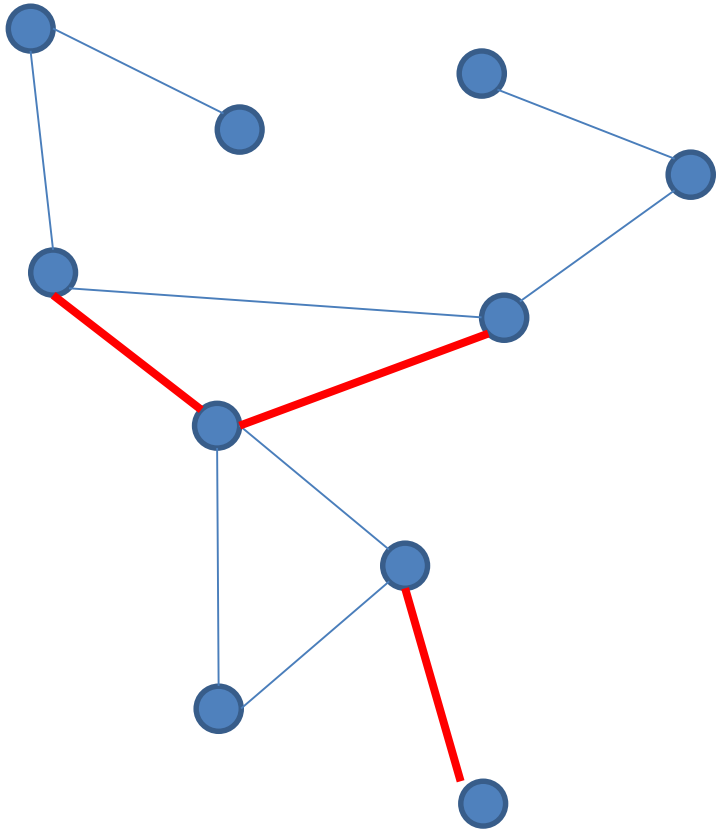
If G is ε-far from connected, then it has $\varepsilon dn/4$ connected components.



Key Claim

Lemma:

If G is ε-far from connected, then it has $\geq \varepsilon dn/4$ connected components.



Proof:

Assume G has $\leq \varepsilon dn/4$ connected components.

Then add $\varepsilon dn/4-1$ edges to build connected graph G'. That requires modifying $\leq \varepsilon dn/2$ entries in adjacency list.

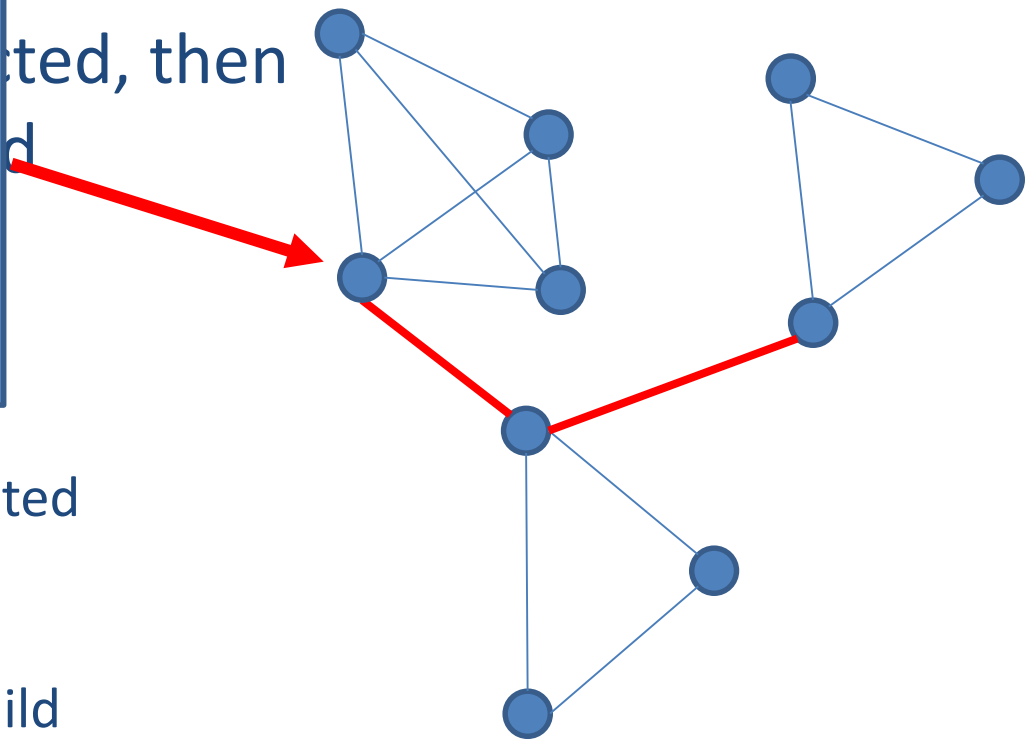
➔

G is ε-close to connected.

Key Claim

Oops!

Cannot always add an edge without increasing the degree of the graph.



Proof:

Assume G has ≤ $\varepsilon dn/4$ connected components.

Then add $\varepsilon dn/4-1$ edges to build connected graph G'. That requires modifying ≤ $\varepsilon dn/2$ entries in adjacency list.

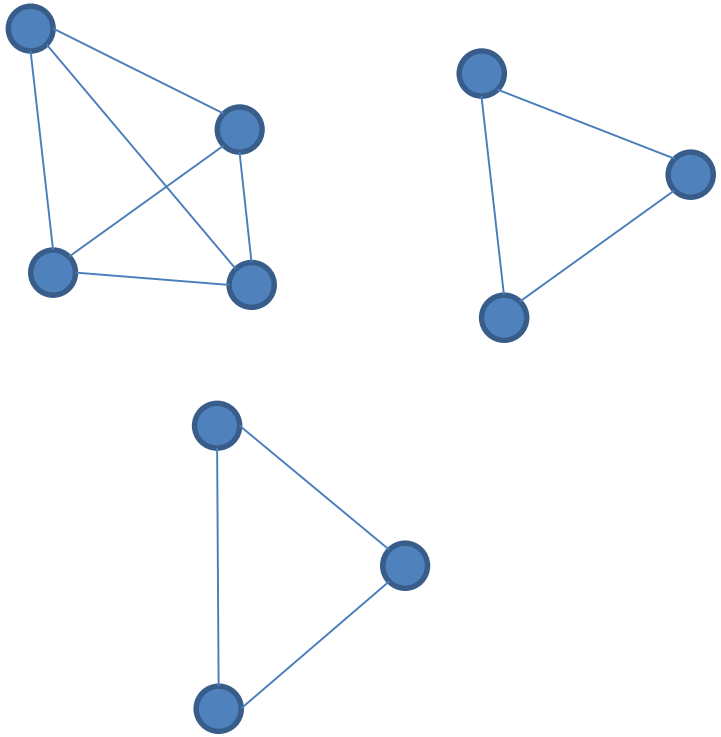
➔

G is ε-close to connected.

Key Claim

Lemma:

If G is ε-far from connected, then it has $\geq \varepsilon dn/4$ connected components.



Proof:

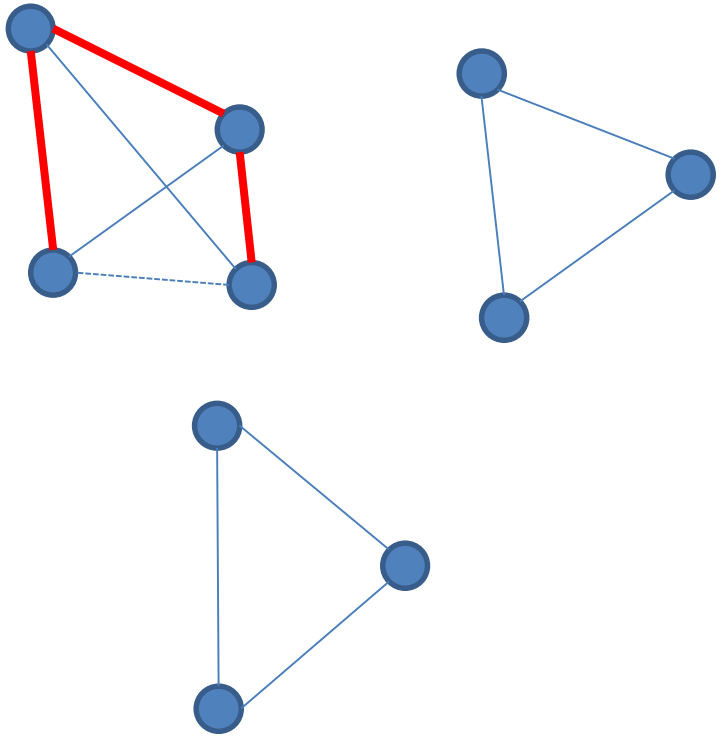
Assume G has $\leq \varepsilon dn/4$ connected components.

For each connected component, if every node has degree d:

Key Claim

Lemma:

If G is ε-far from connected, then it has $\geq \varepsilon dn/4$ connected components.



Proof:

Assume G has $\leq \varepsilon dn/4$ connected components.

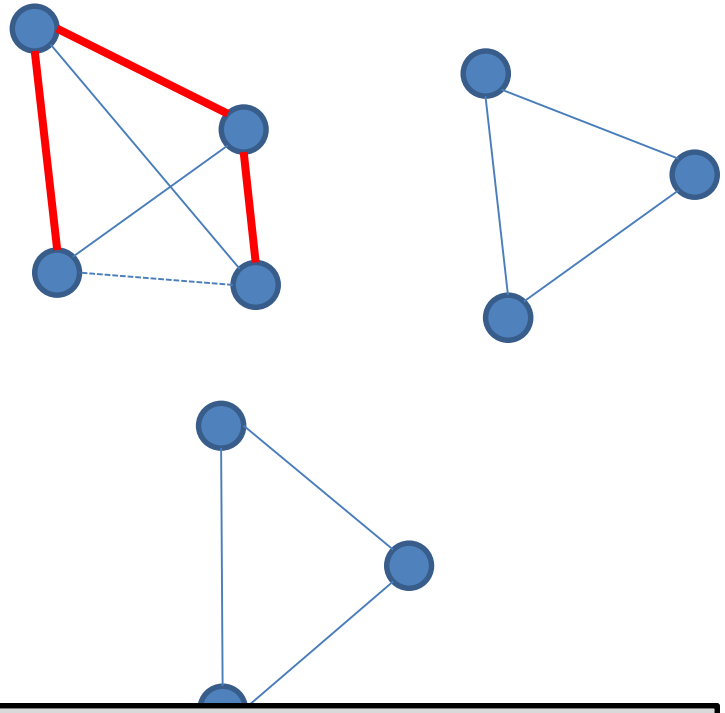
For each connected component, if every node has degree d:

If it has k nodes, find a spanning tree with k-1 edges. Remove any one edge not in spanning tree.

Key Claim

Lemma:

If G is ε-far from connected, then it has $\geq \varepsilon dn/4$ connected components.



Proof:

Assume G has $\leq \varepsilon dn/4$ connected components.

For each connected component, if every node has degree d:

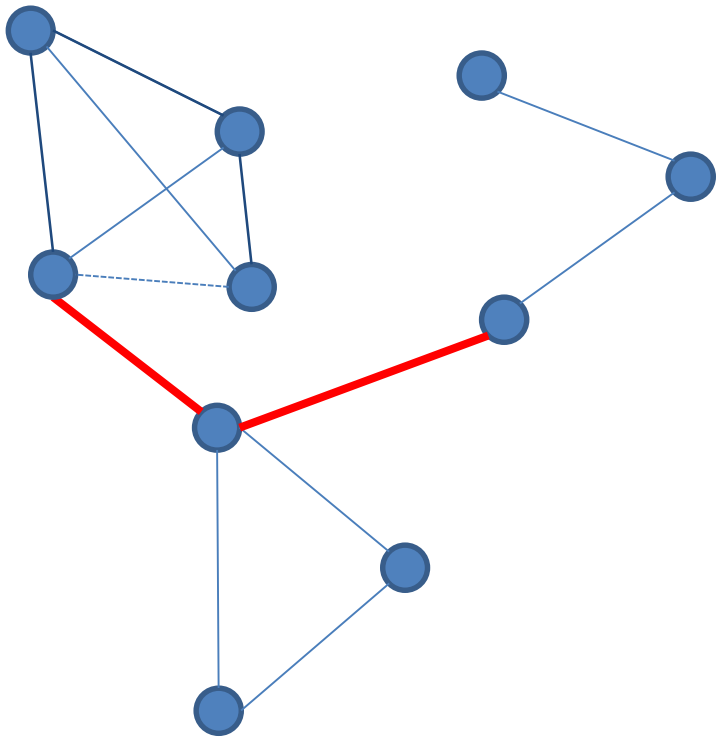
$$\text{number of edges} = k(k-1)/2 > k-1$$

If it has k nodes, find a spanning tree with k-1 edges. Remove any one edge not in spanning tree.

Key Claim

Lemma:

If G is ε-far from connected, then it has $\geq \varepsilon dn/4$ connected components.



Proof:

Assume G has $\leq \varepsilon dn/4$ connected components.

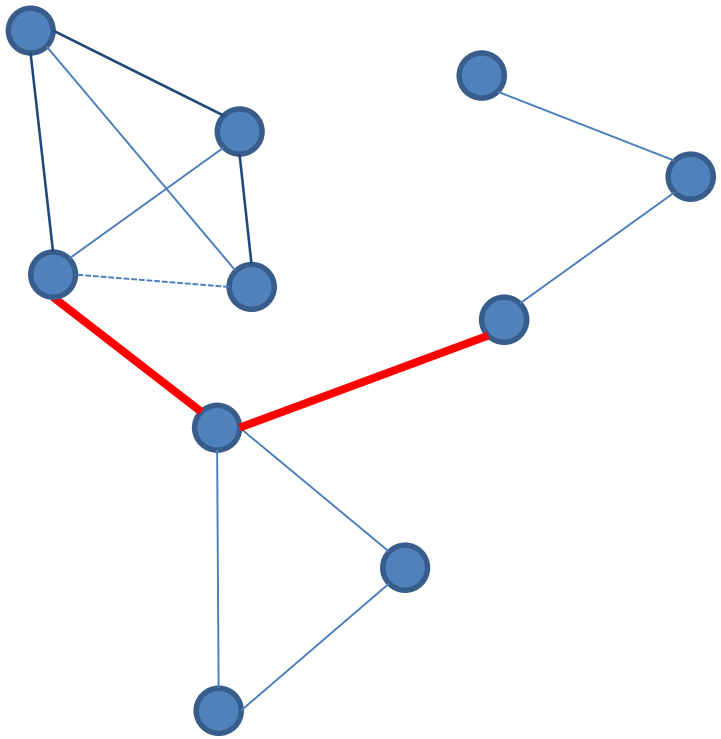
Delete $\leq \varepsilon dn/4$ edges so each connected component has at least one node with degree $< d$.

Then add $\leq \varepsilon dn/4-1$ edges to build connected graph G'.

Key Claim

Lemma:

If G is ε-far from connected, then it has $\geq \varepsilon dn/4$ connected components.



Proof:

Assume G has $\leq \varepsilon dn/4$ connected components.

Delete $\leq \varepsilon dn/4$ edges so each connected component has at least one node with degree $< d$.

Then add $\leq \varepsilon dn/4-1$ edges to build connected graph G'.

Modifies $\leq \varepsilon dn$ entries in adjacency list.

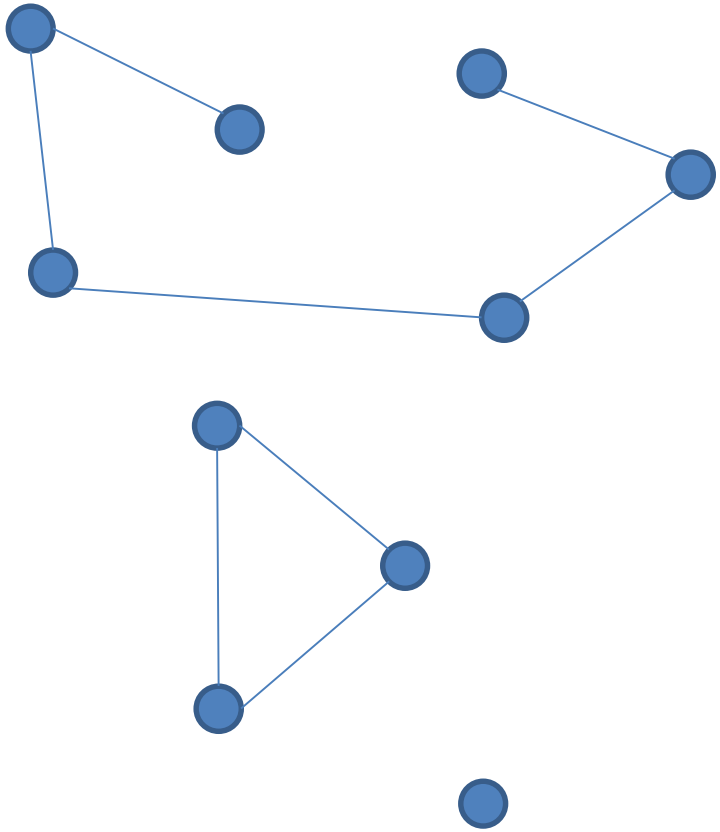
➔

G is ε-close to connected.

Key Claim

Lemma:

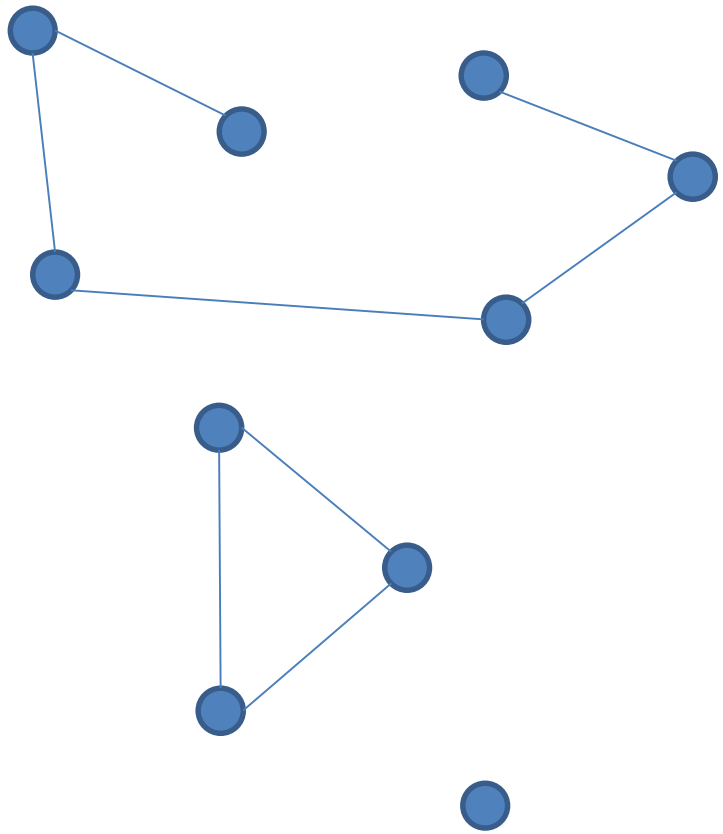
If G is ε-far from connected, then it has $\varepsilon dn/4$ connected components.



Key Claim 2

Lemma:

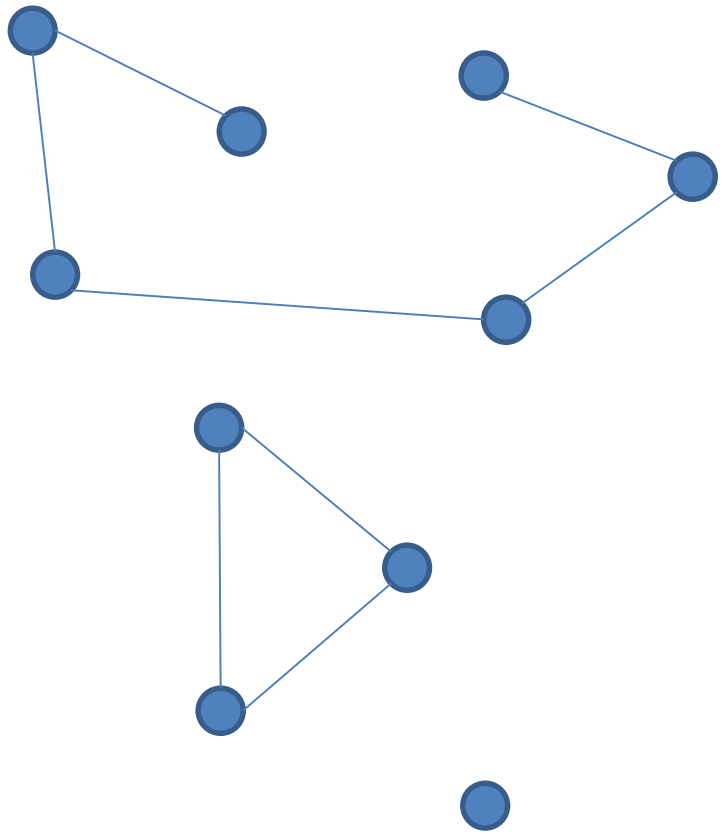
If G is ε-far from connected, then it has $\varepsilon dn/8$ connected components of size $\leq 8/\varepsilon d$.



Key Claim 2

Lemma:

If G is ε-far from connected, then it has $\varepsilon dn/8$ connected components of size $\leq 8/\varepsilon d$.



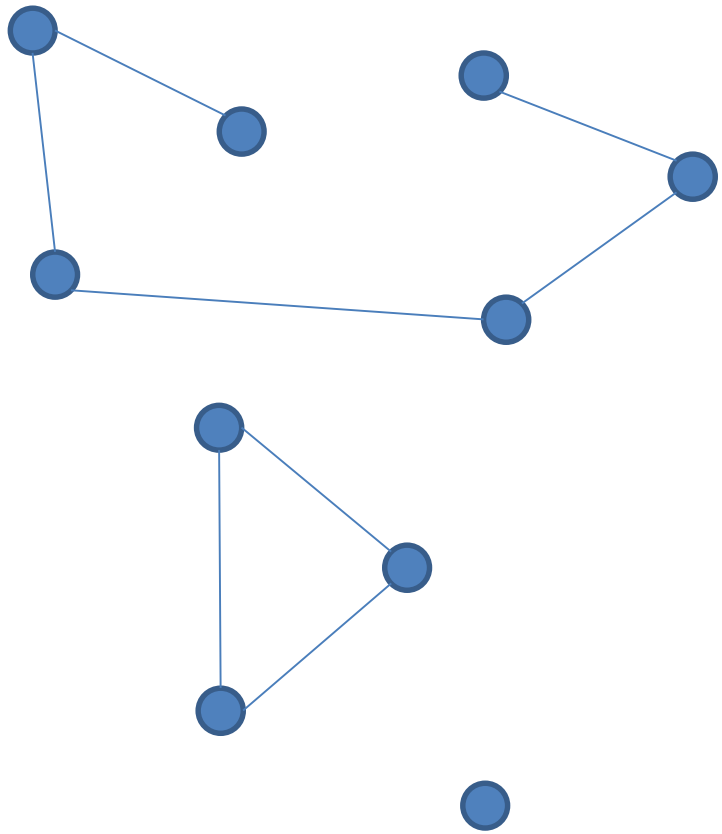
Proof:

Counting argument.

Key Claim 2

Lemma:

If G is ε-far from connected, then it has $\varepsilon dn/8$ connected components of size $\leq 8/\varepsilon d$.



Proof:

Counting argument.

Assume not.

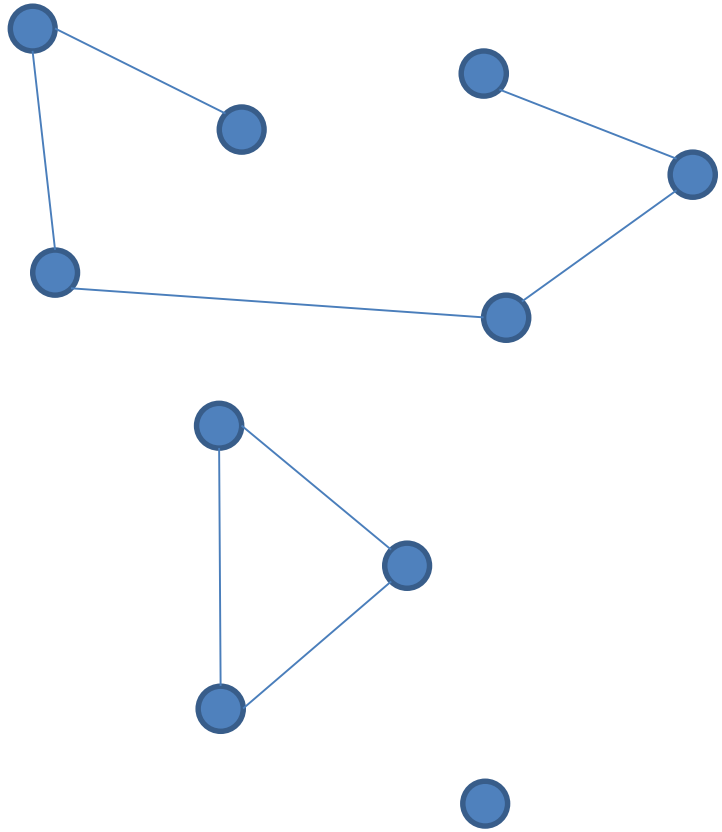
- There are at least $\varepsilon dn/4$ connected components.

At most half can be twice the average size…

Key Claim 2

Lemma:

If G is ε-far from connected, then it has $\varepsilon dn/8$ connected components of size $\leq 8/\varepsilon d$.



Proof:

Counting argument.

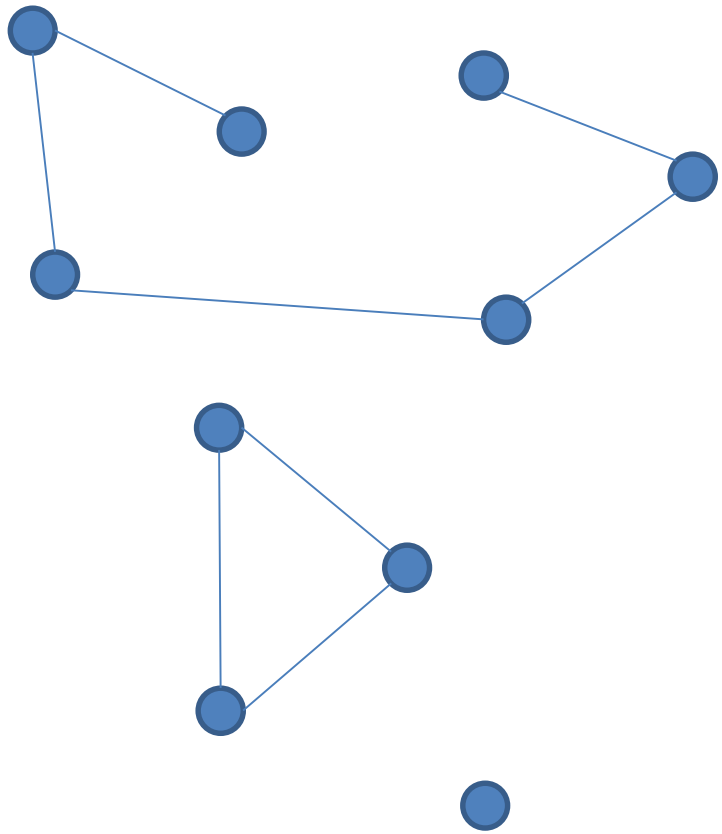
Assume not.

- There are at least $\varepsilon dn/4$ connected components.
- At most $\varepsilon dn/8$ are of size $\leq 8/\varepsilon d$.
- At least $\varepsilon dn/8$ are of size $> 8/\varepsilon d$.

Key Claim 2

Lemma:

If G is ε-far from connected, then it has $\varepsilon dn/8$ connected components of size $\leq 8/\varepsilon d$.



Proof:

Counting argument.

Assume not.

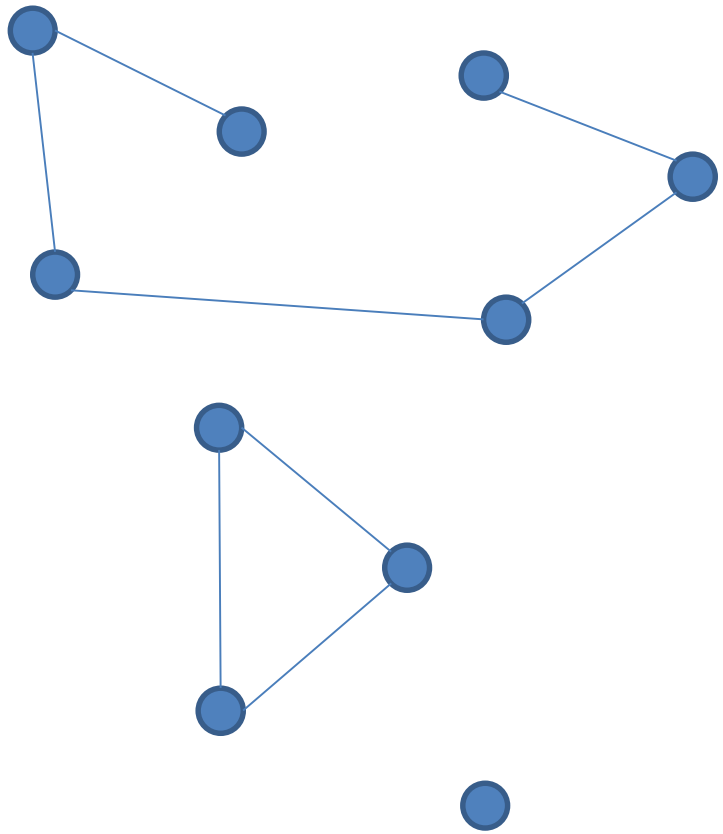
- There are at least $\varepsilon dn/4$ connected components.
- At most $\varepsilon dn/8$ are of size $\leq 8/\varepsilon d$.
- At least $\varepsilon dn/8$ are of size $> 8/\varepsilon d$.

$$\left(\frac{\epsilon dn}{8}\right)\left\lfloor\frac{8}{\epsilon d}+1\right\rfloor > n$$ ➔ CONTRADICTION

Key Claim 2

Lemma:

If G is ε-far from connected, then it has $\varepsilon dn/8$ connected components of size $\leq 8/\varepsilon d$.



Proof:

Counting argument.

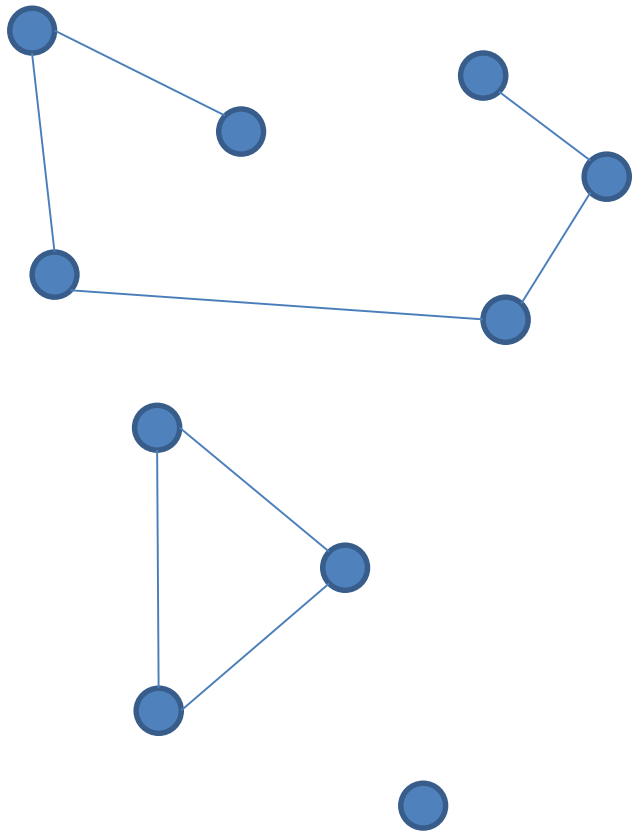
Connectivity

Connected(G, n, d, ε)

Repeat $16/\varepsilon d$ times:

- Choose random node u.
- Do a BFS from u, stopping after $8/\varepsilon d$ nodes are found.
- If CC of u has $\leq 8/\varepsilon d$ nodes, return **FALSE**.

Return **TRUE**



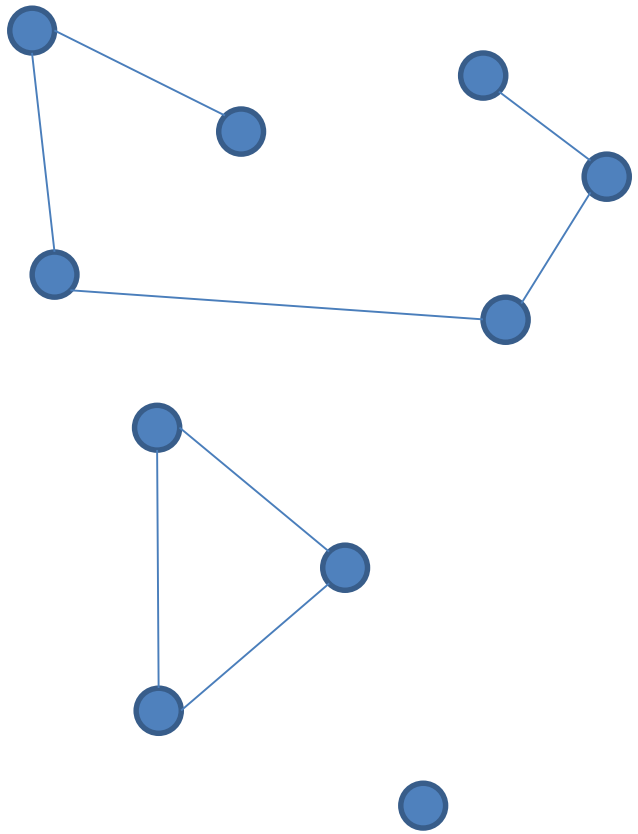
Connectivity

Connected(G, n, d, ε)

Repeat $16/\varepsilon d$ times:

- Choose random node u.
- Do a BFS from u, stopping after $8/\varepsilon d$ nodes are found.
- If CC of u has $\leq 8/\varepsilon d$ nodes, return **FALSE**.

Return **TRUE**



Claim: Each BFS takes time at most $d(8/\varepsilon d)$.

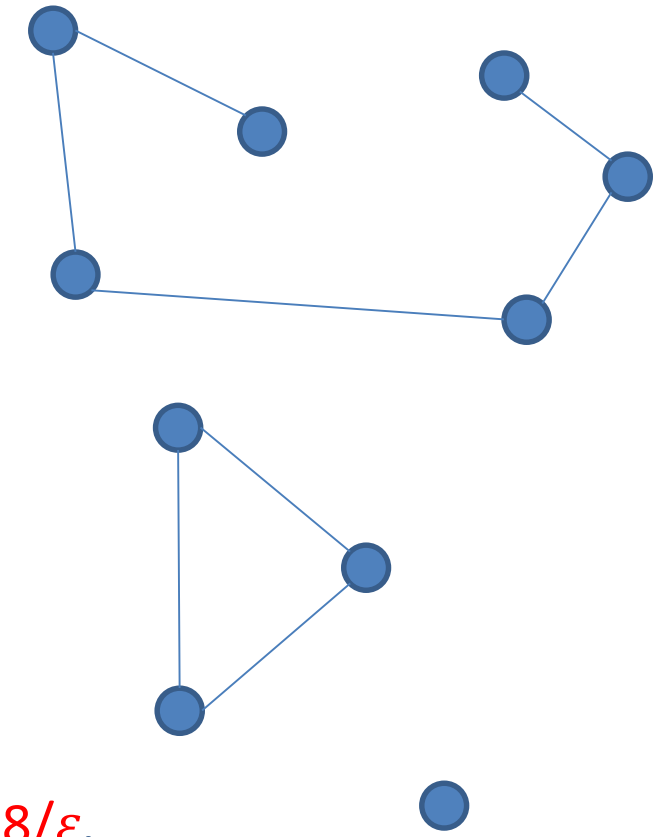
Connectivity

Connected(G, n, d, ε)

Repeat $16/\varepsilon d$ times:

- Choose random node u.
- Do a BFS from u, stopping after $8/\varepsilon d$ nodes are found.
- If CC of u has $\leq 8/\varepsilon d$ nodes, return **FALSE**.

Return **TRUE**



Claim: Each BFS takes time at most $d(8/\varepsilon d) = 8/\varepsilon$.

Proof: Explore at most $(8/\varepsilon d)$ nodes of degree at most d.

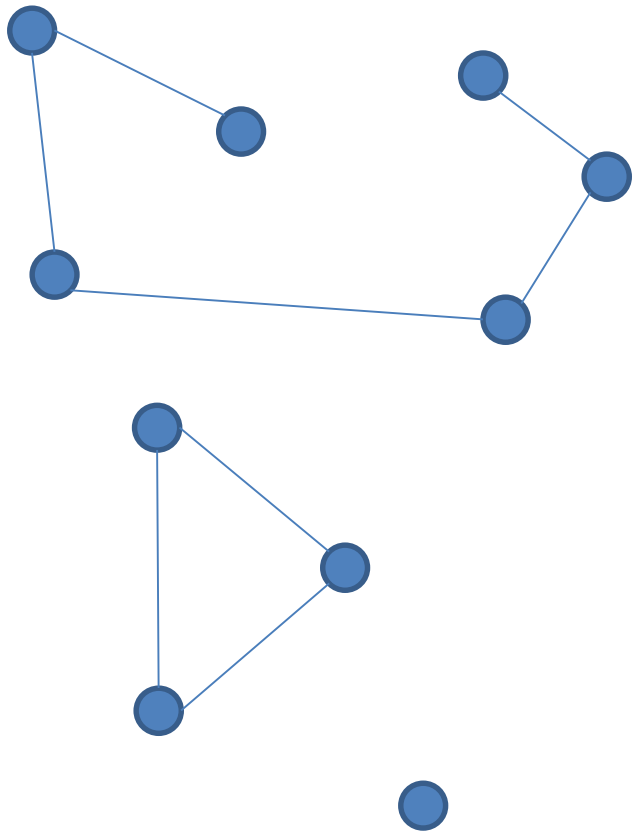
Connectivity

Connected(G, n, d, ε)

Repeat $16/\varepsilon d$ times:

- Choose random node u.
- Do a BFS from u, stopping after $8/\varepsilon d$ nodes are found.
- If CC of u has $\leq 8/\varepsilon d$ nodes, return **FALSE**.

Return **TRUE**



Claim: Total time is $O(1/\varepsilon^2 d)$.

Proof: $\left(\frac{16}{\epsilon d}\right)\left(\frac{8}{\epsilon}\right) = O\left(\frac{1}{\epsilon^2 d}\right)$

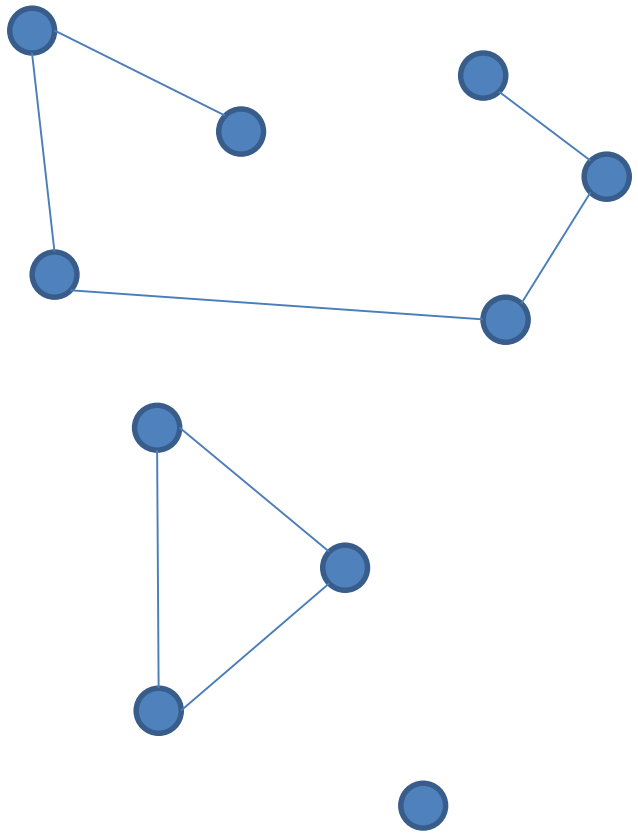
Connectivity

Connected(G, n, d, ε)

Repeat $16/\varepsilon d$ times:

- Choose random node u.
- Do a BFS from u, stopping after $8/\varepsilon d$ nodes are found.
- If CC of u has $\leq 8/\varepsilon d$ nodes, return **FALSE**.

Return **TRUE**



Claim: If G is connected, returns **TRUE**.

Proof: Immediate. No component has $\leq 8/\varepsilon d$ nodes.

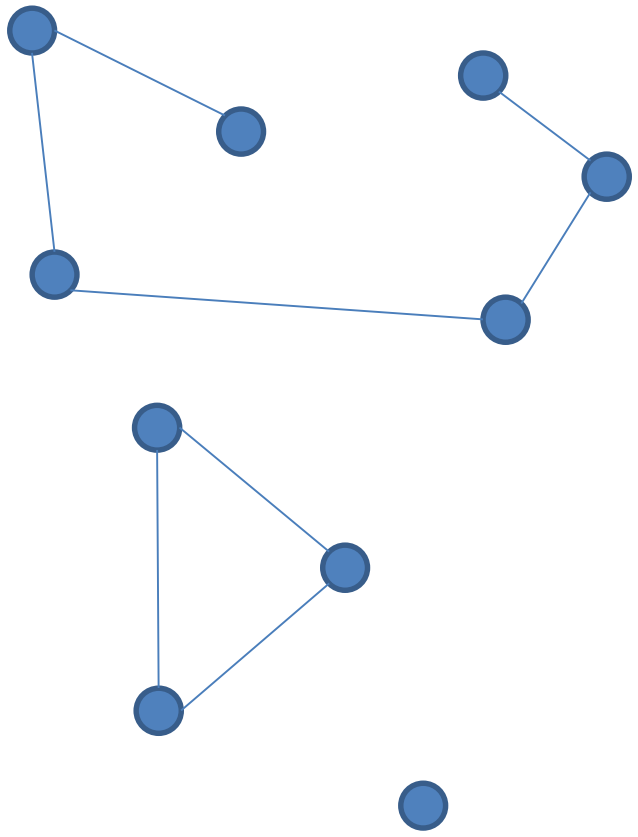
Connectivity

Connected(G, n, d, ε)

Repeat $16/\varepsilon d$ times:

- Choose random node u.
- Do a BFS from u, stopping after $8/\varepsilon d$ nodes are found.
- If CC of u has $\leq 8/\varepsilon d$ nodes, return **FALSE**.

Return **TRUE**



Claim: If G is ε-far from connected, then returns **FALSE** with probability $\geq 2/3$.

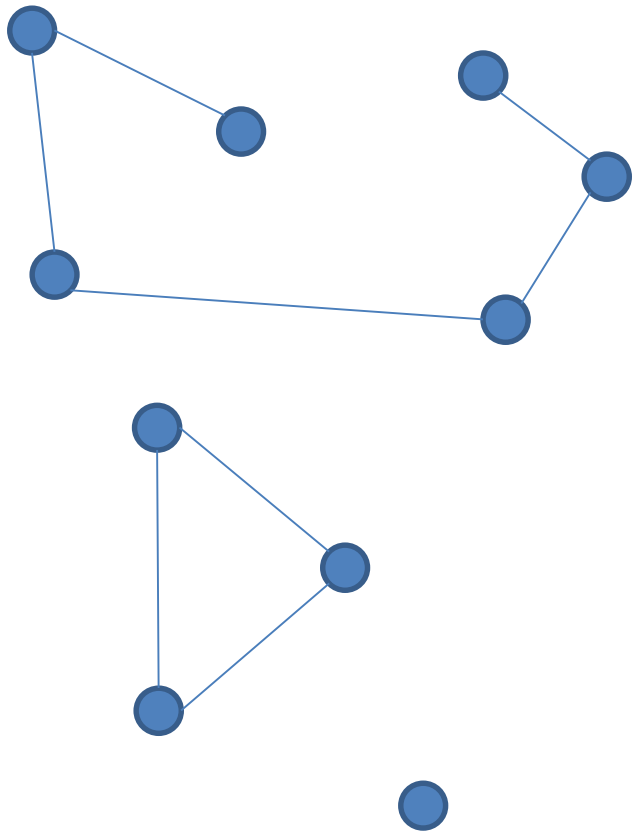
Proof: …

Connectivity

Claim: If G is ε-far from connected, then returns **FALSE** with probability ≥ 2/3.

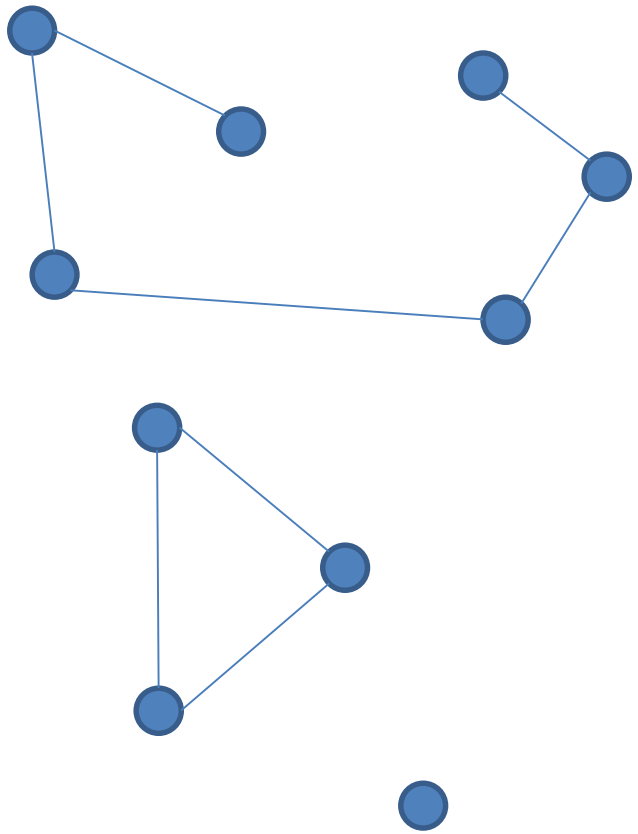
Proof:

If G is ε-far from connected, then it has at least $\varepsilon dn/8$ connected components of size $\leq 8/\varepsilon d$.



Connectivity

Claim: If G is ε-far from connected, then returns **FALSE** with probability ≥ 2/3.



Proof:

If G is ε-far from connected, then it has at least $\varepsilon dn/8$ connected components of size ≥ 1 and ≤ $8/\varepsilon d$.

Each iteration / sample has probability at least $\geq \frac{\frac{\epsilon dn}{8}}{n} \geq \frac{\epsilon d}{8}$ of finding small connected component and returning **FALSE**.

Connectivity

Death Bed Fact: (0 < x < 1)

$$e^{-x} = 1 - x + x^2/2 - \dots$$

$$\frac{1}{e^2} \leq \left(1 - \frac{1}{x}\right)^x \leq \frac{1}{e}$$

Claim: If G is ε-far from connected, then returns **FALSE** with probability ≥ 2/3.

Proof:

Each iteration / sample has probability at least $\geq \frac{\frac{\epsilon d n}{8}}{n} \geq \frac{\epsilon d}{8}$ of finding small connected component and returning **FALSE**.

$$\begin{aligned}
\Pr[\text{no iteration finds a small connected component}] &\leq \left(1 - \frac{\epsilon d}{8}\right)^s \\
&\leq \left(1 - \frac{\epsilon d}{8}\right)^{\frac{16}{\epsilon d}} \\
&\leq e^{-2} \\
&\leq 1/3
\end{aligned}$$

Connectivity

Death Bed Fact: (0 < x < 1)

$$e^{-x} = 1 - x + x^2/2 - \dots$$

$$\frac{1}{e^2} \leq \left(1 - \frac{1}{x}\right)^x \leq \frac{1}{e}$$

Claim: If G is ε-far from connected, then returns **FALSE** with probability ≥ 2/3.

Proof:

Each iteration / sample has probability at least $\geq \frac{\frac{\epsilon d n}{8}}{n} \geq \frac{\epsilon d}{8}$ of finding small connected component and returning false.

$$\begin{aligned}
\Pr[\text{no iteration finds a small connected component}] &\leq \left(1 - \frac{\epsilon d}{8}\right)^s \\
&\leq \left(1 - \frac{\epsilon d}{8}\right)^{\frac{16}{\epsilon d}} \\
&\leq e^{-2} \\
&\leq 1/3
\end{aligned}$$

➔ Some iteration finds a small CC and returns FALSE with probability at least 2/3.

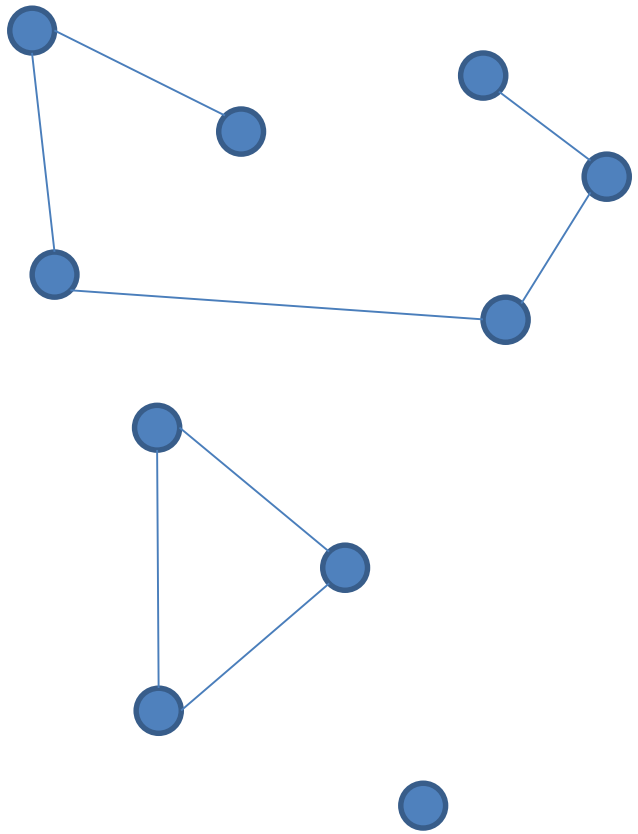
Connectivity

Connected(G, n, d, ε)

Repeat $16/\varepsilon d$ times:

- Choose random node u.
- Do a BFS from u, stopping after $8/\varepsilon d$ nodes are found.
- If CC of u has $\leq 8/\varepsilon d$ nodes, return **FALSE**.

Return **TRUE**



Claim: If G is ε-far from connected, then returns **FALSE** with probability $\geq 2/3$.

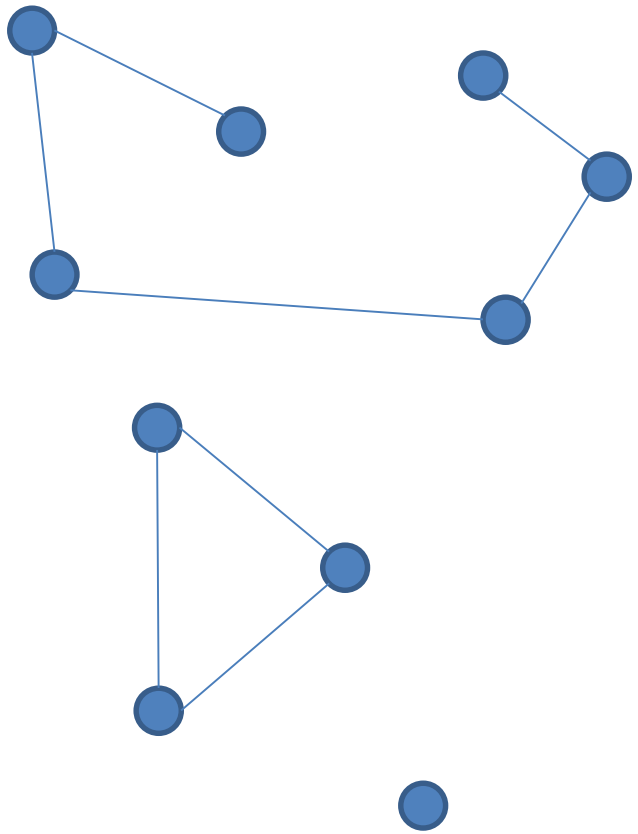
Connectivity

Connected(G, n, d, ε)

Repeat $16/\varepsilon d$ times:

- Choose random node u.
- Do a BFS from u, stopping after $8/\varepsilon d$ nodes are found.
- If CC of u has $\leq 8/\varepsilon d$ nodes, return **FALSE**.

Return **TRUE**



Claim: Total time is $O(1/\varepsilon^2 d)$.

Claim: If G is connected, returns **TRUE**.

Claim: If G is ε-far from connected, then returns **FALSE** with probability $\geq 2/3$.

Connectivity

General idea:

- Use sampling and local approximation to understand global graph properties.
- For what other interesting properties can you do this?

Questions to think about:

- Is gap approximation useful?
- Is there a better notion of “close to connected”?
- For what values of ε and d is this actually fast?
- What happens in dense graphs?
- Can you find a faster algorithm? In theory? In practice?

Announcements / Reminders

Problem sets:

Problem Set 1 will be released tomorrow.

Problem Set 1 will be due next week.

Summary

Last Week:

Toy example 1: array all 0's?

- Gap-style question:
  All 0's or far from all 0's?

Toy example 2: Faction of 1's?

- Additive ± ε approximation
- Hoeffding Bound

Is the graph connected?

- Gap-style question.
- O(1) time algorithm.
- Correct with probability 2/3.

Today:

Number of connected components in a graph.

- Approximation algorithm.

Weight of MST

- Approximation algorithm.