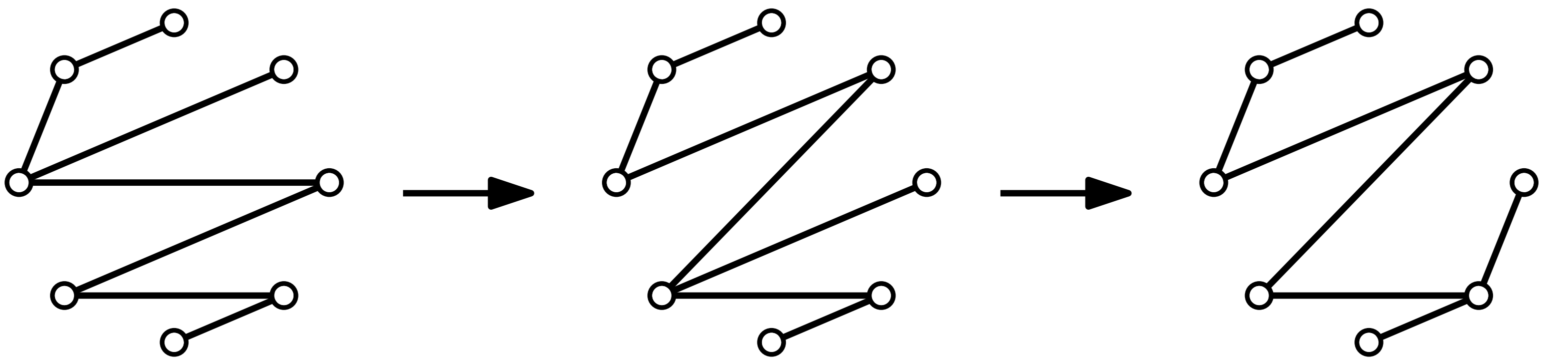
Constrained Flips in Plane Spanning Trees

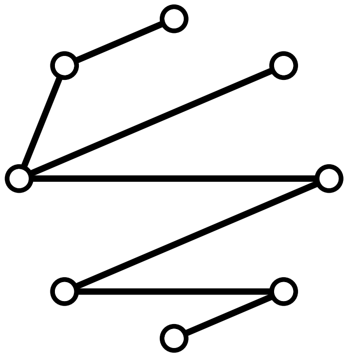
Oswin Aichholzer, Joseph Dorfer, Birgit Vogtenhuber

Graz University of Technology



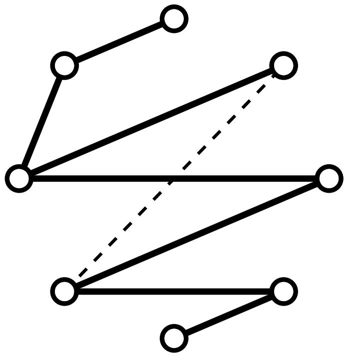
Flips

Flips in plane trees: Remove edge from tree, add different edge, get another plane tree.



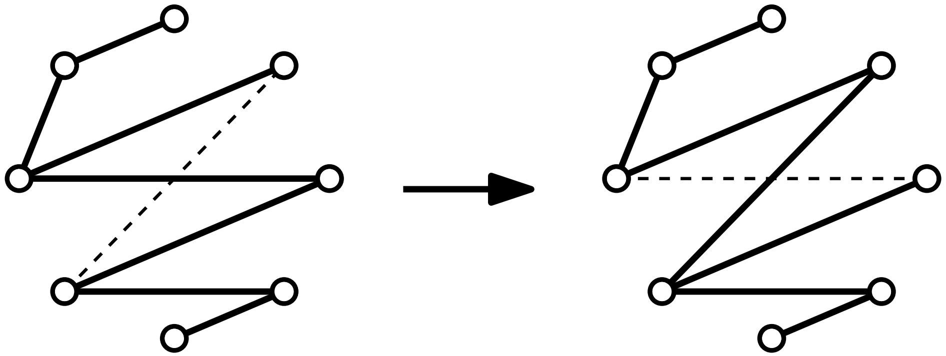
Flips

Flips in plane trees: Remove edge from tree, add different edge, get another plane tree.



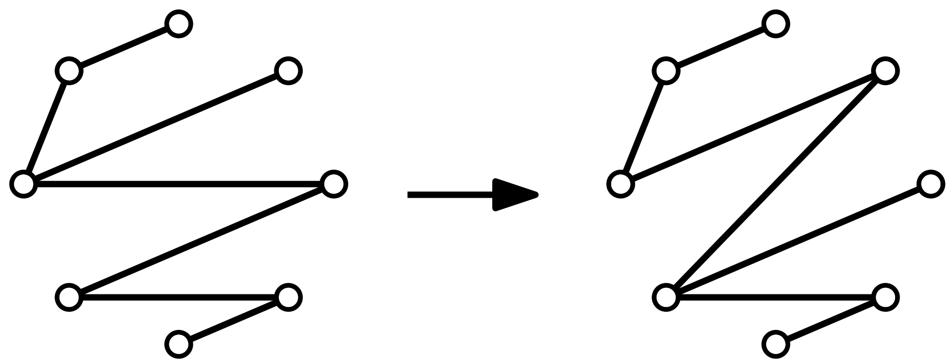
Flips

Flips in plane trees: Remove edge from tree, add different edge, get another plane tree.



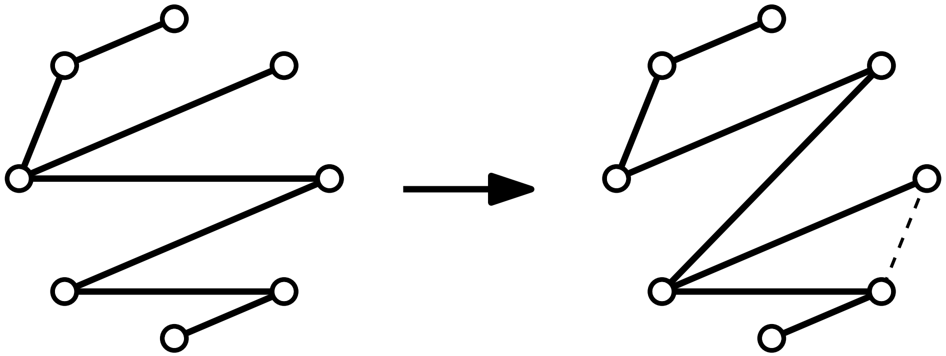
Flips

Flips in plane trees: Remove edge from tree, add different edge, get another plane tree.



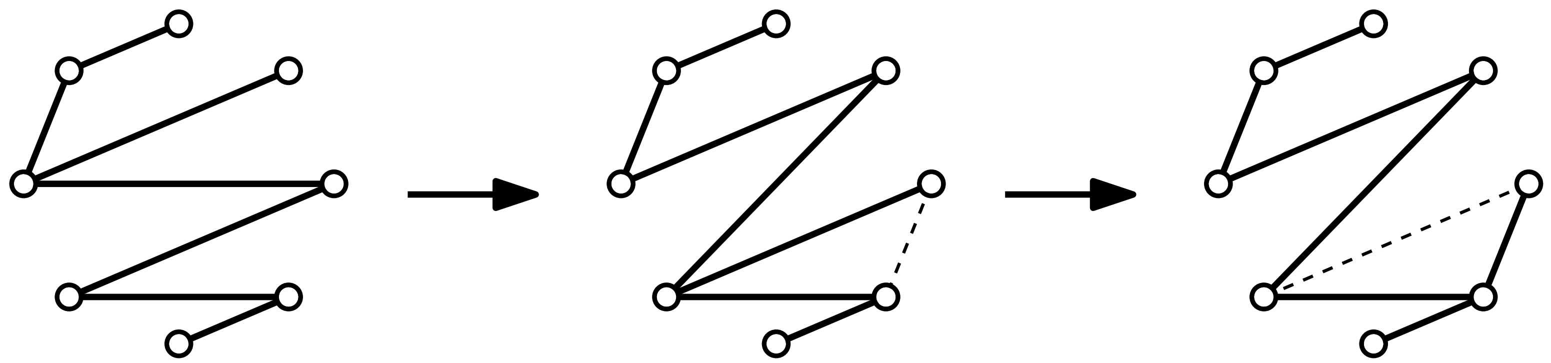
Flips

Flips in plane trees: Remove edge from tree, add different edge, get another plane tree.



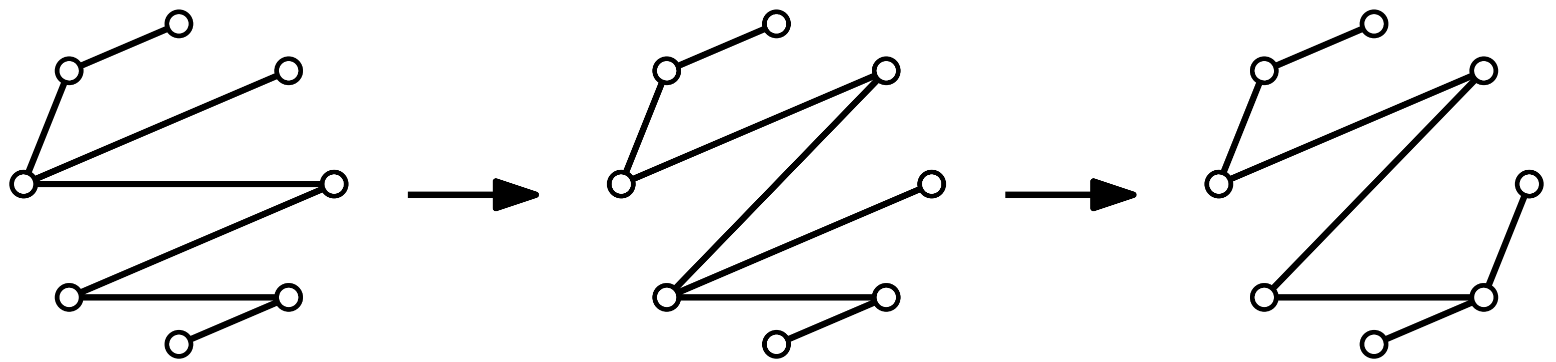
Flips

Flips in plane trees: Remove edge from tree, add different edge, get another plane tree.



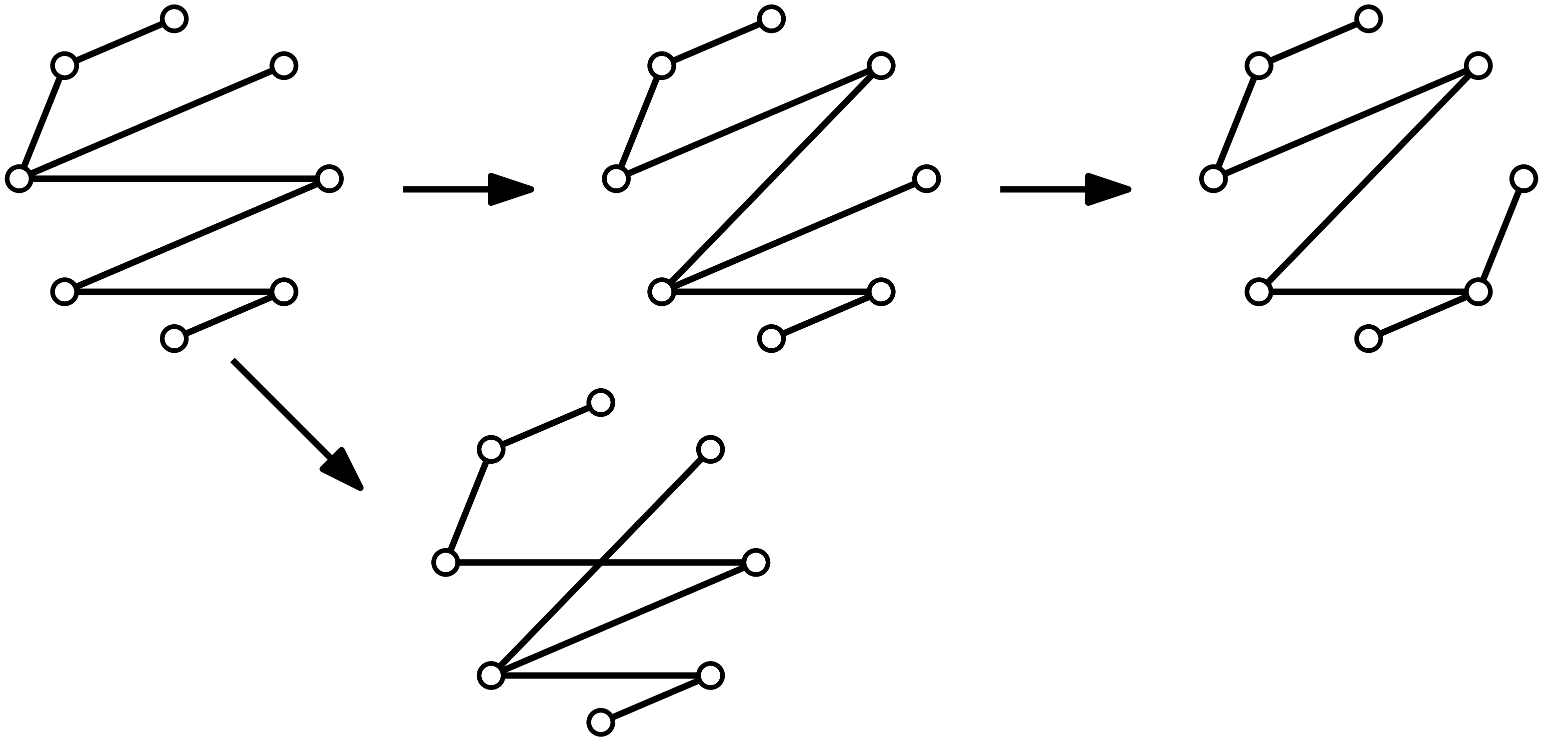
Flips

Flips in plane trees: Remove edge from tree, add different edge, get another plane tree.



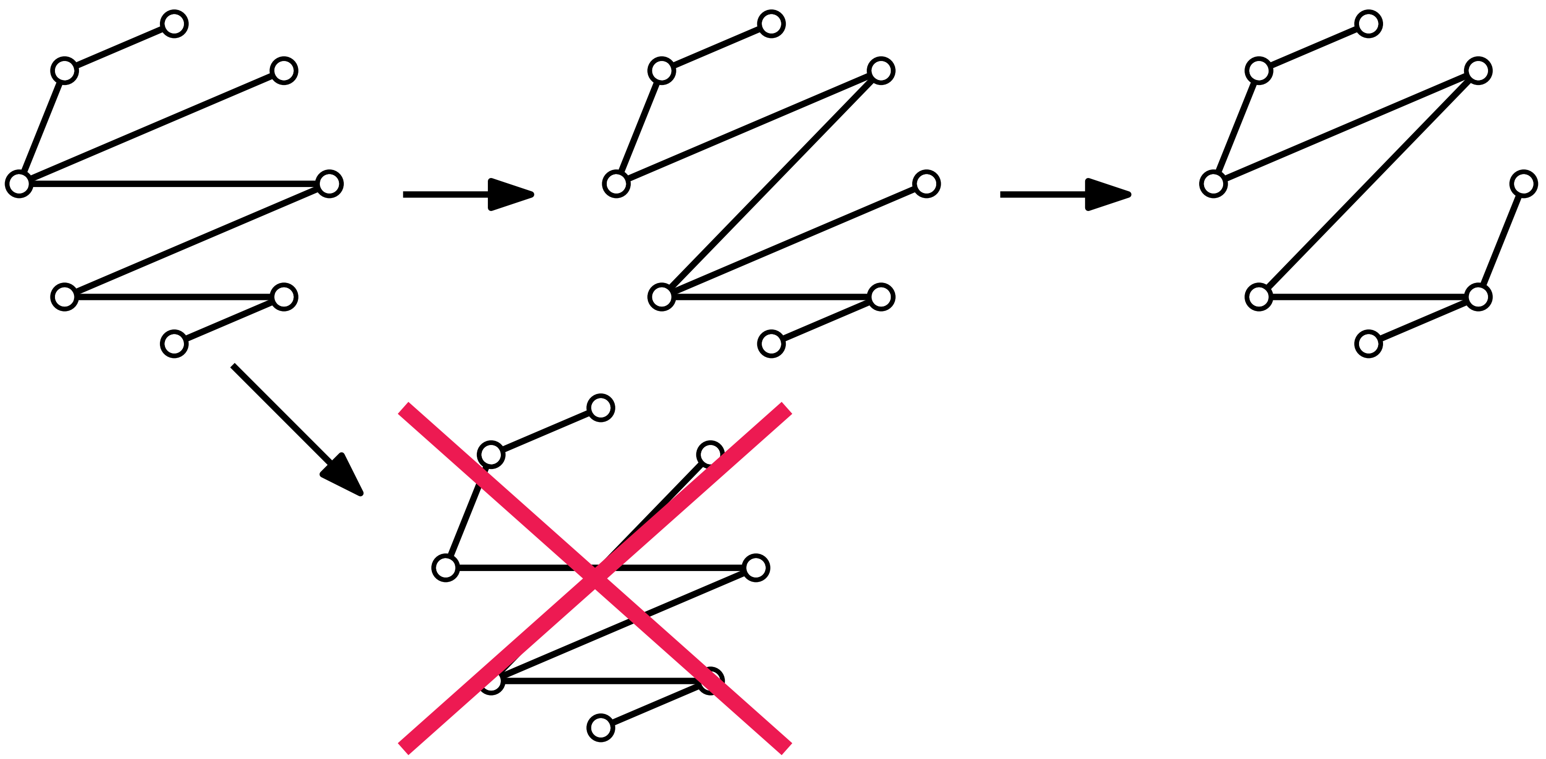
Flips

Flips in plane trees: Remove edge from tree, add different edge, get another plane tree.



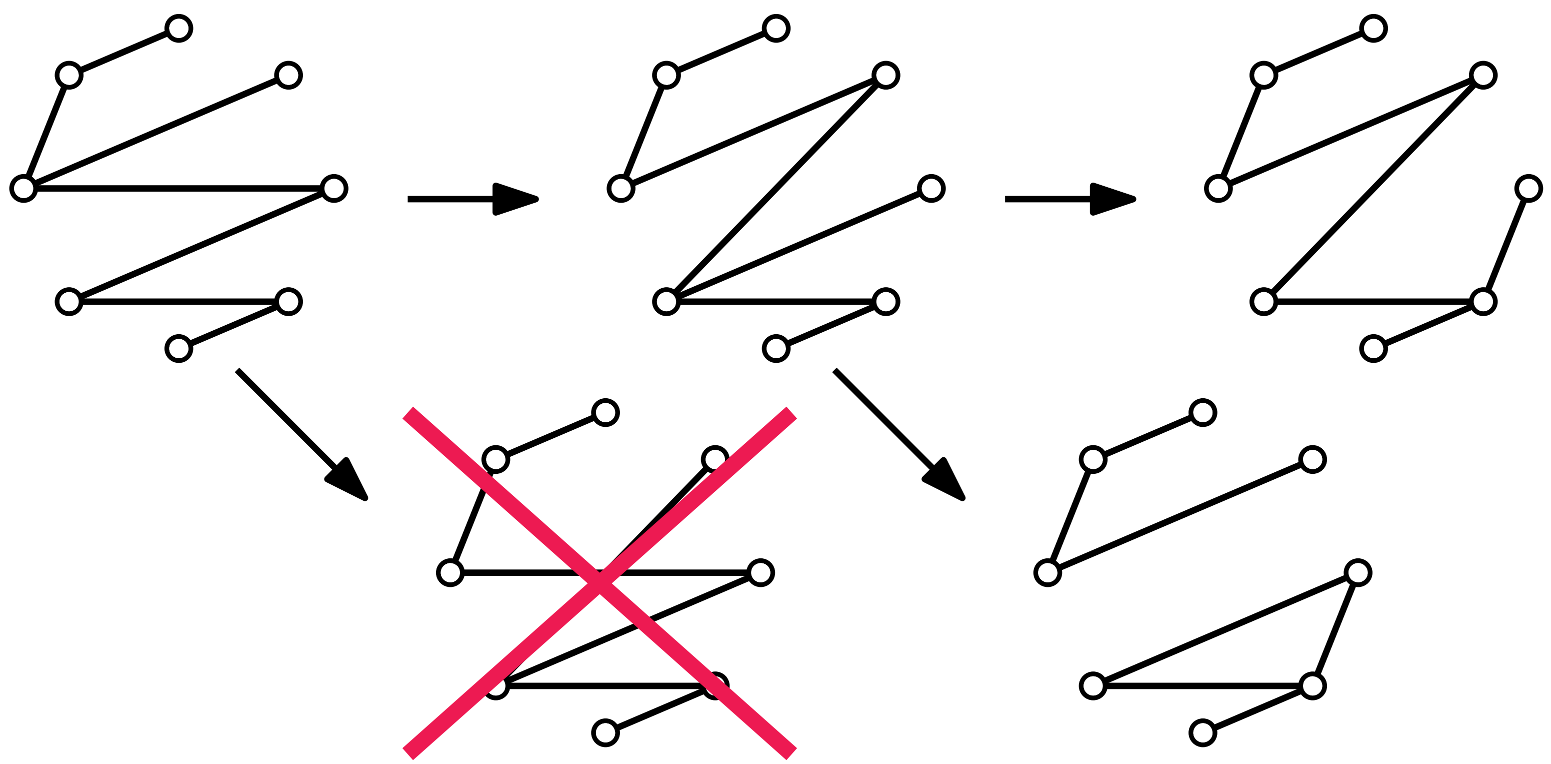
Flips

Flips in plane trees: Remove edge from tree, add different edge, get another plane tree.



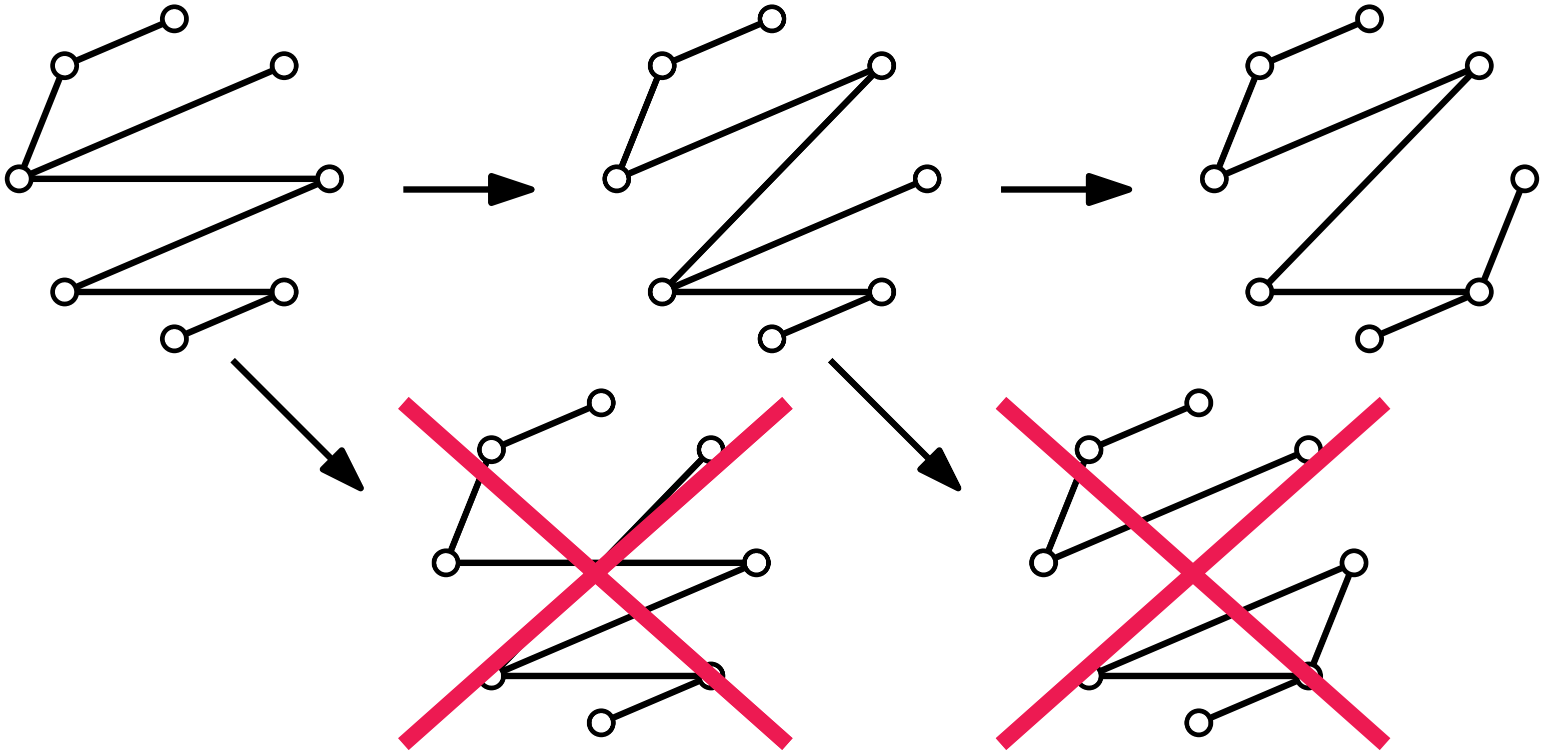
Flips

Flips in plane trees: Remove edge from tree, add different edge, get another plane tree.

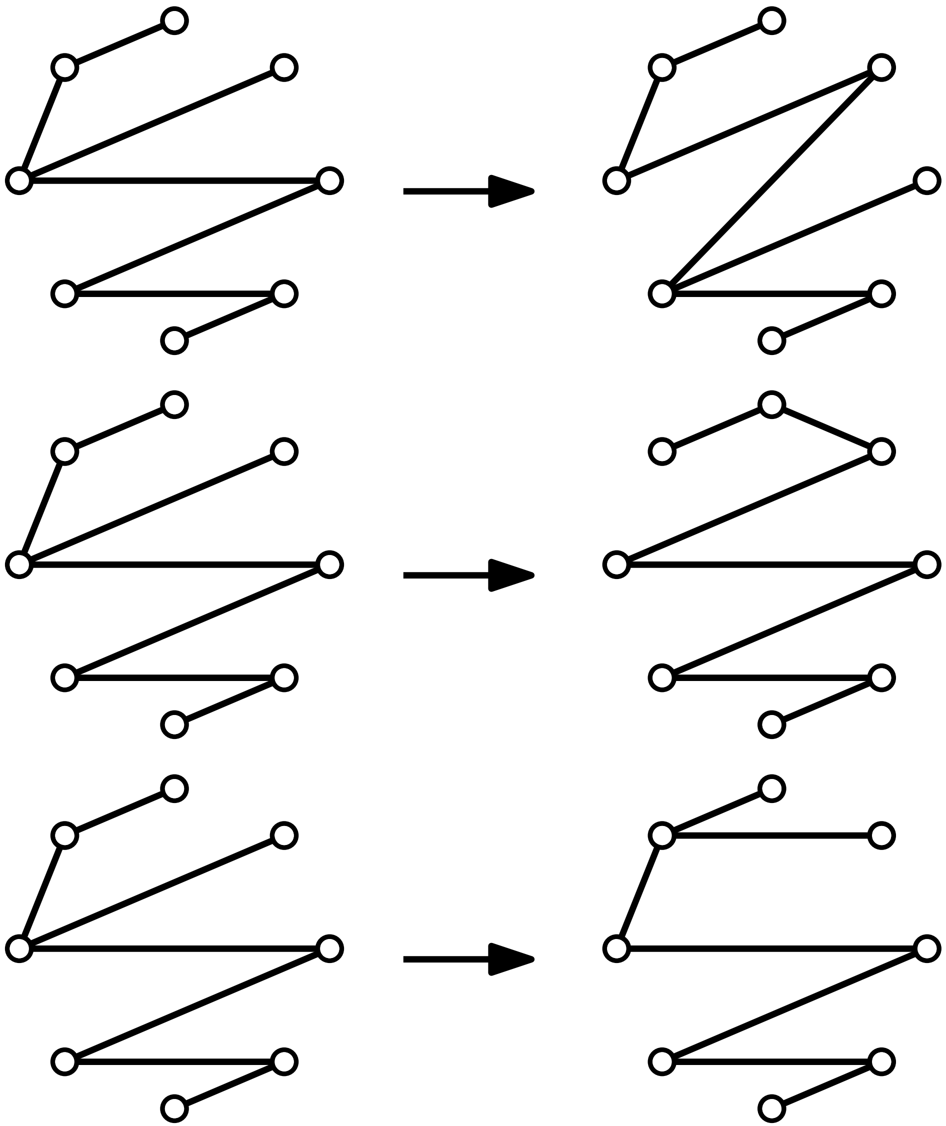


Flips

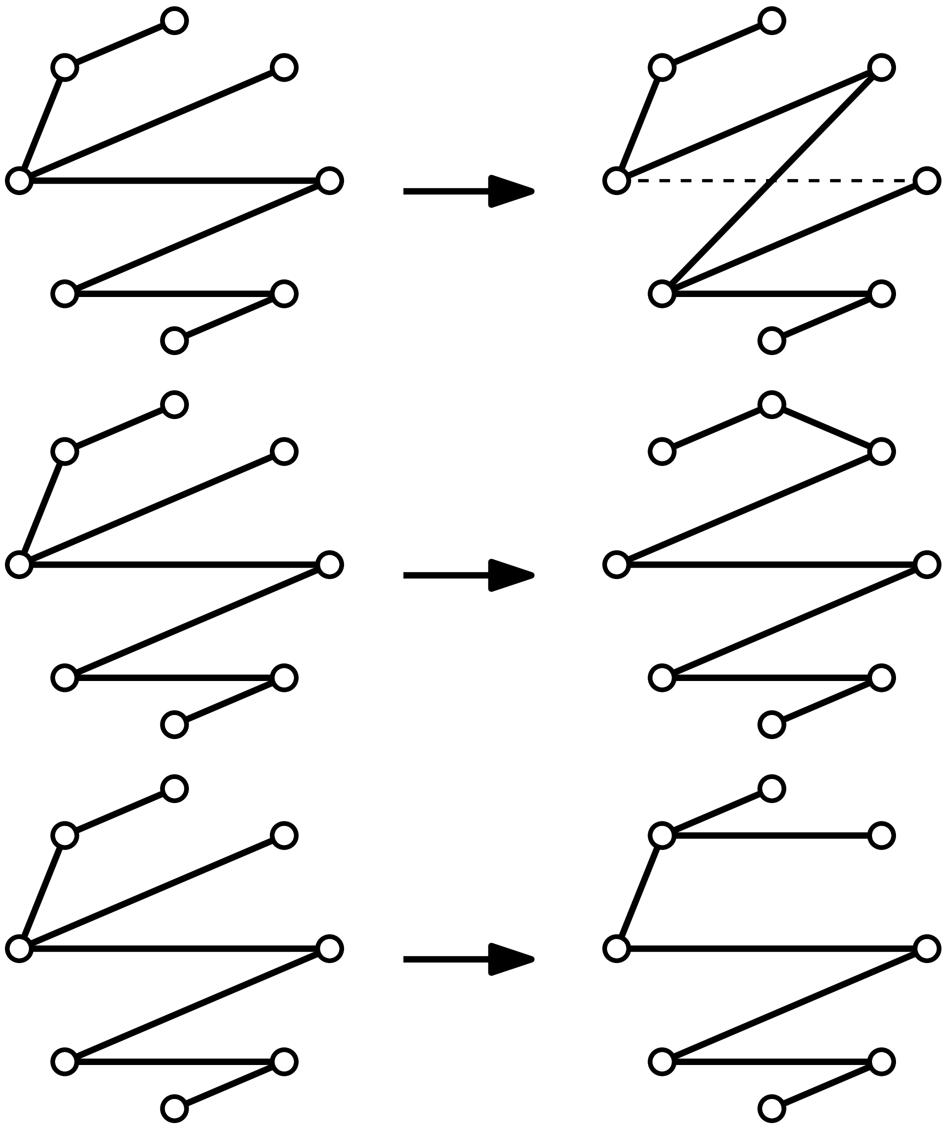
Flips in plane trees: Remove edge from tree, add different edge, get another plane tree.



Types of flips

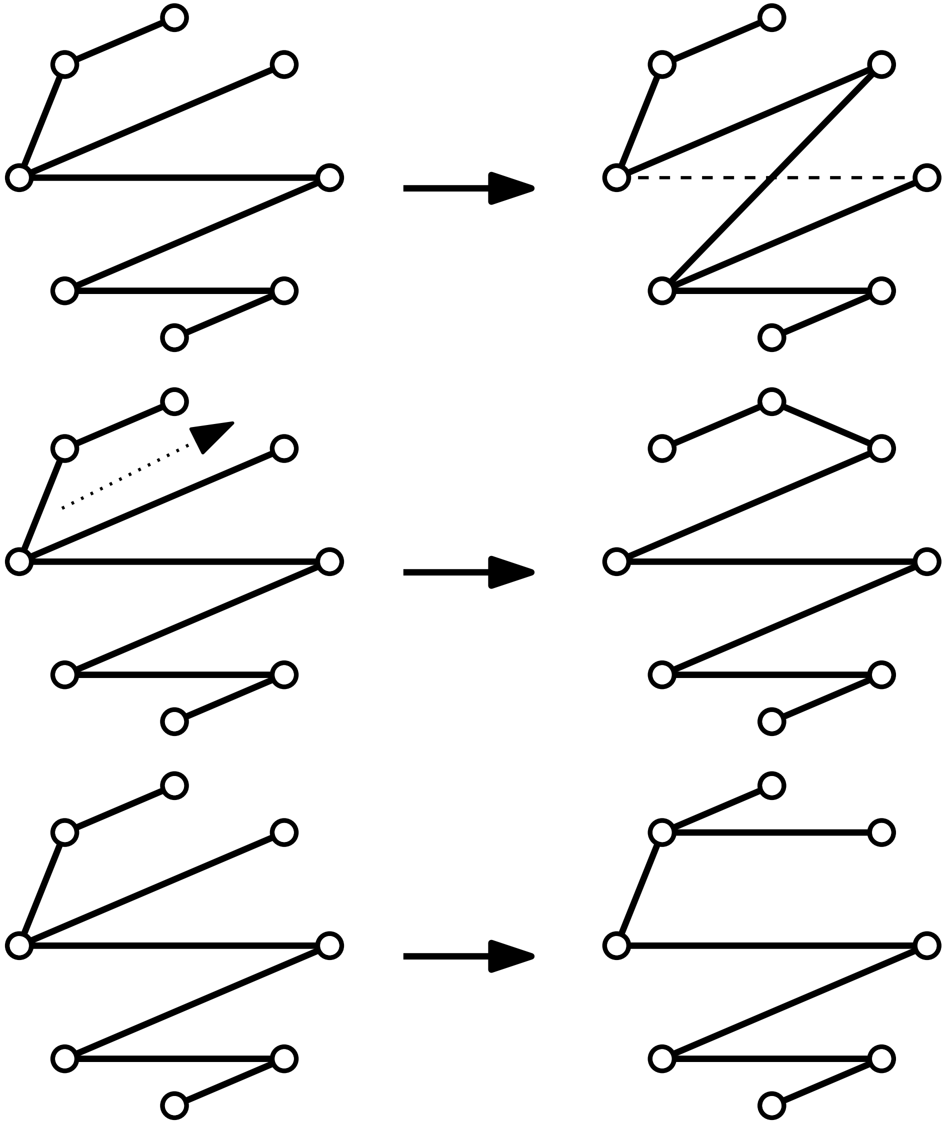


Types of flips



Crossing Flip:
Added and removed edge cross

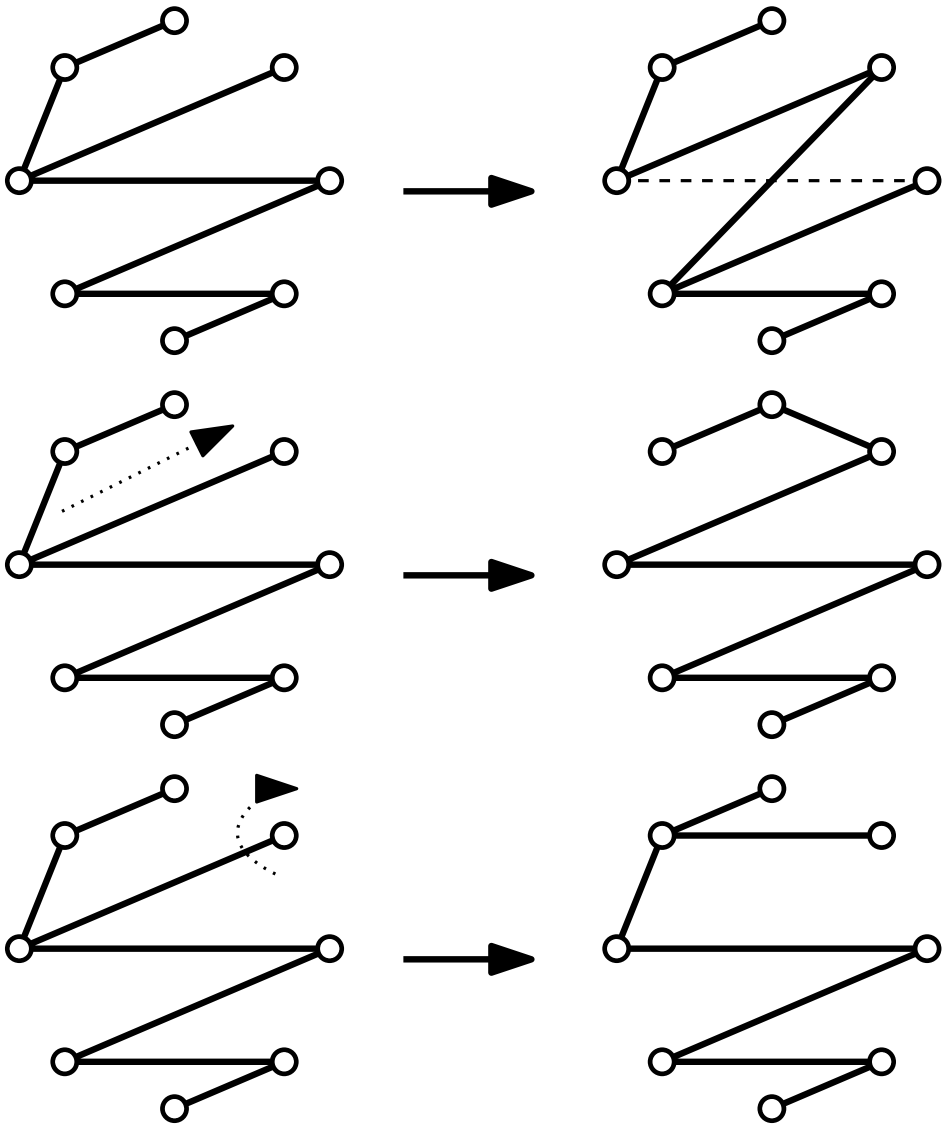
Types of flips



Crossing Flip:
Added and removed edge cross

Compatible Flip:
Added and removed edge do not cross

Types of flips



Crossing Flip:
Added and removed edge cross

Compatible Flip:
Added and removed edge do not cross

Rotation:
Added and removed edge share a vertex

Flipping - Central Questions

Connectedness: Transform any structure into any other via flips?

Flipping - Central Questions

Connectedness: Transform any structure into any other via flips?

Diameter: Worst case number of flips between configurations?

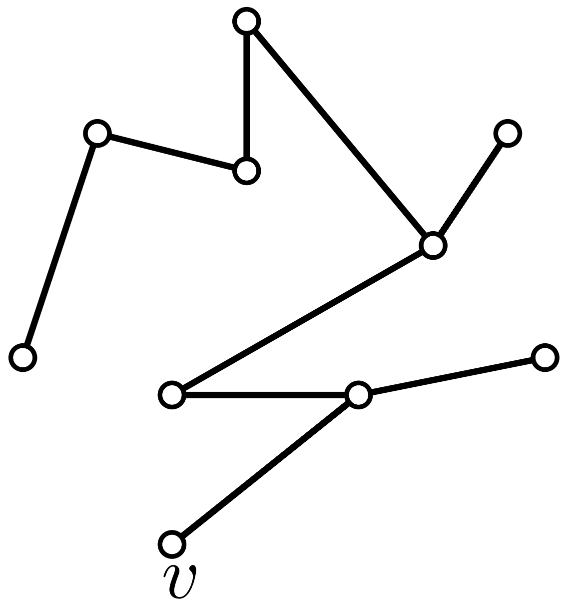
Flipping - Central Questions

Connectedness: Transform any structure into any other via flips?

Diameter: Worst case number of flips between configurations?

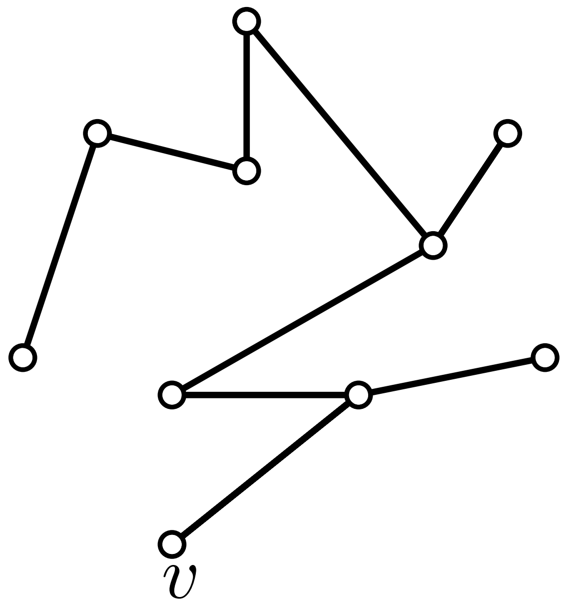
Complexity: Two specific configurations: How many flips needed? How to compute flip sequence?

Connectedness	Diameter	Complexity

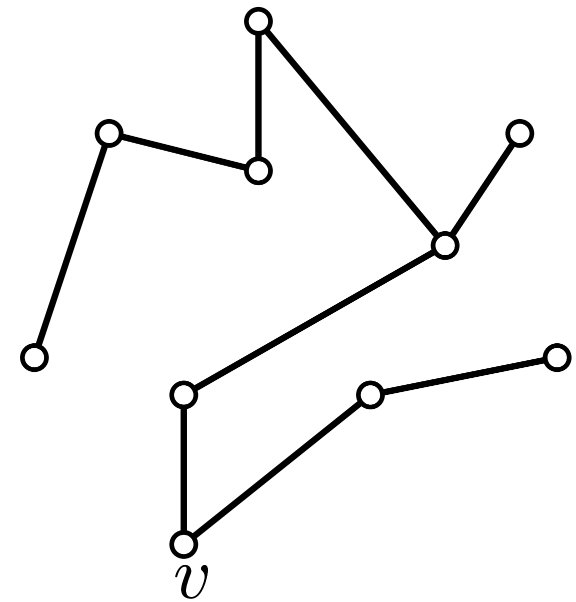


[Avis, Fukuda 1996]

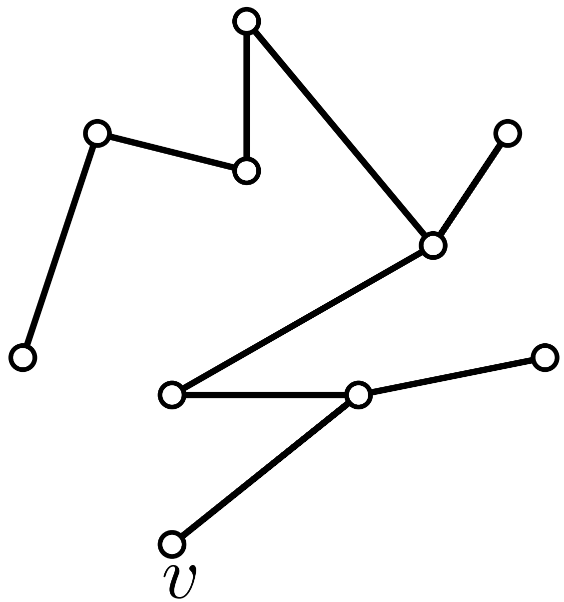
Connectedness	Diameter	Complexity



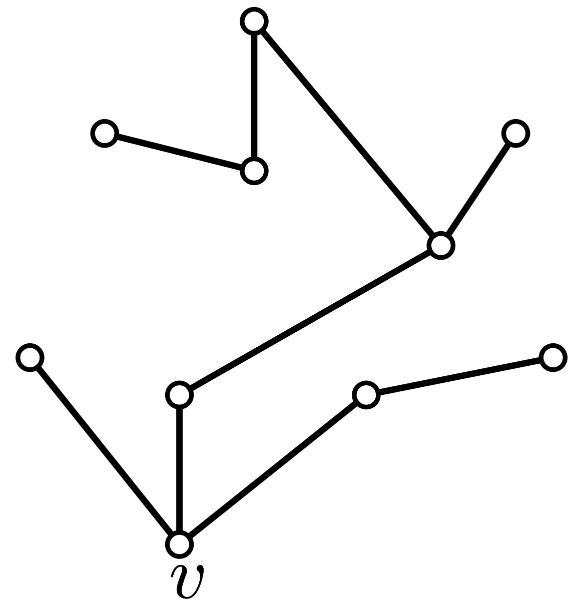
Pick a vertex v on the boundary.
Flip into fan at v.



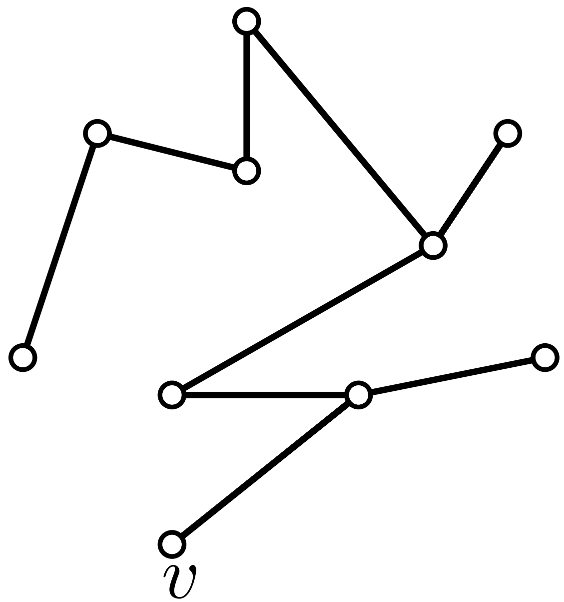
[Avis, Fukuda 1996]



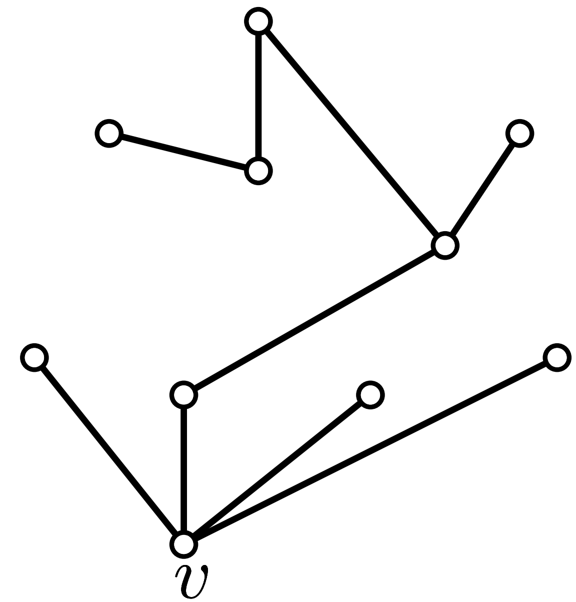
Pick a vertex v on the boundary.
Flip into fan at v.



[Avis, Fukuda 1996]

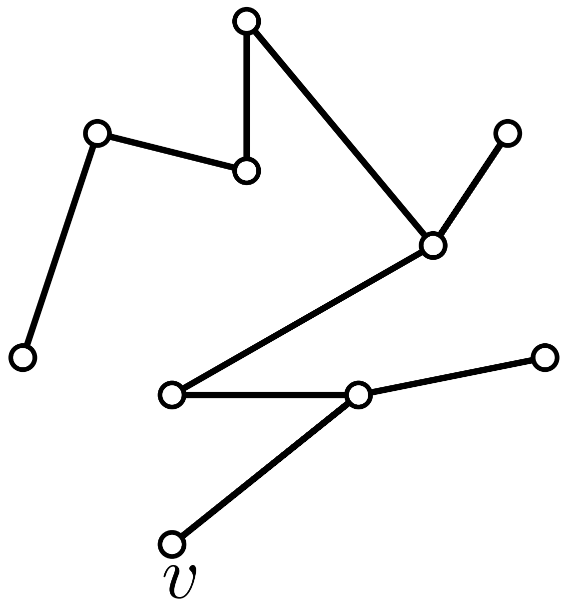


Pick a vertex v on the boundary.
Flip into fan at v.

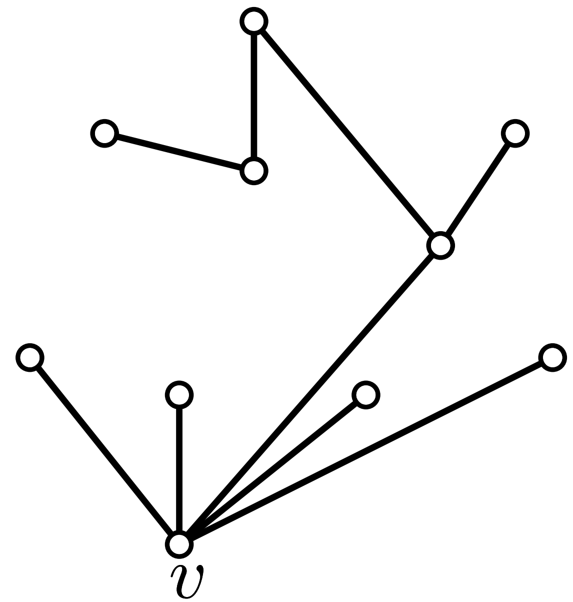


[Avis, Fukuda 1996]

Connectedness	**Diameter**	**Complexity**

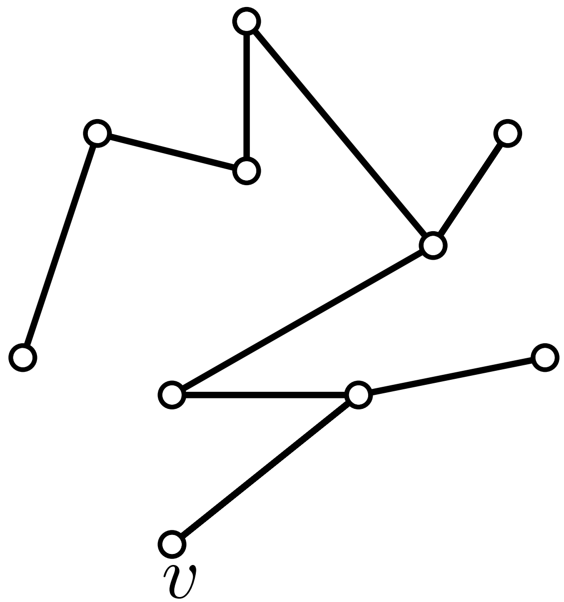


Pick a vertex v on the boundary.
Flip into fan at v.

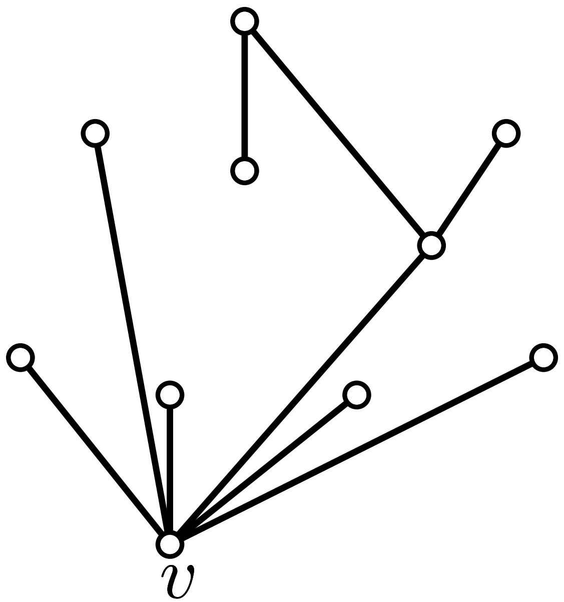


[Avis, Fukuda 1996]

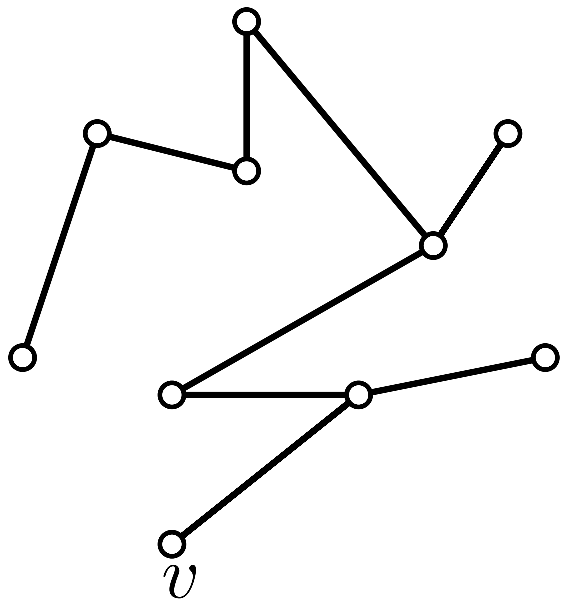
Connectedness	Diameter	Complexity



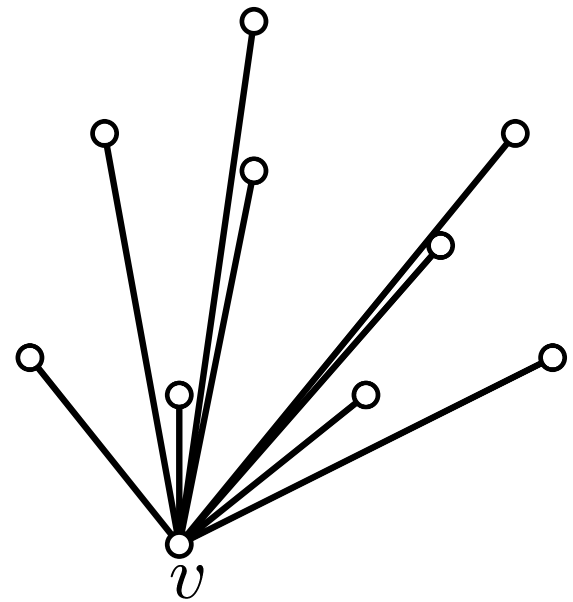
Pick a vertex v on the boundary.
Flip into fan at v.



[Avis, Fukuda 1996]

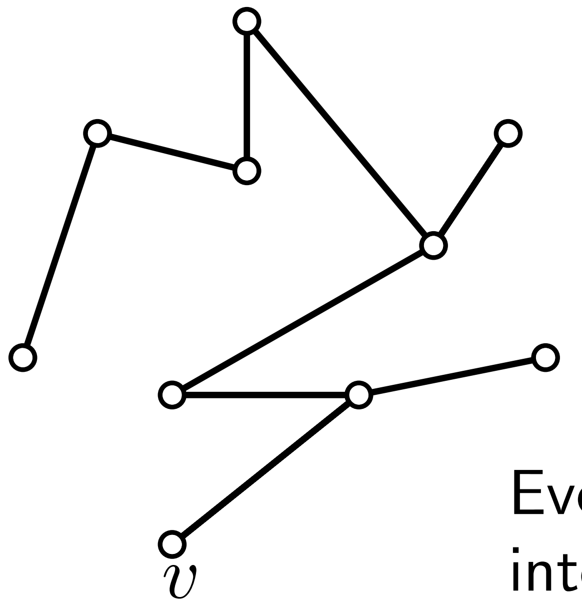


Pick a vertex v on the boundary.
Flip into fan at v.

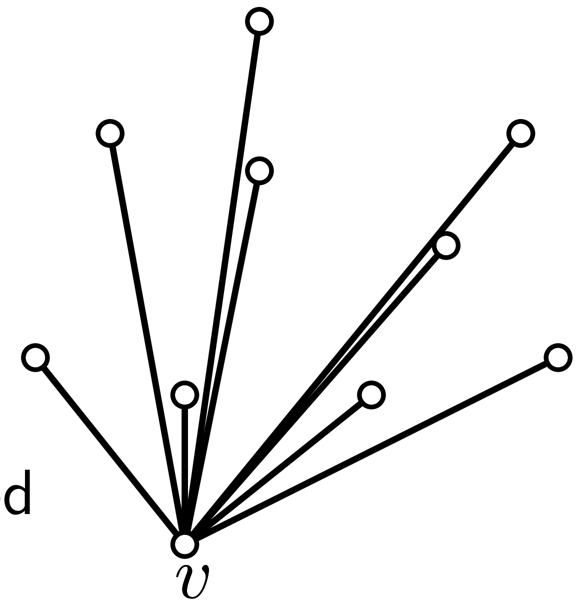


[Avis, Fukuda 1996]

Connectedness	Diameter	Complexity



Pick a vertex v on the boundary.
Flip into fan at v.

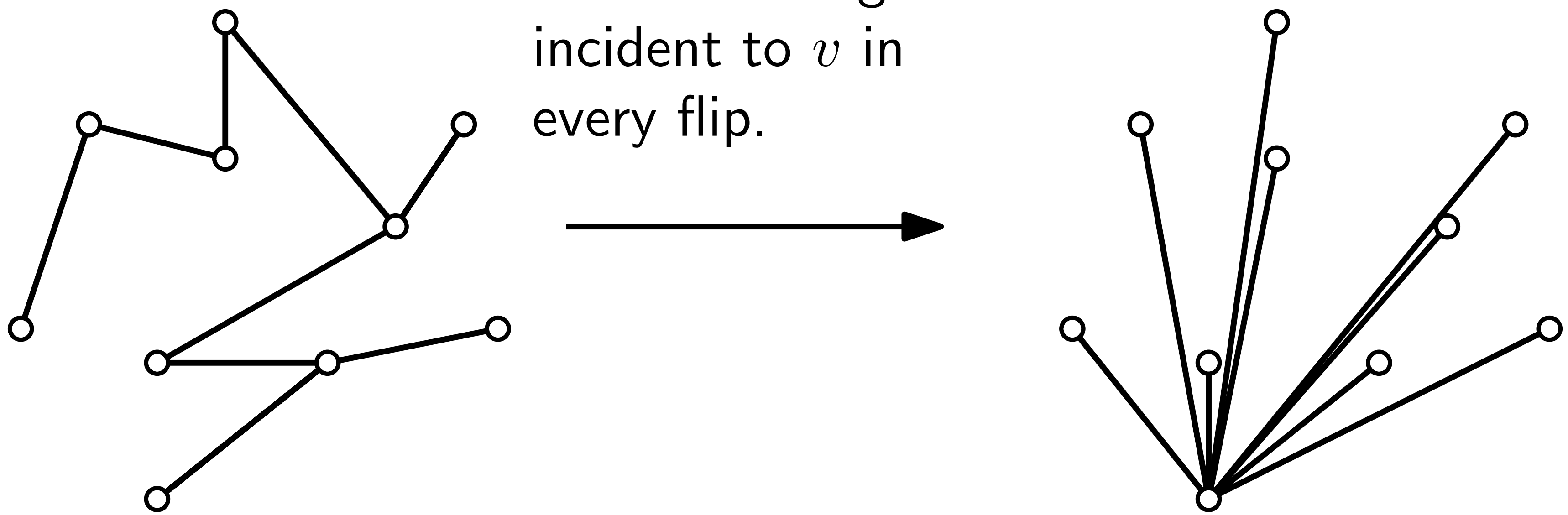


Every tree can be flipped into a fan at some boundary vertex and back.
$\Rightarrow$ **Connectedness**

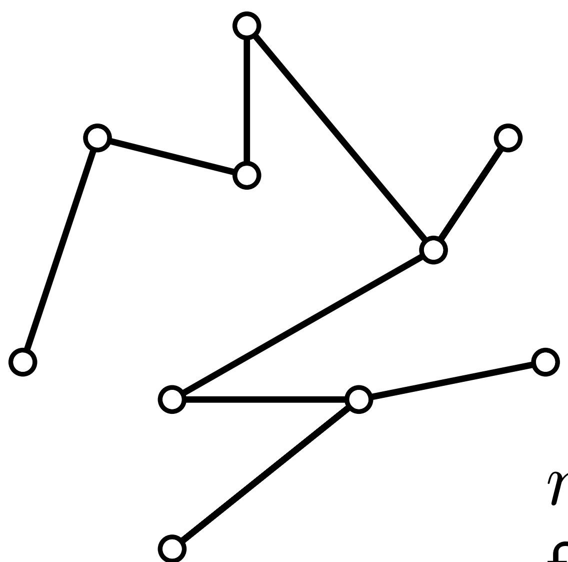
[Avis, Fukuda 1996]

Connectedness	**Diameter**	Complexity

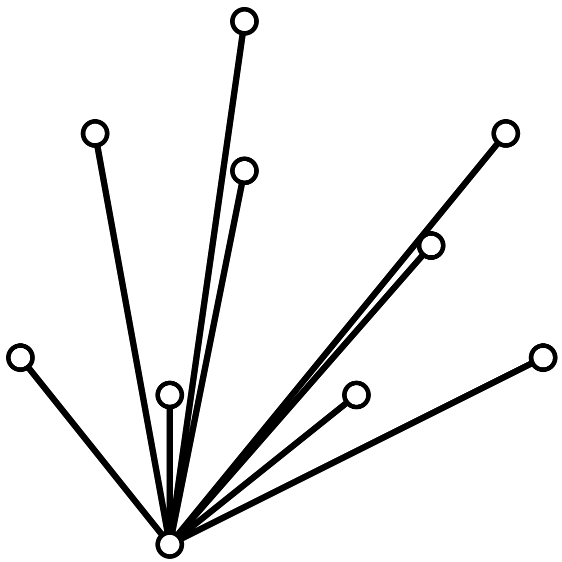
Increase the number of edges incident to v in every flip.



[Avis, Fukuda 1996]



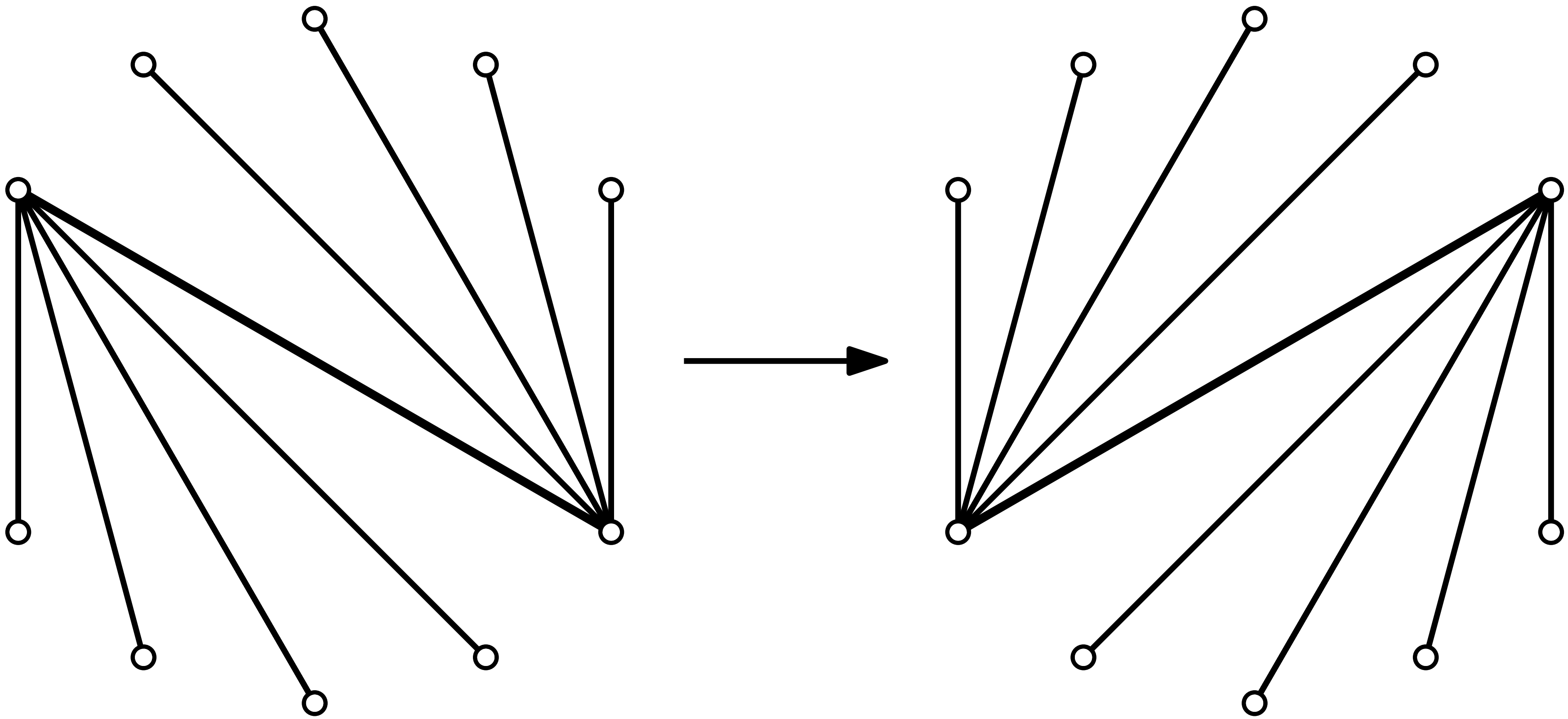
Increase the number of edges incident to v in every flip.



$n-2$ flips to fan for each tree (and back).

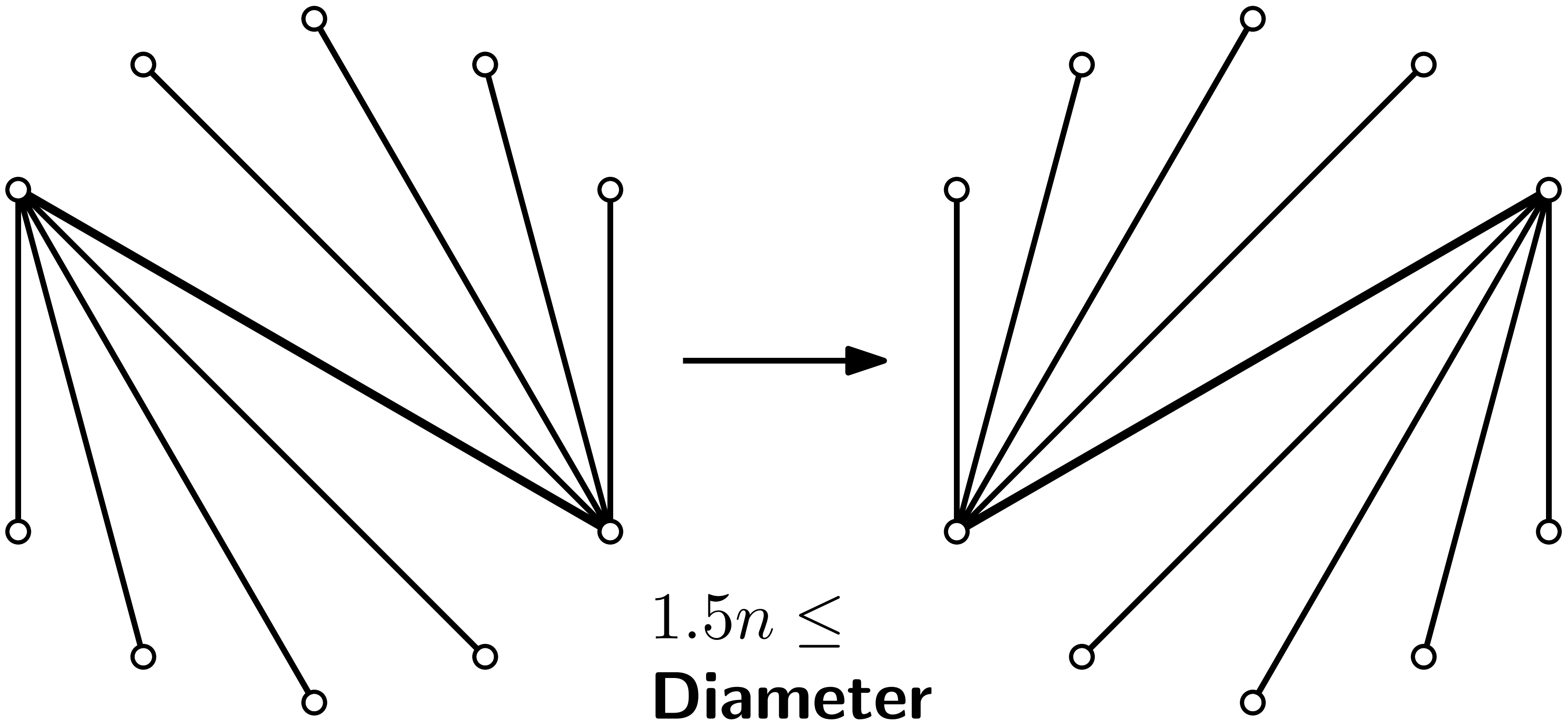
$\Rightarrow$ **Diameter** $\leq 2n-4$

[Avis, Fukuda 1996]



[Hernando, Hurtado, Márquez, Mora, and Noy, 1999]

Connectedness	Diameter	Complexity



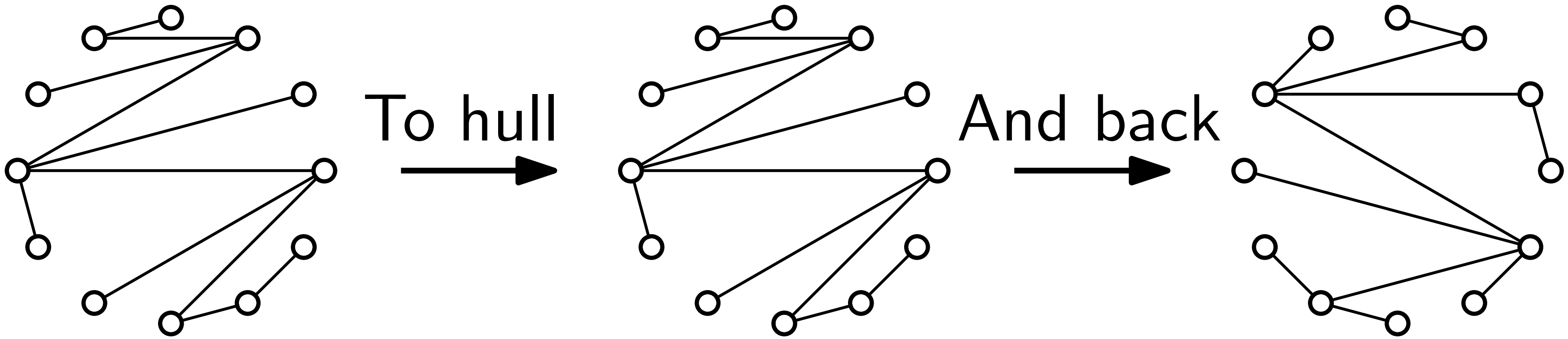
$1.5n \leq$
Diameter

[Hernando, Hurtado, Márquez, Mora, and Noy, 1999]

Attention: From now on: Point set assumed to be in **Convex Position**

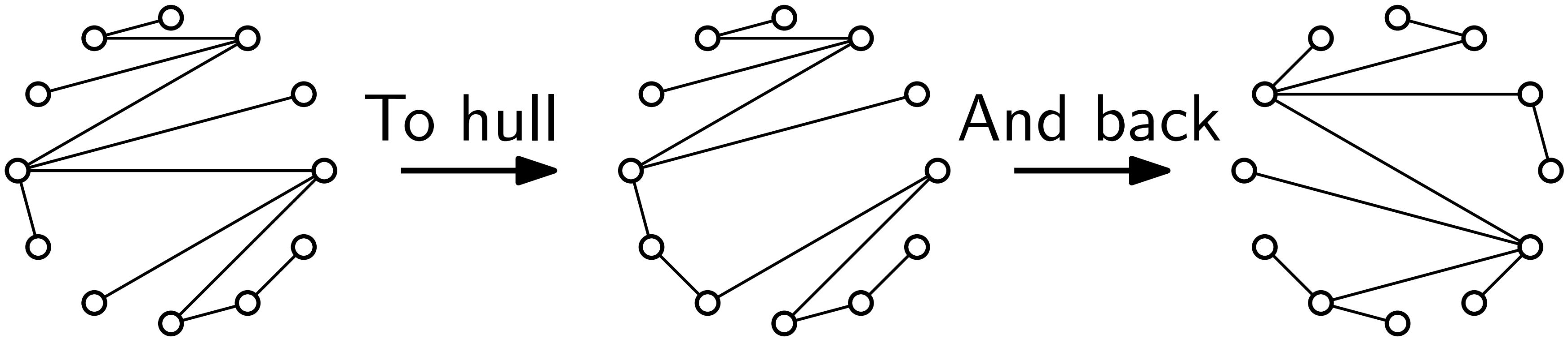
Connectedness	Diameter	Complexity

Convex sets $\Rightarrow$ new options for flip sequences



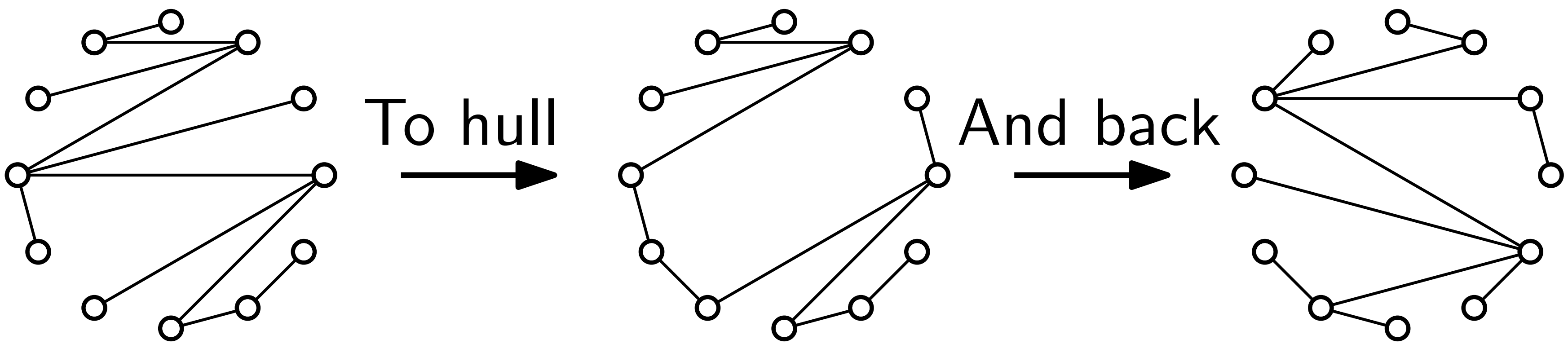
Connectedness	**Diameter**	Complexity

Convex sets $\Rightarrow$ new options for flip sequences



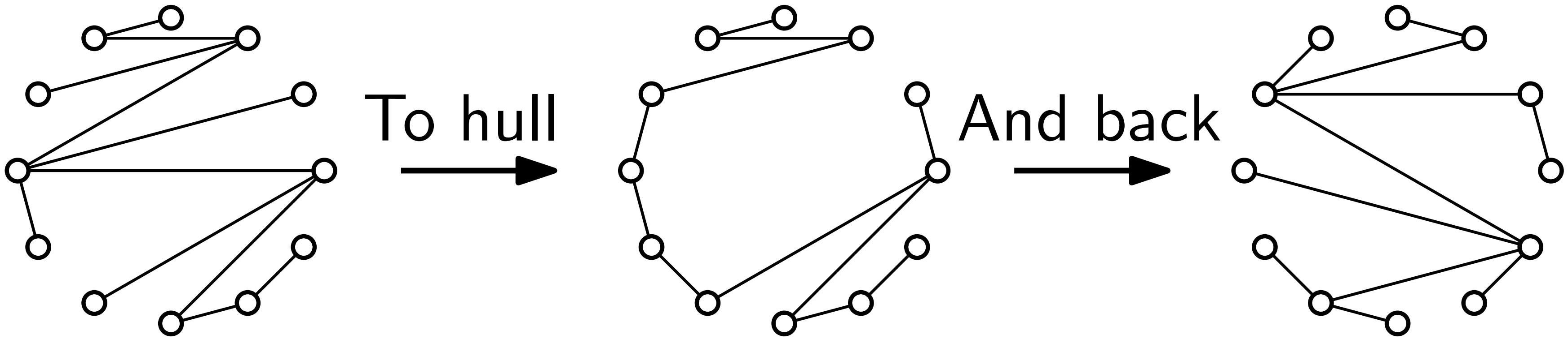
Connectedness	Diameter	Complexity

Convex sets $\Rightarrow$ new options for flip sequences



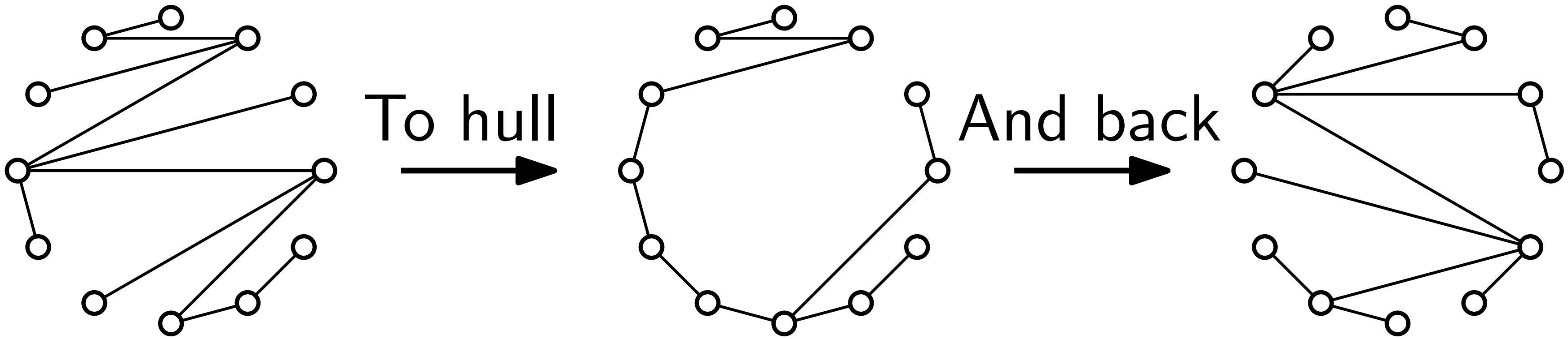
Connectedness	Diameter	Complexity

Convex sets $\Rightarrow$ new options for flip sequences



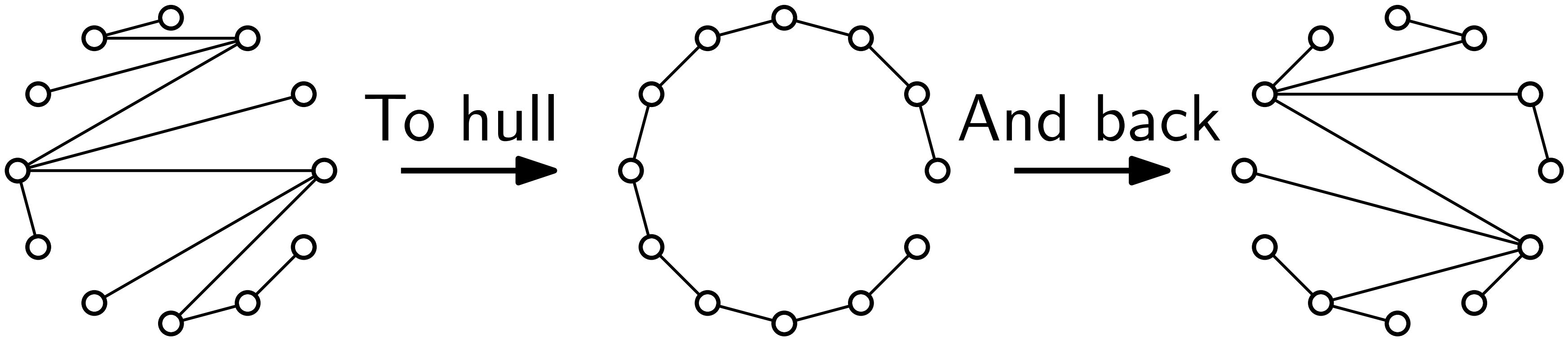
Connectedness	Diameter	Complexity

Convex sets $\Rightarrow$ new options for flip sequences



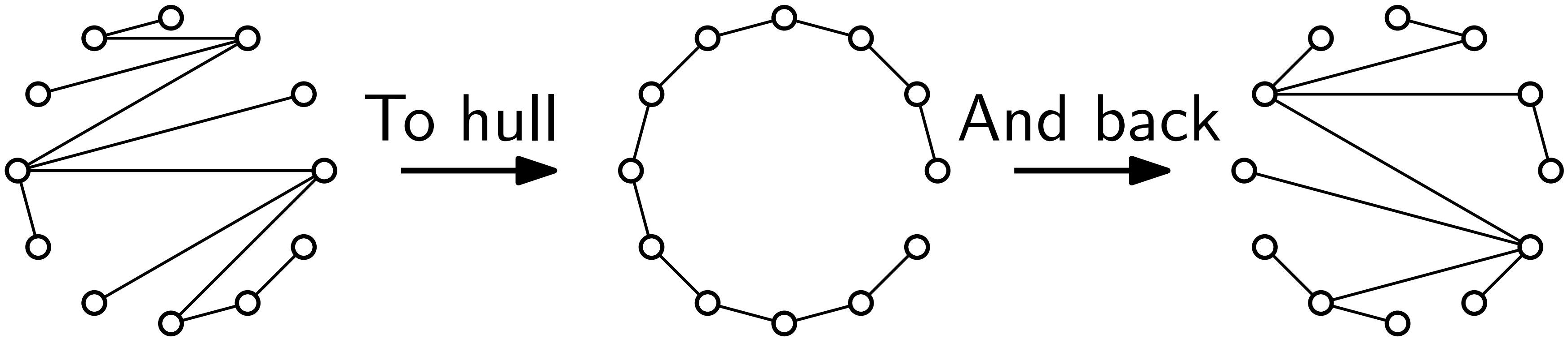
Connectedness	Diameter	Complexity

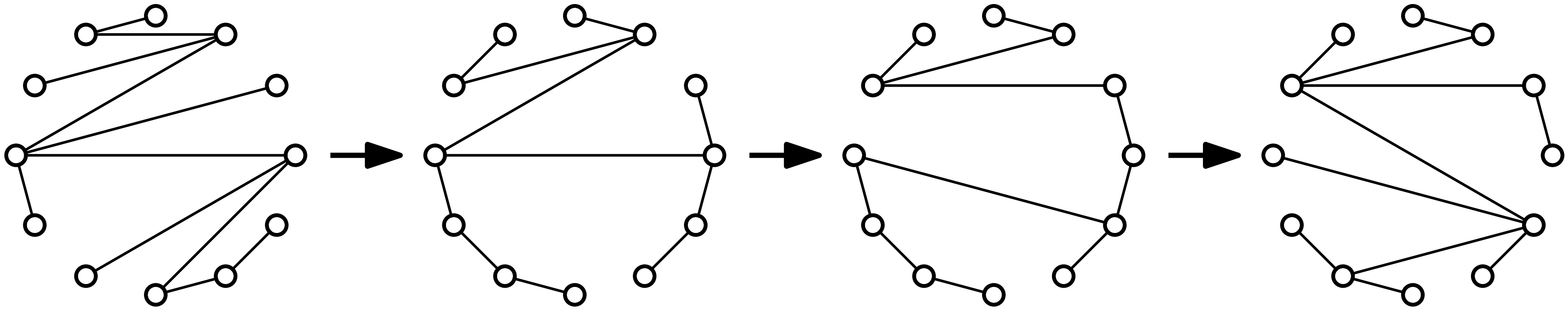
Convex sets $\Rightarrow$ new options for flip sequences



Connectedness	**Diameter**	Complexity

Convex sets $\Rightarrow$ new options for flip sequences

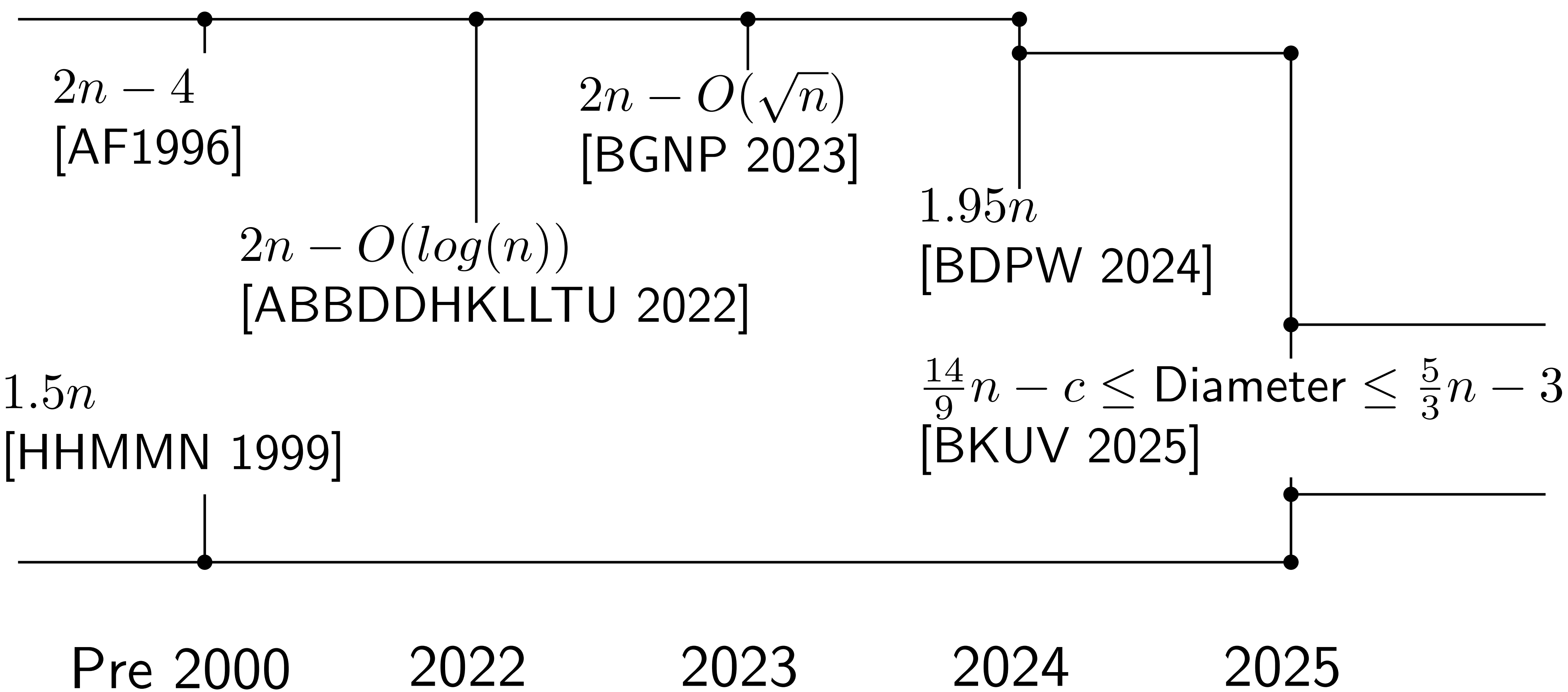




Make some direct flips inbetween

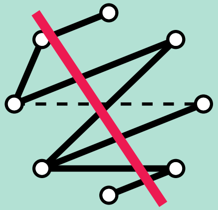
Connectedness | **Diameter** | **Complexity**

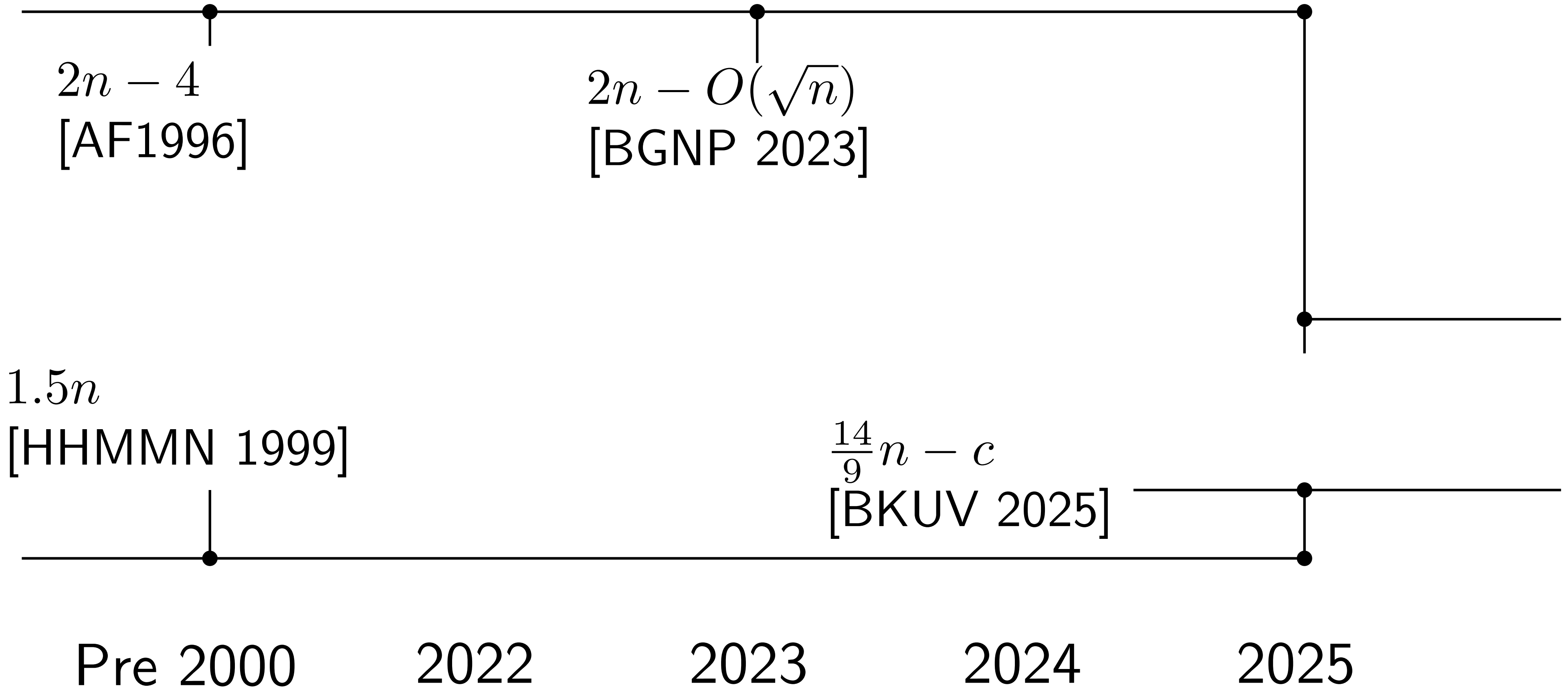
Progression of maximal number of flips.
Who can make the most direct flips?



Connectedness	**Diameter**	Complexity

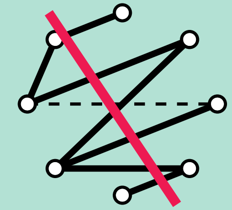
Progression of maximal number of **compatible** flips. (no crossing flips)





Connectedness	**Diameter**	Complexity

Progression of maximal number of **compatible** flips. (no crossing flips)



$2n-4$
[AF1996]

$2n-O(\sqrt{n})$
[BGNP 2023]

$1.5n$
[HHMMN 1999]

$\frac{14}{9}n-c$
[BKUV 2025]

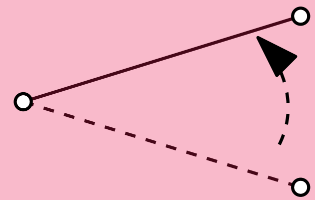
$\frac{5}{3}n-\mathbf{2}$ New Result

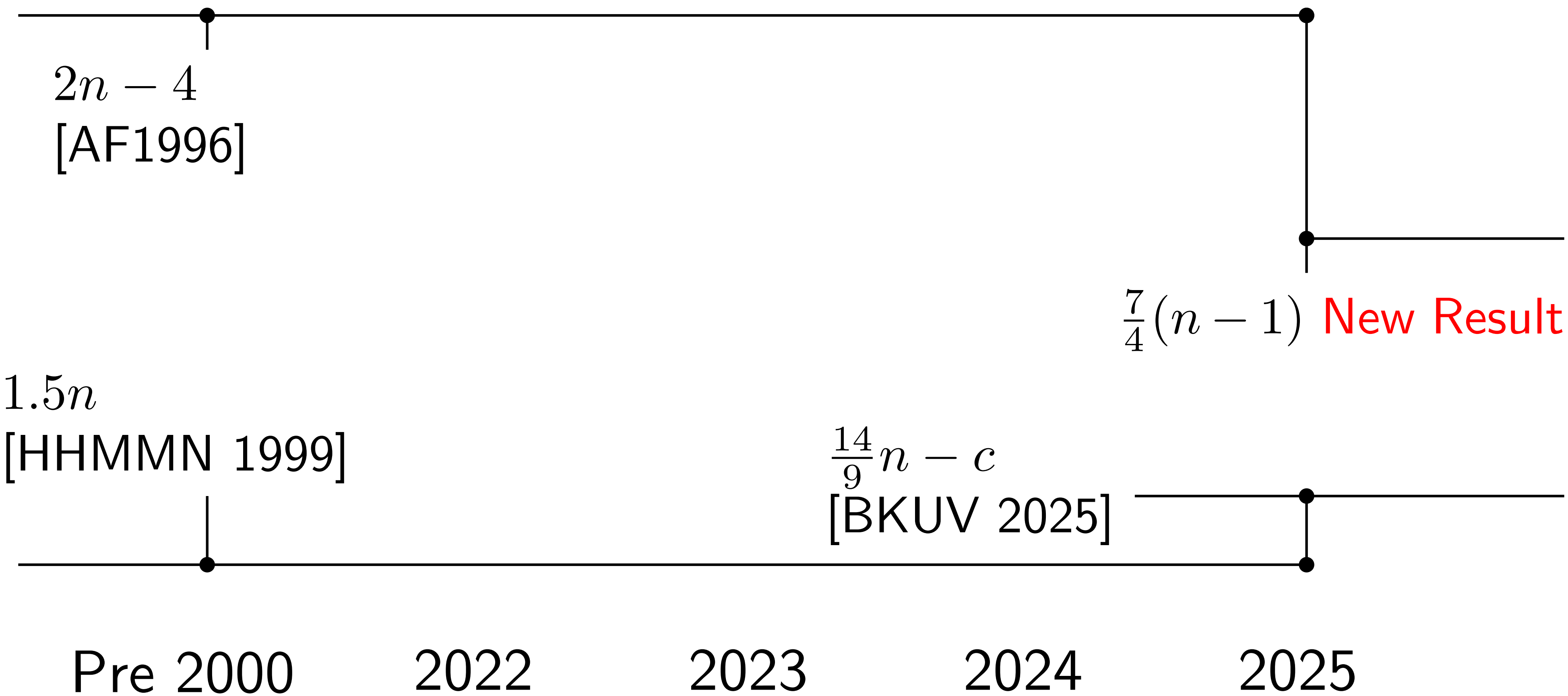
Pre 2000 2022 2023 2024 2025

Connectedness	**Diagram**	Complexity

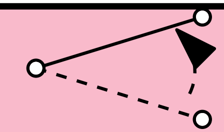
Diameter

Progression of maximal number of **rotations**. (shared vertices)

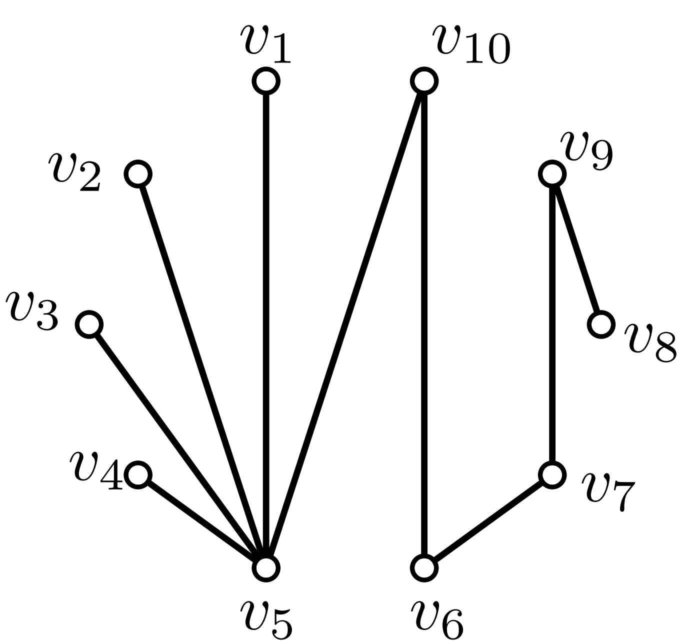


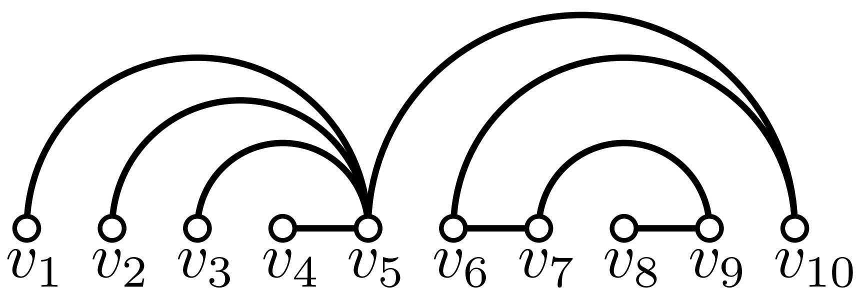


Transforming trees in $\frac{7}{4}(n-1)$ rotations.

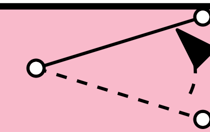


Step 1 Unfold trees onto a horizontal spine.

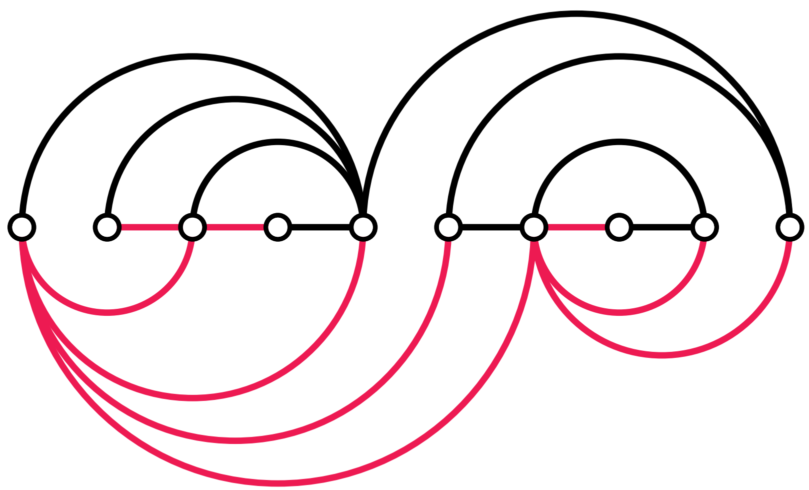




Connectedness	Diameter	Complexity

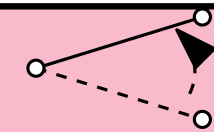
Transforming trees in $\frac{7}{4}(n-1)$ rotations. 

Step 2: Repeat for target tree and draw it on the bottom.

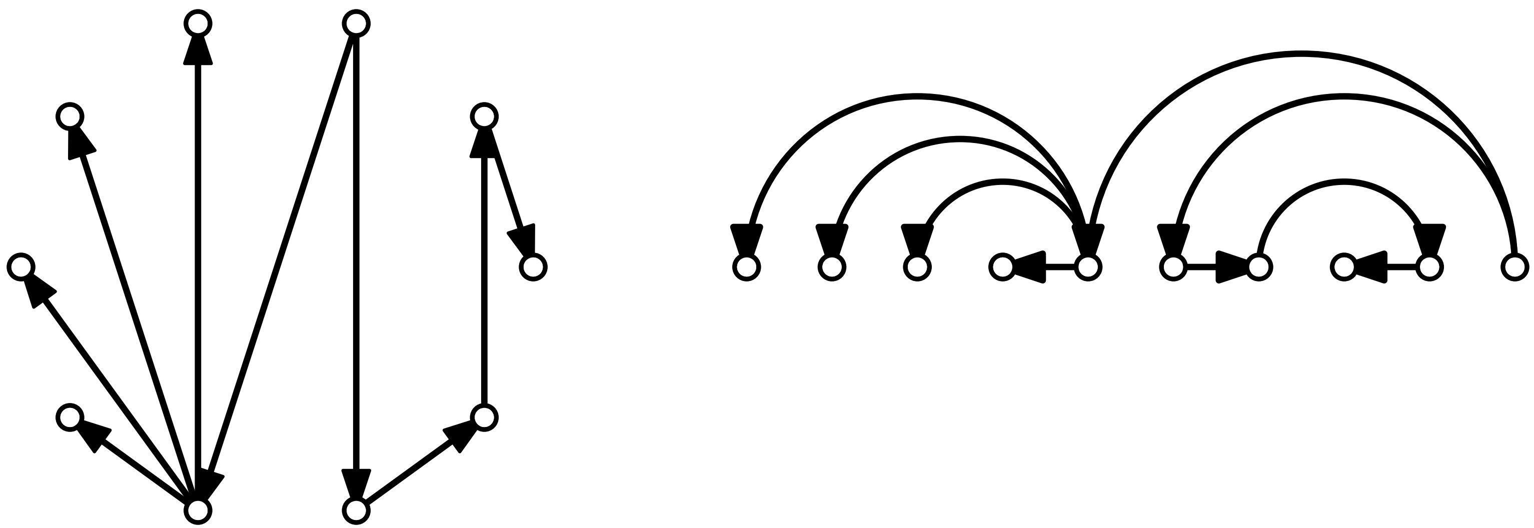


Connectedness	Diametre	Complexity

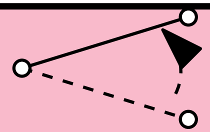
Transforming trees in $\frac{7}{4}(n-1)$ rotations.



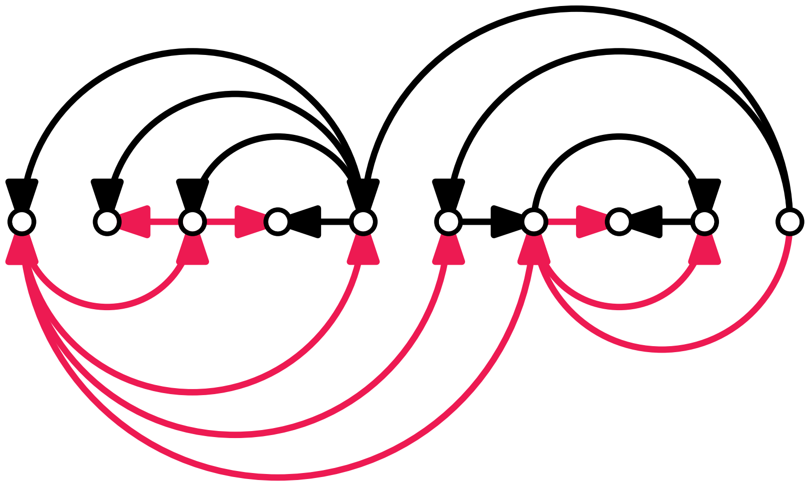
Step 3 Pick a root vertex and orient all edges away from it.



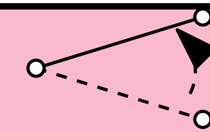
Connectedness	**Diameter**	Complexity

Transforming trees in $\frac{7}{4}(n-1)$ rotations. 

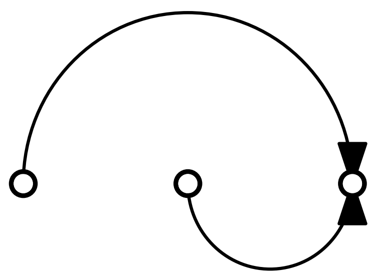
Step 4: Repeat for both trees.



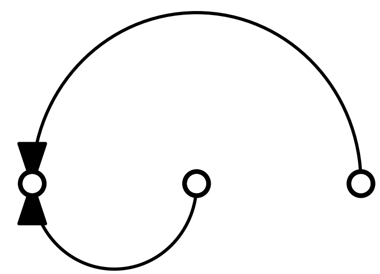
Transforming trees in $\frac{7}{4}(n-1)$ rotations.



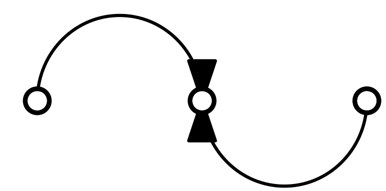
How can pairs of edges look like?



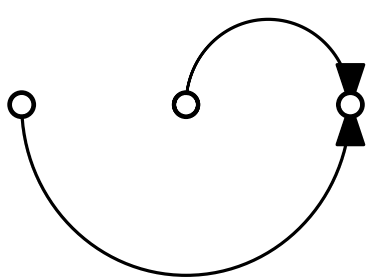
Right-attached above



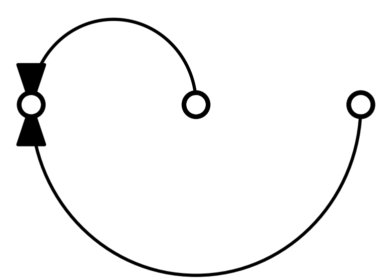
Left-attached above



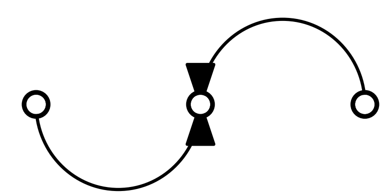
Dive



Right-attached below

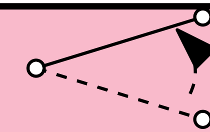


Left-attached below

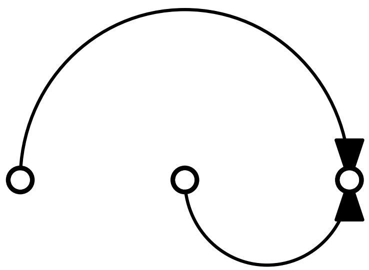


Jump

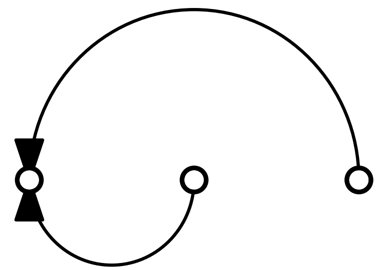
Transforming trees in $\frac{7}{4}(n-1)$ rotations.



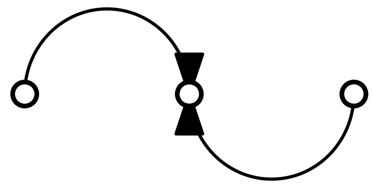
How can pairs of edges look like?



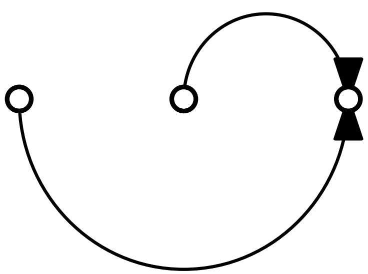
Right-attached above



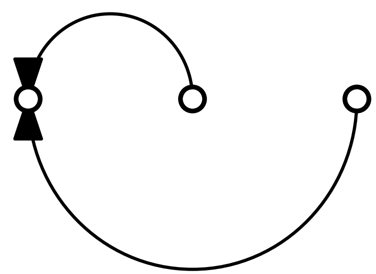
Left-attached above



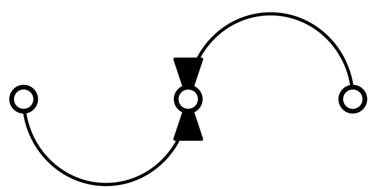
Dive



Right-attached below



Left-attached below



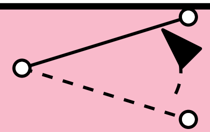
Jump

Right

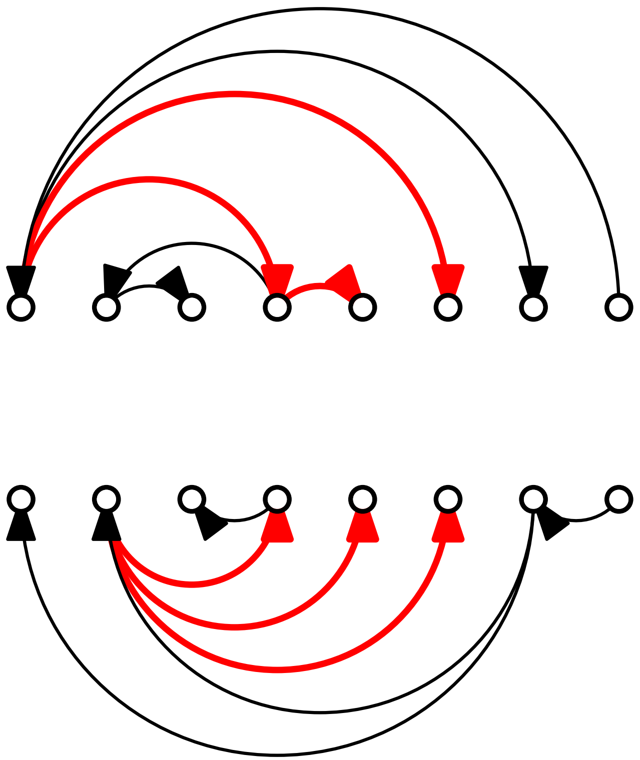
Left

Connectedness	Diameter	Complexity

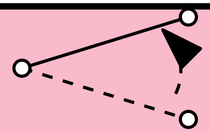
Transforming trees in $\frac{7}{4}(n-1)$ rotations.



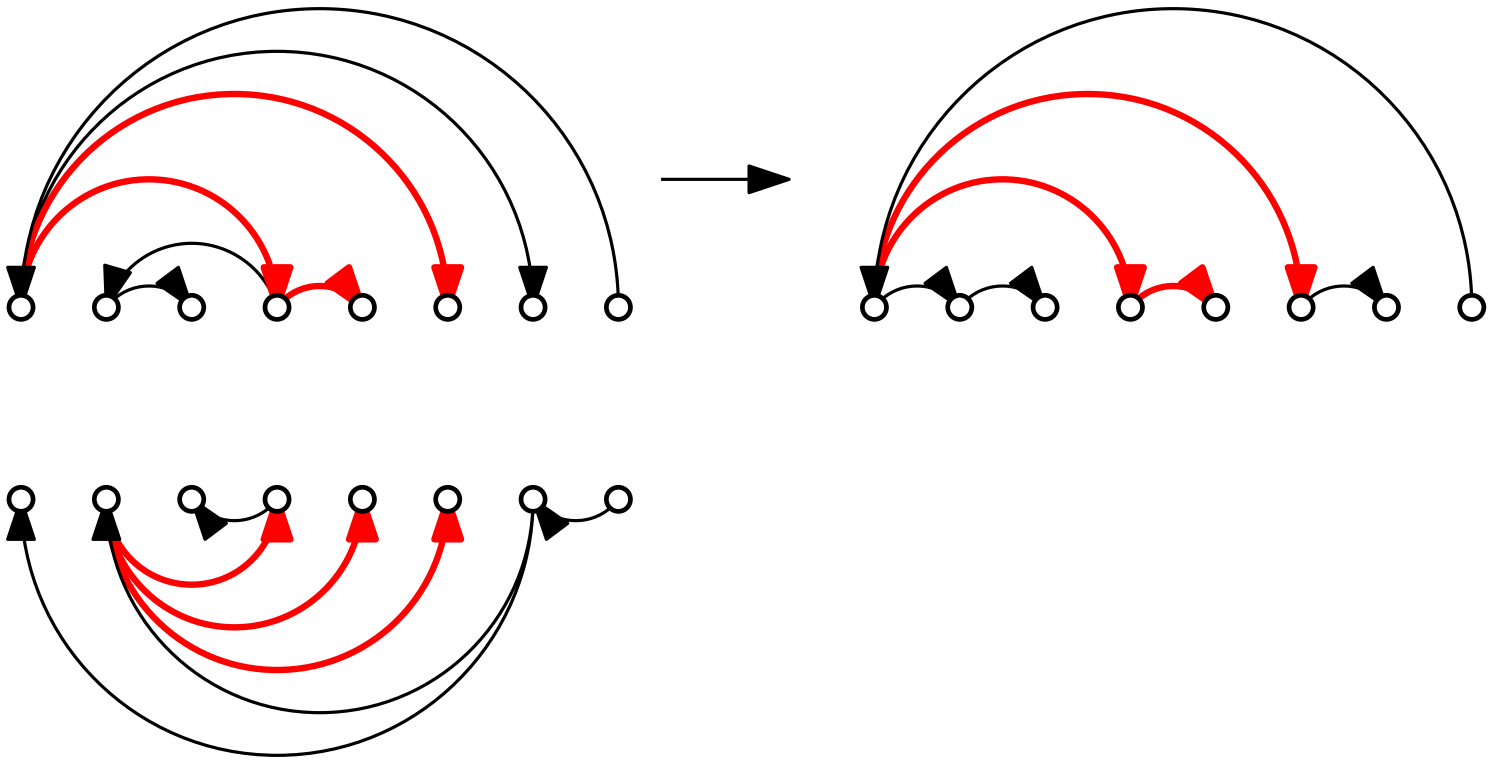
Step 5 (Easy): We can go from one tree to another in $2(n-1)-$ #Right (resp. #Left) rotations.



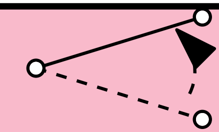
Transforming trees in $\frac{7}{4}(n-1)$ rotations.



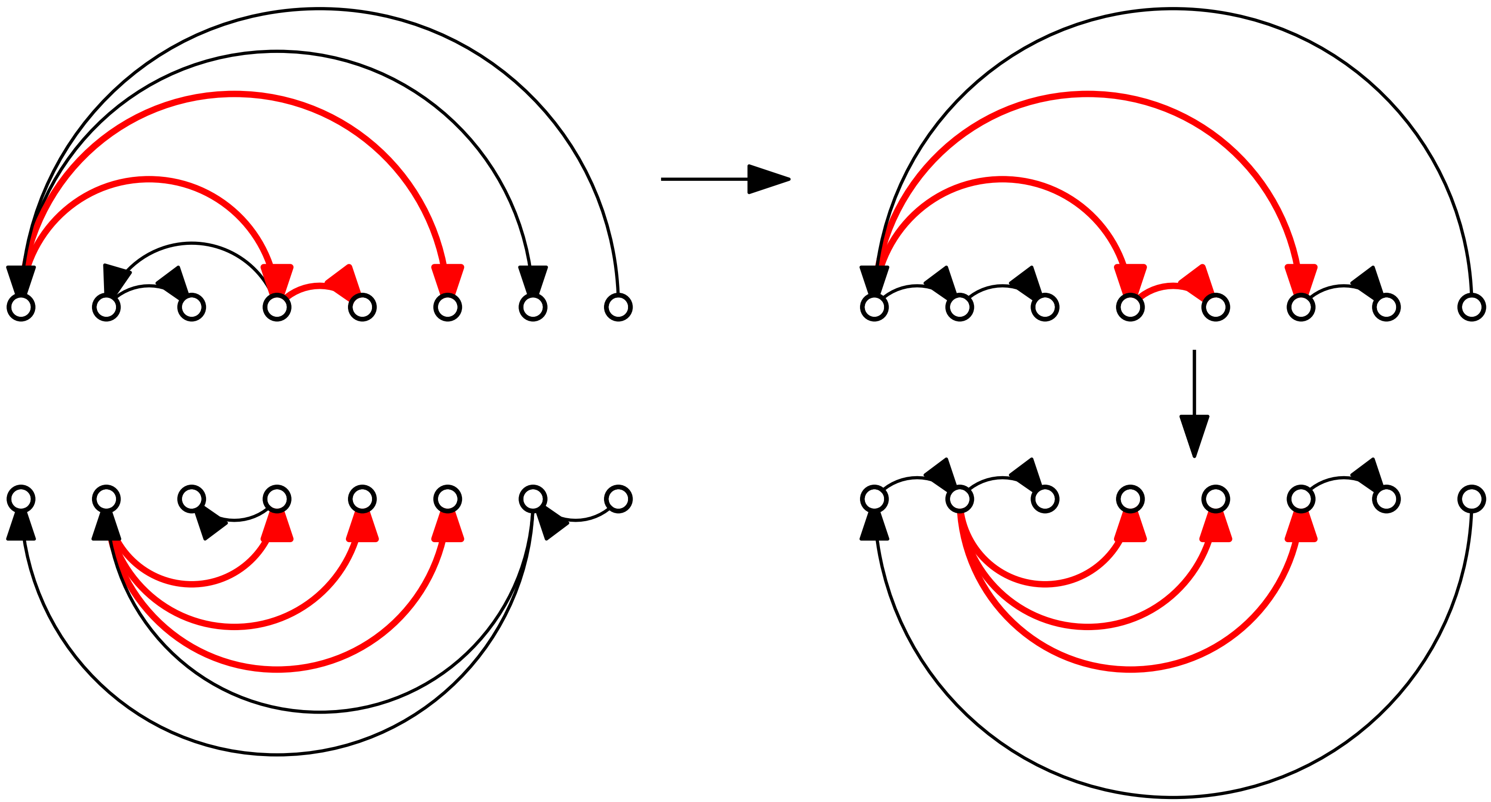
Step 5 (Easy): We can go from one tree to another in $2(n-1)$ − #Right (resp. #Left) rotations.
Step 5.1: Rotate not right-attached edges to convex hull.



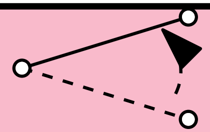
Transforming trees in $\frac{7}{4}(n-1)$ rotations.



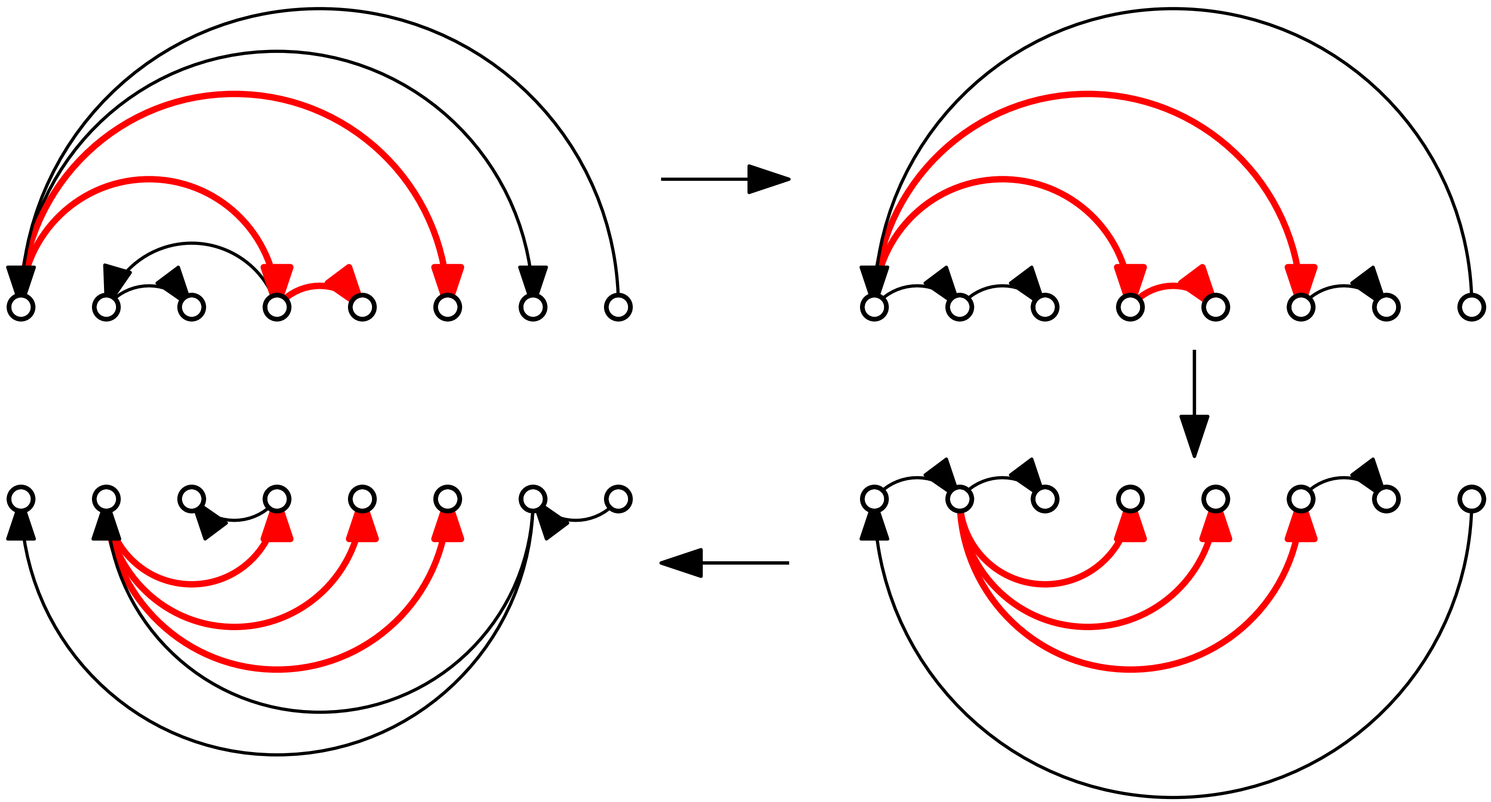
Step 5 (Easy): We can go from one tree to another in $2(n-1) - \#$Right (resp. $\#$Left) rotations.
Step 5.2: Rotate right attached edges to target location.



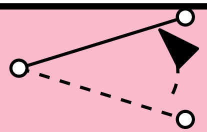
Transforming trees in $\frac{7}{4}(n-1)$ rotations.



Step 5 (Easy): We can go from one tree to another in $2(n-1) - \#\text{Right}$ (resp. $\#\text{Left}$) rotations.
Step 5.3: Rotate remaining edges to target location.

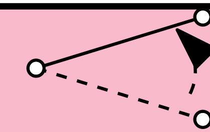


Transforming trees in $\frac{7}{4}(n-1)$ rotations.



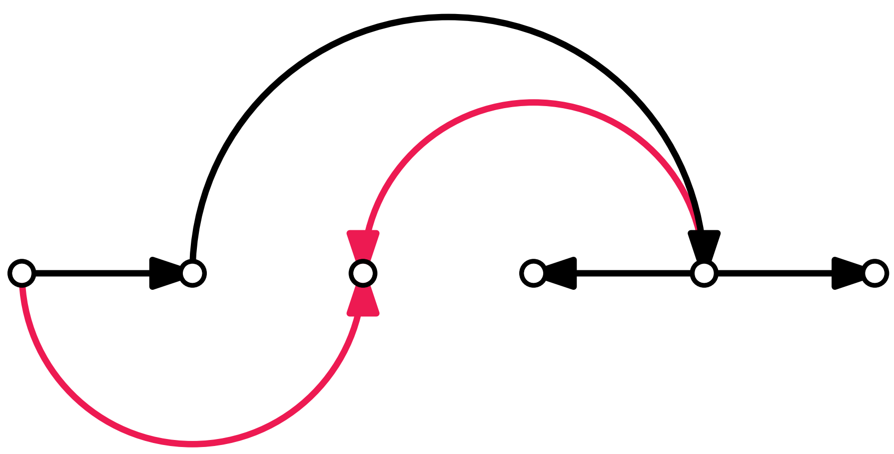
Step 6 (Not so easy): We can go from one tree to another in $2(n-1) - $ #Jump (resp. #Dive) rotations.

Transforming trees in $\frac{7}{4}(n-1)$ rotations.

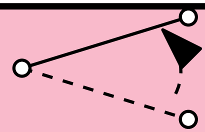


Step 6 (Not so easy): We can go from one tree to another in $2(n-1) - \#$Jump (resp. $\#$Dive) rotations.

Why not so easy?

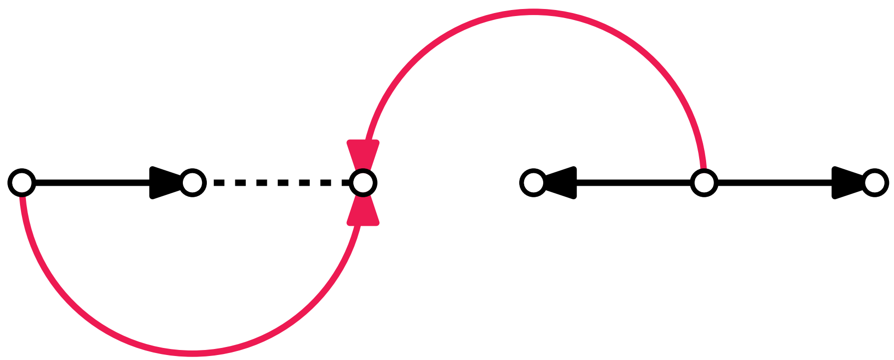


Transforming trees in $\frac{7}{4}(n-1)$ rotations.

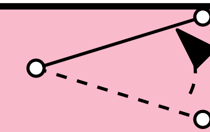


Step 6 (Not so easy): We can go from one tree to another in $2(n-1) - \#$Jump (resp. $\#$Dive) rotations.

Why not so easy?

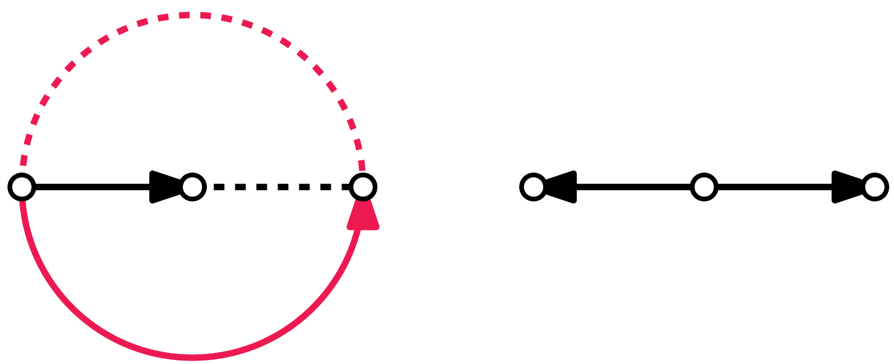


Transforming trees in $\frac{7}{4}(n-1)$ rotations.

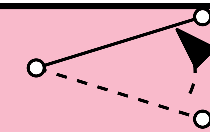


Step 6 (Not so easy): We can go from one tree to another in $2(n-1) - \#$Jump (resp. $\#$Dive) rotations.

Why not so easy?



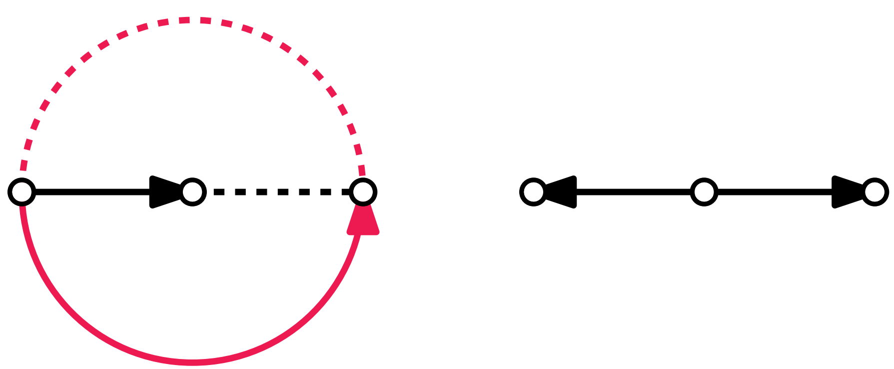
Transforming trees in $\frac{7}{4}(n-1)$ rotations.



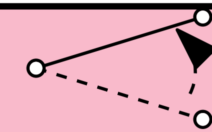
Step 6 (Not so easy): We can go from one tree to another in $2(n-1) - \#$Jump (resp. $\#$Dive) rotations.

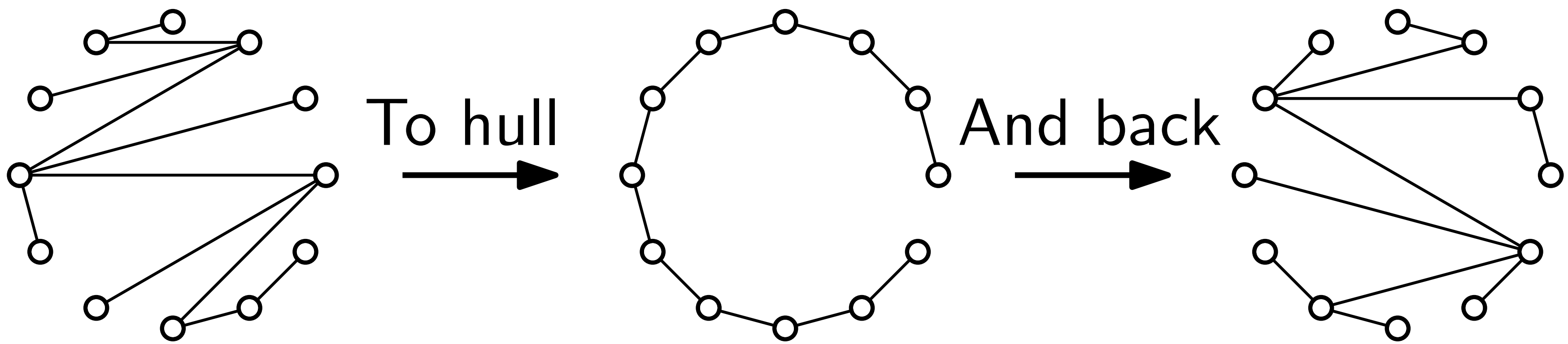
Why not so easy?

Closes Cycle!

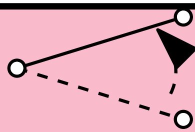


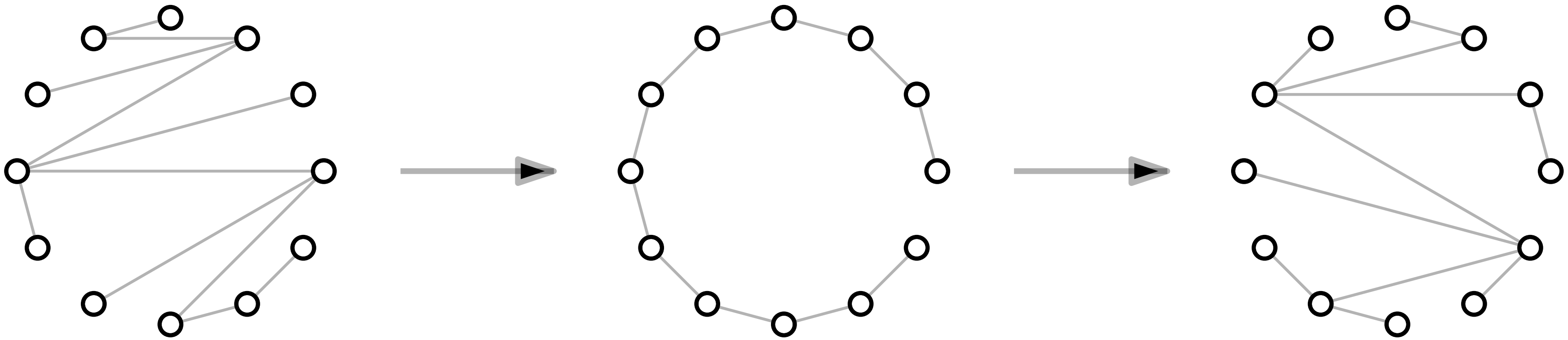
Transforming trees in $\frac{7}{4}(n-1)$ rotations.





Transforming trees in $\frac{7}{4}(n-1)$ rotations.

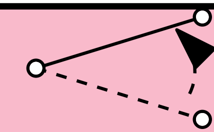


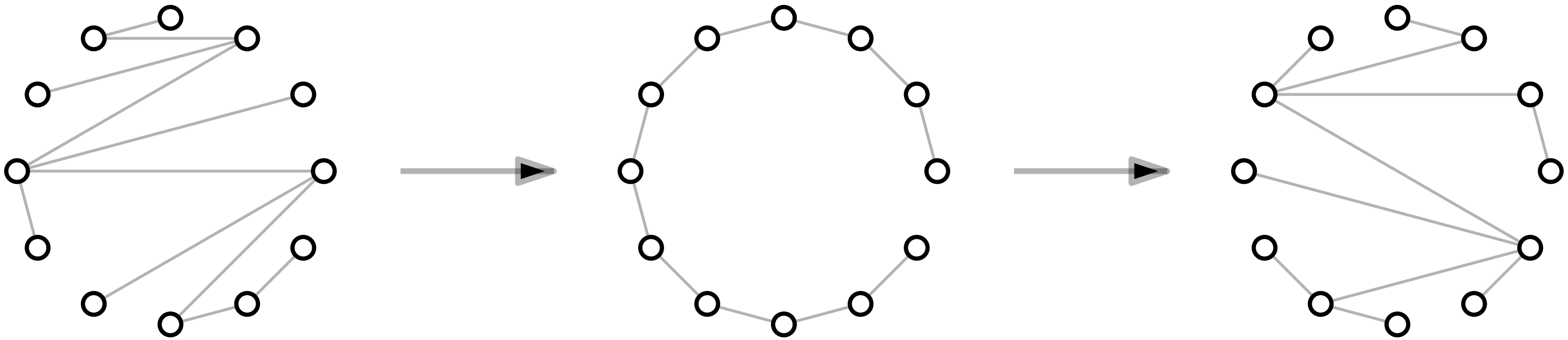


Did not work

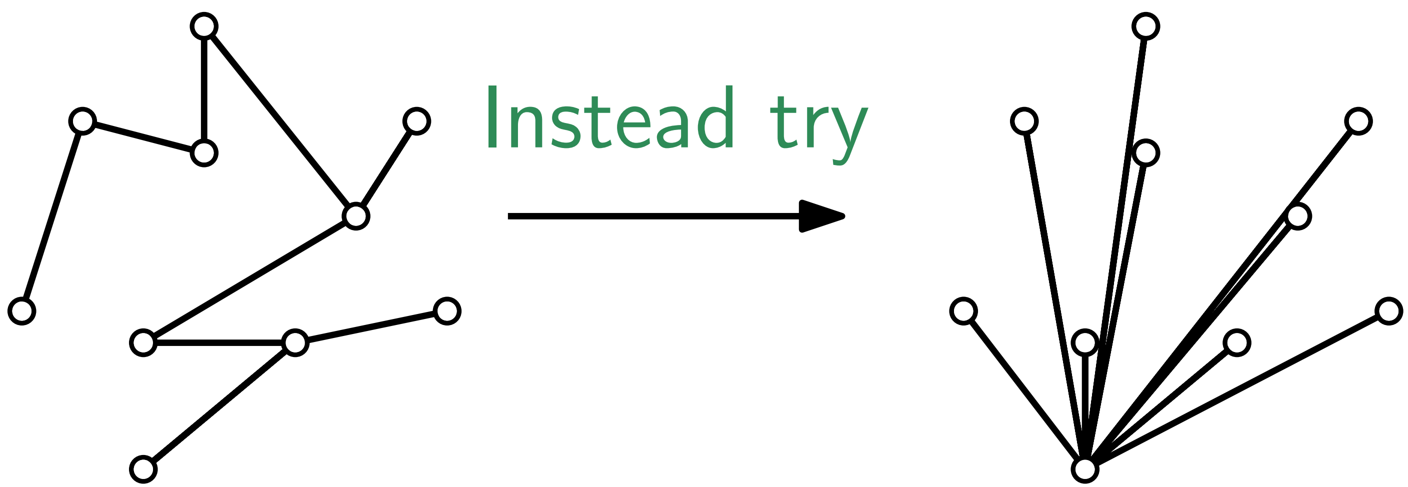
Connectedness	**Diameter**	Complexity

Transforming trees in $\frac{7}{4}(n-1)$ rotations.





Did not work



Transforming trees in $\frac{7}{4}(n-1)$ rotations.

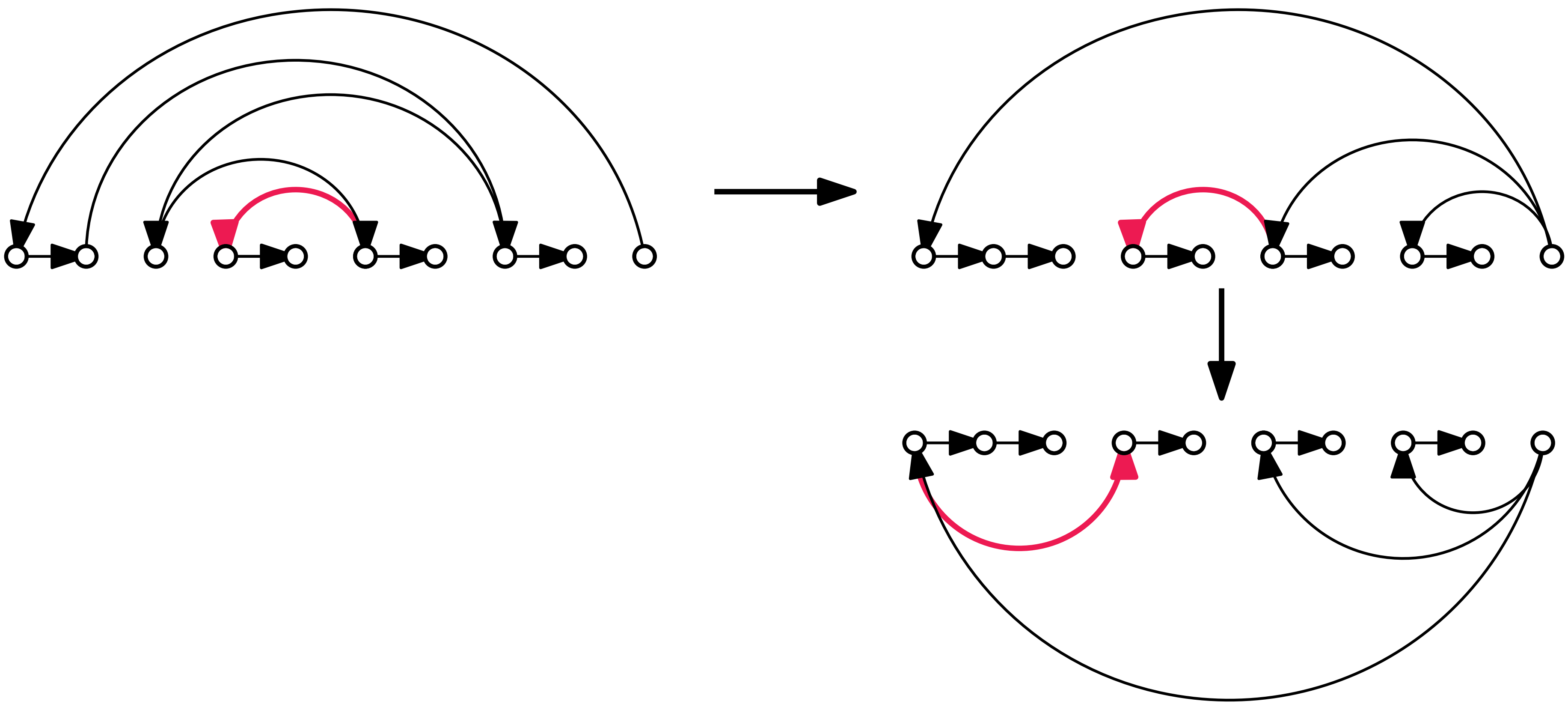
Step 6 (Not so easy): We can go from one tree to another in $2(n-1) - $ #Jump (resp. #Dive) rotations.

Step 6.1: Rotate into mixture of convex hull and star.

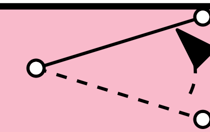
Transforming trees in $\frac{7}{4}(n-1)$ rotations.

Step 6 (Not so easy): We can go from one tree to another in $2(n-1) - \#$Jump (resp. $\#$Dive) rotations

Step 6.2: Rotate Jump edges directly

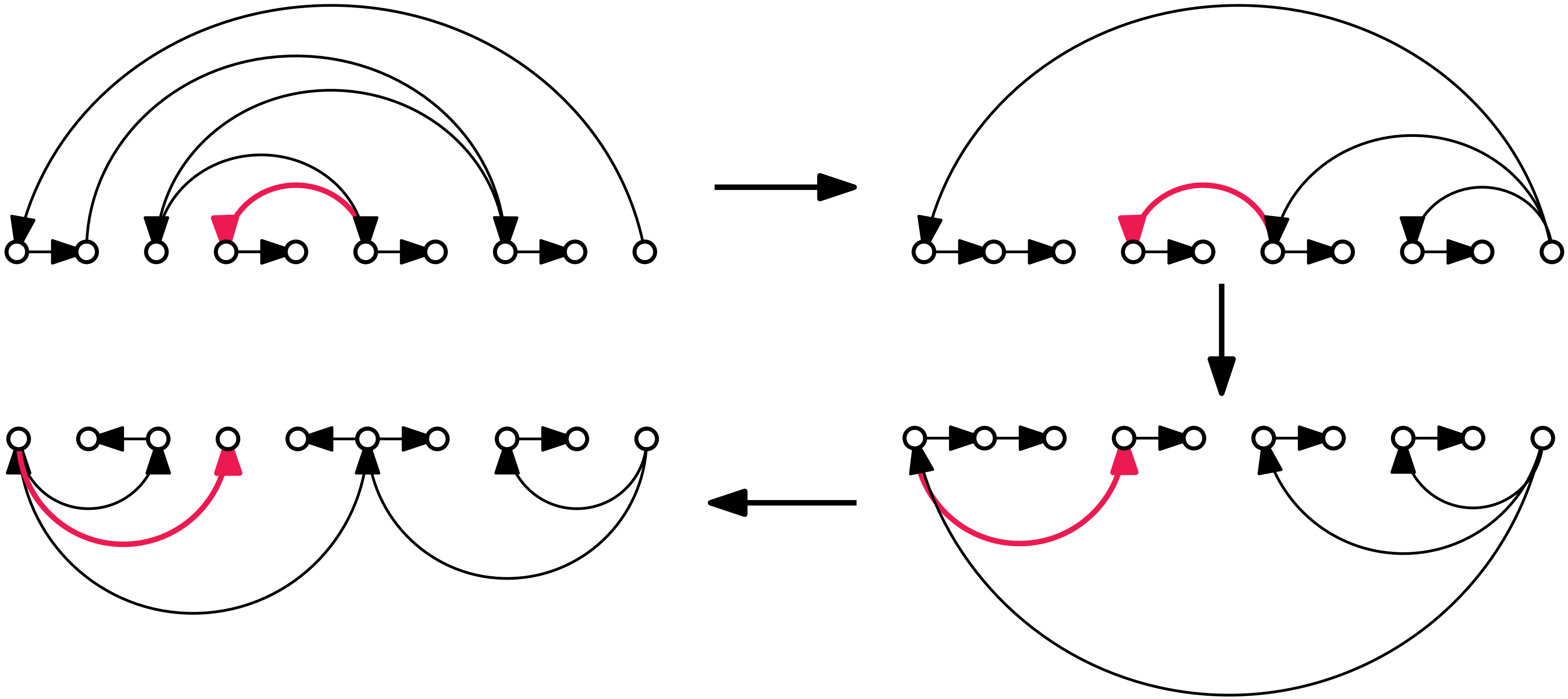


Transforming trees in $\frac{7}{4}(n-1)$ rotations.



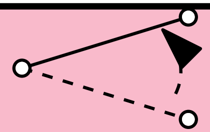
Step 6 (Not so easy): We can go from one tree to another in $2(n-1) - $#Jump (resp. #Dive) rotations

Step 6.3: Rotate to target position



Connectedness	Diameter	Complexity

Transforming trees in $\frac{7}{4}(n-1)$ rotations.



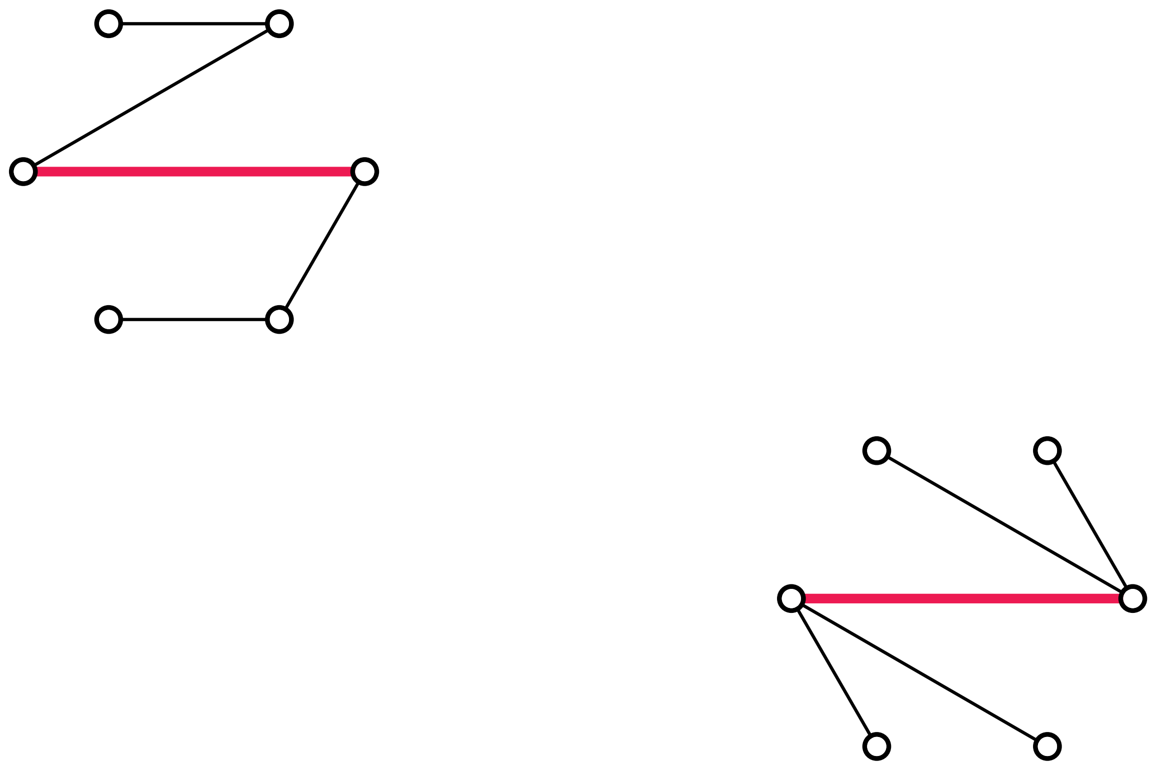
Step 5: We can go from one tree to another in $2(n-1) - \#$Right (resp. $\#$Left) rotations.

Step 6: We can go from one tree to another in $2(n-1) - \#$Jump (resp. $\#$Dive) rotations.

$$\#\text{Right} + \#\text{Left} + \#\text{Jump} + \#\text{Dive} = n - 1$$

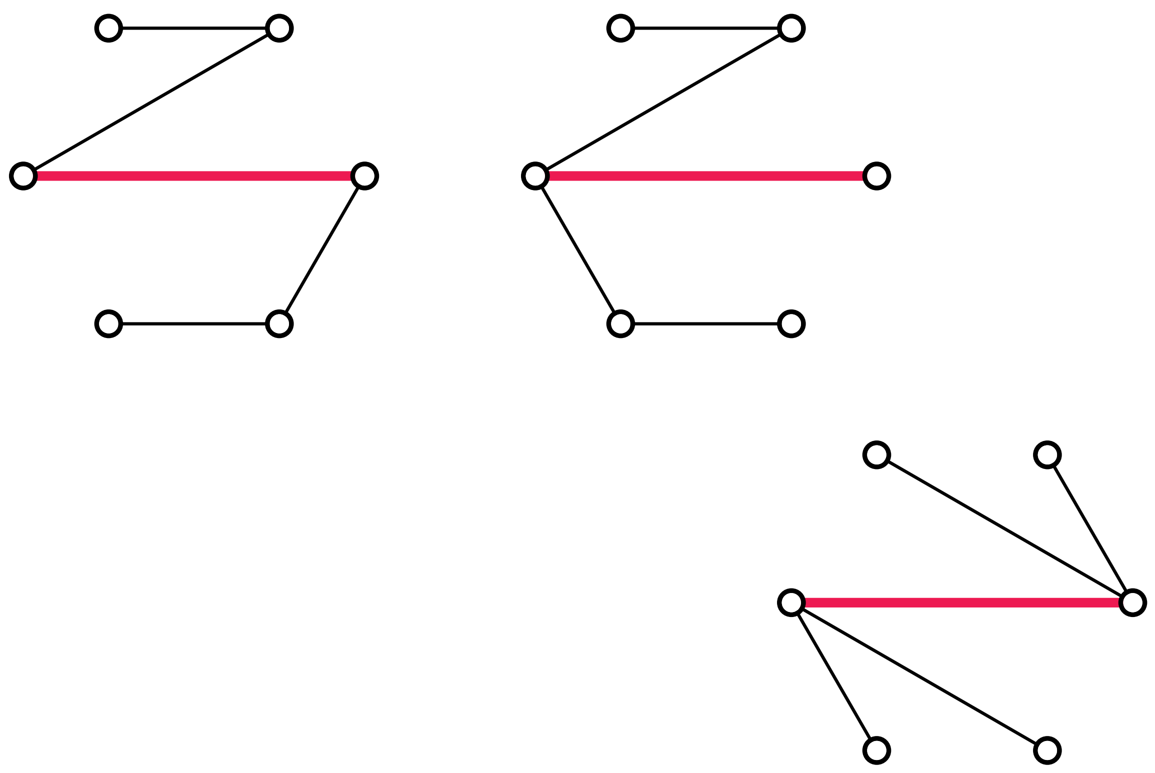
Connectedness	Diameter	Complexity

Happy Edges In the initial structure and target structure



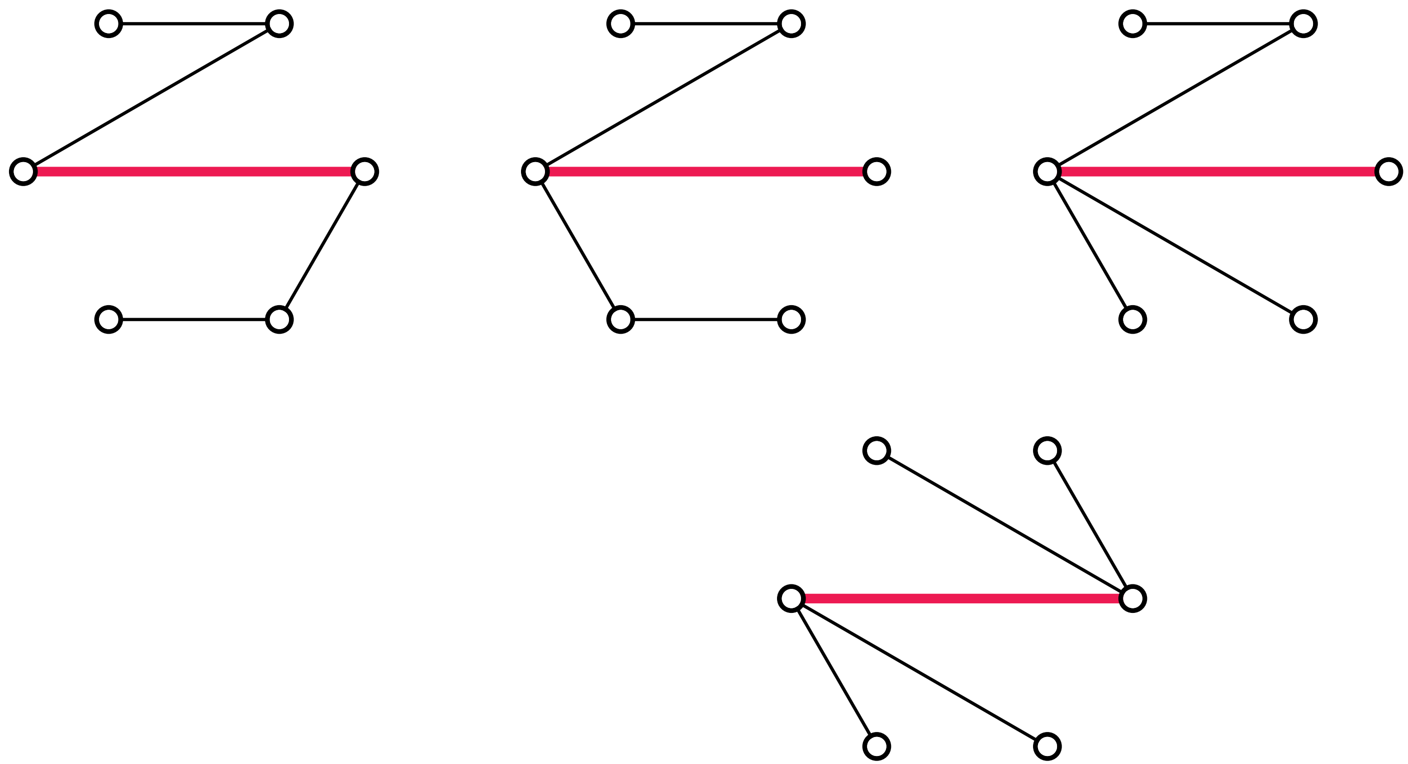
Connectedness	Diameter	Complexity

Happy Edges In the initial structure and target structure



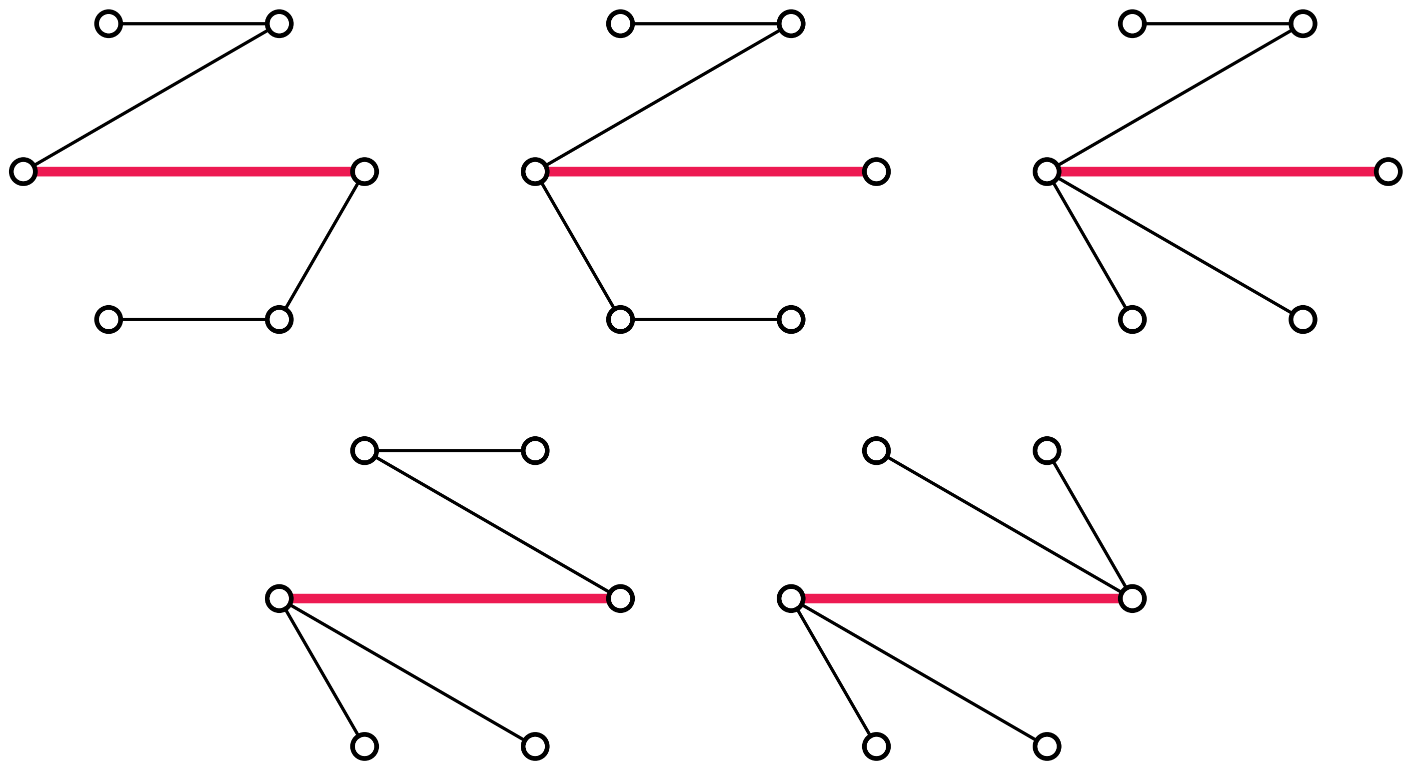
Connectedness	Diameter	Complexity

Happy Edges In the initial structure and target structure



Connectedness	Diametert	**Complexity**

Happy Edges In the initial structure and target structure

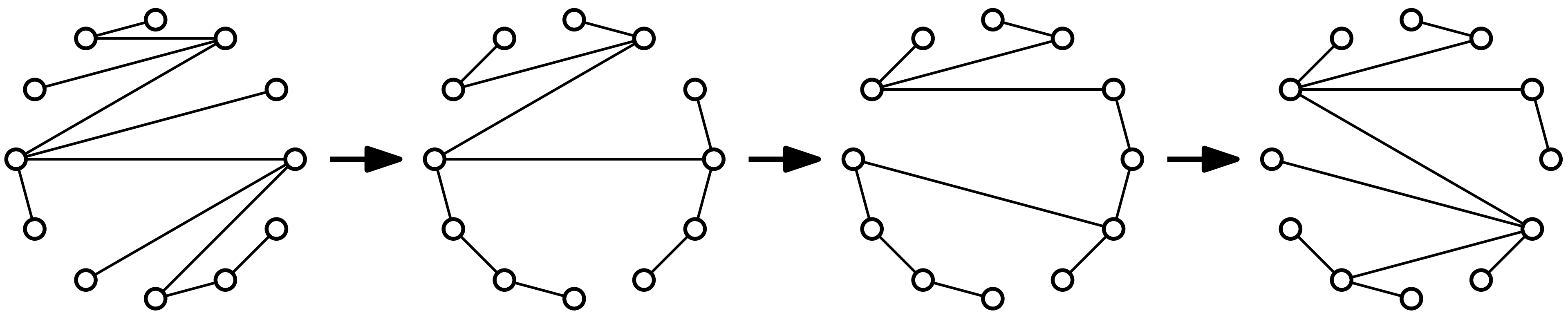


Connectedness	Diameter	Complexity

Conjecture 1 (For any type of flip): Keeping happy edges yields optimal flip sequences.

Conjecture 1 (For any type of flip): Keeping happy edges yields optimal flip sequences.

Conjecture 2 (For any type of flip): Using only intermediate edges from the convex hull yields optimal flip sequences.



Connectedness	Diameter	Complexity

Conjecture 1 (For any type of flip): Keeping happy edges yields optimal flip sequences.

Conjecture 2 (For any type of flip): Using only intermediate edges from the convex hull yields optimal flip sequences.

Conjecture 2 $\Rightarrow$ Conjecture 1 [ABBDDKLLTU 2022]

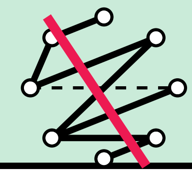
Connectedness	Diameter	Complexity

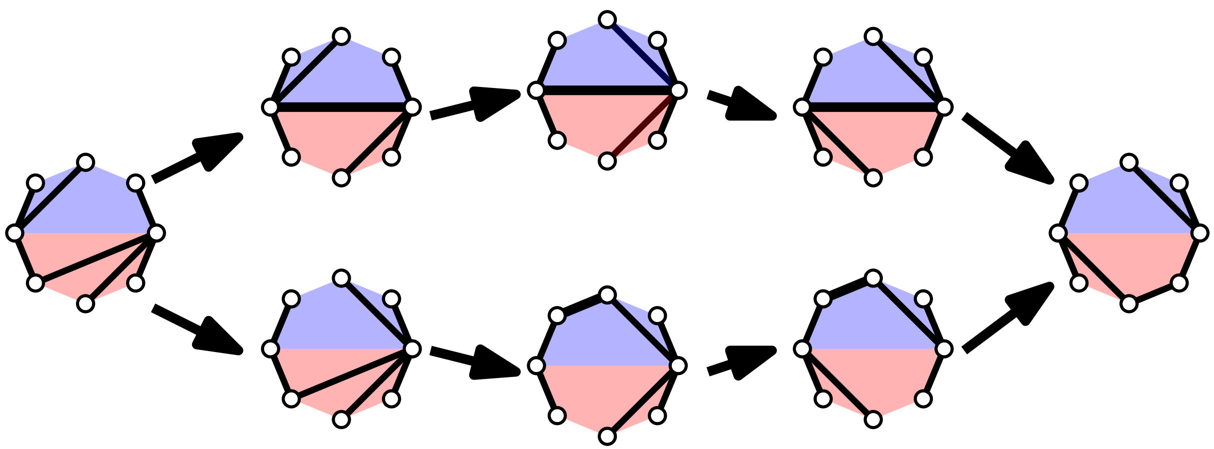
Conjecture 1 (For any type of flip): Keeping happy edges yields optimal flip sequences.

Conjecture 2 (For any type of flip): Using only intermediate edges from the convex hull yields optimal flip sequences.

Conjecture 2 $\Rightarrow$ Conjecture 1 [ABBDDKLLTU 2022]

Conjecture 2 (and consequently Conjecture 1) holds for **compatible** flip sequences. New Result





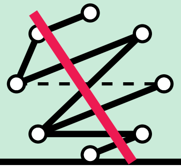
Connectedness	Diameter	Complexity

Conjecture 1 (For any type of flip): Keeping happy edges yields optimal flip sequences.

Conjecture 2 (For any type of flip): Using only intermediate edges from the convex hull yields optimal flip sequences.

Conjecture 2 $\Rightarrow$ Conjecture 1 [ABBDDKLLTU 2022]

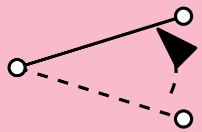
Conjecture 2 (and consequently Conjecture 1) holds for **compatible** flip sequences. New Result



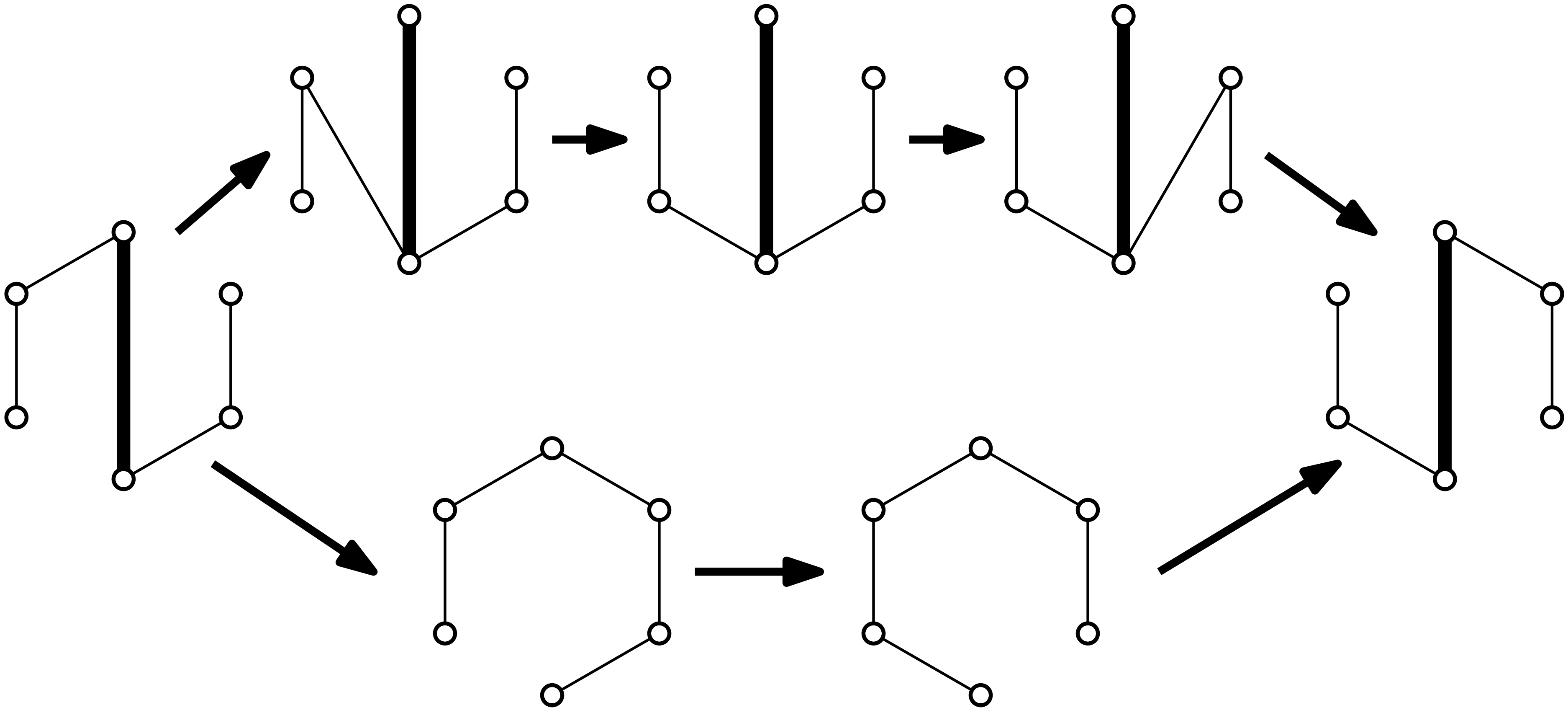
Application: FPT-Algorithm: The flip distance k is FPT when taking k as the parameter. New Result

Connectedness	Diameter	Complexity

The happy edge property does not hold for rotations.

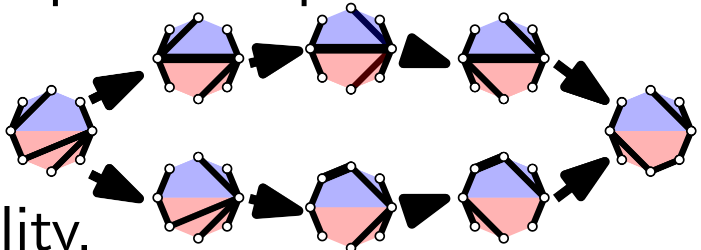


New Result



Recap: Constrained flips

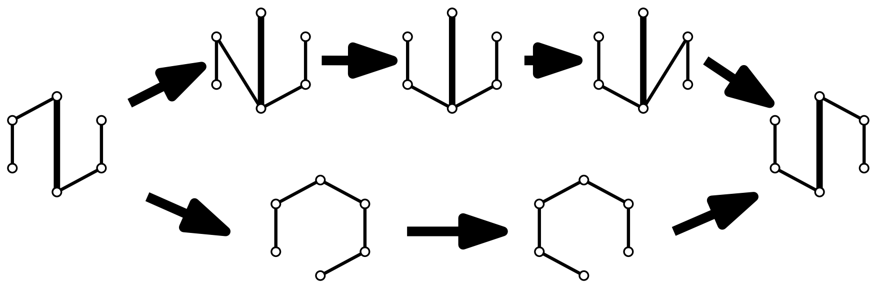
The Happy Edge property holds for compatible flips.



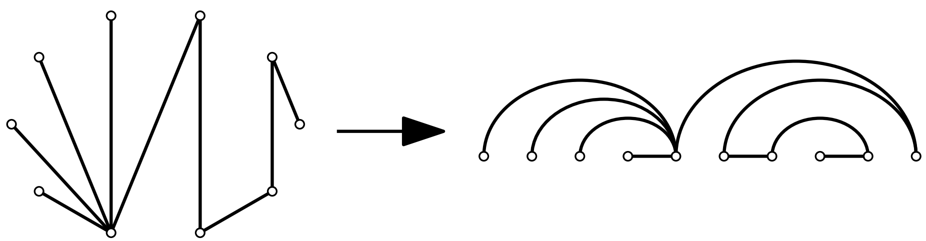
Application: fixed-parameter tractability.

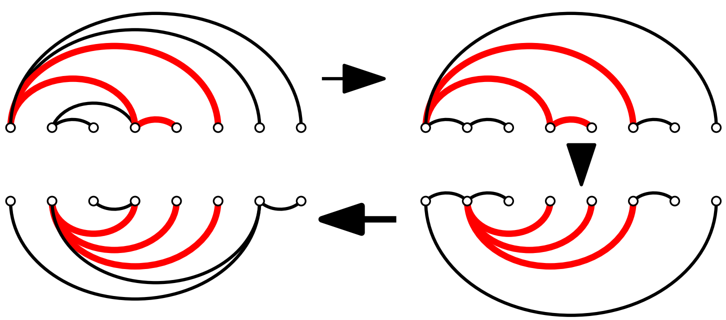
Improved upper bound on length of compatible flip sequences. $\frac{5}{3}n - 2$

The Happy Edge property does not hold for rotations.



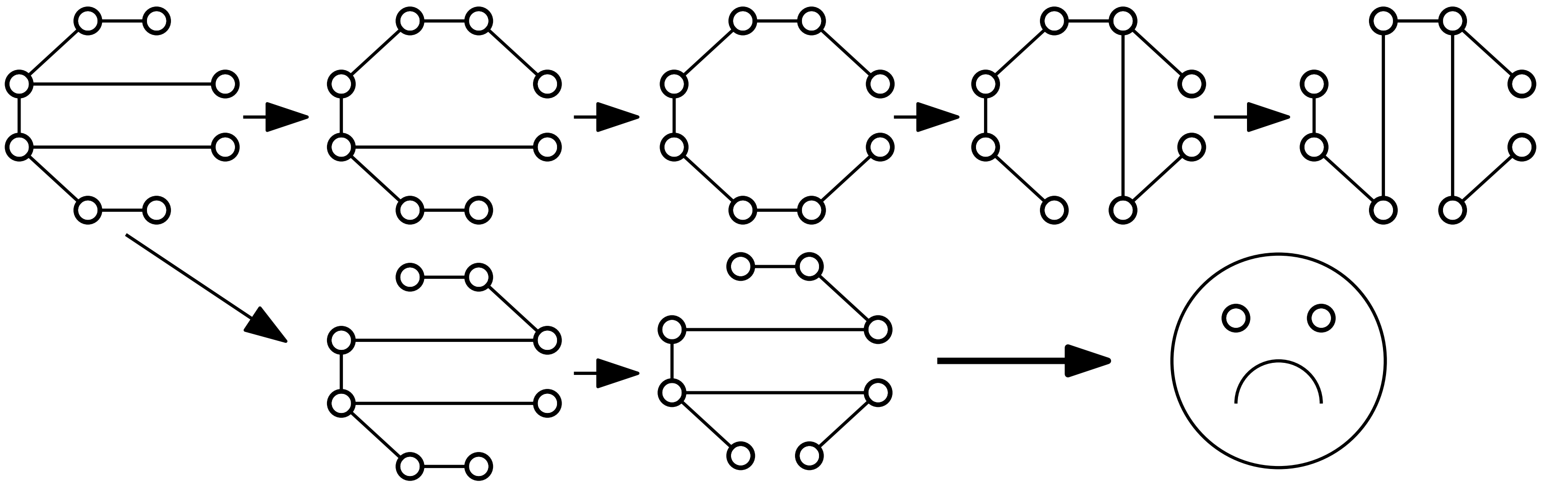
Improved upper bound on the length of rotation sequences. $\frac{7}{4}(n-1)$





Connectedness	Diameter	Complexity

Attention Happy Edge Property $\neq$ Greedy Flips



[ABBDDKLLTU 2022]