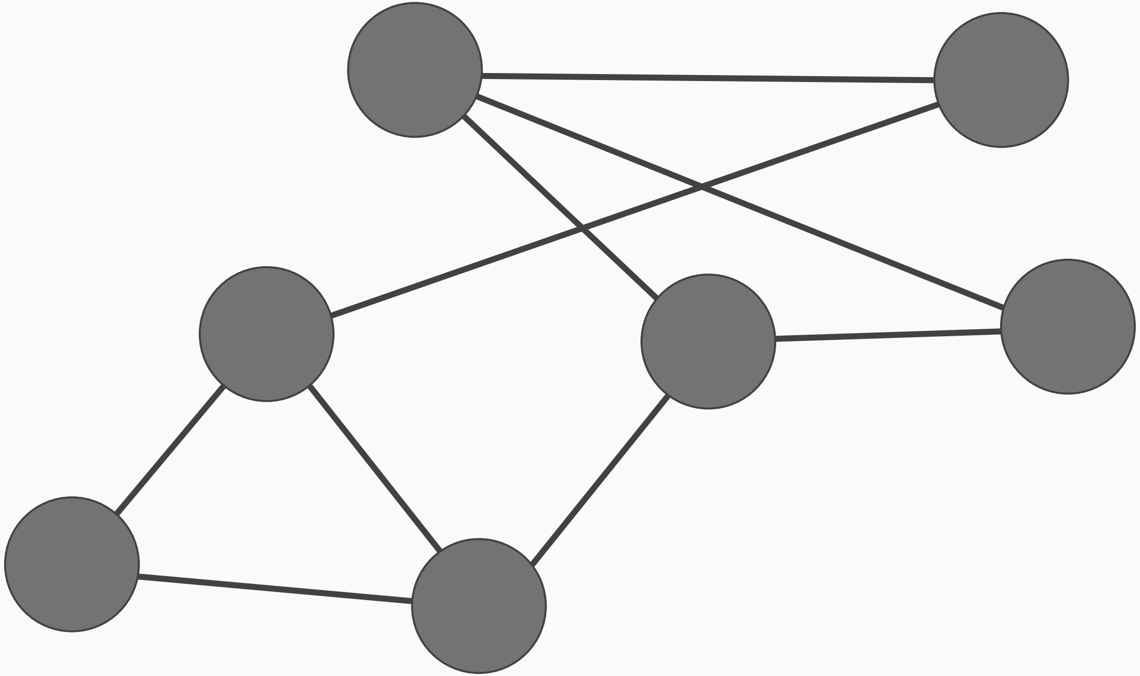
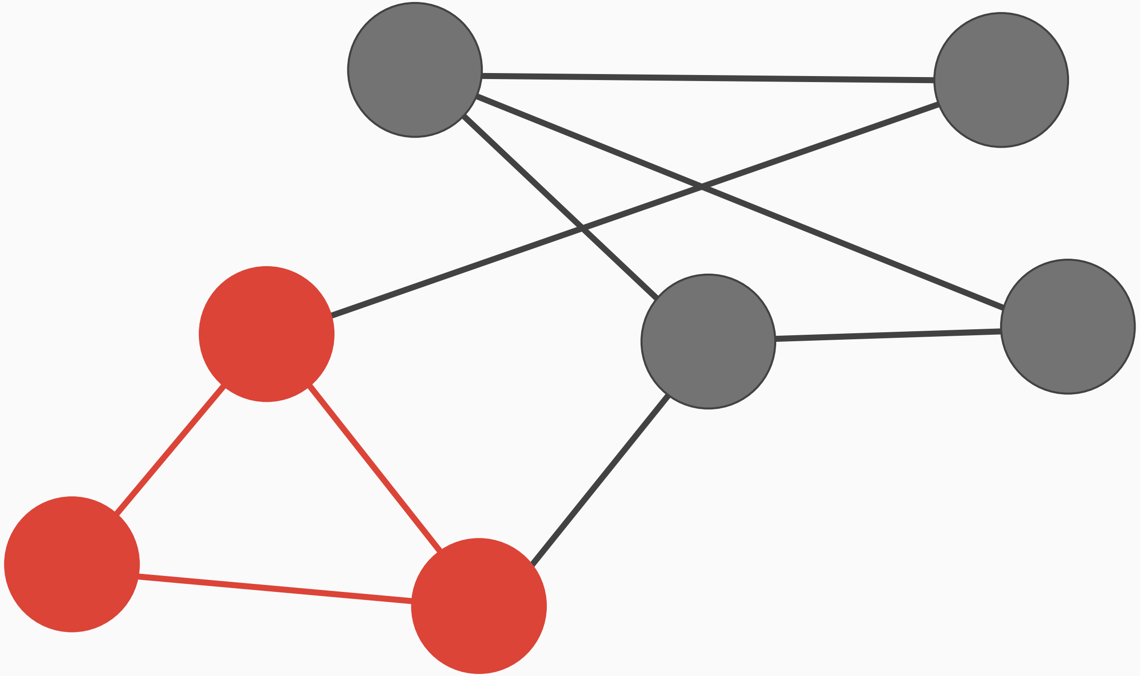
CS5234: Counting Triangles

A tale of three sampling algorithms

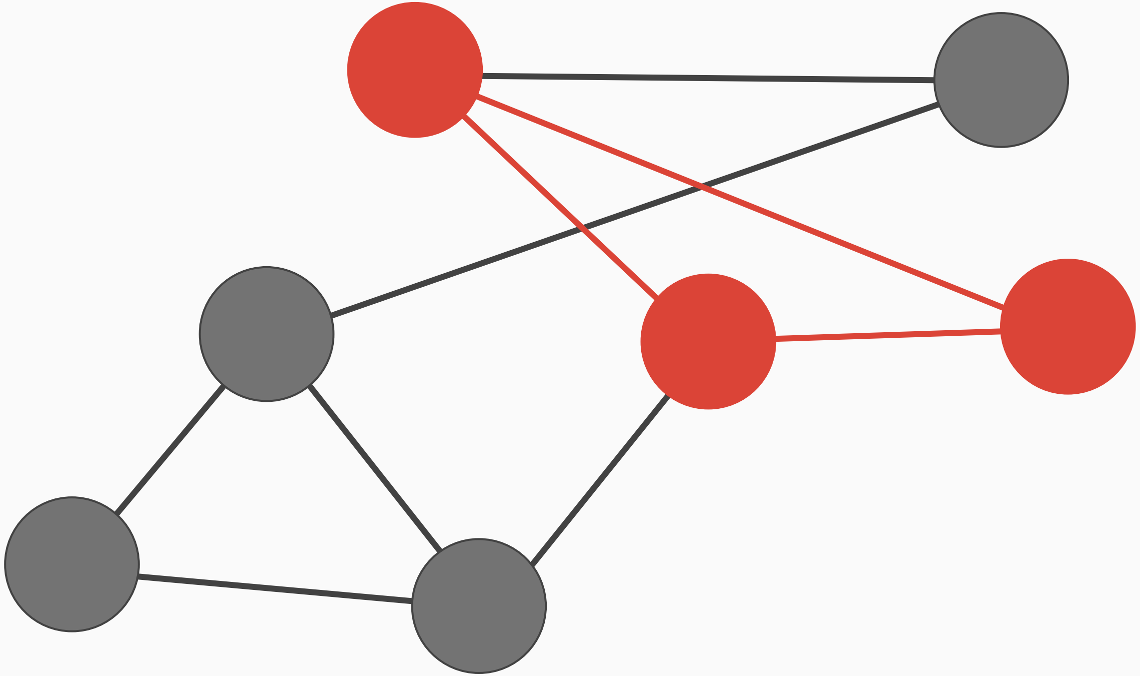
Counting Triangles in a Graph



Counting Triangles in a Graph



Counting Triangles in a Graph



What would this be useful for?

- Computing the transitivity coefficient of a graph.
- Motif detection in protein interaction networks.
- Social network analysis
- Etc

BUT!

Now we want to do this in a stream!

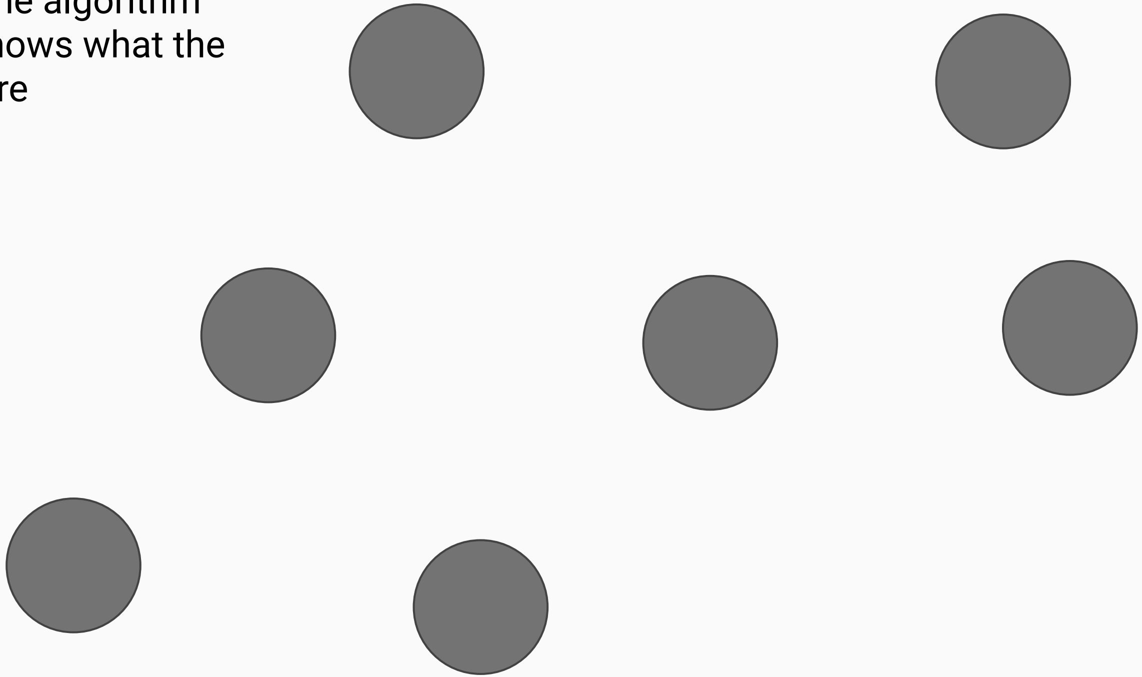
BUT!

Now we want to do this in a stream!

And in one pass!

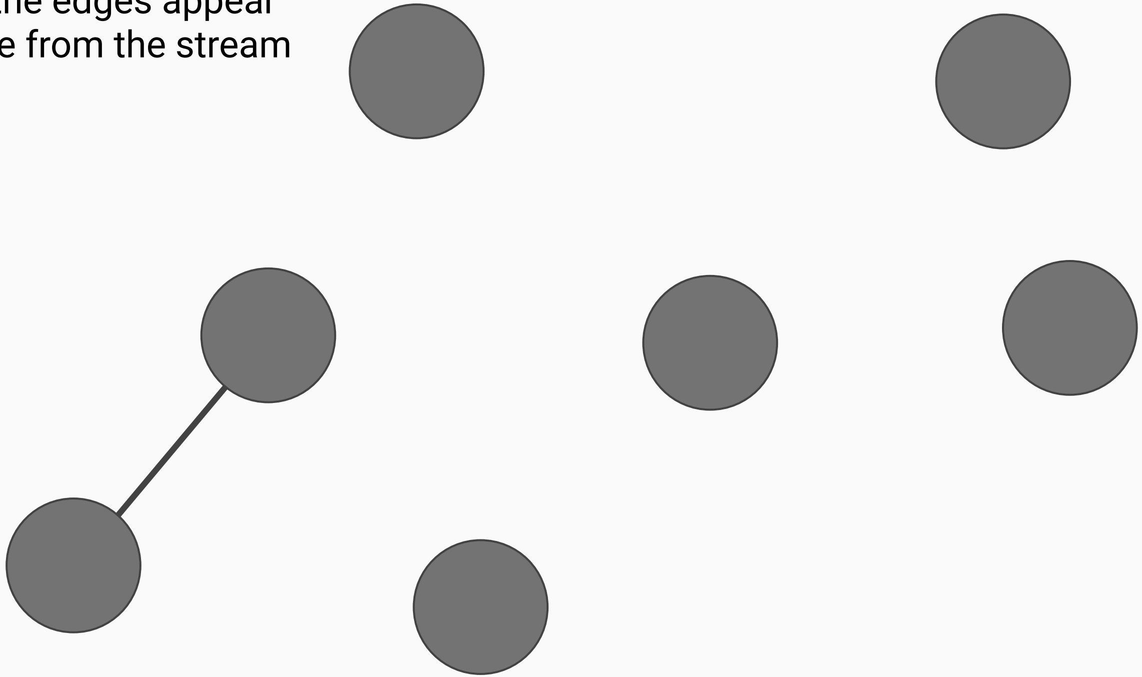
In the streaming model:

Assume the algorithm already knows what the vertices are



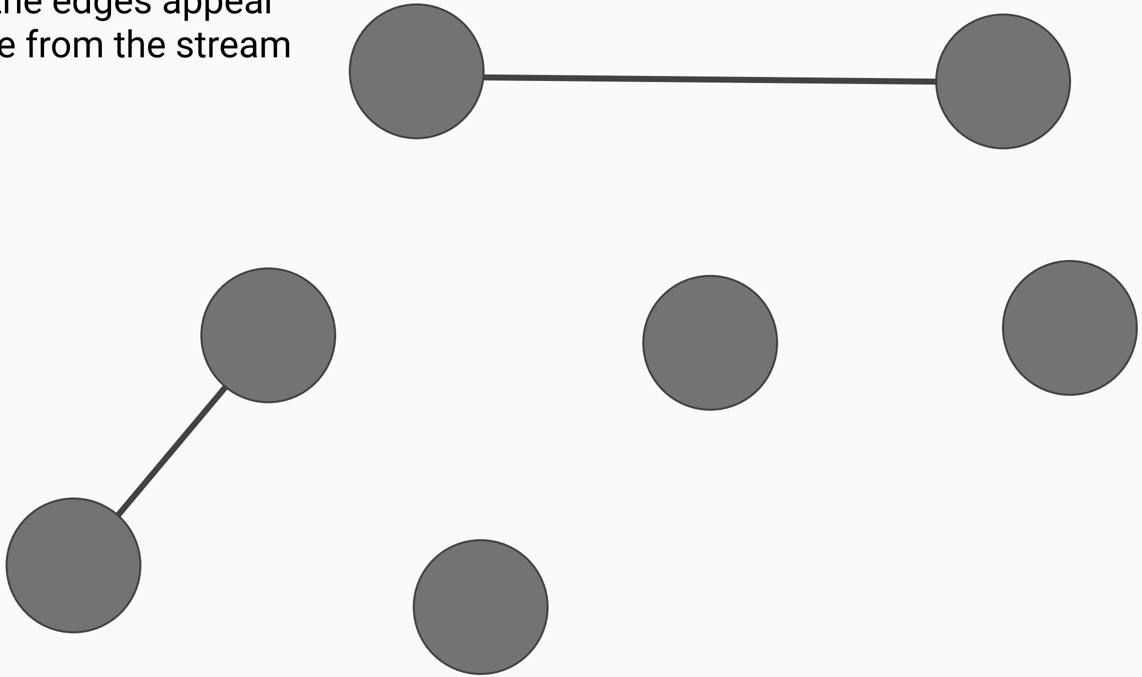
In the streaming model:

And now the edges appear one by one from the stream



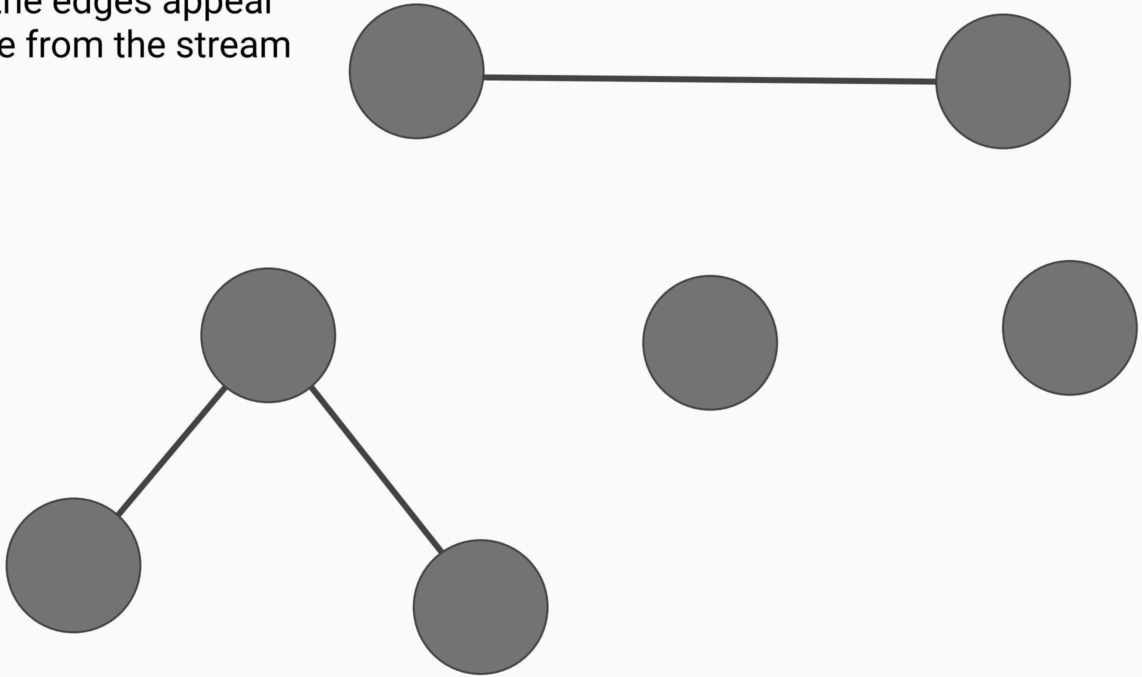
In the streaming model:

And now the edges appear one by one from the stream



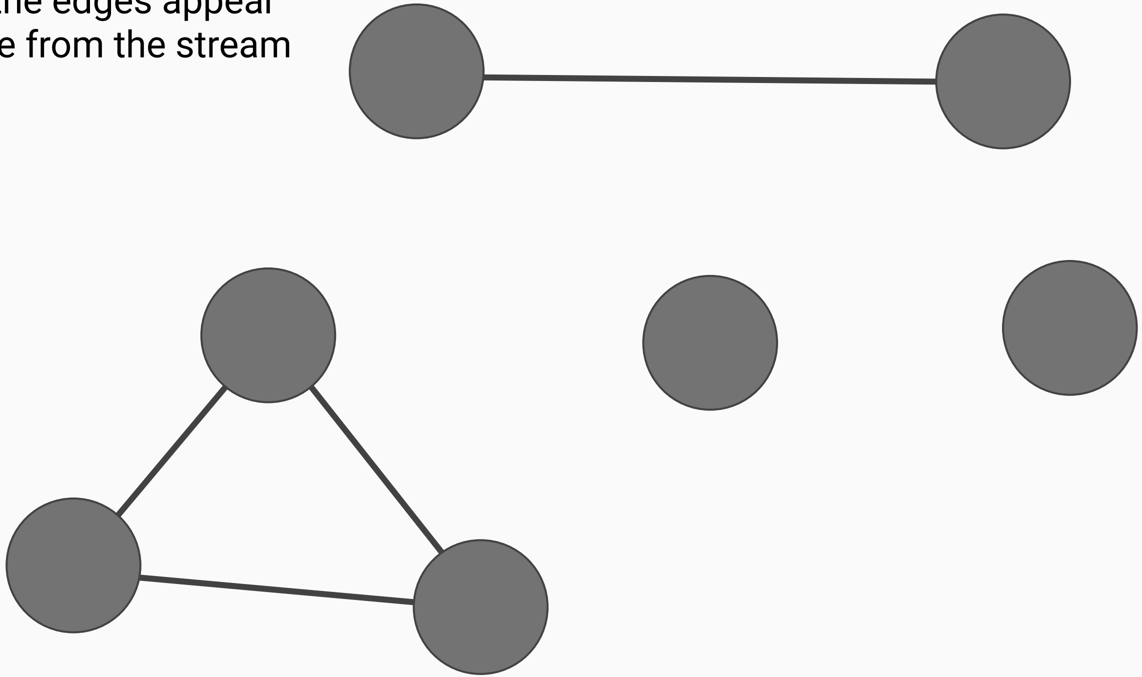
In the streaming model:

And now the edges appear one by one from the stream



In the streaming model:

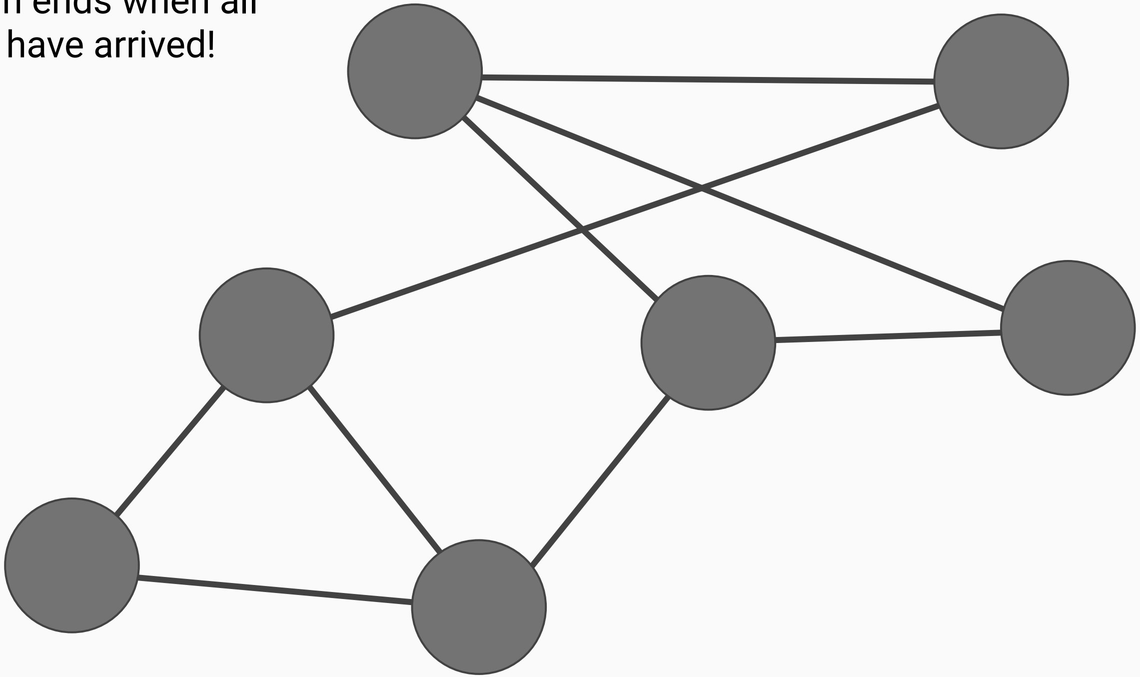
And now the edges appear one by one from the stream



In the streaming model:

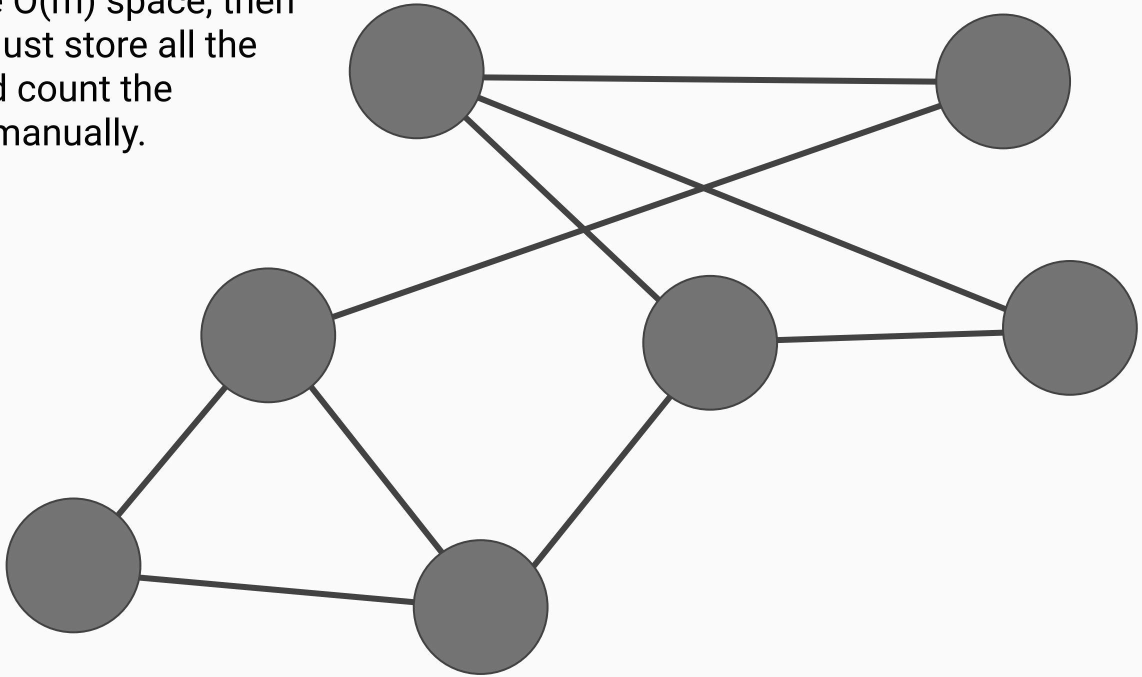
In the streaming model:

The stream ends when all the edges have arrived!



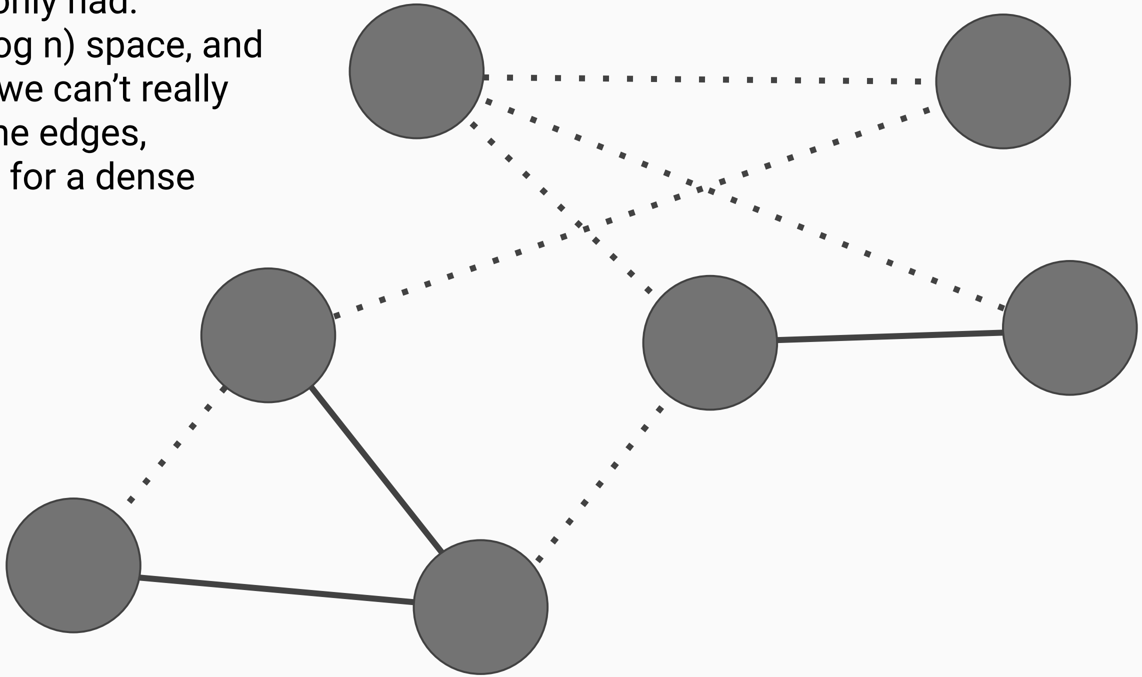
Say we didn't have a space constraint:

If we have O(m) space, then we could just store all the edges and count the triangles manually.



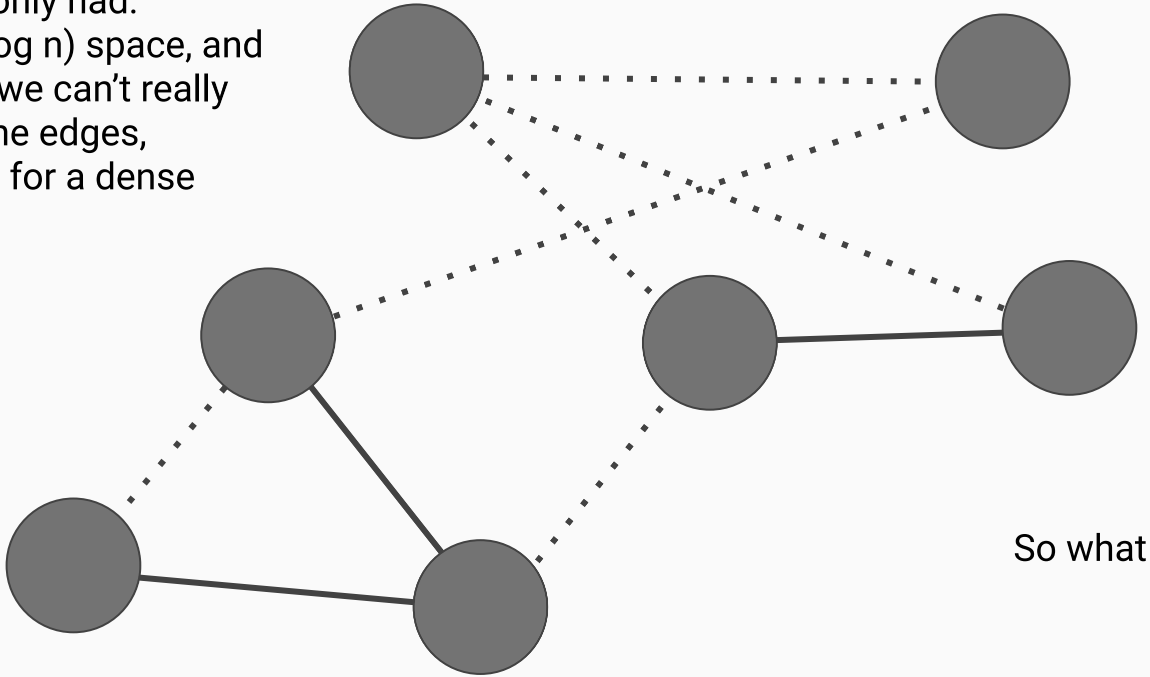
Say we didn’t have a space constraint:

But if we only had:
O(n poly log n) space, and therefore we can’t really store all the edges, especially for a dense graph.



Say we didn't have a space constraint:

But if we only had:
O(n poly log n) space, and therefore we can't really store all the edges, especially for a dense graph.



So what should we do?

Our solution?

RANDOMISE!

Our solution?

Flip some coins!!

RANDOMISE!

Our solution?

Flip some coins!!

Sample edges!

RANDOMISE!

Our solution?

Flip some coins!!

Sample edges!

RANDOMISE!

`Use a Chernoff bound somewhere!`

Okay hold on...

- As it turns out counting triangles in a graph if we only had constant passes on the stream is quite impossible! [Braverman, Ostrovsky, Vilenchik, 13’] (How Hard is Counting Triangles in the Streaming Model?)

- In actual fact, algorithms will need at least $\Omega(m / T)$ space, where T is the number of triangles in the graph.

- As it turns out counting triangles in a graph if we only had constant passes on the stream is quite impossible! [Braverman, Ostrovsky, Vilenchik, 13’] (How Hard is Counting Triangles in the Streaming Model?)

- In actual fact, algorithms will need at least $\Omega(m / T)$ space, where T is the number of triangles in the graph.

It **didn’t** stop people from trying though. :| The algorithms are still performant provided you have a large number of triangles.

Counting Triangles

With lots of space so make of it what you will.

3 Shades of Sampling

- An edge and a vertex

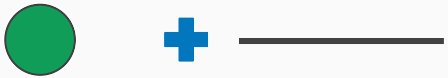
3 Shades of Sampling

- An edge and a vertex

- Two neighbouring edges

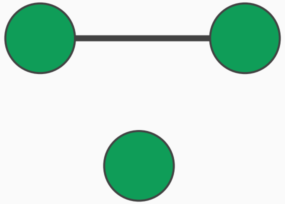
3 Shades of Sampling

- An edge and a vertex



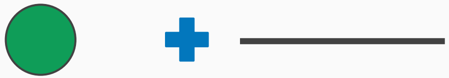
- Two neighbouring edges

- An entire subgraph



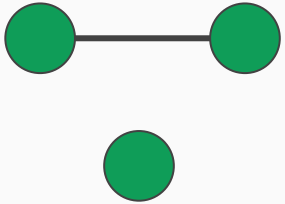
3 Shades of Sampling

- An edge and a vertex



- Two neighbouring edges

- An entire subgraph



Want to return, a count of triangles that is:

Within $(1+\varepsilon)$ factor, with probability $(1-\delta)$.

But first! Reservoir Sampling:

But first! Reservoir Sampling:

But first! Reservoir Sampling:

But first! Reservoir Sampling:

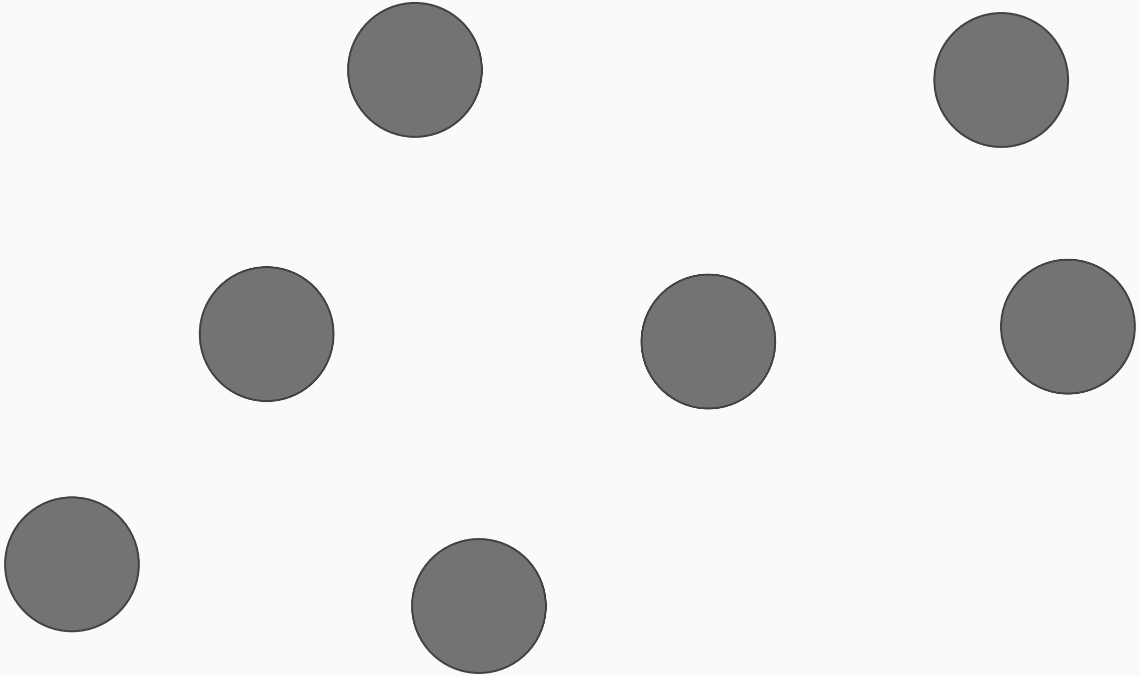
But first! Reservoir Sampling:

Sampling Idea 1

Sampling Idea 1:

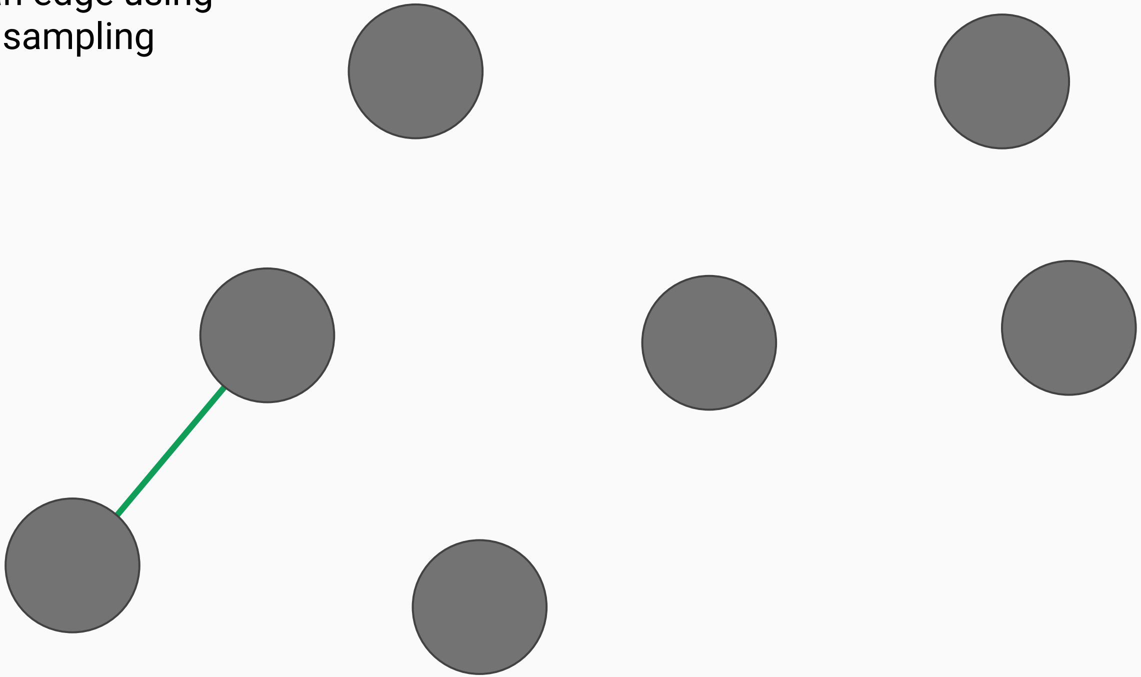
1. Sample an edge at random
2. Sample a vertex at random
3. Now (fingers crossed) we really hope that there are two other edges that will come and connect the vertex and the edge we sampled earlier.

Idea 1 in action:



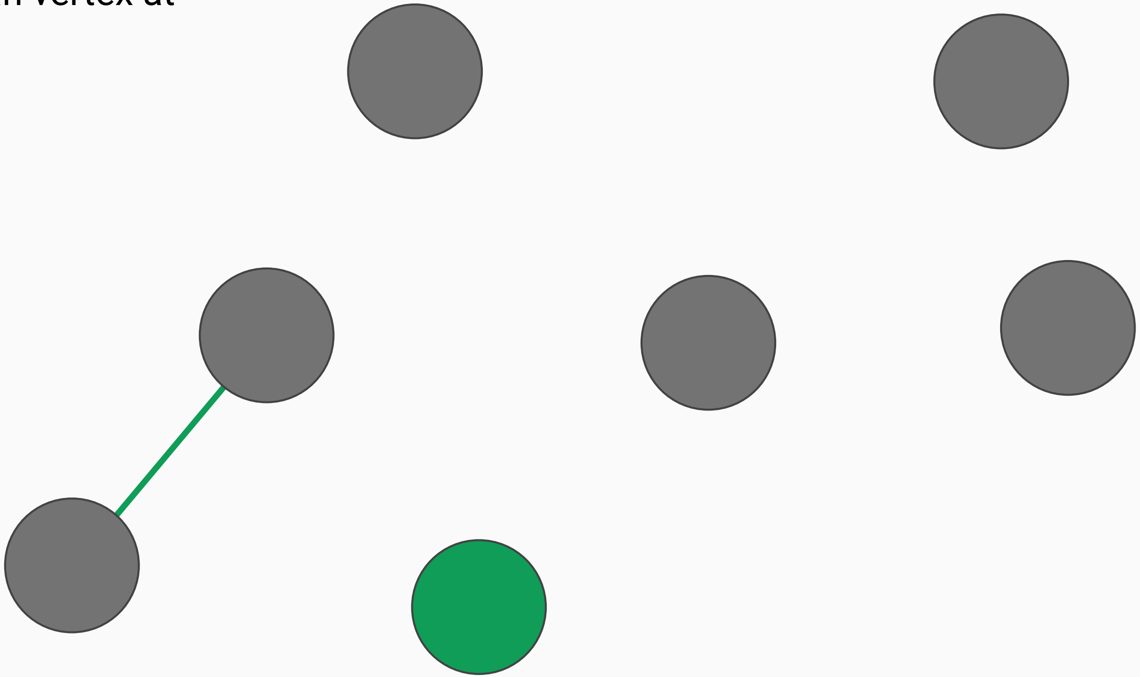
Idea 1 in action:

Sample an edge using reservoir sampling



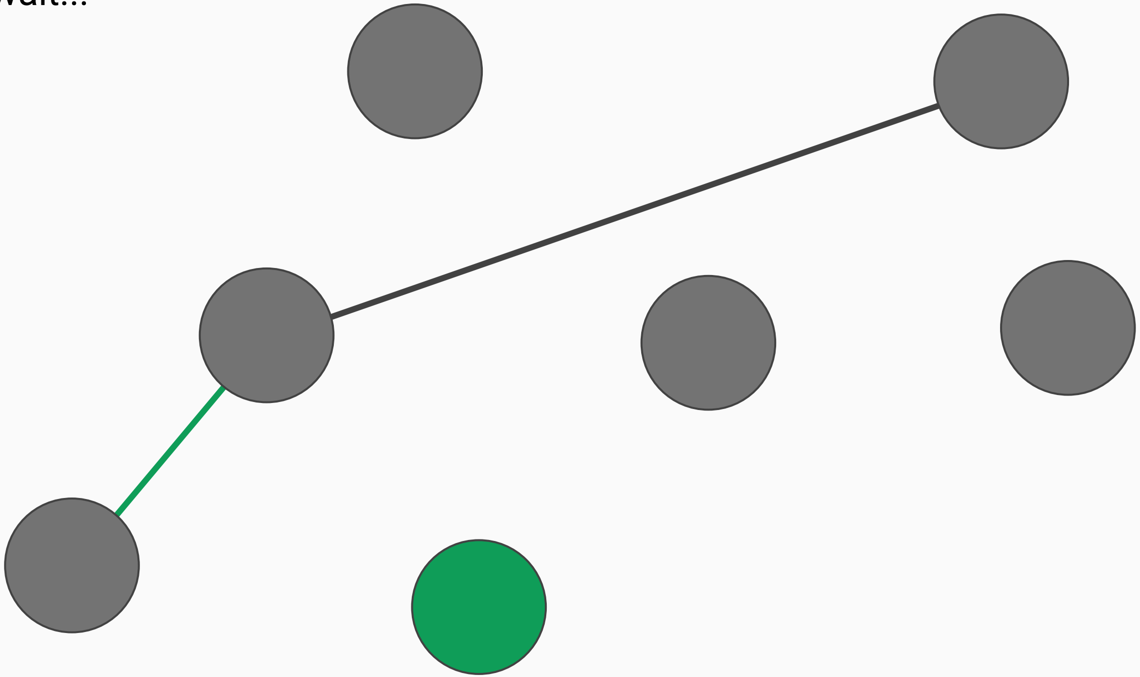
Idea 1 in action:

Sample an vertex at random.



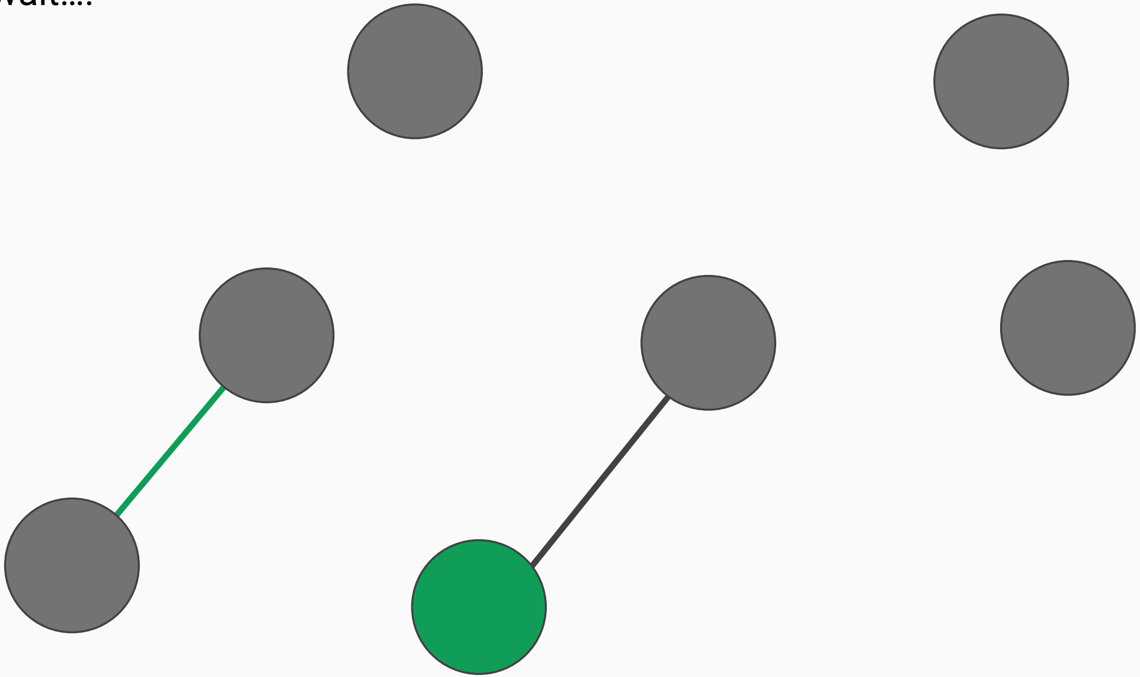
Idea 1 in action:

Now we wait...



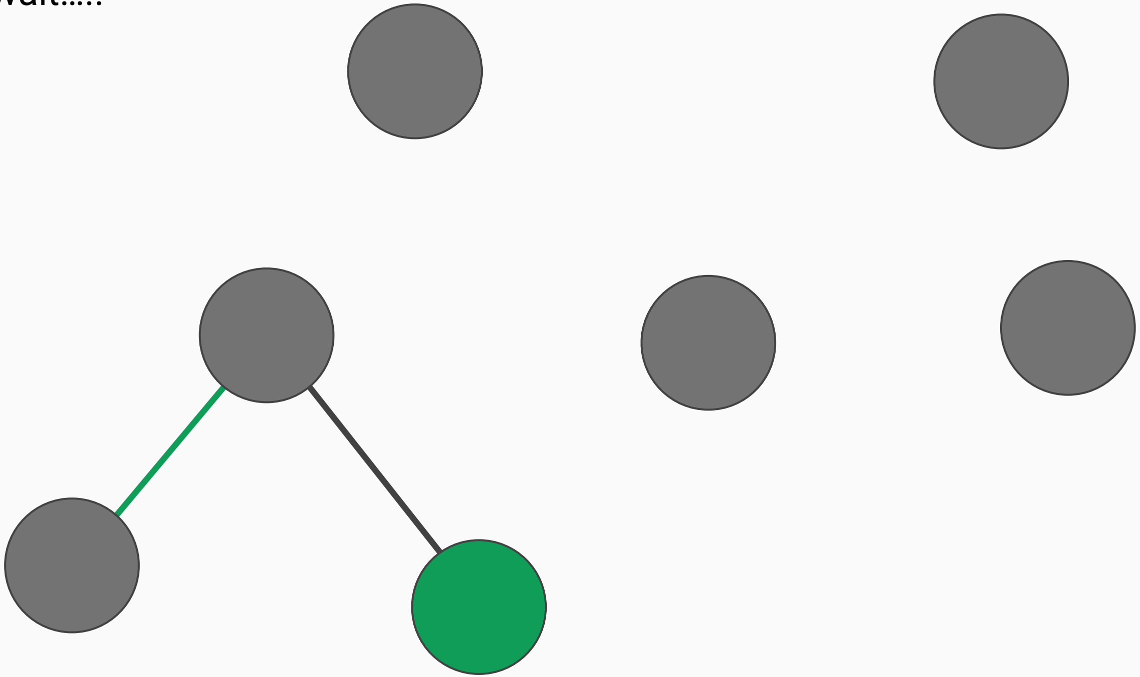
Idea 1 in action:

Now we wait....



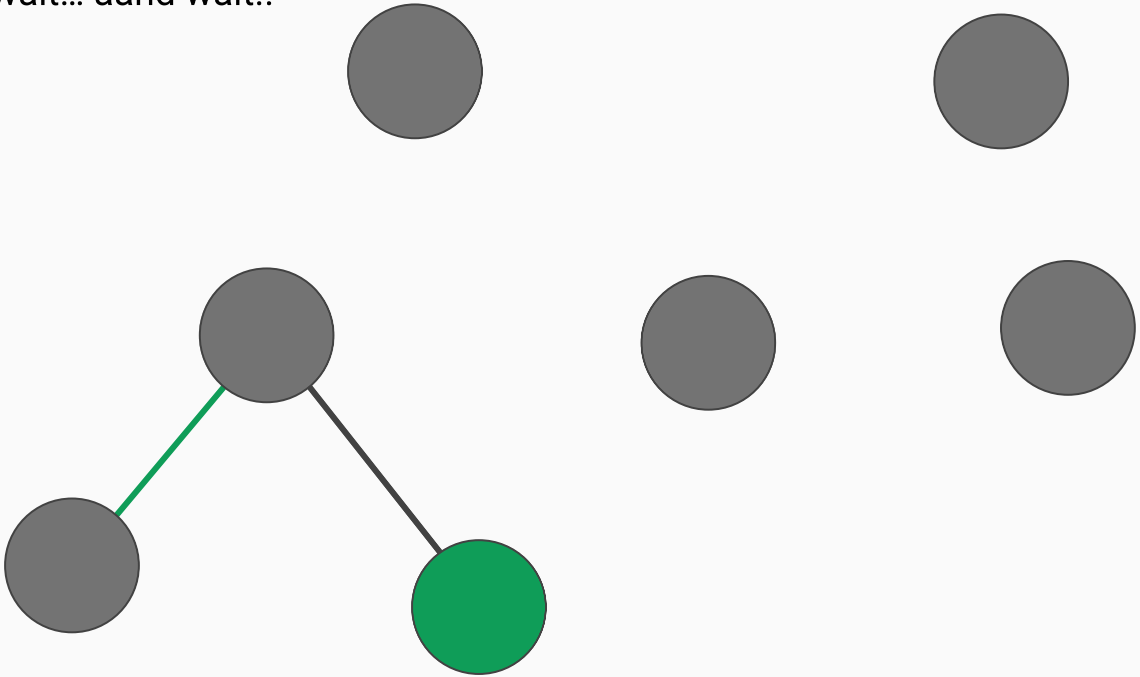
Idea 1 in action:

Now we wait.....



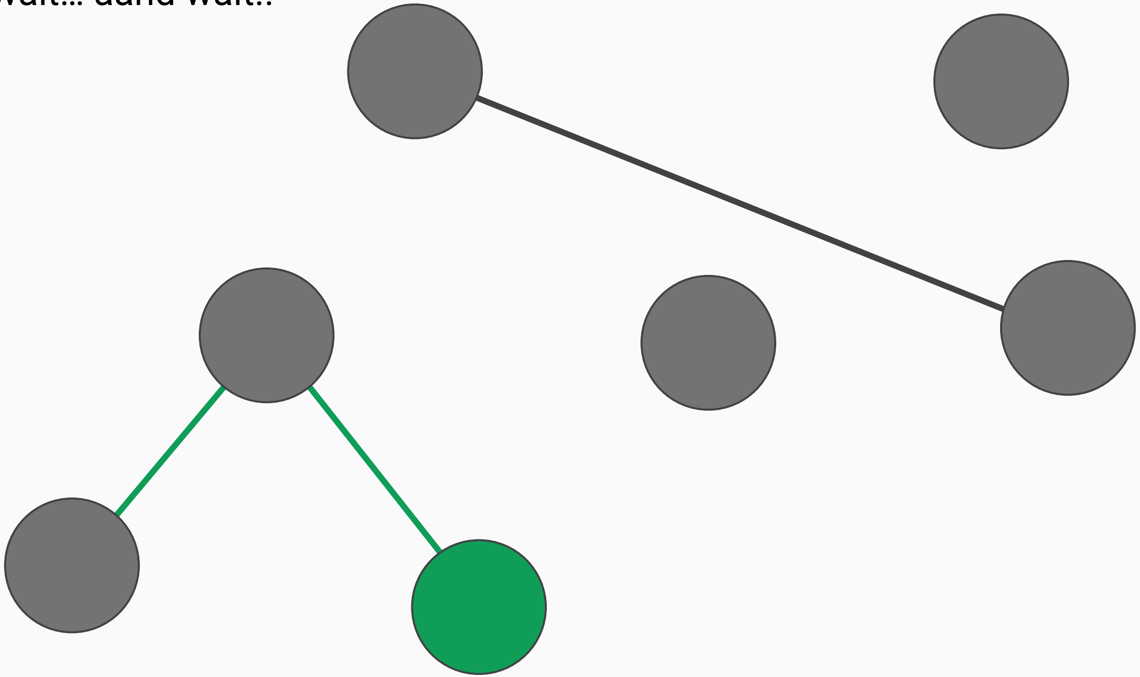
Idea 1 in action:

Now we wait… aand wait..



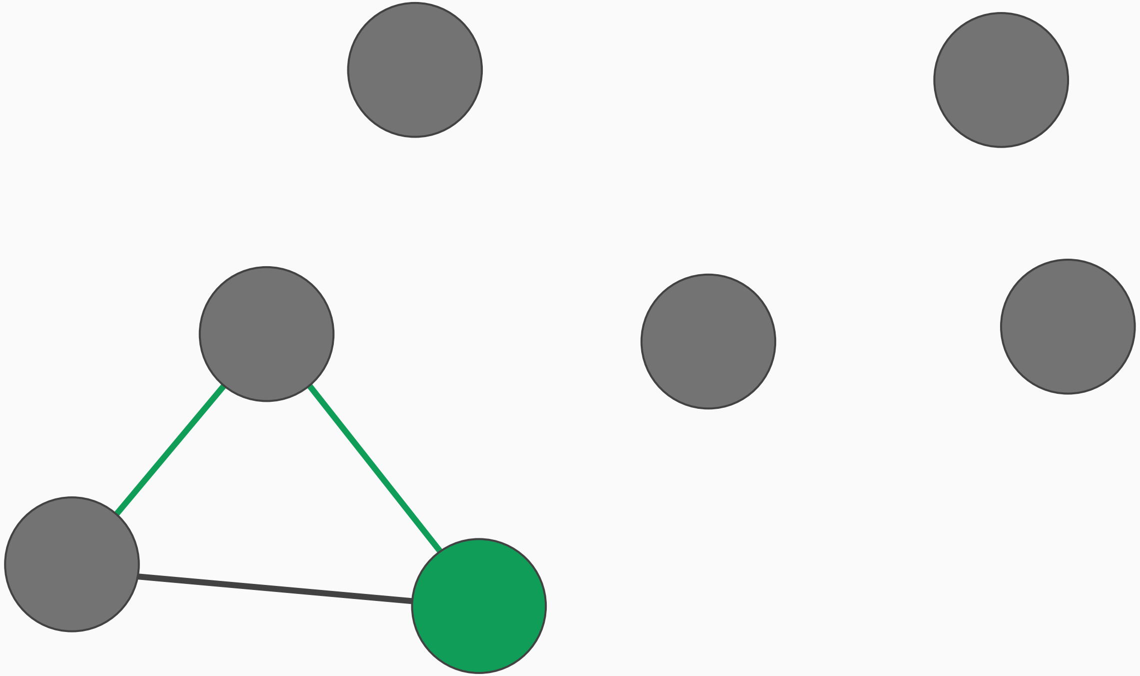
Idea 1 in action:

Now we wait… aand wait..



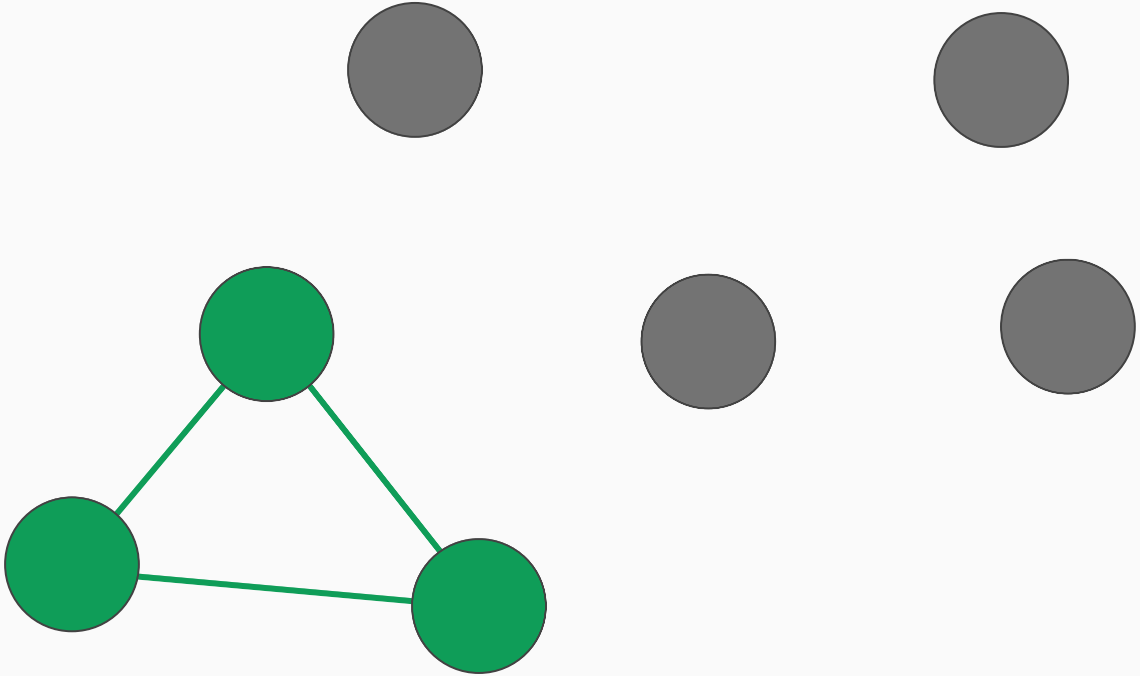
Idea 1 in action:

And zing!



Idea 1 in action:

And zing!



Algorithm 1:

1. Run r copies of this sampling idea, which are independent of each other.

2. Count the number of triangles sampled, and return: count * m * (n - 2) // r

And now, some math

Goal: Show our algorithm in expectation returns the number of triangles.

1. The probability that we sample any edge and vertex pair is give as:

$$\frac{1}{(n-2)m}$$

And now, some math

Goal: Show our algorithm in expectation returns the number of triangles.

1. The probability that we sample any edge and vertex pair is give as:

$$\frac{1}{(n-2)m}$$

2. Now there are T many triangles, so the probability that we sample a triangle is actually:

$$\frac{T}{(n-2)m}$$

And now, some math

Goal: Show our algorithm in expectation returns the number of triangles.

1. The probability that we sample any edge and vertex pair is give as:

$$\frac{1}{(n-2)m}$$

2. Now there are T many triangles, so the probability that we sample a triangle is actually:

$$\frac{T}{(n-2)m}$$

3. So the expected value of our output is:

$$\frac{rT}{(n-2)m} \cdot (n-2)m\frac{1}{r} = T$$

And now, some math

Goal: Show our algo

in expectation return

number of trian

Chernoff Bound!!

1. The probability that we sample any edge and vertex pair is give as:

$$\frac{1}{(n-2)m}$$

es, so the probability that we

$$\frac{\quad}{(n-2)m}$$

3. So the expected value of our output is:

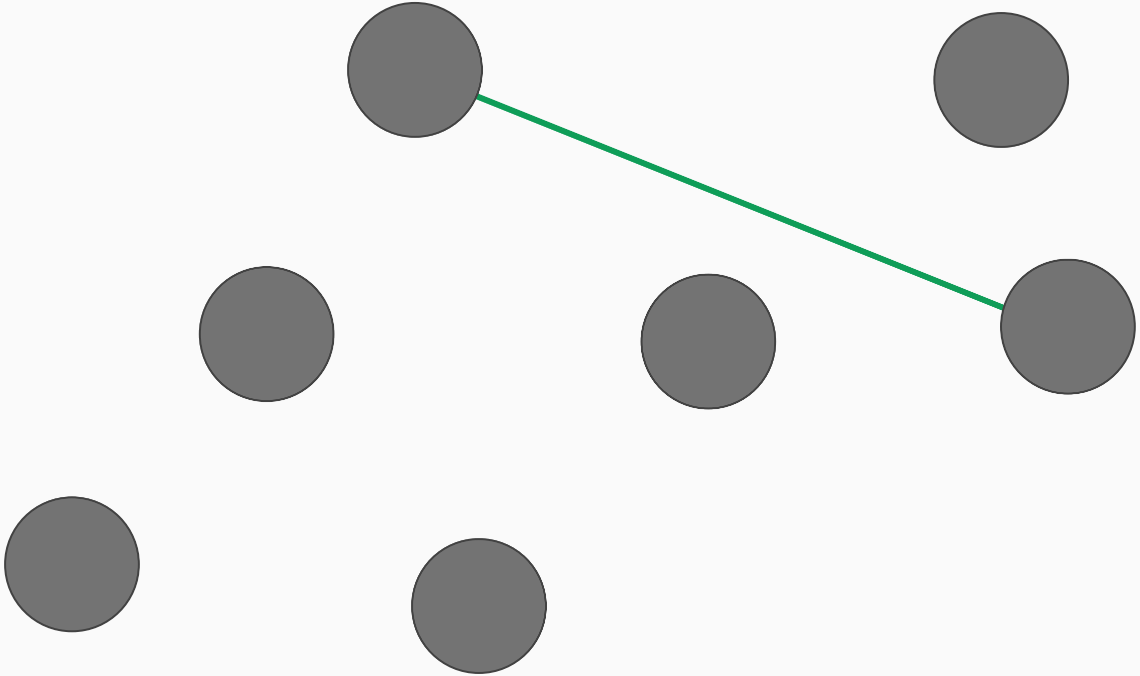
$$\frac{rT}{(n-2)m}\cdot(n-2)m\frac{1}{r}=T$$

Sampling Idea 2

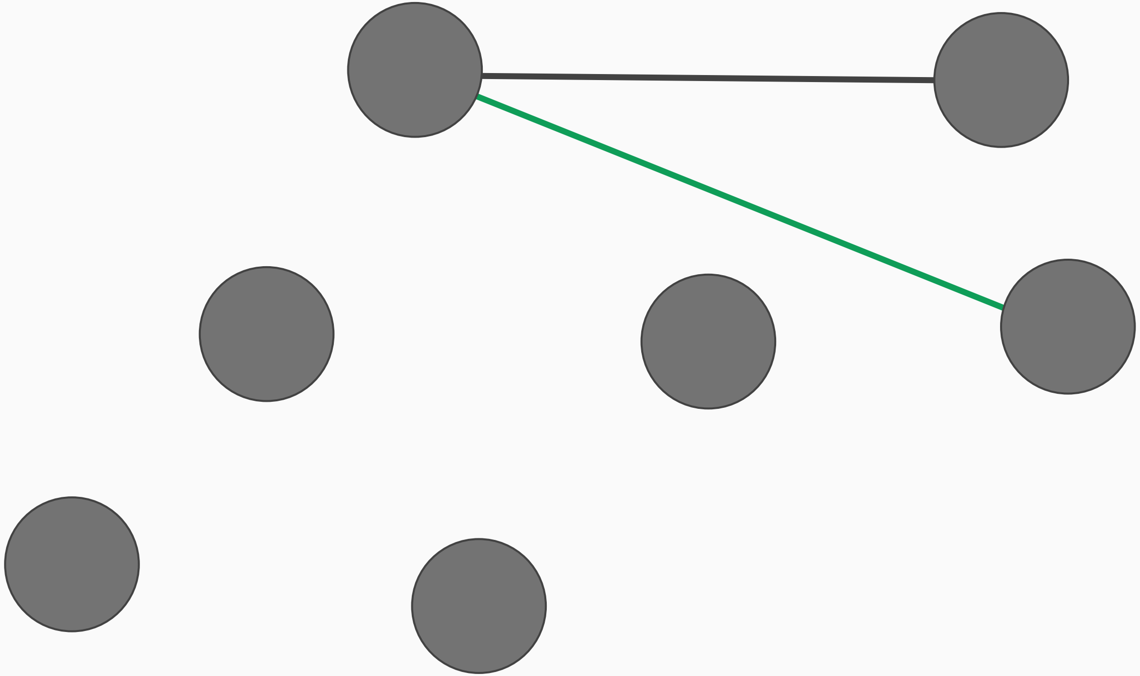
Sampling Idea 2:

1. Sample an edge at random
2. Sample a neighbouring edge at random (also using a separate reservoir algorithm)
3. Keep a count of the number of neighbours the first edge has seen, c.
4. If we stick with a triangle by the end of the stream, we return m*c .
5. Else we return 0.

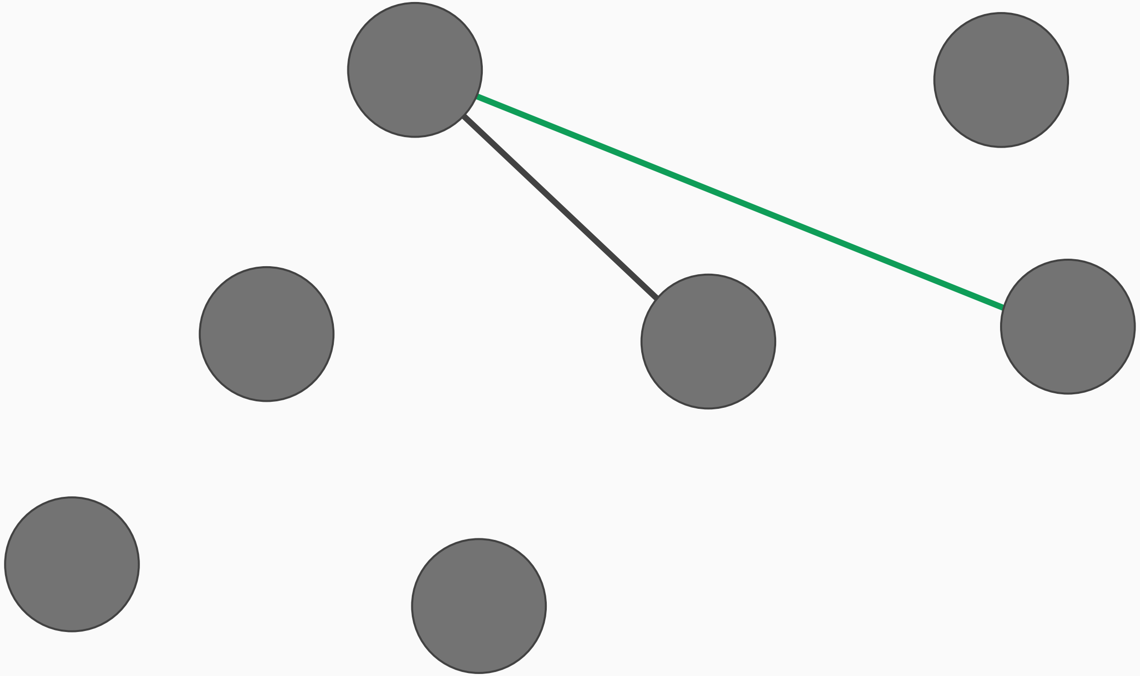
Counting Triangles in a Graph



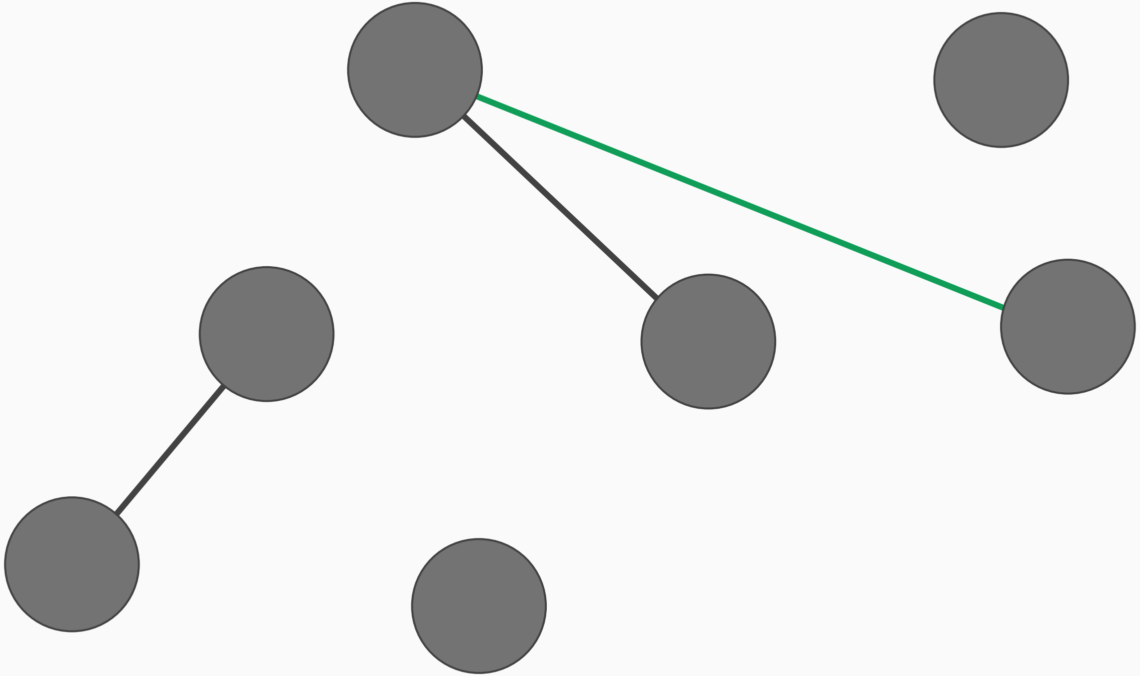
Counting Triangles in a Graph



Counting Triangles in a Graph

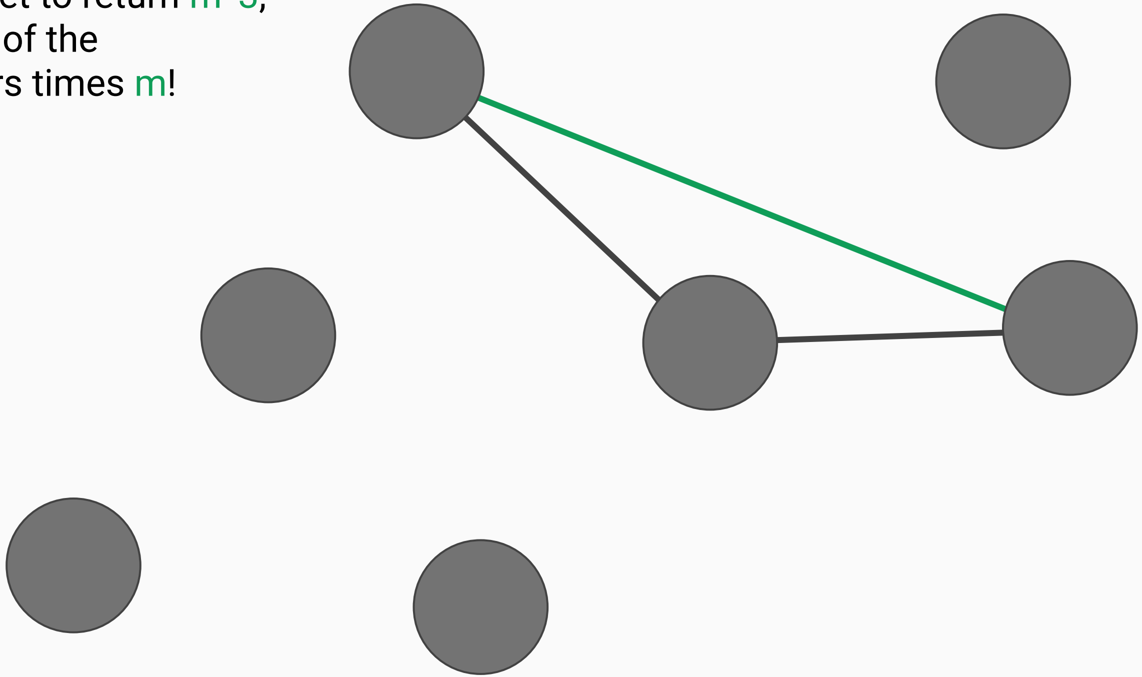


Counting Triangles in a Graph



Counting Triangles in a Graph

Don't forget to return m*3, the count of the neighbours times m!



And now, **more** math

Again we want to show in expectation this value is equals to the number of triangles.

$$
\begin{aligned}
\mathbb{E}[m \cdot c] &= \\
&= \\
&= \\
&= \\
&=
\end{aligned}
$$

And now, **more** math

Again we want to show in expectation this value is equals to the number of triangles.

$$\mathbb{E}[m \cdot c] = m \cdot \mathbb{E}[c] = m \cdot \sum_{t \in \tau(G)} c_t Pr[t \text{ was the sampled}]$$

$$=$$

$$=$$

$$=$$

$$=$$

And now, **more** math

Again we want to show in expectation this value is equals to the number of triangles.

$$\begin{aligned}
\mathbb{E}[m \cdot c] &= m \cdot \mathbb{E}[c] = m \cdot \sum_{t \in \tau(G)} c_t Pr[t \text{ was the sampled}] \\
&= m \cdot \sum_{t \in \tau(G)} c_t \frac{1}{m \cdot N(e_1)} \\
&= \\
&= \\
&=
\end{aligned}$$

And now, **more** math

Again we want to show in expectation this value is equals to the number of triangles.

$$
\begin{aligned}
\mathbb{E}[m \cdot c] = m \cdot \mathbb{E}[c] &= m \cdot \sum_{t \in \tau(G)} c_t Pr[t \text{ was the sampled}] \\
&= m \cdot \sum_{t \in \tau(G)} c_t \frac{1}{m \cdot N(e_1)} \\
&= m \cdot \sum_{t \in \tau(G)} c_t \frac{1}{m \cdot c_t} \\
&= \\
&=
\end{aligned}
$$

And now, **more** math

Again we want to show in expectation this value is equals to the number of triangles.

$$
\begin{aligned}
\mathbb{E}[m \cdot c] = m \cdot \mathbb{E}[c] &= m \cdot \sum_{t \in \tau(G)} c_t Pr[t \text{ was the sampled}] \\
&= m \cdot \sum_{t \in \tau(G)} c_t \frac{1}{m \cdot N(e_1)} \\
&= m \cdot \sum_{t \in \tau(G)} c_t \frac{1}{m \cdot c_t} \\
&= \sum_{t \in \tau(G)} 1 \\
&=
\end{aligned}
$$

And now, **more** math

Again we want to show in expectation this value is equals to the number of triangles.

$$
\begin{aligned}
\mathbb{E}[m \cdot c] = m \cdot \mathbb{E}[c] &= m \cdot \sum_{t \in \tau(G)} c_t Pr[t \text{ was the sampled}] \\
&= m \cdot \sum_{t \in \tau(G)} c_t \frac{1}{m \cdot N(e_1)} \\
&= m \cdot \sum_{t \in \tau(G)} c_t \frac{1}{m \cdot c_t} \\
&= \sum_{t \in \tau(G)} 1 \\
&= |\tau(G)|
\end{aligned}
$$

And now, **more** math

Again we want to show in exp
this value is equals to the numb
triangles.

Chernoff Bound!!

$$\mathbb{E}[m \cdot c] = m \cdot \mathbb{E}[c] = m \cdot \sum_{t \in \tau(G)} c_t Pr[t \text{ was the sampled}]$$

$$= \sum_{t \in \tau(G)} 1$$

$$= |\tau(G)|$$

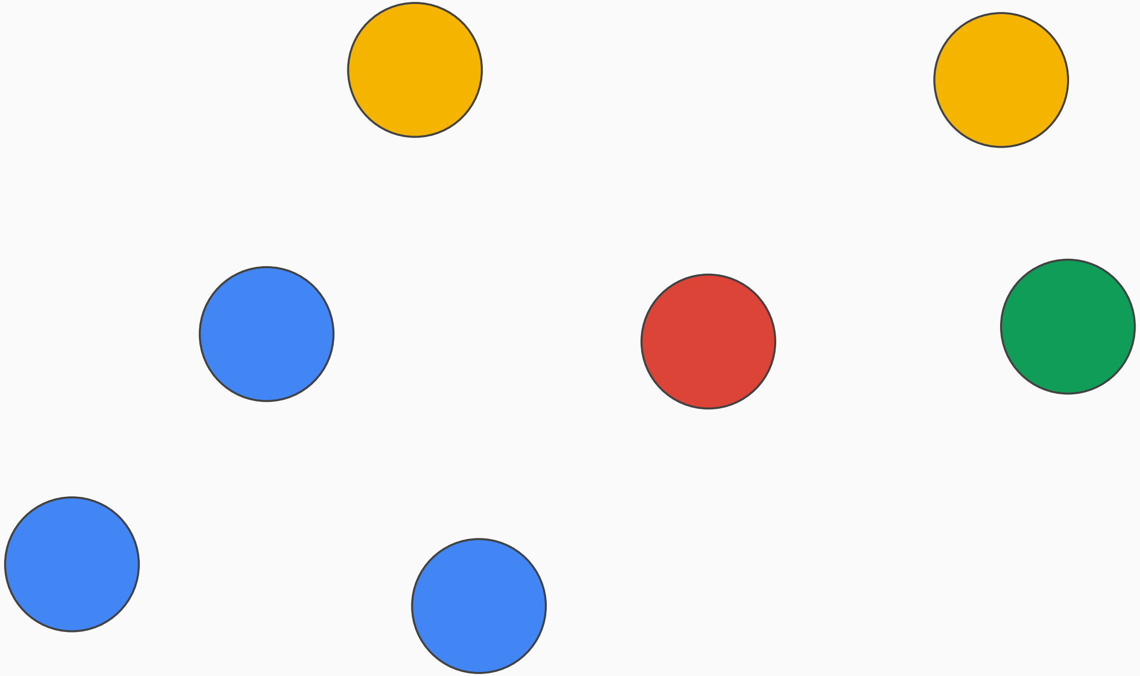
Sampling Idea 3:

Sampling Idea 3:

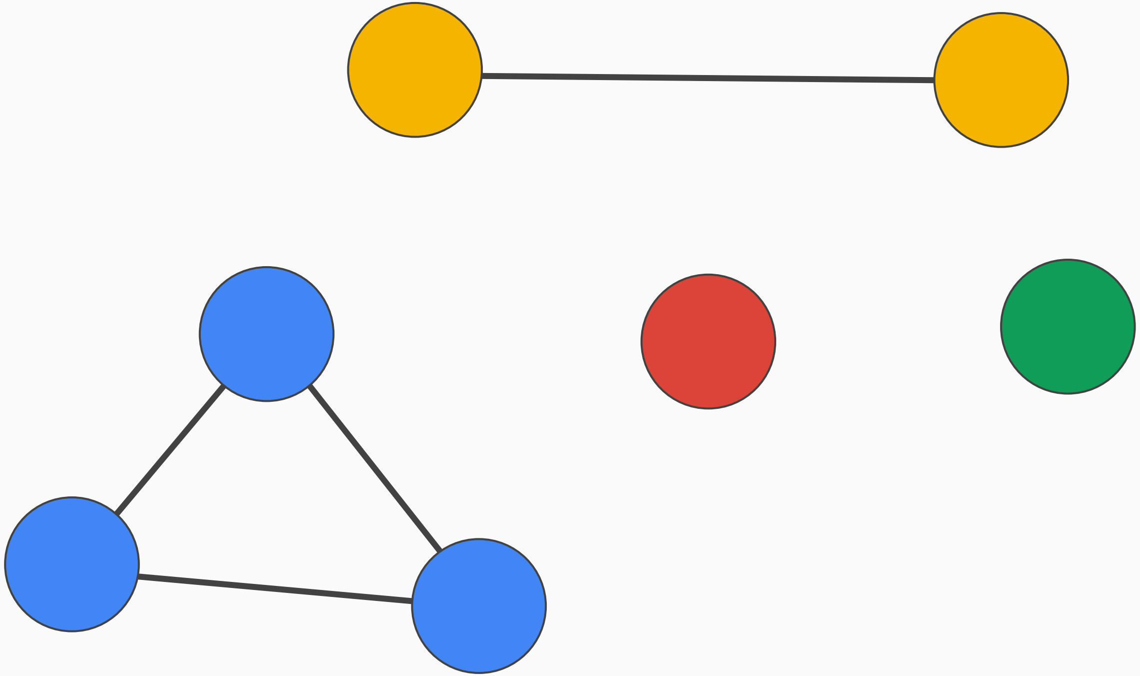
1. Set N colours that we will randomly colour the vertices with.
2. For any edge that arrives in a stream such that both endpoints are the same colour, we keep it in our new subgraph.
3. At the end of the stream, we count the triangles that remain in the subgraph, and output that count, multiplied by N^2

Here N = 1/p.

Counting Triangles in a Graph



Counting Triangles in a Graph



Surprise surprise, **even more** math

Would it be unexpected if I were to again, say that we needed to show in the value returned is in expectation the number of triangles?

$$T = \sum_{i=1}^{\tau(G)} X_i \quad ;$$

Surprise surprise, **even more** math

Would it be unexpected if I were to again, say that we needed to show in the value returned is in expectation the number of triangles?

$$T = \sum_{i=1}^{\tau(G)} X_i \quad ; \quad Pr[X_i = 1] = p^2$$

Surprise surprise, **even more** math

Would it be unexpected if I were to again, say that we needed to show in the value returned is in expectation the number of triangles?

$$T = \sum_{i=1}^{\tau(G)} X_i \quad ; \quad Pr[X_i = 1] = p^2$$

$$\mathbb{E}[T] = \sum_{i=1}^{\tau(G)} \mathbb{E}[X_i] = \tau(G) \cdot p^2$$

Surprise surprise, **even more** math

Would it be unexpected if I we

again, say that we needed to sh

the value retu

number of triangles?

Chernoff Bound!!

$\tau($

$[X_i = 1] = p^2$

$$\mathbb{E}[T] = \sum_{i=1} \mathbb{E}[X_i] = \tau(G) \cdot p^2$$

Surprise surprise, **even more** ma

Would it be unexpected i
again, say that we neede
the value retu
number of triangles?

YOU CAN'T!
THE X_i'S ARE NOT INDEPENDENT!

$$[X_i = 1] = p^2$$

$$= \tau(G) \cdot p^2$$

Use Chebyshev instead.

I guess the last method of bounding varied from the previous two.

$$Var[T] = \mathbb{E}[T^2] - \mathbb{E}[T]^2$$

Use Chebyshev instead.

I guess the last method of bounding varied from the previous two.

$$Var[T] = \mathbb{E}[T^2] - \mathbb{E}[T]^2$$
$$= \mathbb{E}[\left(\sum_{i=1}^{\tau(G)} X_i\right)^2] - \tau(G)^2 \cdot p^4$$

Use Chebyshev instead.

I guess the last method of bounding varied from the previous two.

$$
\begin{aligned}
Var[T] &= \mathbb{E}[T^2] - \mathbb{E}[T]^2 \\
&= \mathbb{E}[\left(\sum_{i=1}^{\tau(G)} X_i\right)^2] - \tau(G)^2 \cdot p^4 \\
&= \mathbb{E}[\sum_{i=1}^{\tau(G)} X_i^2 + \sum_{i \nsim j} X_i \cdot X_j + \sum_{i \sim j} X_i \cdot X_j] - \tau(G)^2 \cdot p^4
\end{aligned}
$$

Use Chebyshev instead.

I guess the last method of bounding <u>varied</u> from the previous two.

$$
\begin{aligned}
Var[T] &= \mathbb{E}[T^2] - \mathbb{E}[T]^2 \\
&= \mathbb{E}[\left(\sum_{i=1}^{\tau(G)} X_i\right)^2] - \tau(G)^2 \cdot p^4 \\
&= \mathbb{E}[\sum_{i=1}^{\tau(G)} X_i^2 + \sum_{i \not\sim j} X_i \cdot X_j + \sum_{i \sim j} X_i \cdot X_j] - \tau(G)^2 \cdot p^4 \\
&= \sum_{i=1}^{\tau(G)} \mathbb{E}[X_i^2] + \sum_{i \not\sim j} \mathbb{E}[X_i \cdot X_j] + \sum_{i \sim j} \mathbb{E}[X_i \cdot X_j] - \tau(G)^2 \cdot p^4
\end{aligned}
$$

Use Chebyshev instead.

I guess the last method of bounding varied from the previous two.

$$
\begin{aligned}
Var[T] &= \mathbb{E}[T^2] - \mathbb{E}[T]^2 \\
&= \mathbb{E}[\left(\sum_{i=1}^{\tau(G)} X_i\right)^2] - \tau(G)^2 \cdot p^4 \\
&= \mathbb{E}[\sum_{i=1}^{\tau(G)} X_i^2 + \sum_{i \not\sim j} X_i \cdot X_j + \sum_{i \sim j} X_i \cdot X_j] - \tau(G)^2 \cdot p^4 \\
&= \sum_{i=1}^{\tau(G)} \mathbb{E}[X_i^2] + \sum_{i \not\sim j} \mathbb{E}[X_i \cdot X_j] + \sum_{i \sim j} \mathbb{E}[X_i \cdot X_j] - \tau(G)^2 \cdot p^4
\end{aligned}
$$

$$
\begin{aligned}
&\leq \tau(G) \cdot p^2 + \tau(G)^2 \cdot p^4 + \sum_{i \sim j} \mathbb{E}[X_i \cdot X_j] - \tau(G)^2 \cdot p^4 \\
&= \tau(G) \cdot p^2 + \sum_{i \sim j} \mathbb{E}[X_i \cdot X_j]
\end{aligned}
$$

Use Chebyshev instead.

I guess the last method of bounding <u>varied</u> from the previous two.

$$Var[T] = \mathbb{E}[T^2] - \mathbb{E}[T]^2$$

$$= \mathbb{E}[\left(\sum_{i=1}^{\tau(G)} X_i\right)^2] - \tau(G)^2 \cdot p^4$$

$$= \mathbb{E}[\sum_{i=1}^{\tau(G)} X_i^2 + \sum_{i \nsim j} X_i \cdot X_j + \sum_{i \sim j} X_i \cdot X_j] - \tau(G)^2 \cdot p^4$$

$$= \sum_{i=1}^{\tau(G)} \mathbb{E}[X_i^2] + \sum_{i \nsim j} \mathbb{E}[X_i \cdot X_j] + \sum_{i \sim j} \mathbb{E}[X_i \cdot X_j] - \tau(G)^2 \cdot p^4$$

$$\leq \tau(G) \cdot p^2 + \tau(G)^2 \cdot p^4 + \sum_{i \sim j} \mathbb{E}[X_i \cdot X_j] - \tau(G)^2 \cdot p^4$$

$$= \tau(G) \cdot p^2 + \sum_{i \sim j} \mathbb{E}[X_i \cdot X_j]$$

$$\leq \tau(G) \cdot p^2 + 3\tau_{max}\tau(G)p^3$$

Use Chebyshev instead.

I guess the last method of bounding varied from the previous two.

$$
\begin{aligned}
Var[T] &= \mathbb{E}[T^2] - \mathbb{E}[T]^2 \\
&= \mathbb{E}[\left(\sum_{i=1}^{\tau(G)} X_i\right)^2] - \tau(G)^2 \cdot p^4 \\
&= \mathbb{E}[\sum_{i=1}^{\tau(G)} X_i^2 + \sum_{i \not\sim j} X_i \cdot X_j + \sum_{i \sim j} X_i \cdot X_j] - \tau(G)^2 \cdot p^4 \\
&= \sum_{i=1}^{\tau(G)} \mathbb{E}[X_i^2] + \sum_{i \not\sim j} \mathbb{E}[X_i \cdot X_j] + \sum_{i \sim j} \mathbb{E}[X_i \cdot X_j] - \tau(G)^2 \cdot p^4
\end{aligned}
$$

$$
\begin{aligned}
&\leq \tau(G) \cdot p^2 + \tau(G)^2 \cdot p^4 + \sum_{i \sim j} \mathbb{E}[X_i \cdot X_j] - \tau(G)^2 \cdot p^4 \\
&= \tau(G) \cdot p^2 + \sum_{i \sim j} \mathbb{E}[X_i \cdot X_j] \\
&\leq \tau(G) \cdot p^2 + 3\tau_{max}\tau(G)p^3
\end{aligned}
$$

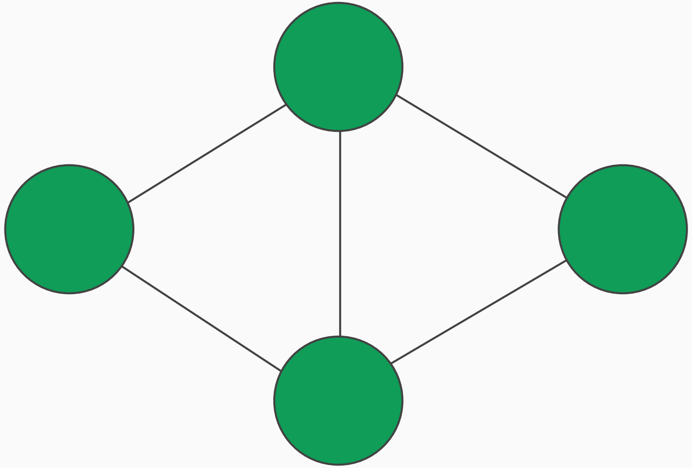
Use Chebyshev instead.

I guess the last method of bounding varied from the previous two.

$$\sum_{i\sim j} \mathbb{E}[X_i \cdot X_j]$$

Only 1 if both triangles are included in the subgraph

= Only if all 4 vertices have the same colour



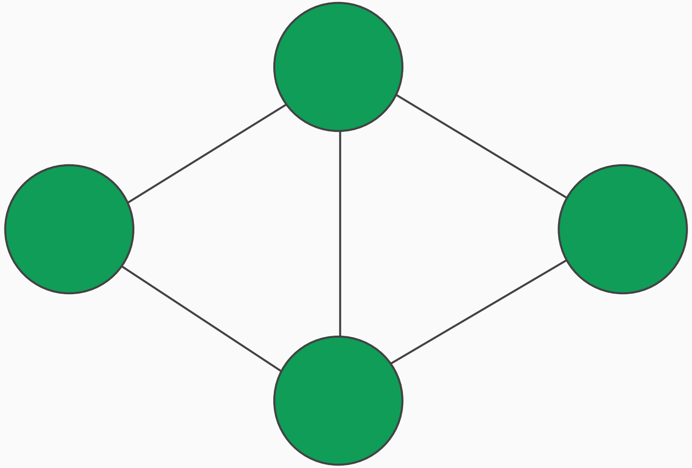
Use Chebyshev instead.

I guess the last method of bounding varied from the previous two.

$$\underbrace{\sum_{i \sim j} p^3}$$

Only 1 if both triangles are included in the subgraph

= Only if all 4 vertices have the same colour

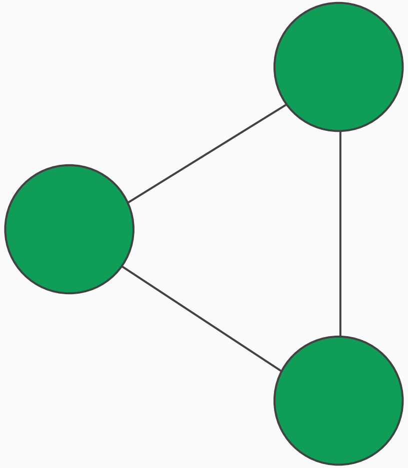


Use Chebyshev instead.

I guess the last method of bounding varied from the previous two.

$$\underbrace{\sum_{i \sim j} p^3}_{}$$

Worst case how many triangles correlated?

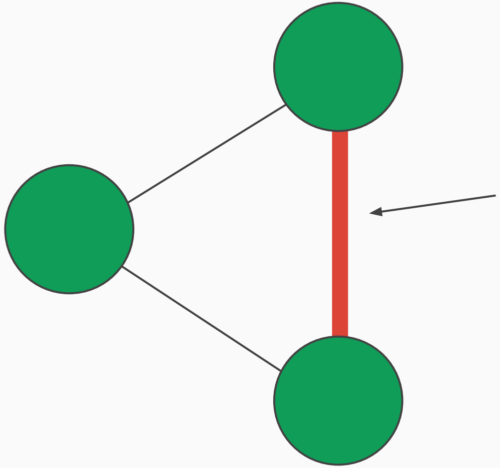


Use Chebyshev instead.

I guess the last method of bounding varied from the previous two.

$$\sum_{i \sim j} p^3$$

Worst case how many triangles correlated?

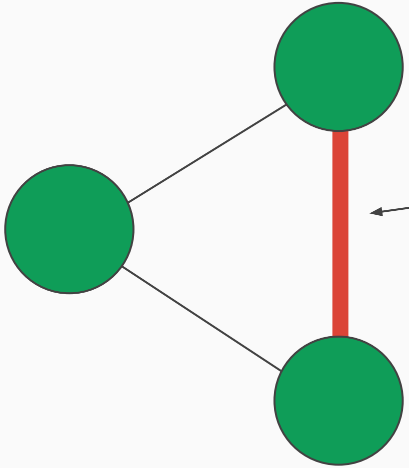


Every edge of every triangle is also shared with every other triangle.

Use Chebyshev instead.

I guess the last method of bounding varied from the previous two.

$$3 \cdot \tau(G) \cdot \tau_{max} \cdot p^3$$



Every edge of every triangle is also shared with every other triangle.

Use Chebyshev instead.

I guess the last method of bounding varied from the previous two.

$$\tau(G) \cdot p^2 + 3\tau_{max}\tau(G)p^3 = o(\tau(G)^2 \cdot p^4)$$

$$\therefore\ 1 + 3\tau_{max}p = o(\tau(G) \cdot p^2)$$

Use Chebyshev instead.

I guess the last method of bounding varied from the previous two.

$$\tau(G) \cdot p^2 + 3\tau_{max}\tau(G)p^3 = o(\tau(G)^2 \cdot p^4)$$

$$\therefore\ 1 + 3\tau_{max}p = o(\tau(G) \cdot p^2)$$

$$p \cdot \tau_{max} < 1/3$$

Use Chebyshev instead.

I guess the last method of bounding varied from the previous two.

$$\tau(G) \cdot p^2 + 3\tau_{max}\tau(G)p^3 = o(\tau(G)^2 \cdot p^4)$$

$$\therefore\ 1 + 3\tau_{max}p = o(\tau(G) \cdot p^2)$$

$$p \cdot \tau_{max} < 1/3$$

$$\implies p = \frac{\tau(G)}{\sqrt{\log n}}$$

Use Chebyshev instead.

I guess the last method of bounding varied from the previous two.

$$\tau(G) \cdot p^2 + 3\tau_{max}\tau(G)p^3 = o(\tau(G)^2 \cdot p^4)$$

$$\therefore \; 1 + 3\tau_{max}p = o(\tau(G) \cdot p^2)$$

$$p \cdot \tau_{max} < 1/3$$

$$\implies p = \frac{\tau(G)}{\sqrt{\log n}}$$

$$p \cdot \tau_{max} \geq 1/3$$

Use Chebyshev instead.

I guess the last method of bounding varied from the previous two.

$$\tau(G)\cdot p^2 + 3\tau_{max}\tau(G)p^3 = o(\tau(G)^2\cdot p^4)$$

$$\therefore\ 1 + 3\tau_{max}p = o(\tau(G)\cdot p^2)$$

$$p\cdot\tau_{max} < 1/3$$

$$\implies p = \frac{\tau(G)}{\sqrt{\log n}}$$

$$p\cdot\tau_{max} \geq 1/3$$

$$\implies p = p \geq \frac{6\tau_{max}\log n}{\tau(G)}$$