# Computing Quartet Distance Is Equivalent to Counting 4-Cycles

Bartłomiej Dudek[1] Paweł Gawrychowski[1]

[1] University of Wrocław

# Trees are everywhere... especially in biology

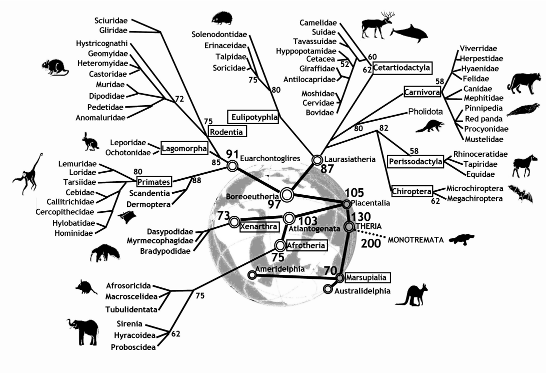


Figure sources: Wikipedia and

California Museum of Paleontology's Understanding Evolution

# Trees are everywhere... especially in biology

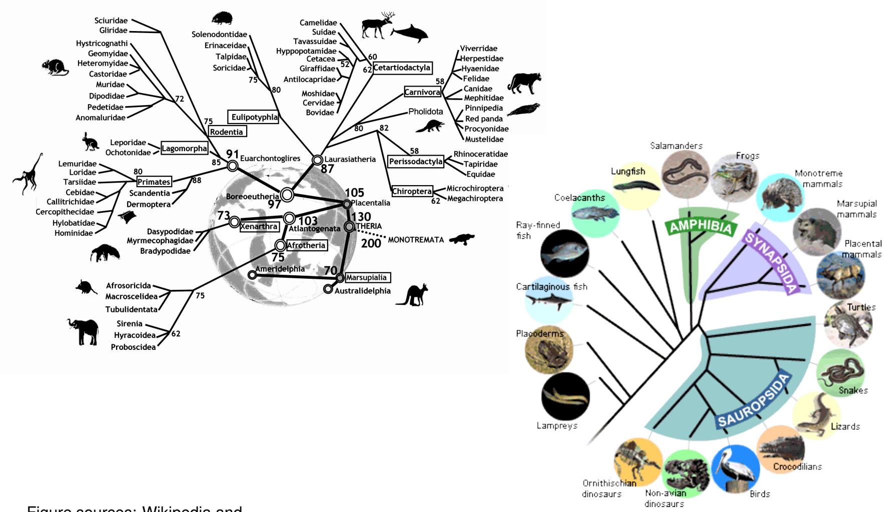
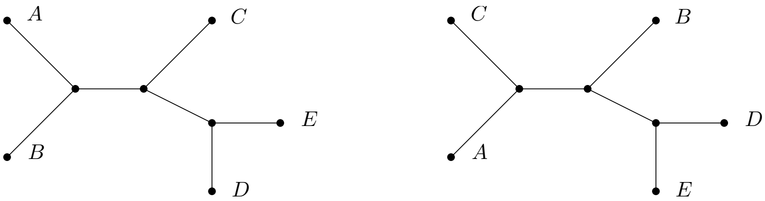


Figure sources: Wikipedia and

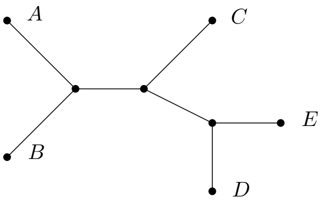
California Museum of Paleontology's Understanding Evolution

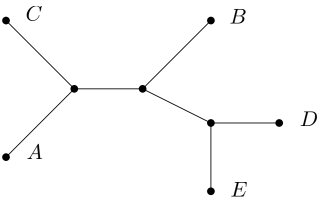
# But how to compare them?



Consider subtrees induced by all possible 4 leaves.

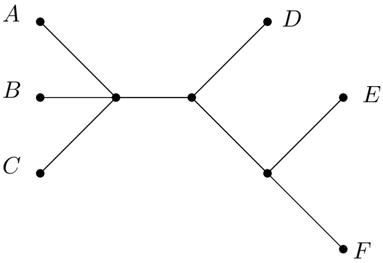
# But how to compare them?



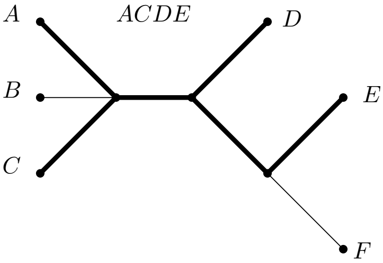


Consider subtrees induced by all possible 4 leaves.

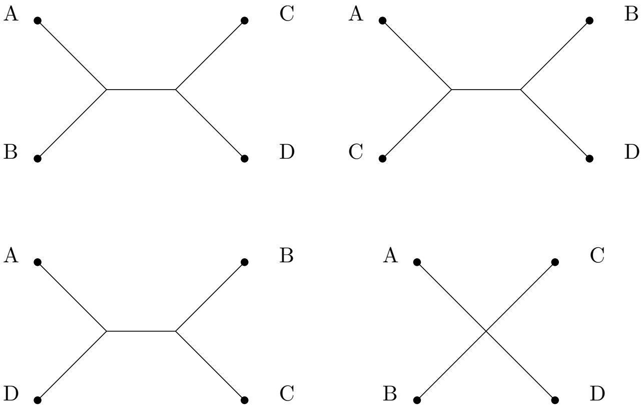
# Possible topologies on 4 leaves



# Possible topologies on 4 leaves

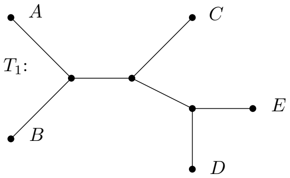


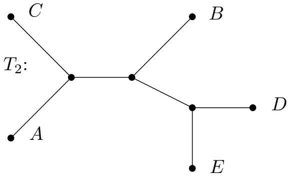
# Possible topologies on 4 leaves



# Quartet distance

Input: two trees with the same set of leaves
Output: number of quartets of leaves inducing different topologies

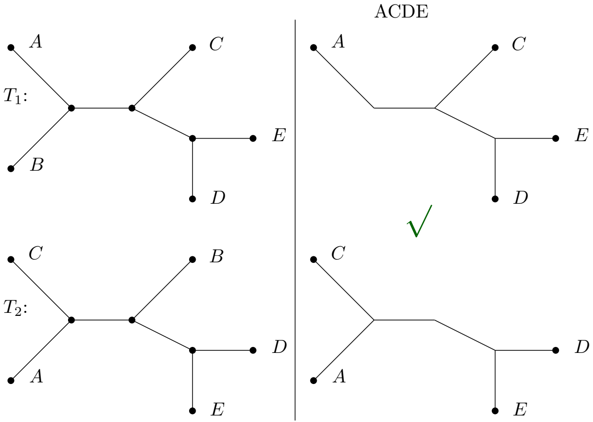




# Quartet distance

Input: two trees with the same set of leaves

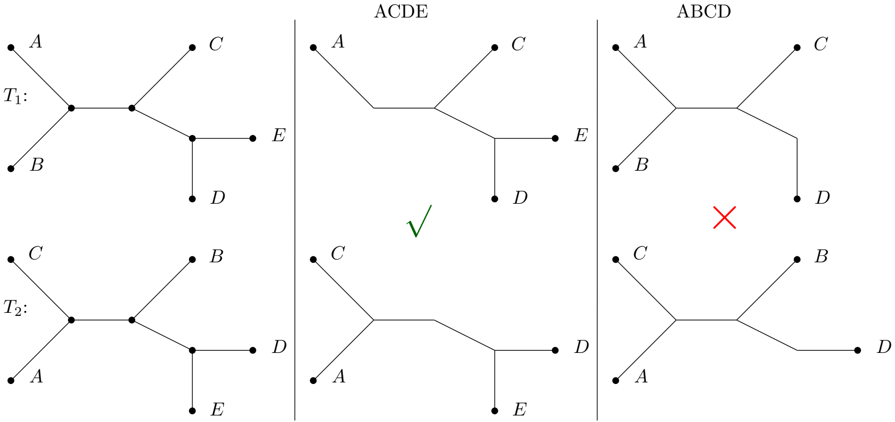
Output: number of quartets of leaves inducing different topologies



# Quartet distance

Input: two trees with the same set of leaves

Output: number of quartets of leaves inducing different topologies



# History of quartet distance

| Year | Authors | Runtime | Arbitrary degree |
|---|---|---|---|
| | Folklore | $\mathcal{O}(n^4)$ | ✓ |
| 1993 | Steel and Penny | $\mathcal{O}(n^3)$ | × |
| 2000 | Bryant et al. | $\mathcal{O}(n^2)$ | × |
| 2001 | Brodal et al. | $\mathcal{O}(n \log^2 n)$ | × |
| 2004 | Brodal et al. | $\mathcal{O}(n \log n)$ | × |
| 2007 | Stissing et al. | $\mathcal{O}(d^9 n \log n)$ | ✓ |
| 2011 | Nielsen et al. | $\mathcal{O}(n^{2.688})$ | ✓ |
| 2013 | Brodal et al. | $\mathcal{O}(dn \log n)$ | ✓ |

Can we do better?

| | | | |
|---|---|---|---|
| | | $\mathcal{O}(n^{1.48})$ | ✓ |
| | | $\mathcal{O}(d^{0.77} n)$ | ✓ |

Can we do even better?

| | | | |
|---|---|---|---|
| | | no $\mathcal{O}(n^{4/3-\varepsilon})$<br>(probably) | ✓ |

# History of quartet distance

| Year | Authors | Runtime | Arbitrary degree |
|---|---|---|---|
| | Folklore | $\mathcal{O}(n^4)$ | ✓ |
| 1993 | Steel and Penny | $\mathcal{O}(n^3)$ | × |
| 2000 | Bryant et al. | $\mathcal{O}(n^2)$ | × |
| 2001 | Brodal et al. | $\mathcal{O}(n \log^2 n)$ | × |
| 2004 | Brodal et al. | $\mathcal{O}(n \log n)$ | × |
| 2007 | Stissing et al. | $\mathcal{O}(d^9 n \log n)$ | ✓ |
| 2011 | Nielsen et al. | $\mathcal{O}(n^{2.688})$ | ✓ |
| 2013 | Brodal et al. | $\mathcal{O}(dn \log n)$ | ✓ |

Can we do better?

| | | | |
|---|---|---|---|
| | | $\mathcal{O}(n^{1.48})$ | ✓ |
| | | $\mathcal{O}(d^{0.77} n)$ | ✓ |

Can we do even better?

| | | | |
|---|---|---|---|
| | | no $\mathcal{O}(n^{4/3-\varepsilon})$<br>(probably) | ✓ |

# History of quartet distance

| Year | Authors | Runtime | Arbitrary degree |
|---|---|---|---|
|  | Folklore | $\mathcal{O}(n^4)$ | √ |
| 1993 | Steel and Penny | $\mathcal{O}(n^3)$ | × |
| 2000 | Bryant et al. | $\mathcal{O}(n^2)$ | × |
| 2001 | Brodal et al. | $\mathcal{O}(n \log^2 n)$ | × |
| 2004 | Brodal et al. | $\mathcal{O}(n \log n)$ | × |
| 2007 | Stissing et al. | $\mathcal{O}(d^9 n \log n)$ | √ |
| 2011 | Nielsen et al. | $\mathcal{O}(n^{2.688})$ | √ |
| 2013 | Brodal et al. | $\mathcal{O}(dn \log n)$ | √ |

Can we do better?

|  |  |  |  |
|---|---|---|---|
|  |  | $\mathcal{O}(n^{1.48})$ | √ |
|  |  | $\mathcal{O}(d^{0.77} n)$ | √ |

Can we do even better?

|  |  |  |  |
|---|---|---|---|
|  |  | no $\mathcal{O}(n^{4/3-\varepsilon})$<br>(probably) | √ |

# History of quartet distance

| Year | Authors | Runtime | Arbitrary degree |
|---|---|---|---|
| | Folklore | $\mathcal{O}(n^4)$ | ✓ |
| 1993 | Steel and Penny | $\mathcal{O}(n^3)$ | × |
| 2000 | Bryant et al. | $\mathcal{O}(n^2)$ | × |
| 2001 | Brodal et al. | $\mathcal{O}(n \log^2 n)$ | × |
| 2004 | Brodal et al. | $\mathcal{O}(n \log n)$ | × |
| 2007 | Stissing et al. | $\mathcal{O}(d^9 n \log n)$ | ✓ |
| 2011 | Nielsen et al. | $\mathcal{O}(n^{2.688})$ | ✓ |
| 2013 | Brodal et al. | $\mathcal{O}(dn \log n)$ | ✓ |

Can we do better?

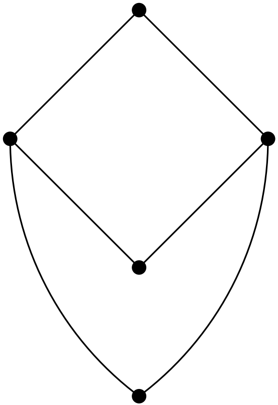
| | | | |
|---|---|---|---|
| | | $\mathcal{O}(n^{1.48})$ | ✓ |
| | | $\mathcal{O}(d^{0.77} n)$ | ✓ |

Can we do even better?

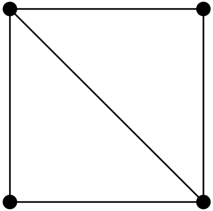
| | | | |
|---|---|---|---|
| | | no $\mathcal{O}(n^{4/3-\varepsilon})$<br>(probably) | ✓ |

# Counting 4-cycles

Input: simple, undirected graph
Output: number of simple cycles of length 4



$3$



$1$

# History of 2$k$-cycles

| Year | Authors | Runtime | Variant |
|---|---|---|---|
| | Folklore | $\mathcal{O}(n^3)$ | |
| 1997 | Alon et al. | $\mathcal{O}(n^\omega)$ | count 4-cycles |
| | | $\mathcal{O}(m^{4/3})$ | find a 4-cycle |
| 1997 | Yuster and Zwick | $\mathcal{O}(n^2)$ | find a 2$k$-cycle |
| 2015 | Vassilevska Williams et al. | $\mathcal{O}(m^{1.48})$ | count 4-cycles |
| 2017 | Dahlgaard et al. | $\mathcal{O}(m^{2k/(k+1)})$ | find a 2$k$-cycle |

**Conjecture [Yuster and Zwick, J. Discr. Math.'97]**

For every $\varepsilon > 0$ no algorithm detects 4-cycles in $\mathcal{O}(n^{2-\varepsilon})$ time.

⇓

**Conjecture [Dahlgaard et al., STOC'17]**

For every $\varepsilon > 0$ no algorithm detects 4-cycles in $\mathcal{O}(m^{4/3-\varepsilon})$ time.

# History of 2$k$-cycles

| Year | Authors | Runtime | Variant |
|---|---|---|---|
| | Folklore | $\mathcal{O}(n^3)$ | |
| 1997 | Alon et al. | $\mathcal{O}(n^\omega)$ | count 4-cycles |
| | | $\mathcal{O}(m^{4/3})$ | find a 4-cycle |
| 1997 | Yuster and Zwick | $\mathcal{O}(n^2)$ | find a 2$k$-cycle |
| 2015 | Vassilevska Williams et al. | $\mathcal{O}(m^{1.48})$ | count 4-cycles |
| 2017 | Dahlgaard et al. | $\mathcal{O}(m^{2k/(k+1)})$ | find a 2$k$-cycle |

**Conjecture [Yuster and Zwick, J. Discr. Math.'97]**

For every $\varepsilon > 0$ no algorithm detects 4-cycles in $\mathcal{O}(n^{2-\varepsilon})$ time.

$$\Downarrow$$

**Conjecture [Dahlgaard et al., STOC'17]**

For every $\varepsilon > 0$ no algorithm detects 4-cycles in $\mathcal{O}(m^{4/3-\varepsilon})$ time.

# History of $2k$-cycles

| Year | Authors | Runtime | Variant |
|---|---|---|---|
| | Folklore | $\mathcal{O}(n^3)$ | |
| 1997 | Alon et al. | $\mathcal{O}(n^\omega)$ | count 4-cycles |
| | | $\mathcal{O}(m^{4/3})$ | find a 4-cycle |
| 1997 | Yuster and Zwick | $\mathcal{O}(n^2)$ | find a $2k$-cycle |
| 2015 | Vassilevska Williams et al. | $\mathcal{O}(m^{1.48})$ | count 4-cycles |
| 2017 | Dahlgaard et al. | $\mathcal{O}(m^{2k/(k+1)})$ | find a $2k$-cycle |

**Conjecture [Yuster and Zwick, J. Discr. Math.'97]**

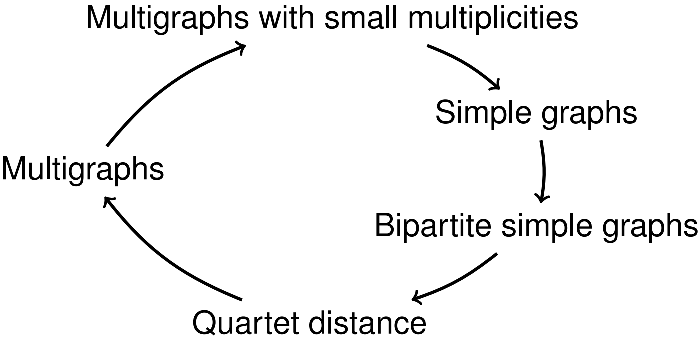
For every $\varepsilon > 0$ no algorithm detects 4-cycles in $\mathcal{O}(n^{2-\varepsilon})$ time.

$$\Downarrow$$

**Conjecture [Dahlgaard et al., STOC'17]**

For every $\varepsilon > 0$ no algorithm detects 4-cycles in $\mathcal{O}(m^{4/3-\varepsilon})$ time.

# Our contribution



Multigraphs with small multiplicities

Simple graphs

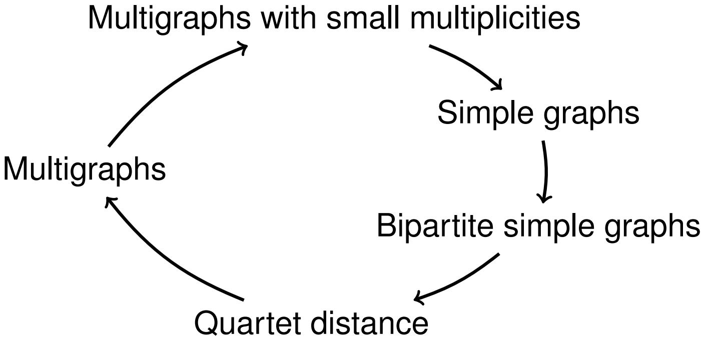
Bipartite simple graphs

Quartet distance

Multigraphs

(*) All reductions are up to polylogarithmic factors.

# Our contribution



Multigraphs with small multiplicities

Simple graphs

Bipartite simple graphs

Quartet distance

Multigraphs

(*) All reductions are up to polylogarithmic factors.

# Warm-up: $\#\Diamond$(simple graphs) $\Longrightarrow$ $\#$⧖(bipartite)

Input: simple graph $G$

Output: bipartite $G'$ such that $\#\Diamond(G)$ can be obtained from $\#$⧖$(G')$

A node $v$ → two nodes $v^{(1)}, v^{(2)}$.

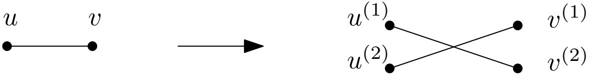
An edge $\{u, v\}$ → two edges $\{u^{(1)}, v^{(2)}\}, \{u^{(2)}, v^{(1)}\}$.

$\#\Diamond(G) = \frac{1}{2}\#$⧖$(G')$

# Warm-up: $\#\Diamond$(simple graphs) $\implies$ #⧖(bipartite)

Input: simple graph $G$
Output: bipartite $G'$ such that $\#\Diamond(G)$ can be obtained from #⧖$(G')$

A node $v$ $\rightarrow$ two nodes $v^{(1)}, v^{(2)}$.
An edge $\{u, v\}$ $\rightarrow$ two edges $\{u^{(1)}, v^{(2)}\}, \{u^{(2)}, v^{(1)}\}$.

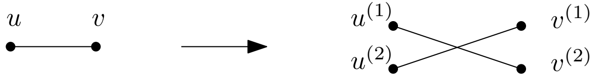


$\#\Diamond(G) = \frac{1}{2}$#⧖$(G')$

# Warm-up: $\#\Diamond$(simple graphs) $\Longrightarrow$ $\#$⧖(bipartite)

Input: simple graph $G$
Output: bipartite $G'$ such that $\#\Diamond(G)$ can be obtained from $\#$⧖$(G')$

A node $v$ → two nodes $v^{(1)}, v^{(2)}$.
An edge $\{u, v\}$ → two edges $\{u^{(1)}, v^{(2)}\}, \{u^{(2)}, v^{(1)}\}$.



$\#\Diamond(G) = \frac{1}{2}\#$⧖$(G')$

# Counting 4-edge subgraphs of a bipartite graph

How many quadruples of edges form $\lessdot$?

$$\# \lessdot = \sum_{v \in V_1} \binom{\deg(v)}{4}$$

Some shapes are more involved to count:

$$\# ≣ = \frac{1}{4}\left((m-3)(\# \equiv) - (\# \underline{\not<}) - 2(\# \underline{\leq}) - 2(\# \underline{\geq})\right)$$

# Counting 4-edge subgraphs of a bipartite graph

How many quadruples of edges form $\not<$?

$$\# \not< = \sum_{v \in V_1} \binom{\deg(v)}{4}$$

Some shapes are more involved to count:

$$\# \equiv\!\!\equiv \; = \frac{1}{4}\left( (m-3)(\# \equiv) - (\# \overset{/}{\underline{>}}) - 2(\# \leqq) - 2(\# \geqq) \right)$$

# Two types of 4-edge shapes

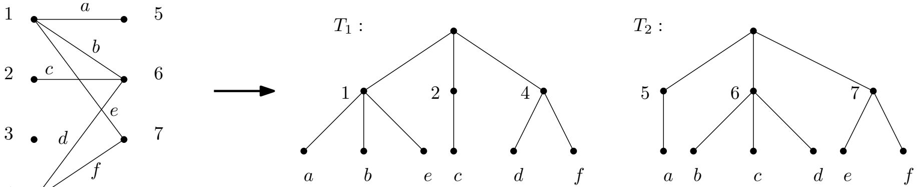
Shapes counted in linear time: $\not\Leftarrow, \ngeq, \underline{\nleftarrow}$ and $\lessgtr$

$\# \ngeq = \sum_{(u,v)\in E} \binom{d(u)-1}{2}(d(v)-1)$

Shapes equivalent to 4-cycles: $\gtrless, \overset{<}{<}, \underline{\ngtr}, \overset{<}{=}$

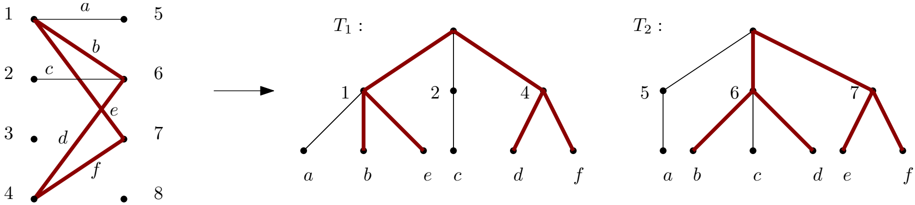
$\# \gtrless = \ldots - 2(\# \text{⧖})$
$\# \equiv = \ldots + 1(\# \text{⧖})$

# #⧖(bipartite graphs) $\Longrightarrow$ quartet distance



$$\binom{\#\text{ of leaves}}{4} - \mathtt{QD}(T_1, T_2) = (\#\,\lessdot)+(\#\,\gtrdot)+(\#\,\leq)+(\#\,\geq)+(\#\,\equiv)+(\#\,\lessgtr)$$

# #⧖(bipartite graphs) $\Longrightarrow$ quartet distance



$$\mathtt{QD}(T_1, T_2) = \ldots + (\# \text{⧖})$$

# #⧖(bipartite graphs) $\Longrightarrow$ quartet distance

**Theorem**

*Counting 4-cycles in a graph with m edges can be reduced in linear time to computing the quartet distance between two trees on $\mathcal{O}(m)$ leaves.*

**Conjecture [Dahlgaard et al. STOC'17]**

For every $\varepsilon > 0$ no algorithm detects 4-cycles in $\mathcal{O}(m^{4/3-\varepsilon})$ time.

$$\Downarrow$$

**Conjecture**

For any $\varepsilon > 0$, there is no $\mathcal{O}(n^{4/3-\varepsilon})$ algorithm for quartet distance.

# #⧖(bipartite graphs) $\implies$ quartet distance

### Theorem

*Counting 4-cycles in a graph with m edges can be reduced in linear time to computing the quartet distance between two trees on $\mathcal{O}(m)$ leaves.*

### Conjecture [Dahlgaard et al. STOC'17]

For every $\varepsilon > 0$ no algorithm detects 4-cycles in $\mathcal{O}(m^{4/3-\varepsilon})$ time.

$\Downarrow$

### Conjecture

For any $\varepsilon > 0$, there is no $\mathcal{O}(n^{4/3-\varepsilon})$ algorithm for quartet distance.

# #⧖(bipartite graphs) $\Longrightarrow$ quartet distance

### Theorem

*Counting 4-cycles in a graph with m edges can be reduced in linear time to computing the quartet distance between two trees on $\mathcal{O}(m)$ leaves.*

### Conjecture [Dahlgaard et al. STOC'17]

For every $\varepsilon > 0$ no algorithm detects 4-cycles in $\mathcal{O}(m^{4/3-\varepsilon})$ time.

$$\Downarrow$$

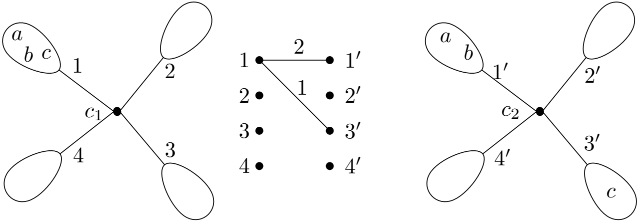
### Conjecture

For any $\varepsilon > 0$, there is no $\mathcal{O}(n^{4/3-\varepsilon})$ algorithm for quartet distance.

# Quartet distance $\Longrightarrow$ (# ≣)(multigraphs)

Shared )−( : $\mathcal{O}(n \log n)$ algorithm by Brodal et al. [SODA'13].

Shared ╳ : consider all pairs $(c_1, c_2)$ of central nodes



and count 4-matchings ≣.

# Counting shared stars ( $\times$ )

Problems:

- Cannot have $\Theta(n^2)$ subproblems
- Need to control the size of subproblems

Techniques:

1. top tree (hierarchical) decomposition
2. heavy-light decomposition
3. extended LCA
4. orthogonal range queries
5. ...

**Theorem**

*Counting all shared stars $\implies$ many instances of $(\# \equiv)$ in multigraphs of total size $\tilde{O}(n)$.*

# Counting shared stars ( ╳ )

Problems:

- Cannot have $\Theta(n^2)$ subproblems
- Need to control the size of subproblems

Techniques:

1. top tree (hierarchical) decomposition
2. heavy-light decomposition
3. extended LCA
4. orthogonal range queries
5. ...

**Theorem**

*Counting all shared stars $\implies$ many instances of $(\# \equiv)$ in multigraphs of total size $\tilde{O}(n)$.*

# Counting shared stars ( ╳ )

Problems:

- Cannot have $\Theta(n^2)$ subproblems
- Need to control the size of subproblems

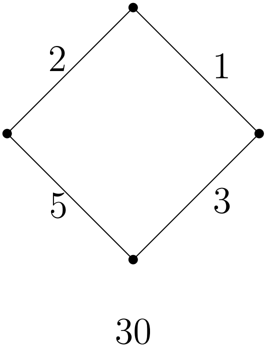
Techniques:

1. top tree (hierarchical) decomposition
2. heavy-light decomposition
3. extended LCA
4. orthogonal range queries
5. ...

## Theorem

*Counting all shared stars $\implies$ many instances of $(\# \equiv)$ in multigraphs of total size $\tilde{\mathcal{O}}(n)$.*

# Counting 4-cycles in multigraphs



arbitrary multiplicities ($\leq U$)

$\Downarrow$

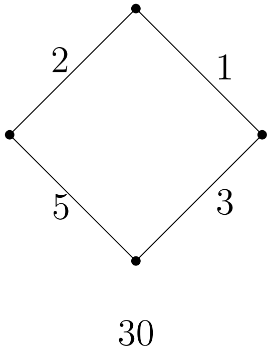
small multiplicities ($\leq c$)

$\Downarrow$

simple graphs

$c = 501^{125}$

# Counting 4-cycles in multigraphs



arbitrary multiplicities ($\leq U$)

$$\Downarrow$$

small multiplicities ($\leq c$)

$$\Downarrow$$

simple graphs

$c = 501^{125}$

# $\#\Diamond$(multigraphs) $\implies$ $\#\Diamond$(small multiplicities)

How to group the cycles?

How many 4-cycles have multiedges with multiplicities $3, 5, 8$ and $10$?

**Colorful cycles**

For every coloring of edges $K : E \to \{1, 2, 3, 4, \perp\}$ into 4 colors, we can compute $f_K(a, b, c, d)$ in $\mathcal{O}(1)$ black-box calls to counting 4-cycles in multigraphs with small multiplicities.

Naïve application: check all possible $U^4$ multisets of multiplicities.

# $\#\Diamond$(multigraphs) $\implies$ $\#\Diamond$(small multiplicities)

How to group the cycles?

How many 4-cycles have multiedges with multiplicities $3, 5, 8$ and 10?

### Colorful cycles

For every coloring of edges $K : E \to \{1, 2, 3, 4, \perp\}$ into 4 colors, we can compute $f_K(a, b, c, d)$ in $\mathcal{O}(1)$ black-box calls to counting 4-cycles in multigraphs with small multiplicities.

Naïve application: check all possible $U^4$ multisets of multiplicities.

# $\#\diamond$(multigraphs) $\implies$ $\#\diamond$(small multiplicities)

How to group the cycles?

How many 4-cycles have multiedges with multiplicities $3, 5, 8$ and 10?

## Colorful cycles

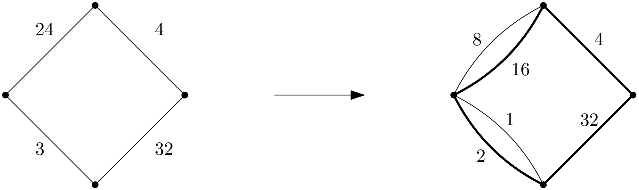
For every coloring of edges $K : E \to \{1, 2, 3, 4, \perp\}$ into 4 colors, we can compute $f_K(a, b, c, d)$ in $\mathcal{O}(1)$ black-box calls to counting 4-cycles in multigraphs with small multiplicities.

Naïve application: check all possible $U^4$ multisets of multiplicities.

# Using powers of 2

Aim: $\mathcal{O}(\log^4 U)$ groups.

Check all multisets of powers of 2.



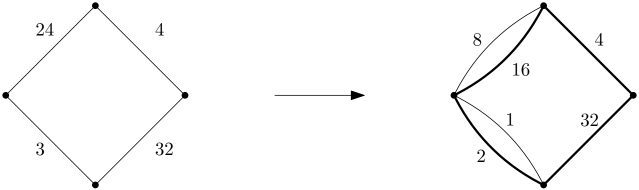
No edge contains two chosen powers of 2 $\implies$ colorful cycles. $\surd$

What if there is more than one chosen power of 2 in one edge? $\times$

# Using powers of 2

Aim: $\mathcal{O}(\log^4 U)$ groups.

Check all multisets of powers of 2.



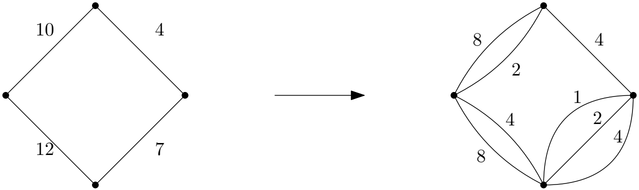
No edge contains two chosen powers of 2 $\implies$ colorful cycles. $\surd$

What if there is more than one chosen power of 2 in one edge? ×

# Using powers of 2

Aim: $\mathcal{O}(\log^4 U)$ groups.

Check all multisets of powers of 2.



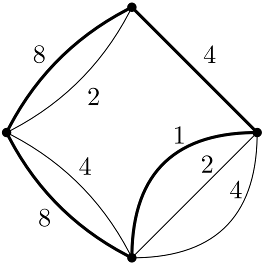
No edge contains two chosen powers of 2 $\implies$ colorful cycles. √

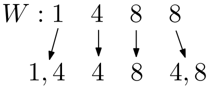
What if there is more than one chosen power of 2 in one edge? ×

# Smarter grouping of cycles

Choose:

1. multiset of weights $W = \{p_1, p_2, p_3, p_4\}$
2. for every $i$: set $M_i$ such that $p_i \in M_i$ and $M_i = \textsc{Bin}(\textsc{Mult}(e)) \cap W$





$W : 1 \quad 4 \quad 8 \quad 8$

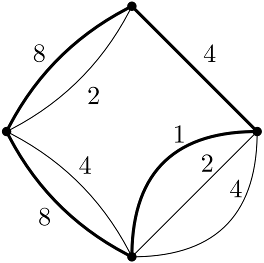
$1,4 \quad 4 \quad 8 \quad 4,8$

Sets $M_i$ are the new colors $\implies$ colorful cycles. $\surd$

# Smarter grouping of cycles

Choose:

1. multiset of weights $W = \{p_1, p_2, p_3, p_4\}$
2. for every $i$: set $M_i$ such that $p_i \in M_i$ and $M_i = \textsc{Bin}(\textsc{Mult}(e)) \cap W$



$$\begin{tikzcd}[column sep=small] W : 1 \arrow[d] & 4 \arrow[d] & 8 \arrow[d] & 8 \arrow[d] \\ 1,4 & 4 & 8 & 4,8 \end{tikzcd}$$

Sets $M_i$ are the new colors $\implies$ colorful cycles. $\surd$

# Summary

## Theorem

*An $\mathcal{O}(n^\delta)$-time algortithm for quartet distance gives $\mathcal{O}(m^\delta)$-time algorithm for counting 4-cycles in simple graphs.*

implies probably no $\mathcal{O}(n^{4/3-\varepsilon})$-time algorithm for QD

## Theorem

*Counting 4-cycles in simple graphs in $\mathcal{O}(m^\delta)$ time gives $\tilde{\mathcal{O}}(n^\delta)$-time algorithm for quartet distance.*

implies an $\mathcal{O}(n^{1.48})$-time algorithm for QD

Questions?

# Summary

**Theorem**

*An $\mathcal{O}(n^\delta)$-time algortithm for quartet distance gives $\mathcal{O}(m^\delta)$-time algorithm for counting 4-cycles in simple graphs.*

implies probably no $\mathcal{O}(n^{4/3-\varepsilon})$-time algorithm for QD

**Theorem**

*Counting 4-cycles in simple graphs in $\mathcal{O}(m^\delta)$ time gives $\tilde{\mathcal{O}}(n^\delta)$-time algorithm for quartet distance.*

implies an $\mathcal{O}(n^{1.48})$-time algorithm for QD

Questions?

# Summary

## Theorem

*An $\mathcal{O}(n^{\delta})$-time algortithm for quartet distance gives $\mathcal{O}(m^{\delta})$-time algorithm for counting 4-cycles in simple graphs.*

implies probably no $\mathcal{O}(n^{4/3-\varepsilon})$-time algorithm for $\mathrm{QD}$

## Theorem

*Counting 4-cycles in simple graphs in $\mathcal{O}(m^{\delta})$ time gives $\tilde{\mathcal{O}}(n^{\delta})$-time algorithm for quartet distance.*

implies an $\mathcal{O}(n^{1.48})$-time algorithm for $\mathrm{QD}$

Questions?

# Summary

**Theorem**

*An $\mathcal{O}(n^{\delta})$-time algortithm for quartet distance gives $\mathcal{O}(m^{\delta})$-time algorithm for counting 4-cycles in simple graphs.*

implies probably no $\mathcal{O}(n^{4/3-\varepsilon})$-time algorithm for $\texttt{QD}$

**Theorem**

*Counting 4-cycles in simple graphs in $\mathcal{O}(m^{\delta})$ time gives $\tilde{\mathcal{O}}(n^{\delta})$-time algorithm for quartet distance.*

implies an $\mathcal{O}(n^{1.48})$-time algorithm for $\texttt{QD}$

Questions?

# Summary

**Theorem**

*An $\mathcal{O}(n^\delta)$-time algortithm for quartet distance gives $\mathcal{O}(m^\delta)$-time algorithm for counting 4-cycles in simple graphs.*

implies probably no $\mathcal{O}(n^{4/3-\varepsilon})$-time algorithm for $\texttt{QD}$

**Theorem**

*Counting 4-cycles in simple graphs in $\mathcal{O}(m^\delta)$ time gives $\tilde{\mathcal{O}}(n^\delta)$-time algorithm for quartet distance.*

implies an $\mathcal{O}(n^{1.48})$-time algorithm for $\texttt{QD}$

Questions?