Some approximation results for the Maximum Agreement Forest (MAF) problem

Inspired by a problem of genetic transfer detection and phylogeny in biology

Joint work with Estela Maris Rodrigues and
Yoshiko Wakabayashi
University of São Paulo, Brazil

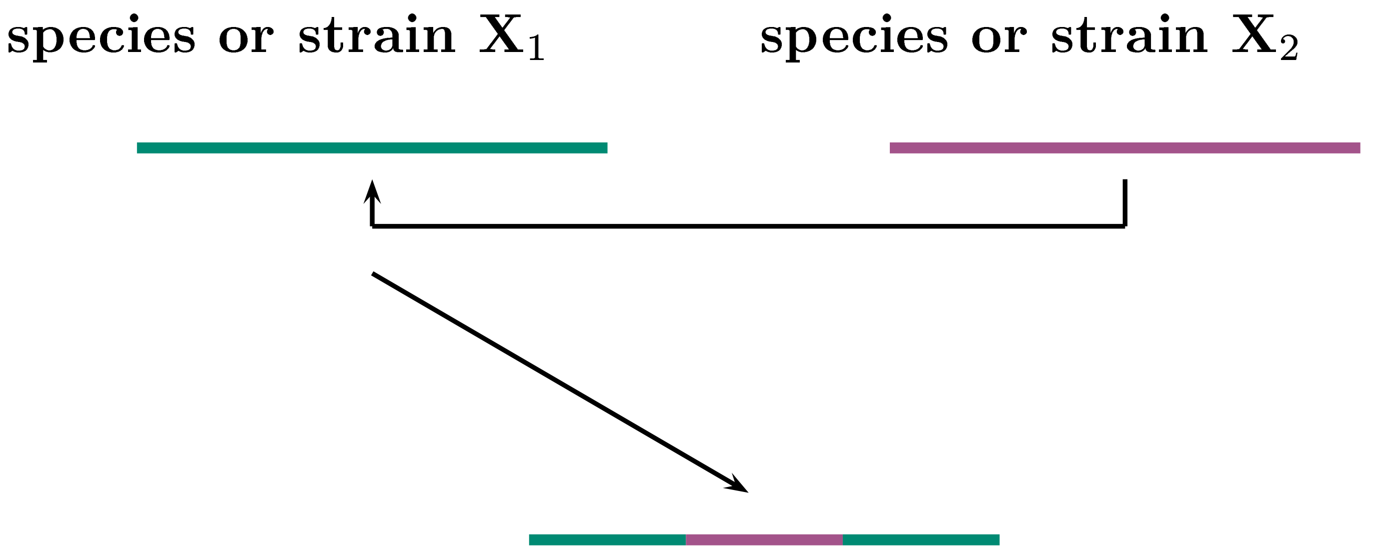
(Approx'01, LNCS, vol. 2129, pages 159-169, Springer-Verlag)

One possible motivation in biology

Genetic transfer detection and location

Genetic transfer

Horizontal transfer



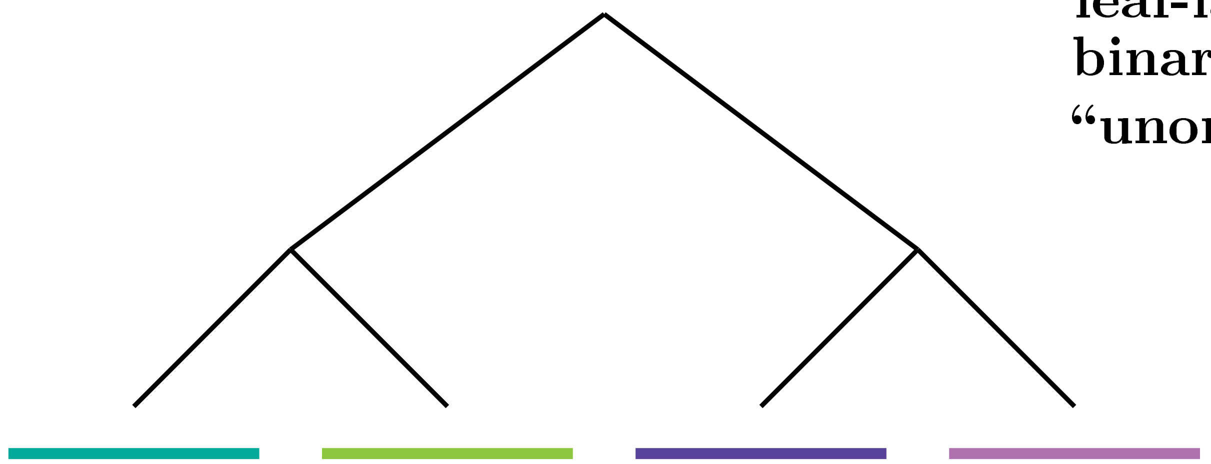
Genetic transfer

Conversion

Effect of transfer on the history of the organisms

Reminder: Evolutionary history represented by means of a so-called phylogenetic tree

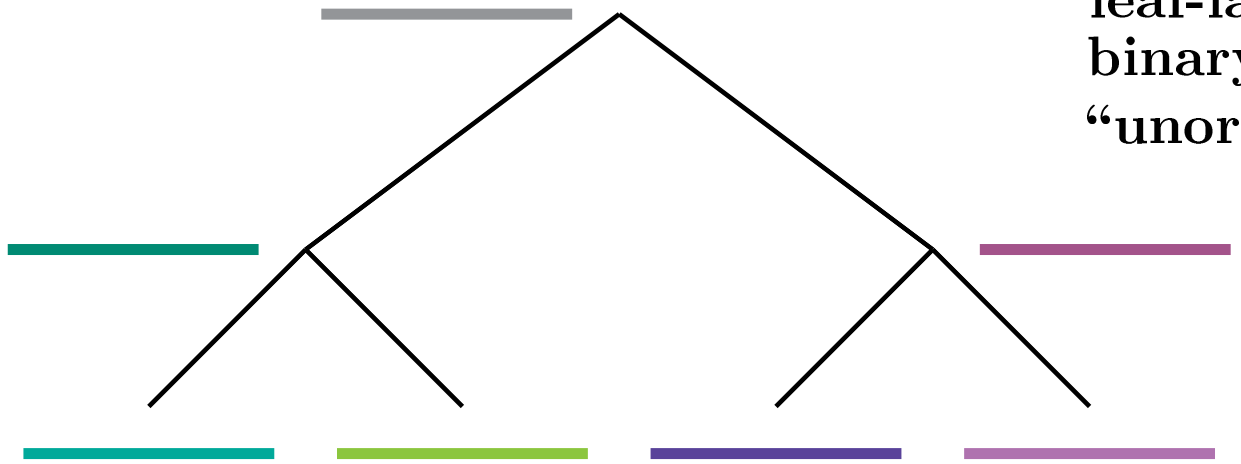
rooted
leaf-labeled
binary
“unordered”



Effect of transfer on the history of the organisms

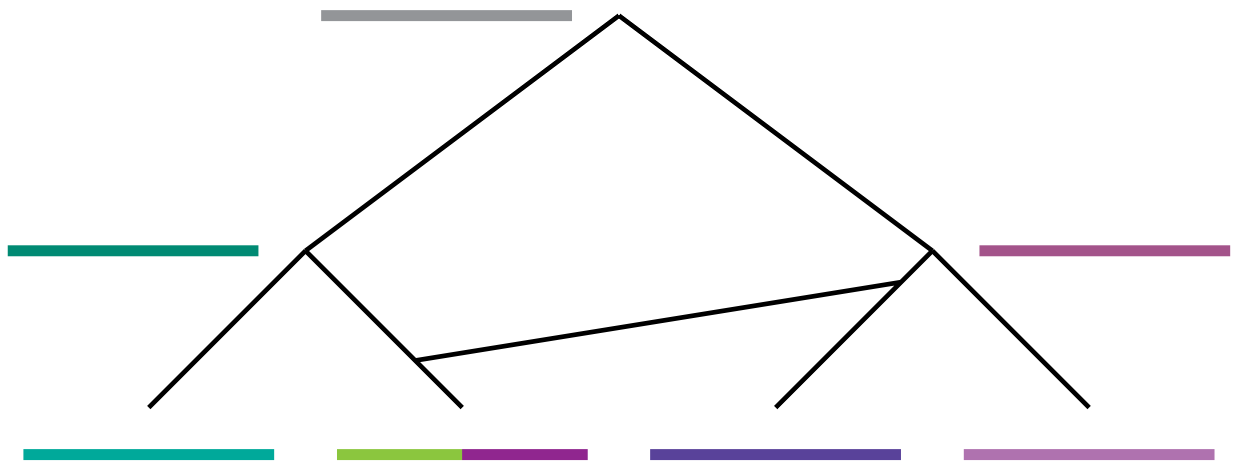
Reminder: Evolutionary history represented by means of a so-called phylogenetic tree

rooted
leaf-labeled
binary
"unordered"



Effect of transfer on the history of the organisms

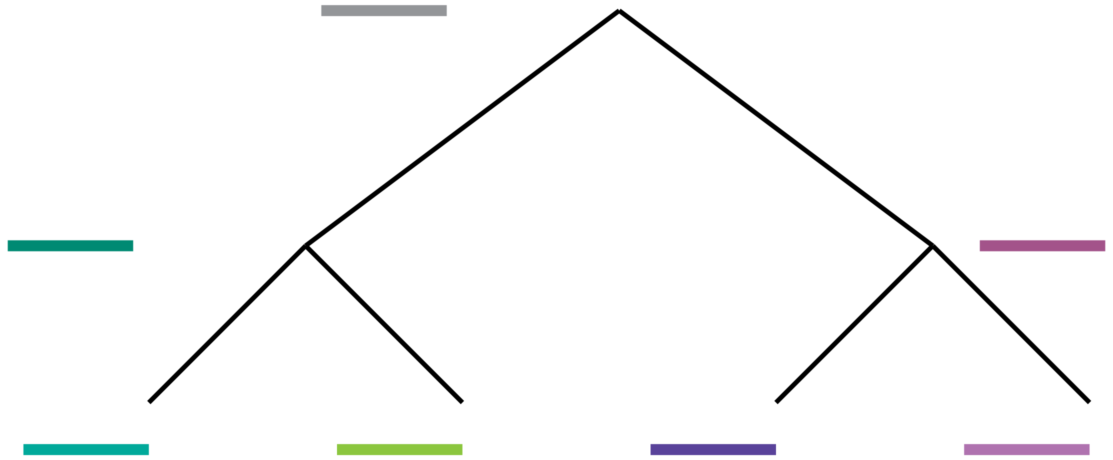
Not a single tree anymore



Effect of transfer on the history of the organisms

Not a single tree anymore but two ! (or more)

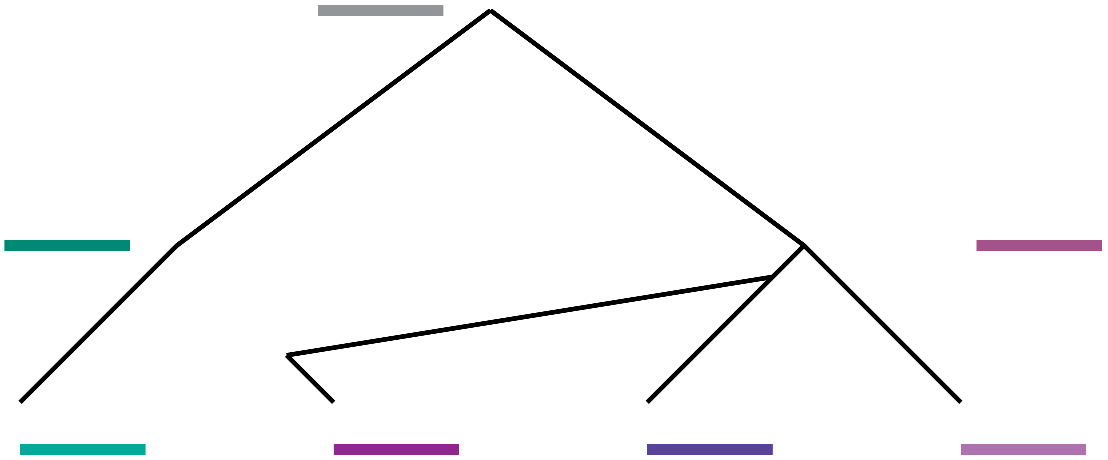
the first:



Effect of transfer on the history of the organisms

Not a single tree anymore but two ! (or more)

the second:



Genetic transfer identification and location

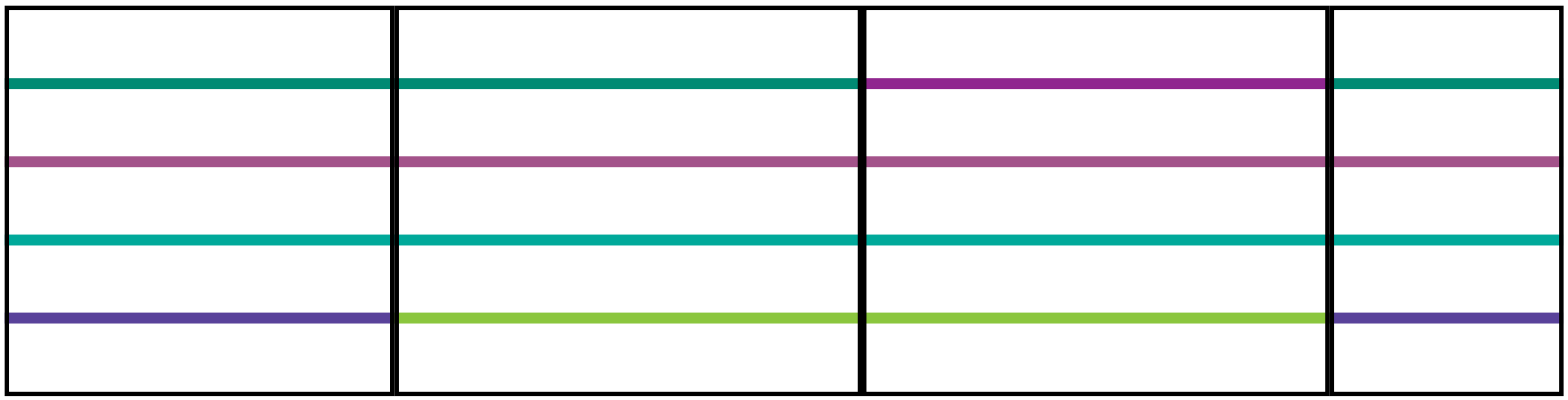
Data

a set of phylogenetically "correctly" aligned sequences from closely related organisms

Genetic transfer identification and location as an optimisation problem

The (ideal ?) result

alignement



tree 1 **tree 2** **tree 3** **tree 4**

“optimal” partition: sum of lengths of trees + penalty for genetic transfers is MINIMUM

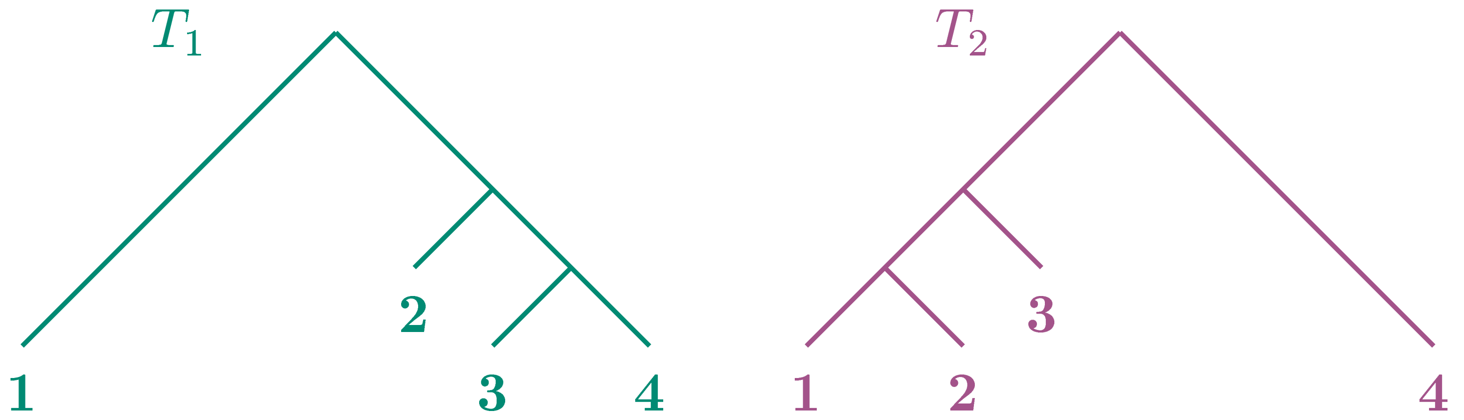
A couple of fundamental questions

How to calculate the genetic transfer distance between two phylogenetic trees ?

How to evaluate whether the obtained (by whatever means) distance is statistically significant ?

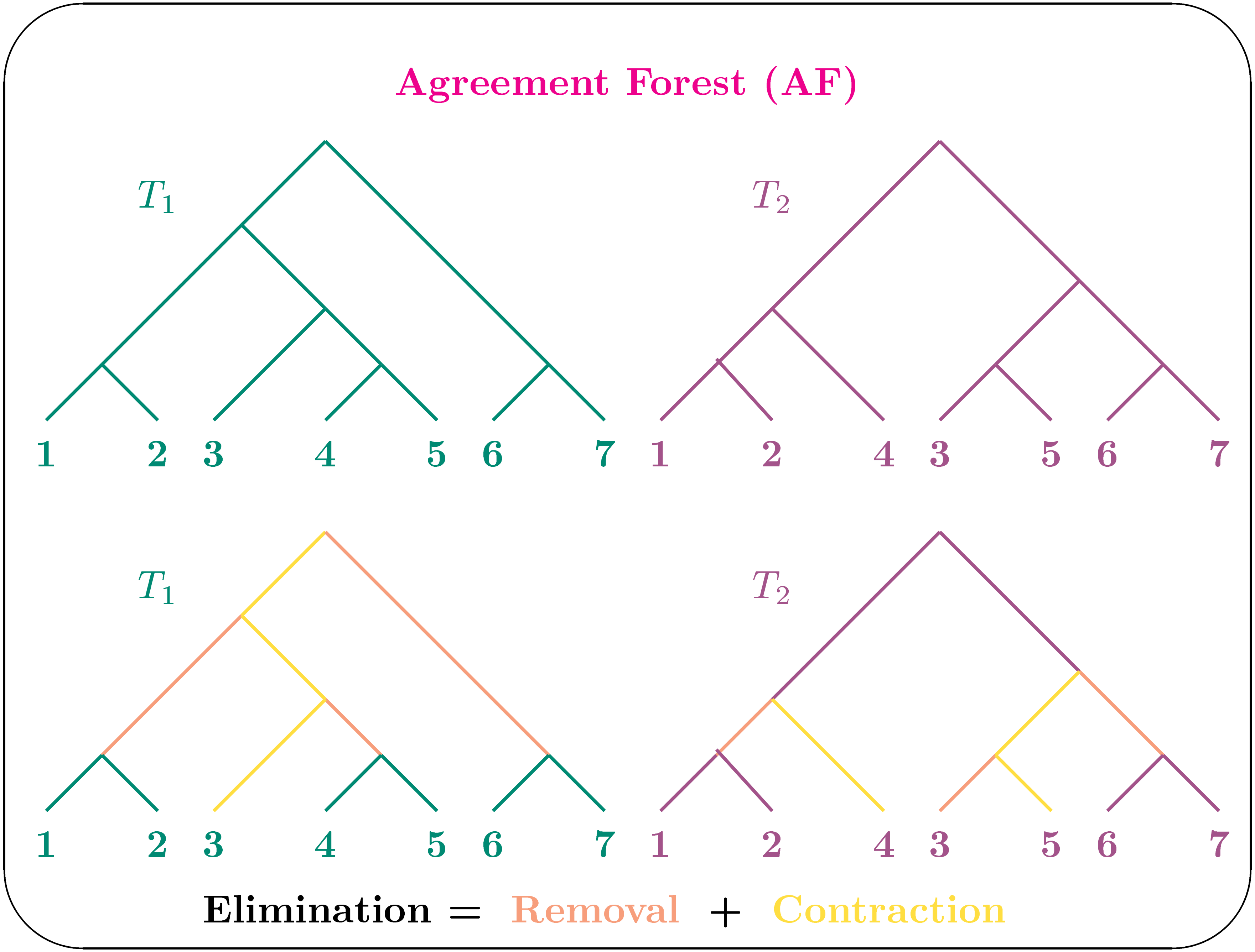
Genetic transfer distance between two phylogenetic trees

The “best” (?) measure: Subtree Prune and Regraft (SPR)



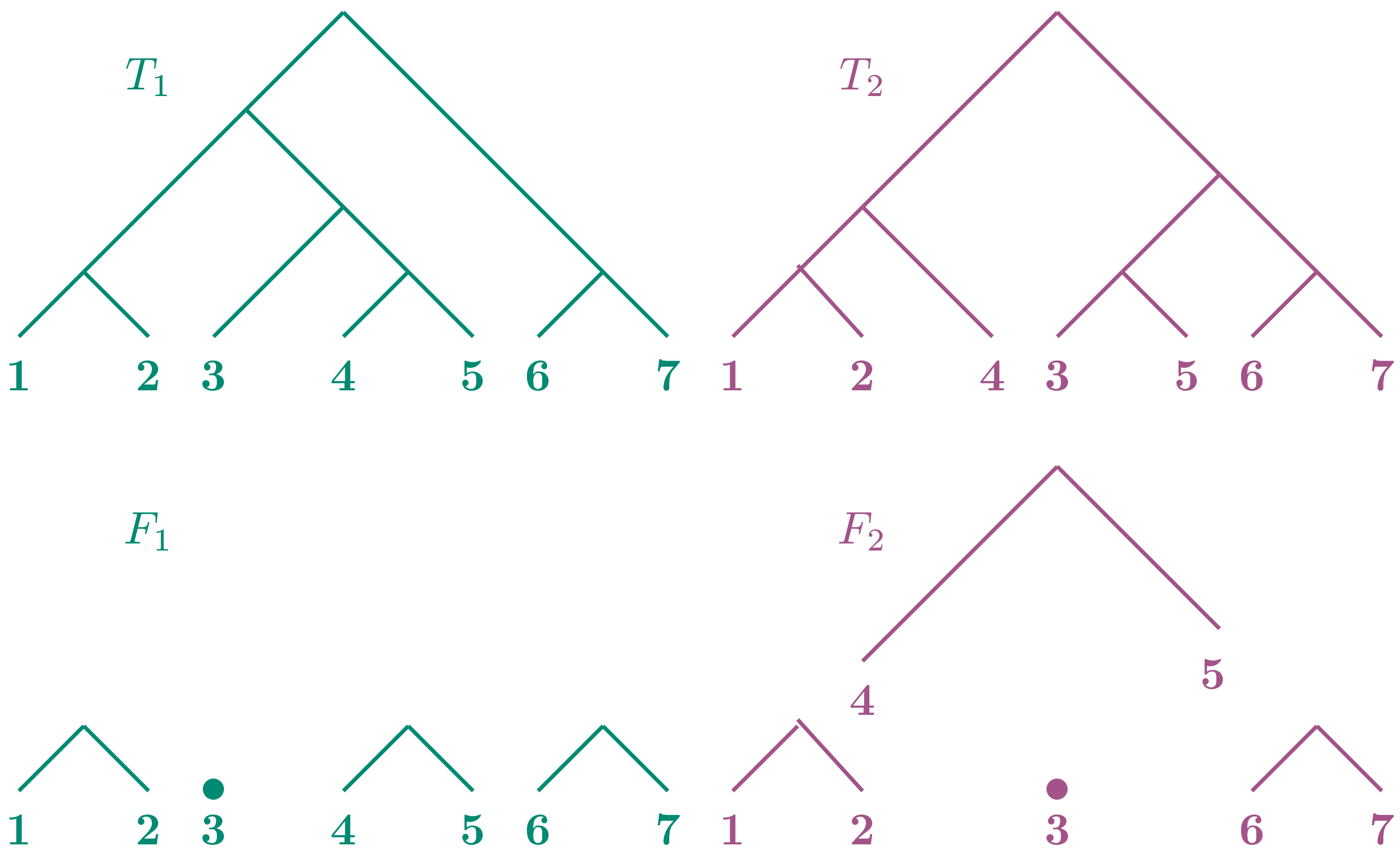
Number of subtree prunes (cuts) and regrafts needed to get from T_1 to T_2 is MINIMAL

here 2 moves are needed



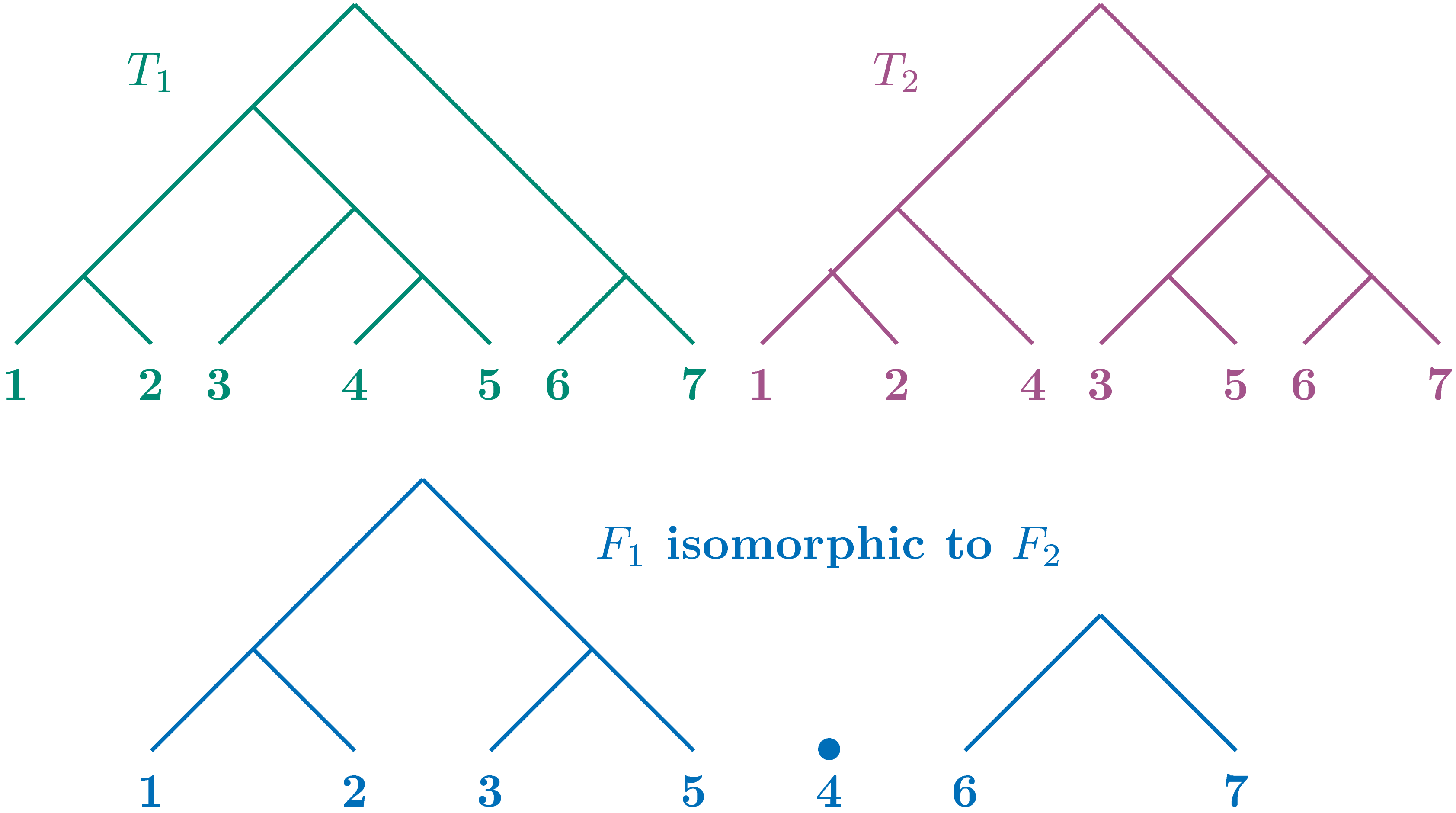
Agreement Forest (AF)
T_1
T_2
1 2 3 4 5 6 7
1 2 4 3 5 6 7
T_1
T_2
1 2 3 4 5 6 7
1 2 4 3 5 6 7
Elimination = Removal + Contraction

Agreement Forest (AF)



size (*i.e.* number of components) = 4

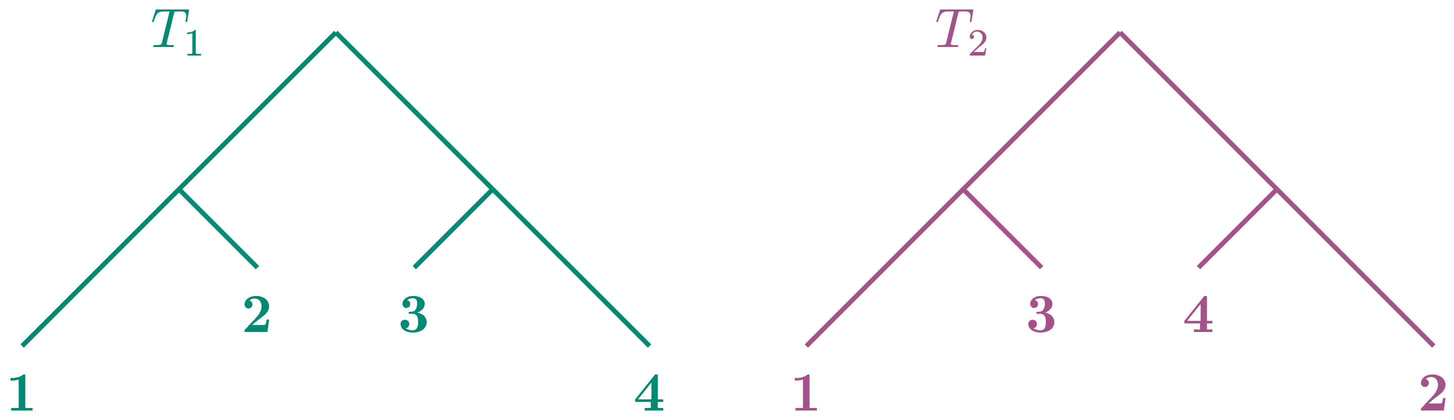
Maximum Agreement Forest (MAF)



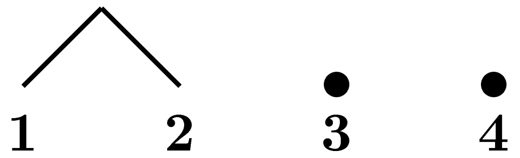
size of MAF = 3

Subtree Prune and Regraft (SPR)

Initially thought to be equal to MAF-1



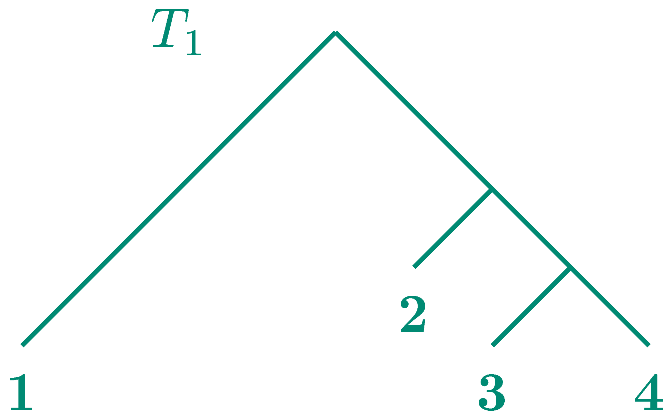
MAF of T_1 and T_2 **size 3**

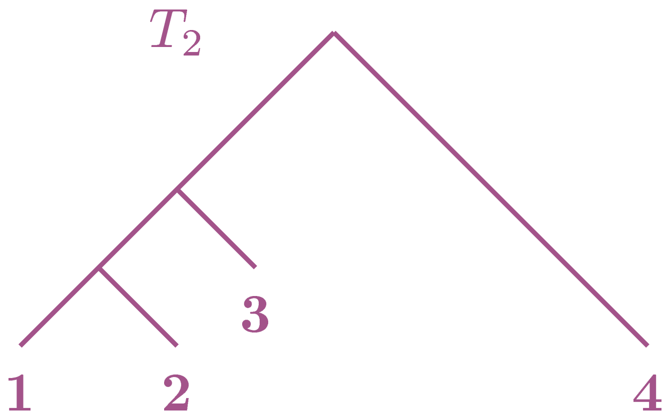


SPR of T_1 and T_2 **2 moves required**

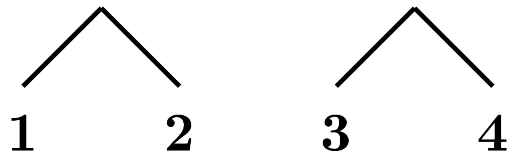
Subtree Prune and Regraft (SPR)

Initially thought to be equal to MAF-1 but in fact it is not (Allen and Steel, 2001)





MAF of T_1 and T_2



size 2

SPR of T_1 and T_2

2 moves required

Maximum Agreement Forest problem for binary trees

Shown to be NP-Hard by Hein *et al.*, 1996

(Suggested to be MAX-SNP-Hard)

Who also provided a 3-approximation algorithm for the problem

Maximum Agreement Forest problem for binary trees

Shown to be NP-Hard by Hein *et al.*, 1996

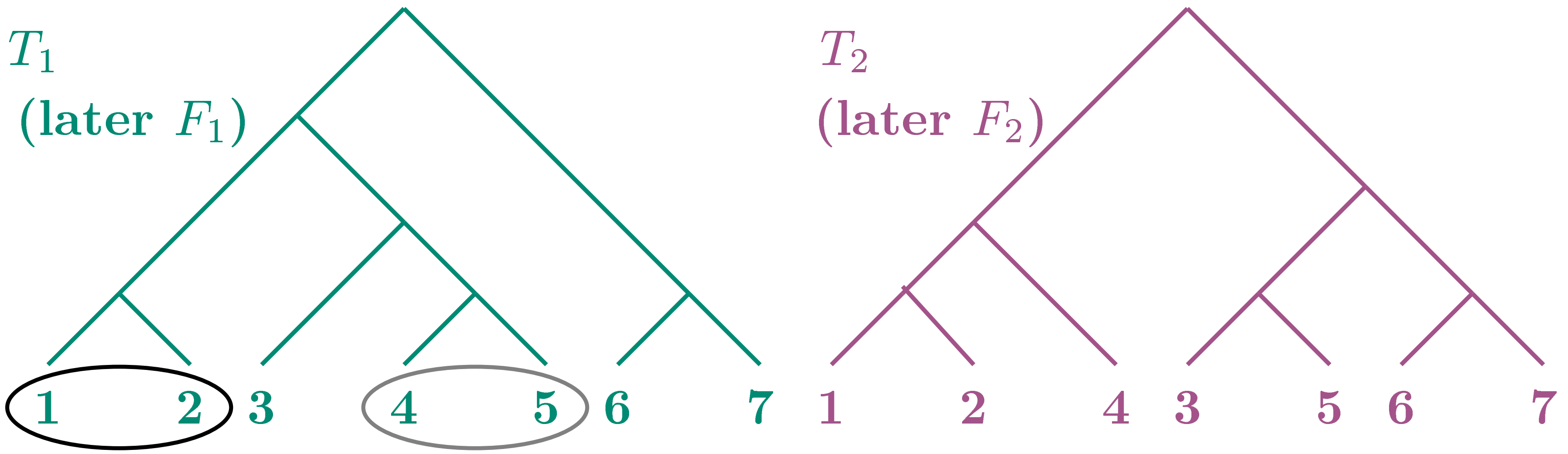
(Suggested to be MAX-SNP-Hard)

Who also provided a 3-approximation algorithm for the problem

Which, in fact, we show is a 4-approximation

We also provide an algorithm which is, indeed, a 3-approximation

Our algorithm for the MAF problem



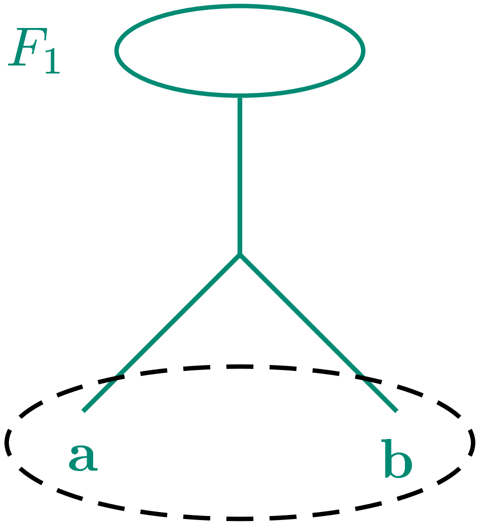
Search for pairs of sibling leaves in T_1 (F_1)

For each pair of sibling leaves in T_1 (F_1)
Consider the positions of the corresponding leaves in T_2 (F_2)

Various cases must be considered

The easiest cases

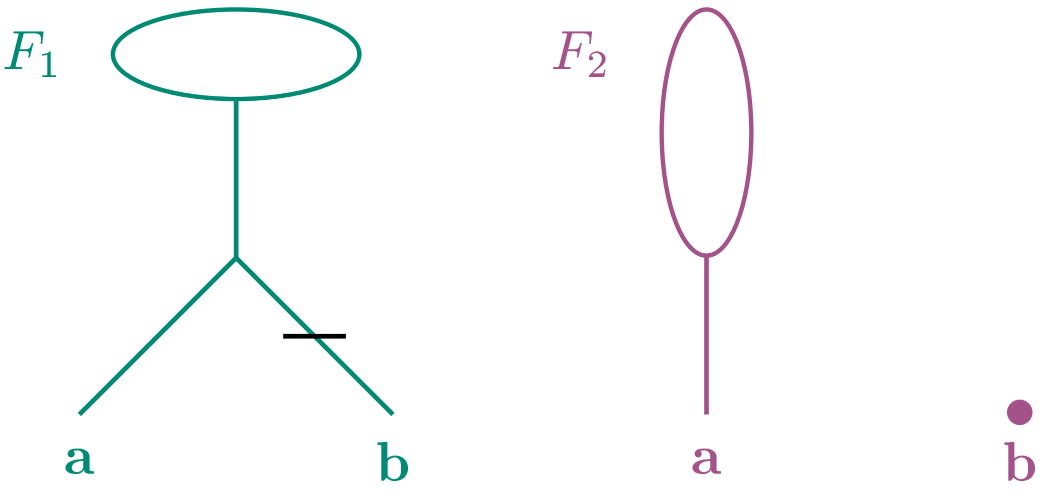
Case 1:



shrink

The easiest cases

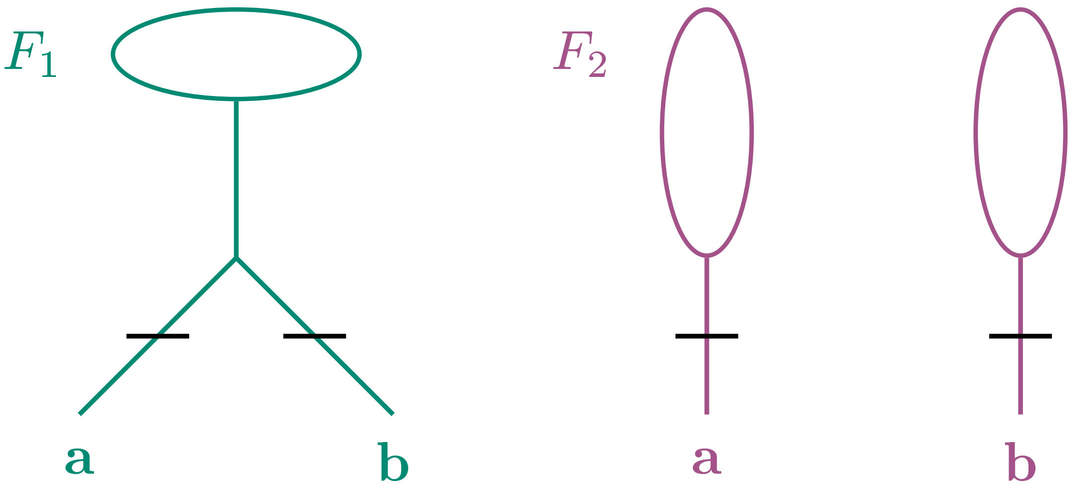
Case 2:



cut an edge in F_1 and contract

The easiest cases

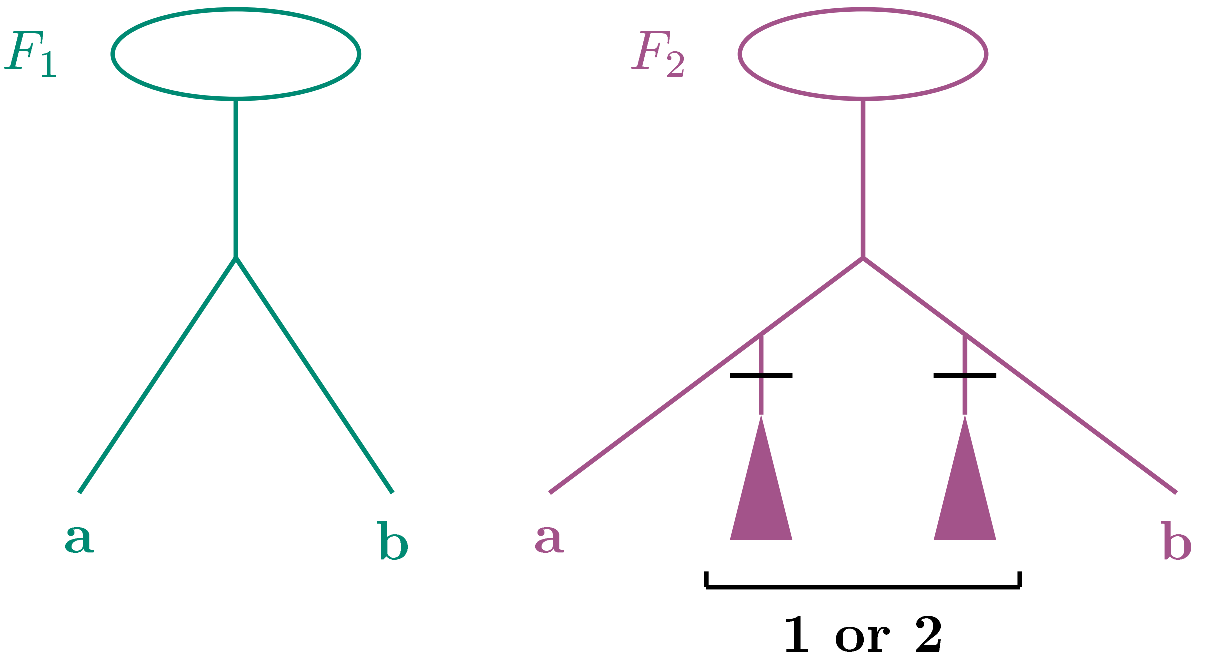
Case 3:



cut two edges each in F_1 and F_2 and contract in F_1

The two more delicate cases

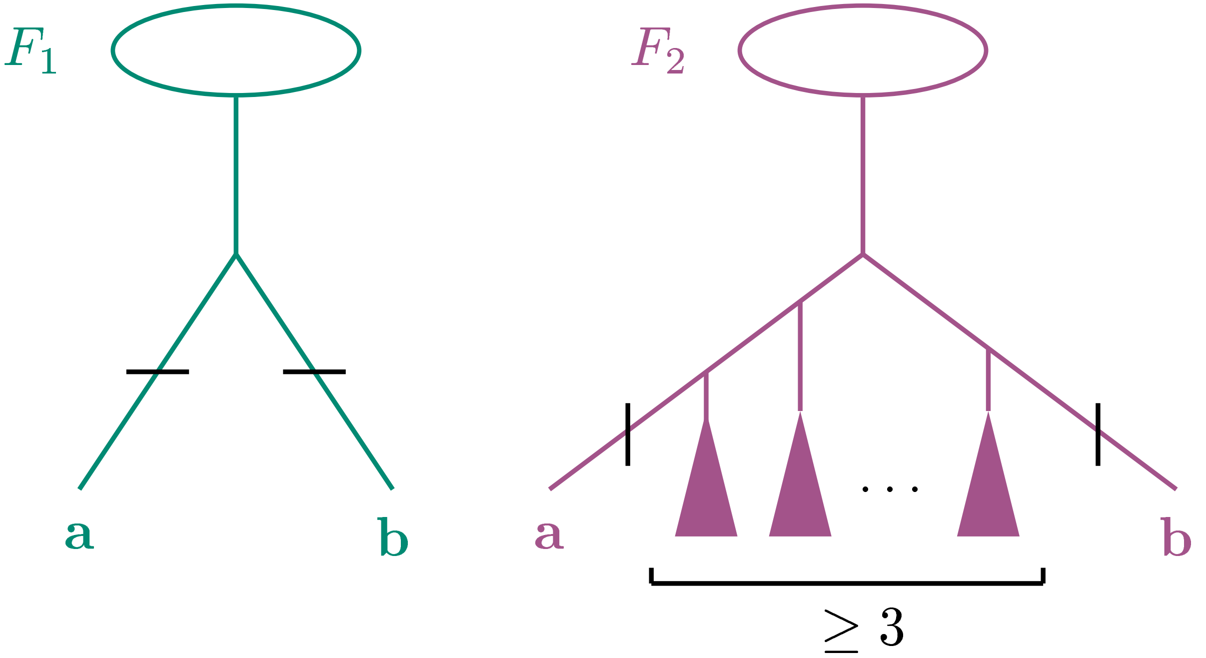
Case 4:



cut the subtrees hanging from the path from a to b in F_2

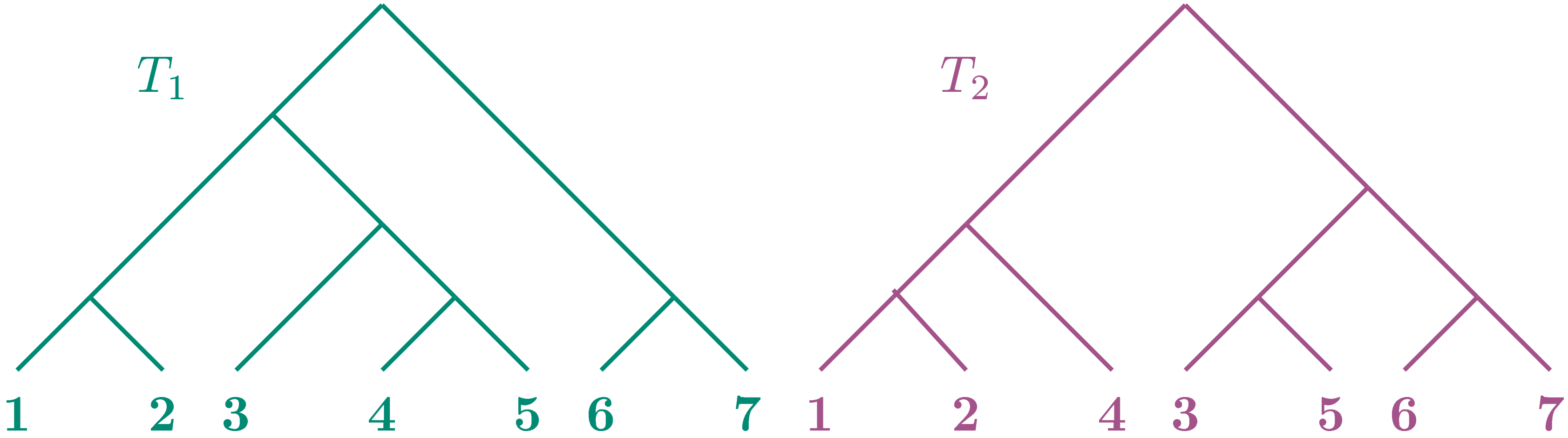
The two more delicate cases

Case 5:

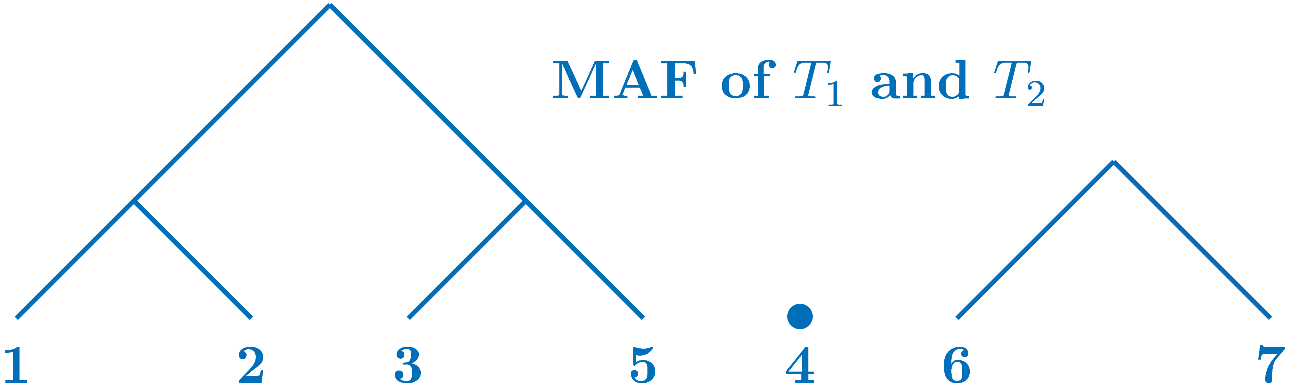


cut two edges each in F_1 and F_2
(those leading to the leaves labelled a and b)
and contract in F_1 and, possibly, F_2

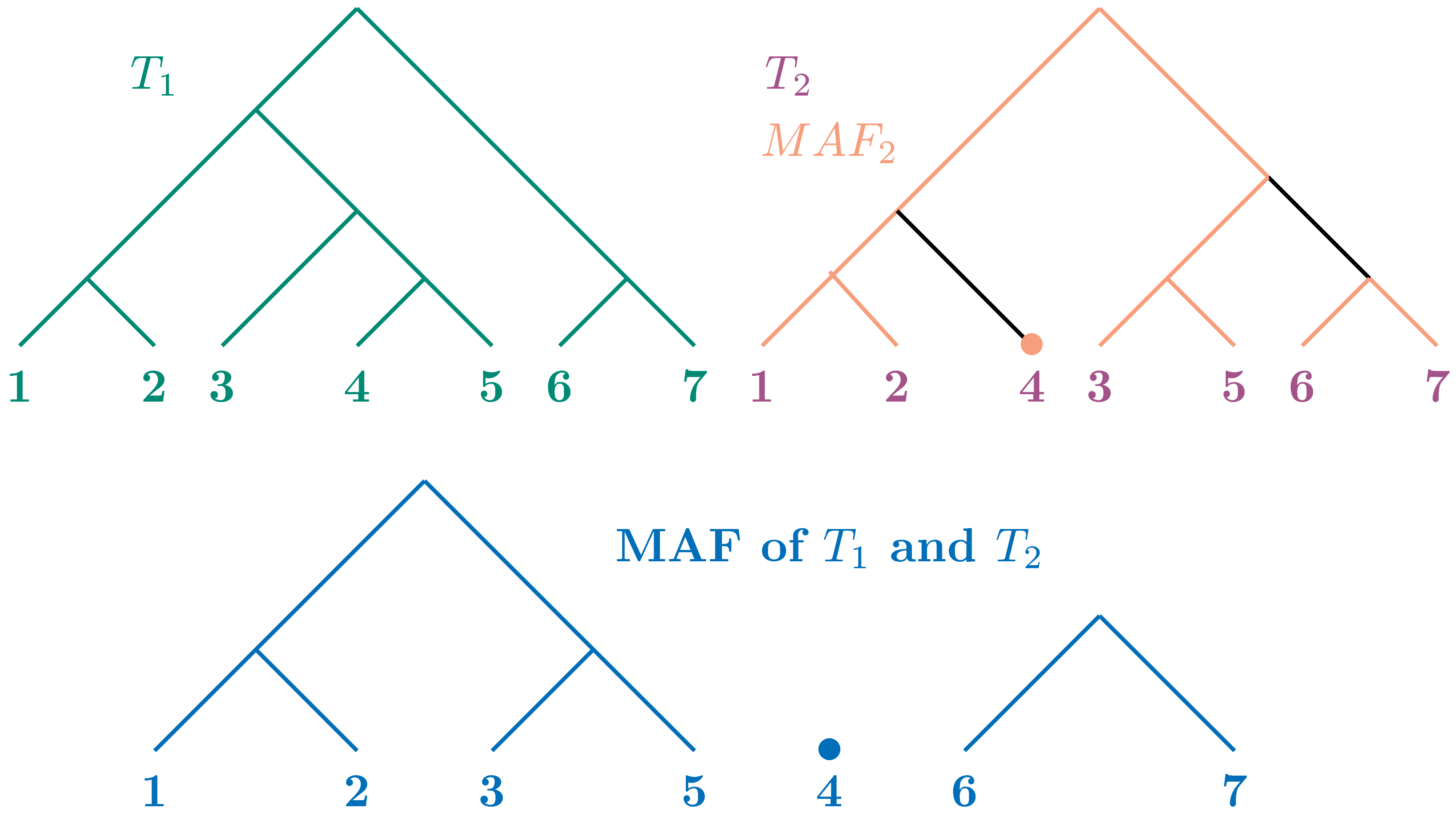
Outline of the ratio analysis



MAF of T_1 and T_2

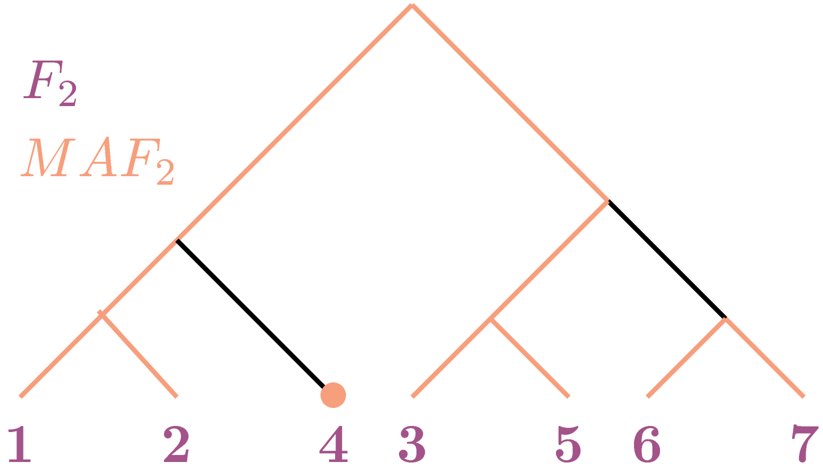


Outline of the ratio analysis



edges NOT belonging to MAF_2 are called links

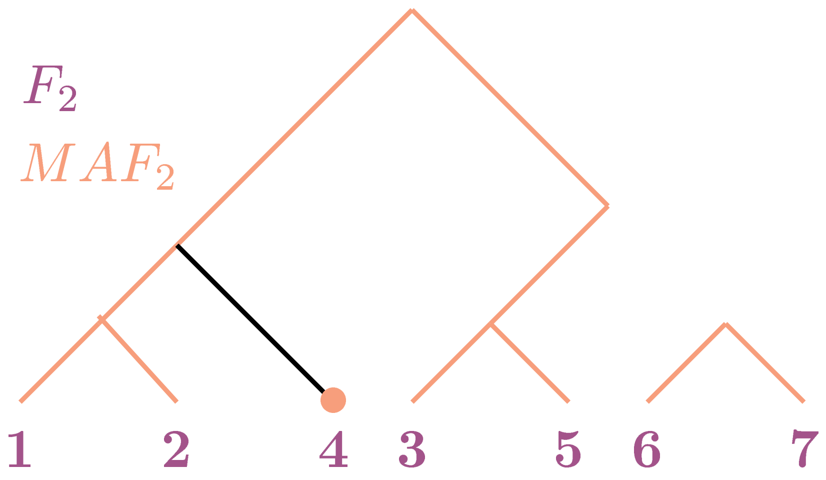
Outline of the ratio analysis



algorithm does "good" and "bad" eliminations

good elimination: elimination of a link

Outline of the ratio analysis



algorithm does "good" and "bad" eliminations

good elimination: elimination of a link

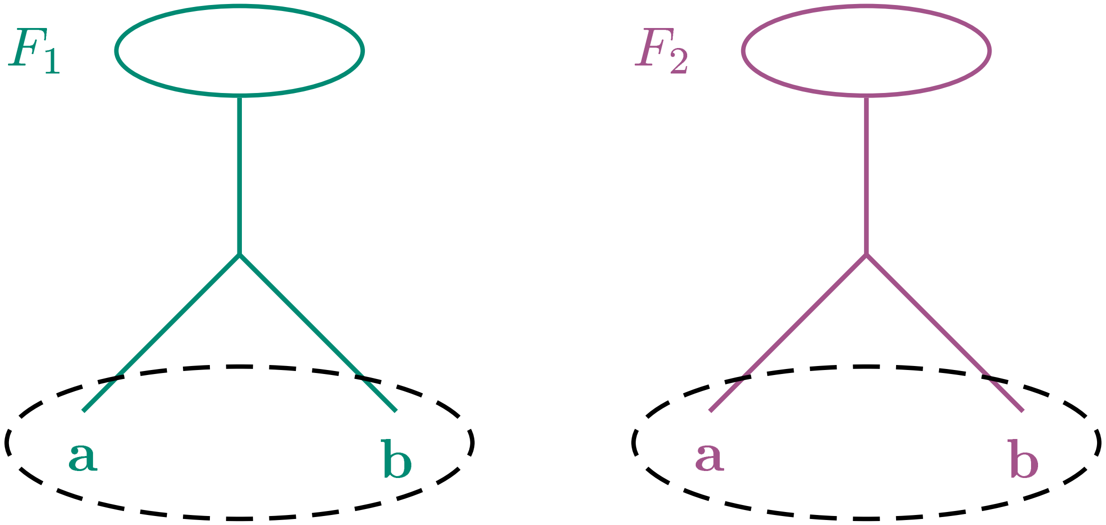
number of good eliminations is MAF-1

idea: prove at least one third of all eliminations are good

again various cases to analyse

The easiest cases

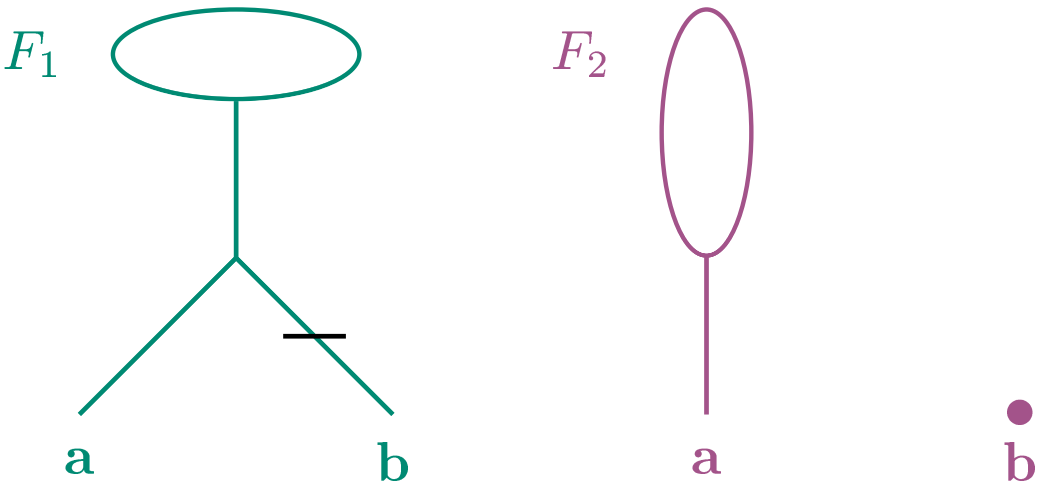
Case 1:



no eliminations

The easiest cases

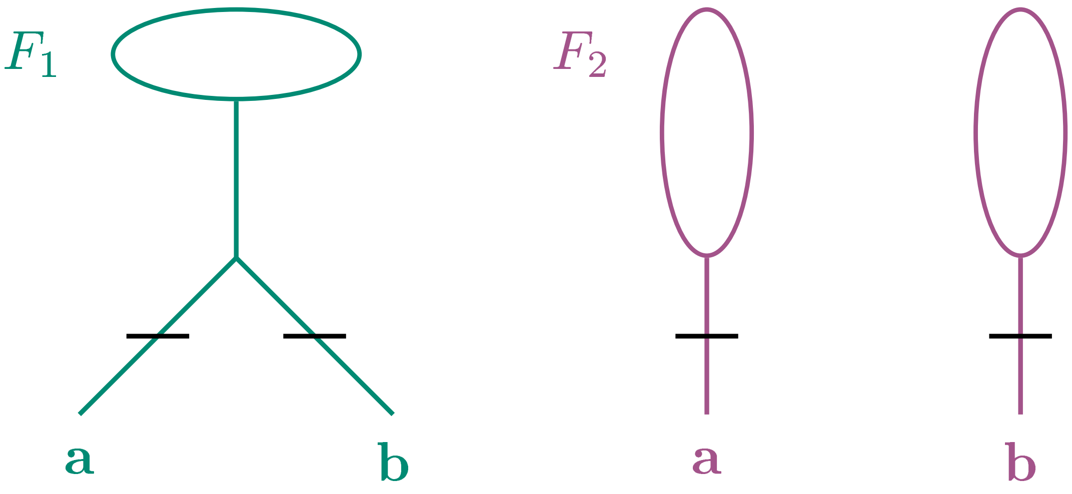
Case 2:



no eliminations (in F_2)

The easiest cases

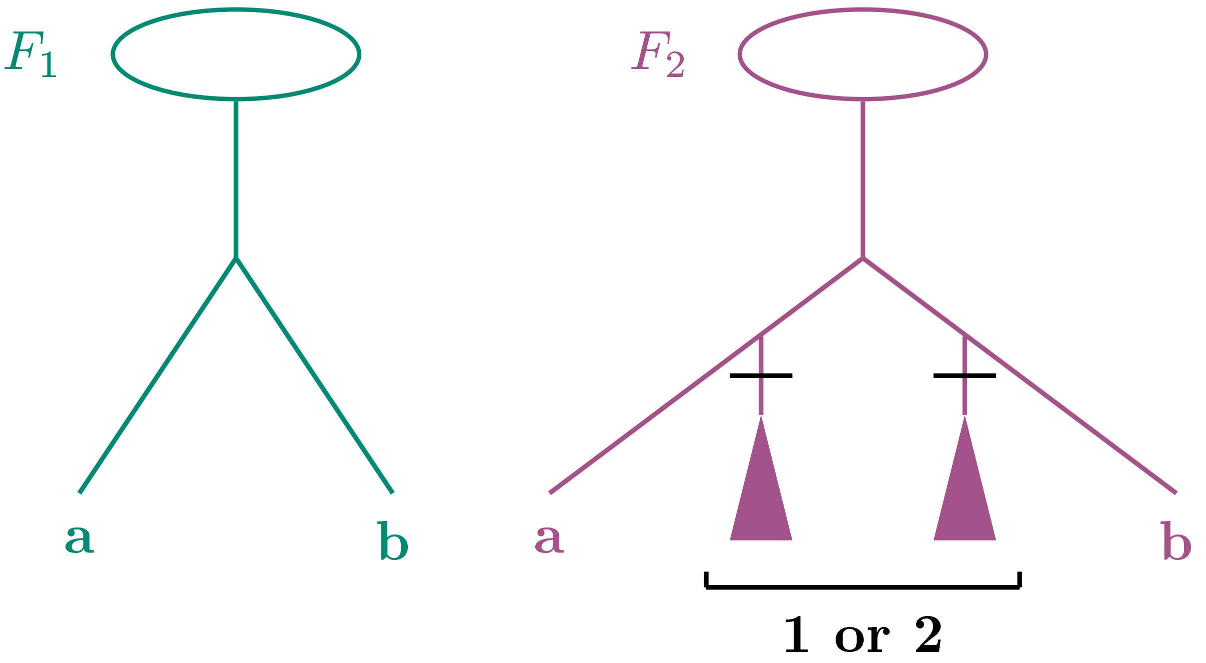
Case 3:



at least one good elimination out of the two performed

The two more delicate cases

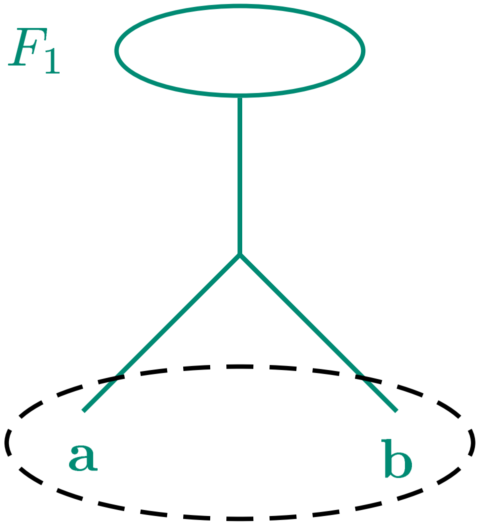
Case 4: **Iteration i**



possibly both are bad eliminations but...

The two more delicate cases

Case 4: **Iteration $i+1$**



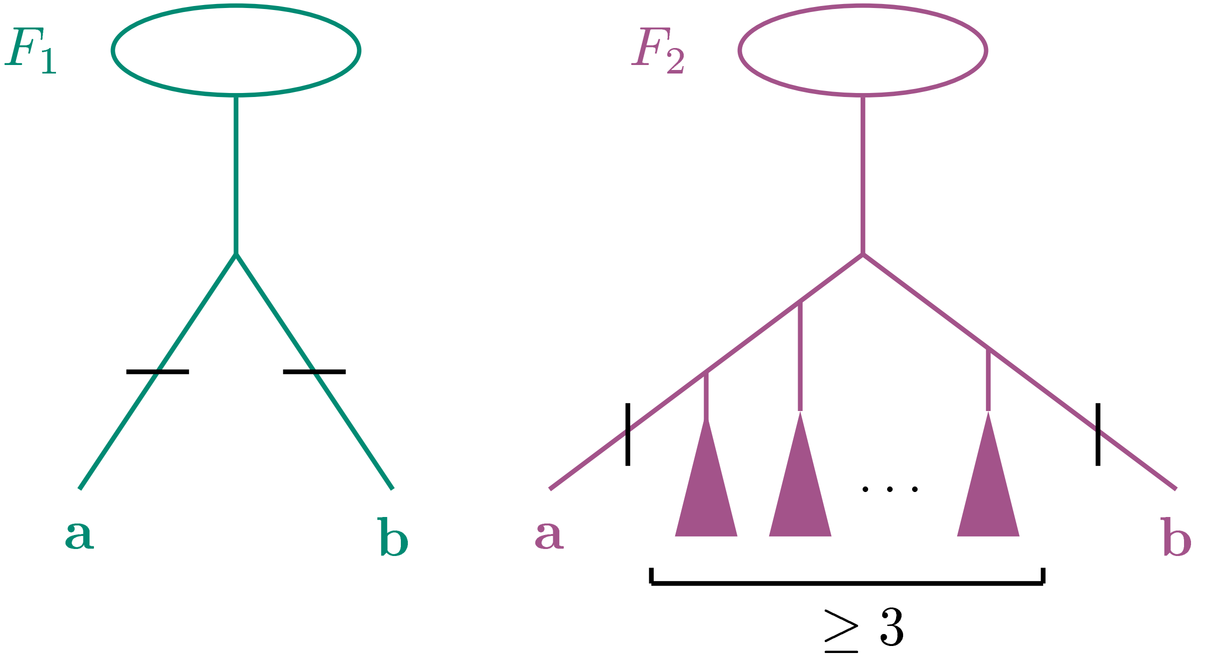
shrink and, therefore, no eliminations performed

but at least one shrinked edge is a link

so that we have the "equivalent" of a good elimination

The two more delicate cases

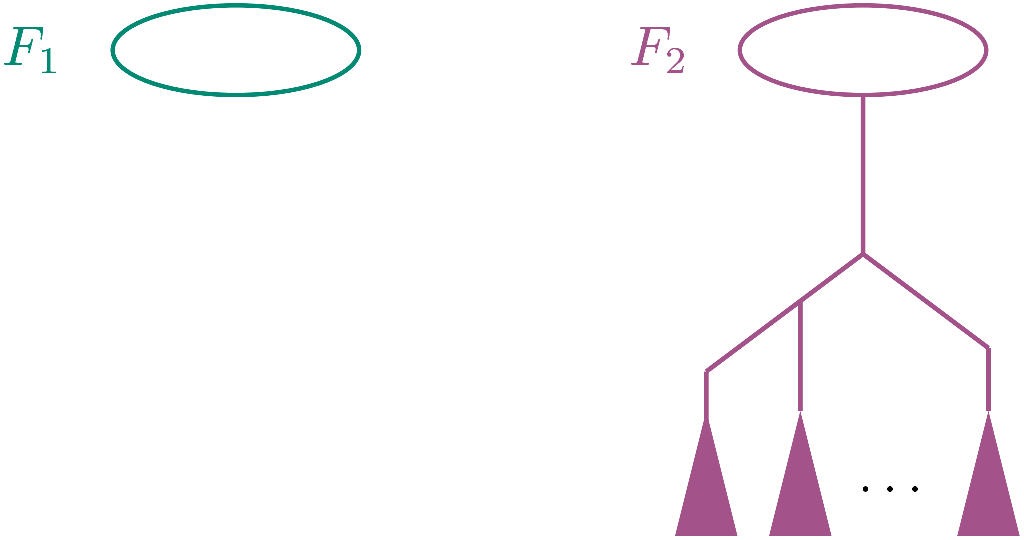
Case 5: **Iteration i**



possibly both are bad eliminations but...

The two more delicate cases

Case 5: **Iteration** $i+1$



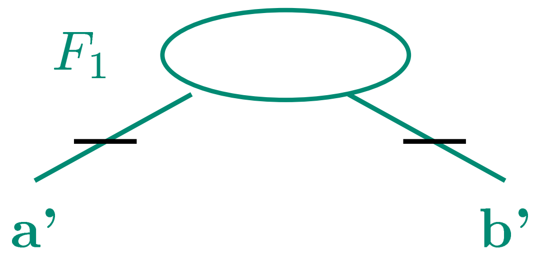
suppose both cuts were bad

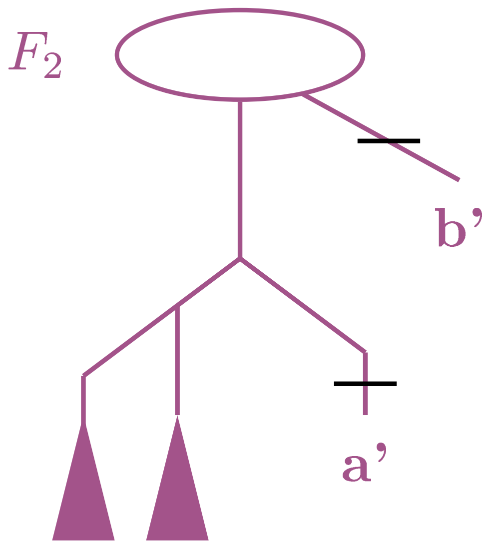
it can be proven that, in this case,

all the edges connecting hanging subtrees to component are links

The two more delicate cases

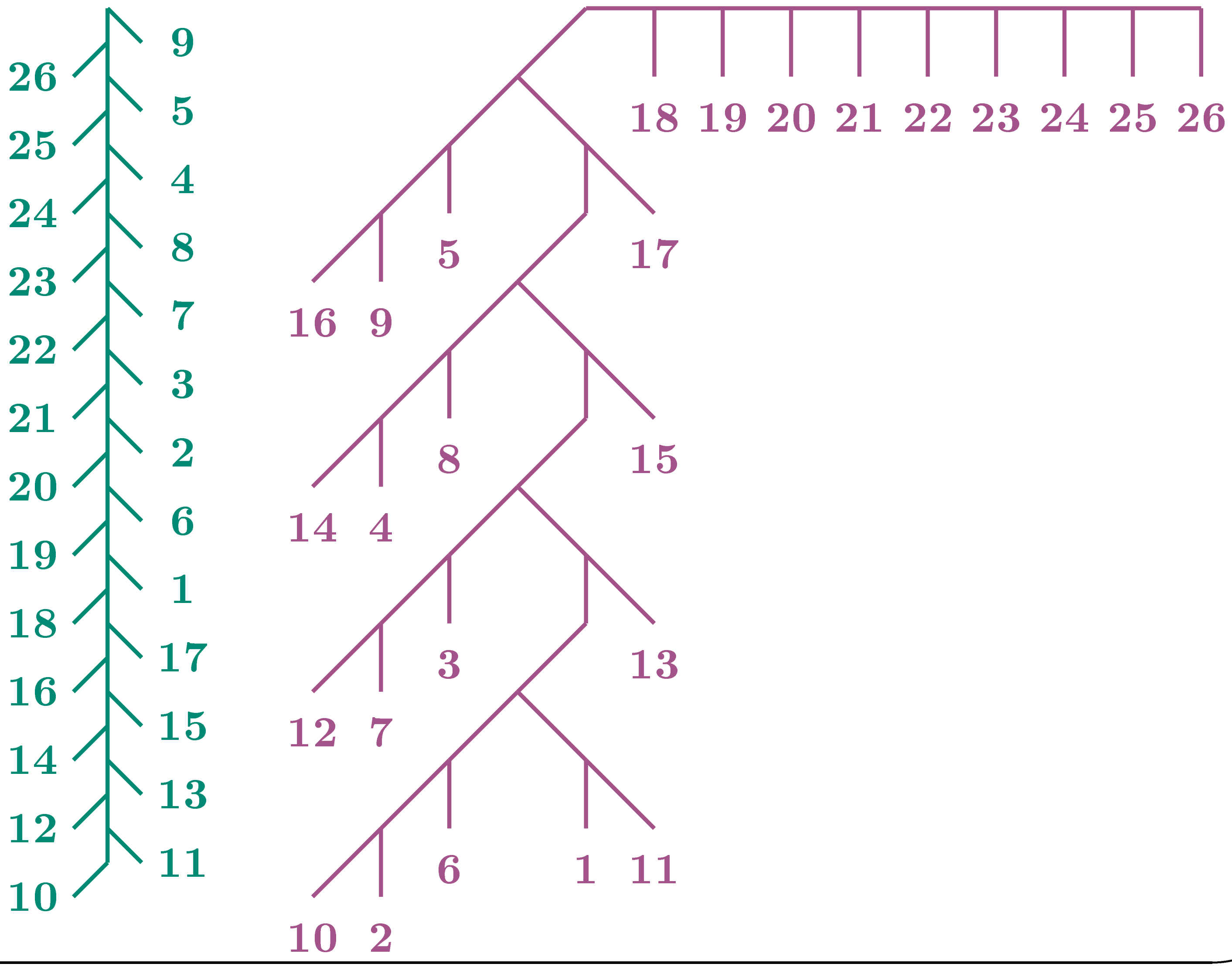
Case 5: **Iteration $i' > i$**



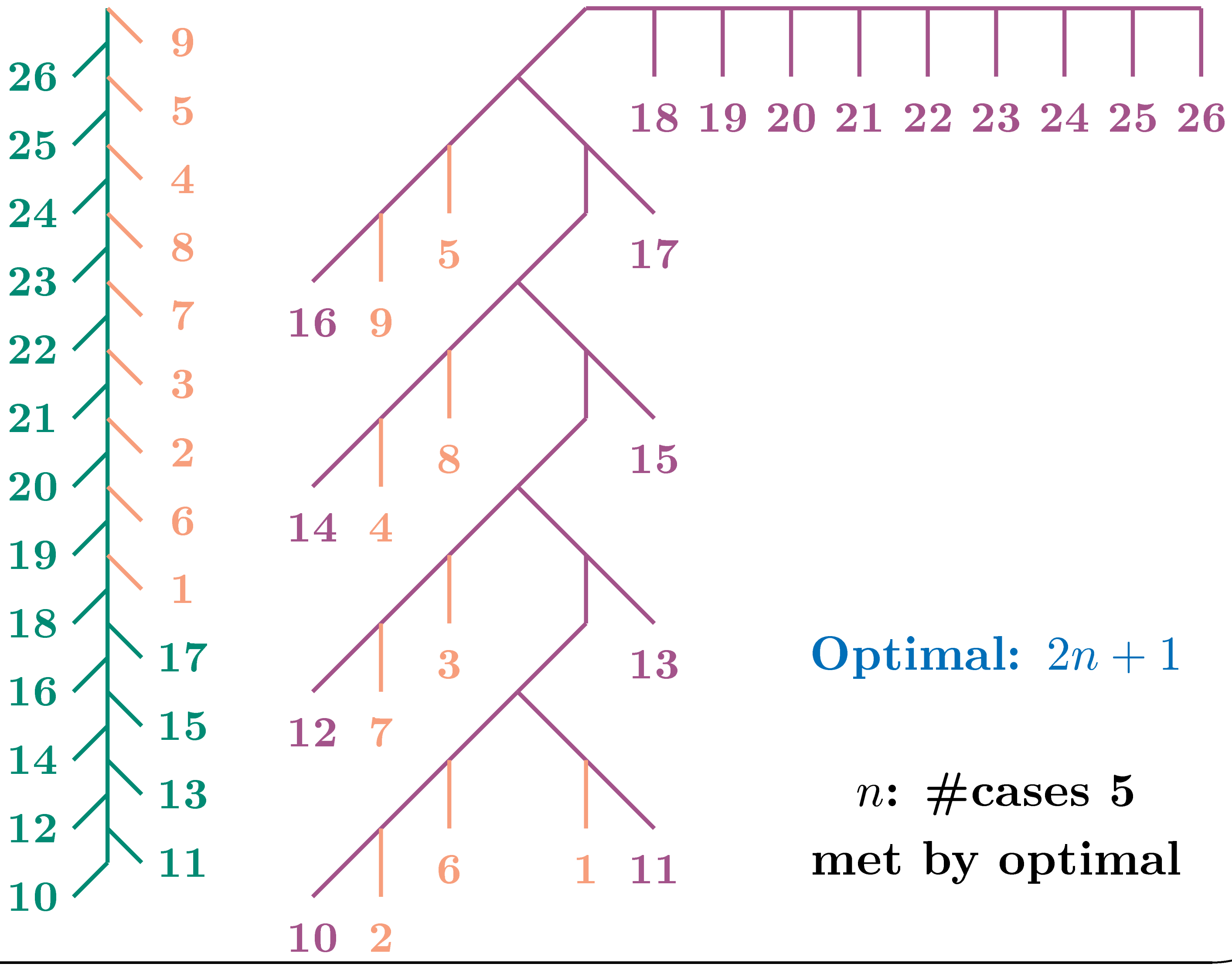


there is one good eliminination
which compensates for the other,
possibly bad elimination, in the same move
and for one of the two bad eliminations at step i

This is a tight ratio



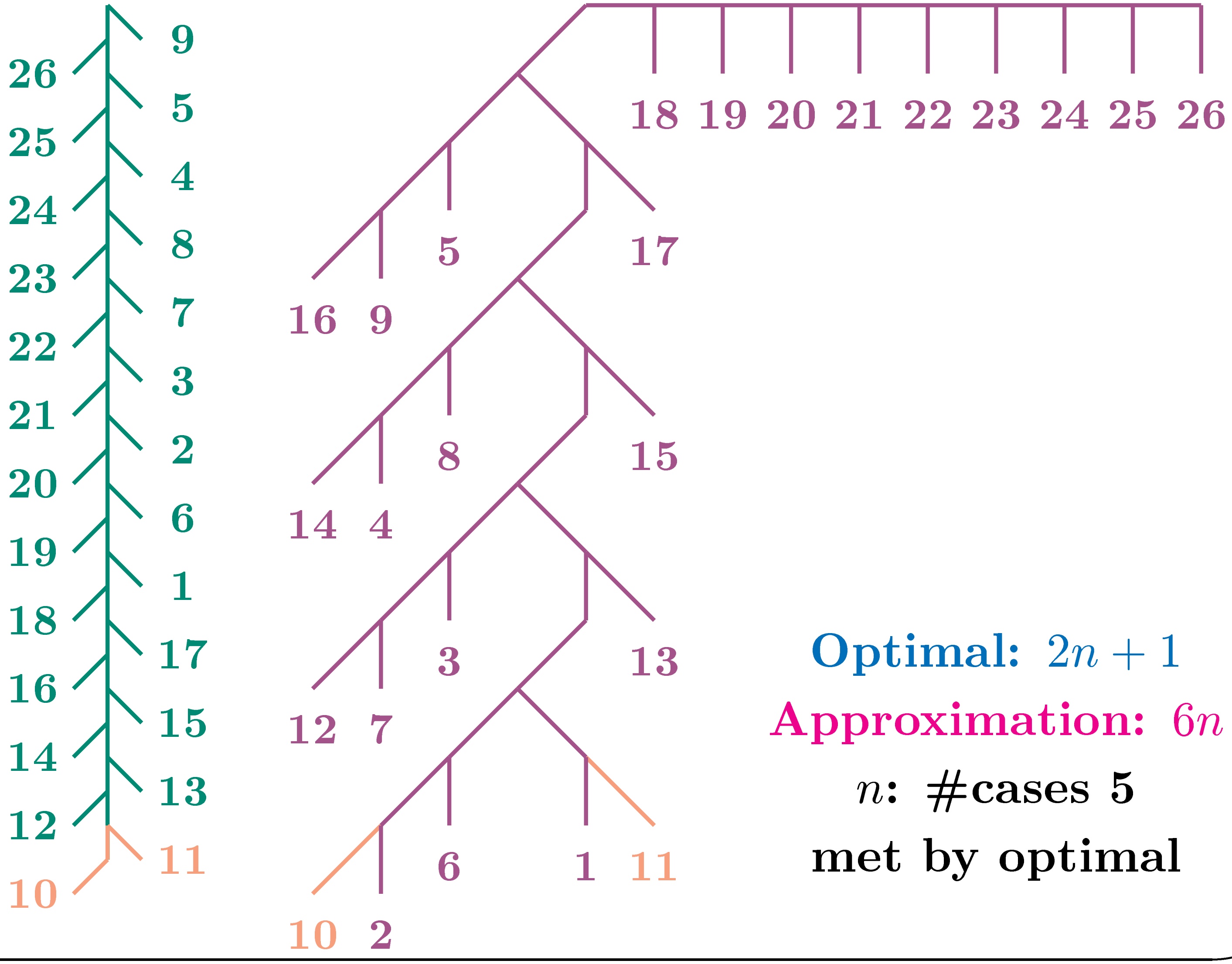
This is a tight ratio



Optimal: $2n+1$

n**: #cases 5**
met by optimal

This is a tight ratio



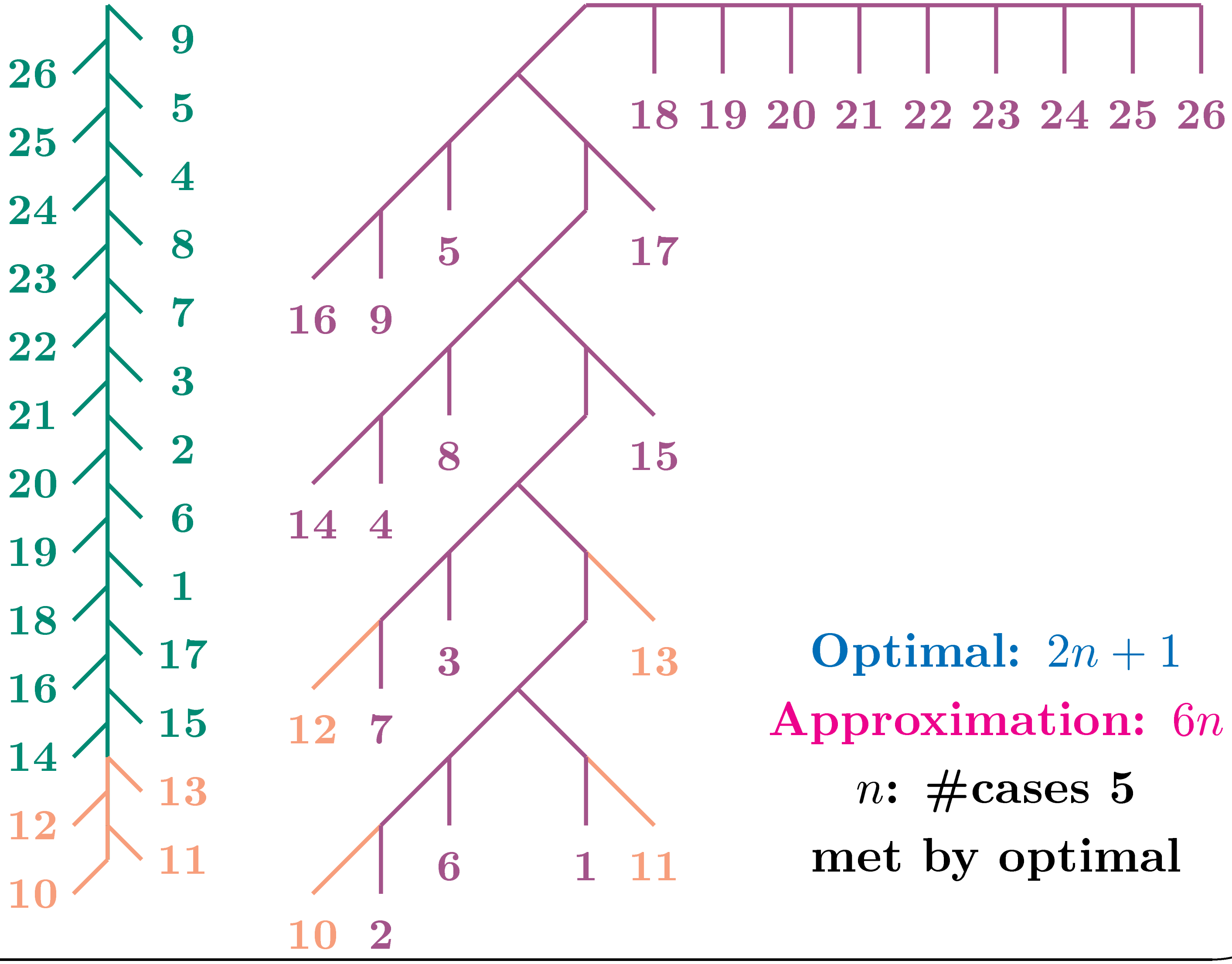
Optimal: $2n+1$

Approximation: $6n$

n**: #cases 5**

met by optimal

This is a tight ratio



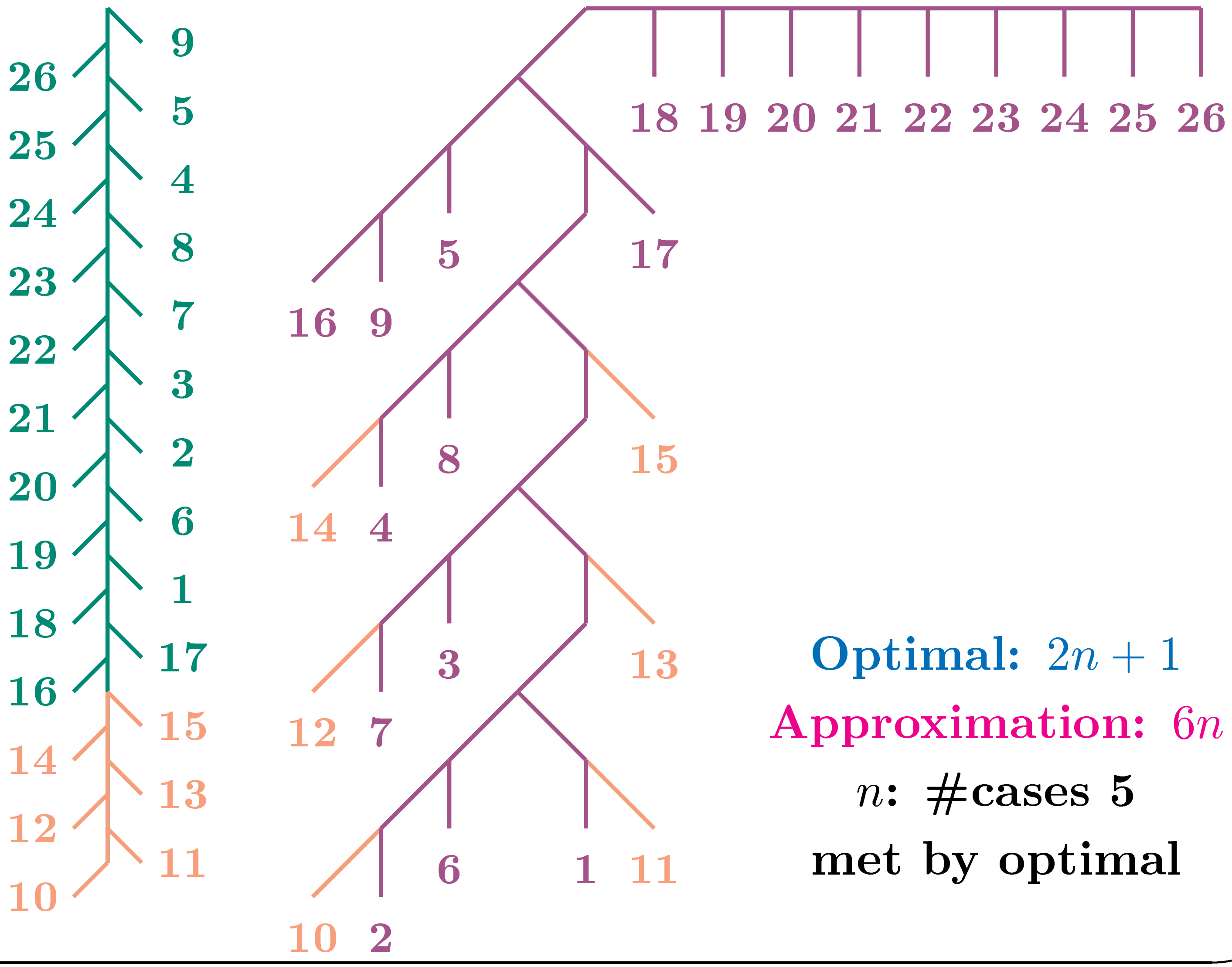
Optimal: $2n+1$

Approximation: $6n$

n**: #cases 5**

met by optimal

This is a tight ratio



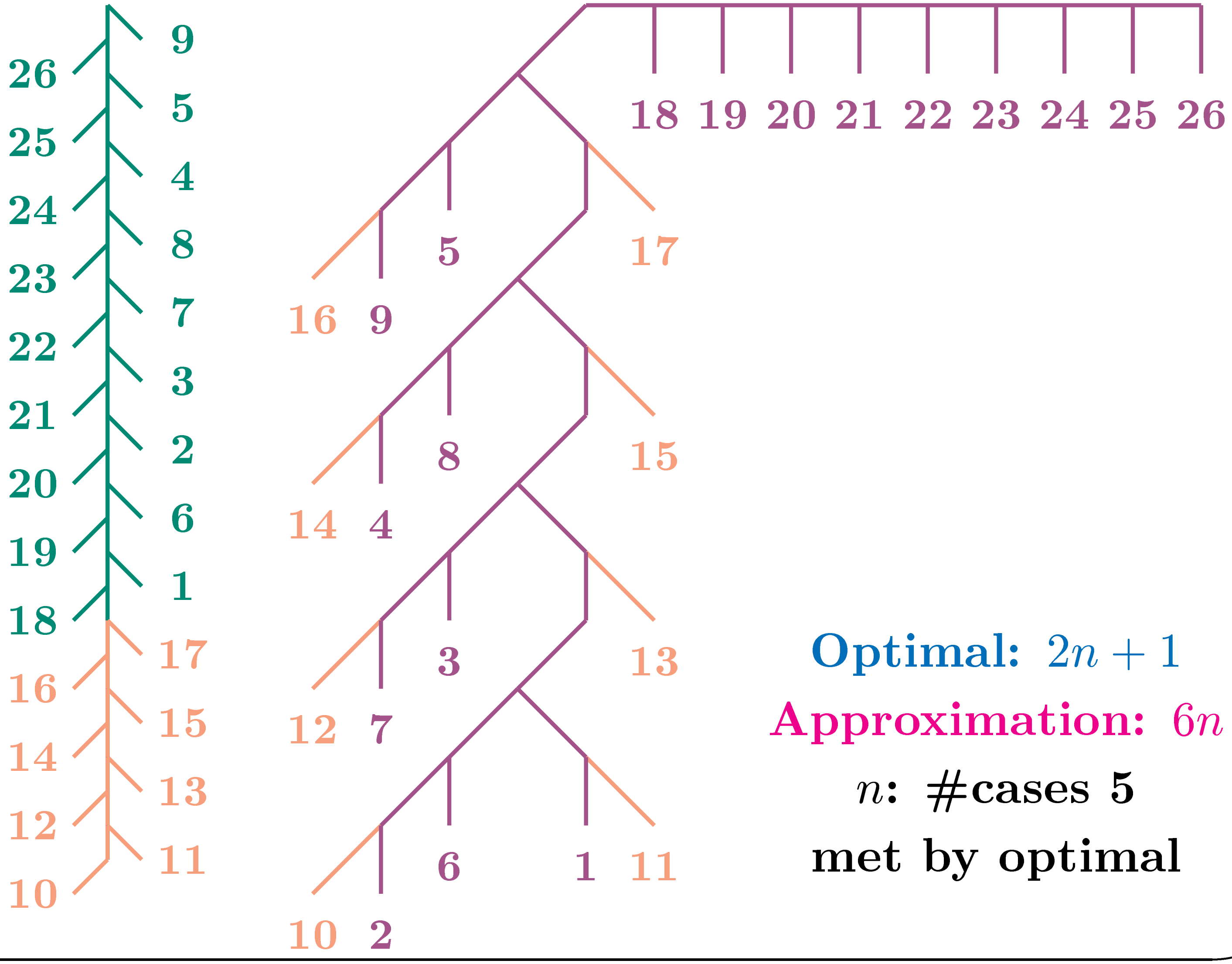
Optimal: $2n+1$

Approximation: $6n$

n**: #cases 5**

met by optimal

This is a tight ratio



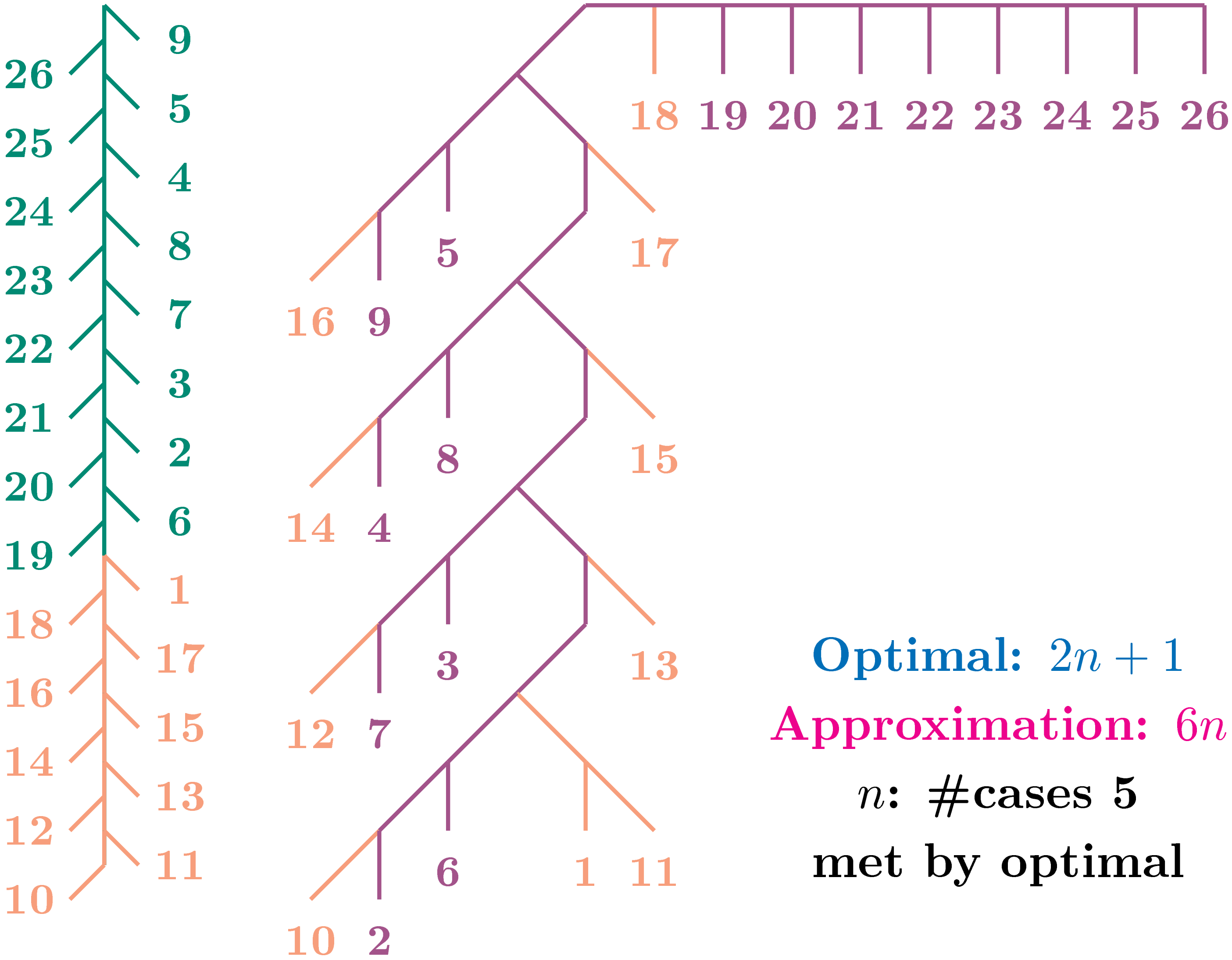
Optimal: $2n + 1$

Approximation: $6n$

n**: #cases 5**

met by optimal

This is a tight ratio



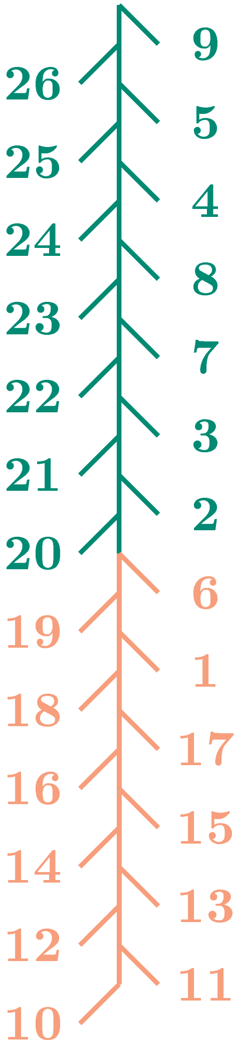
Optimal: $2n + 1$

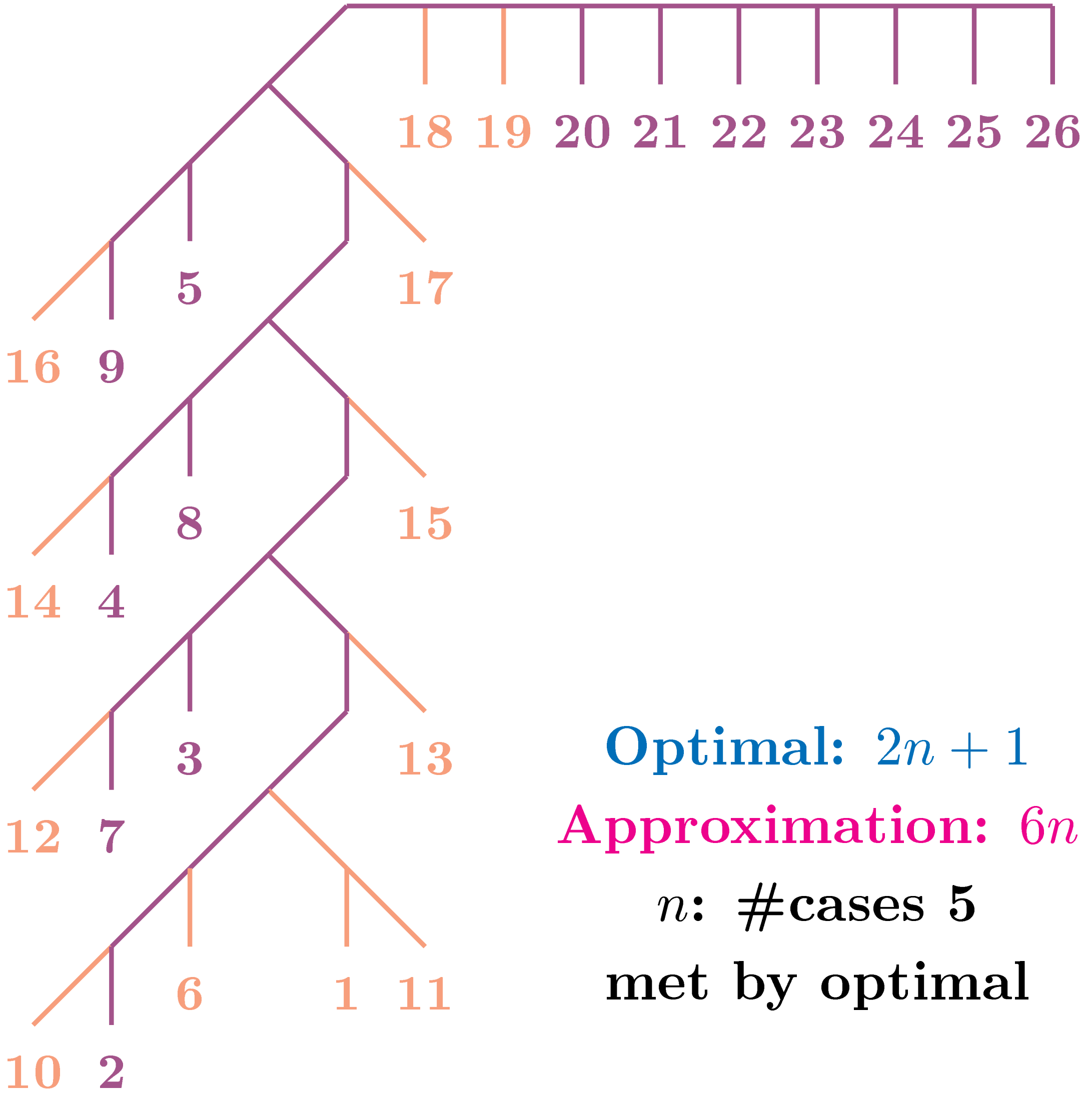
Approximation: $6n$

n**: #cases 5**

met by optimal

This is a tight ratio





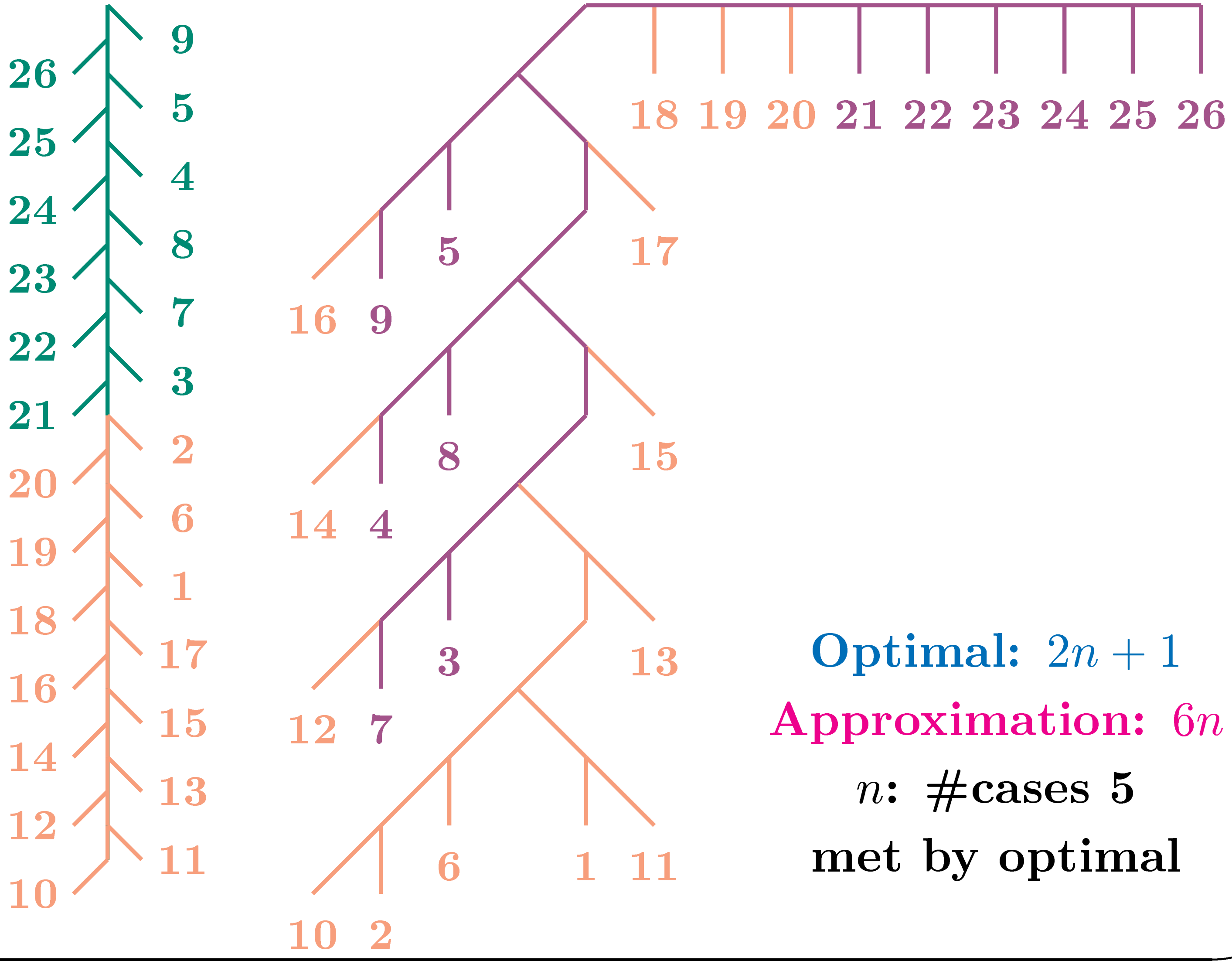
Optimal: $2n + 1$

Approximation: $6n$

n**: #cases 5**

met by optimal

This is a tight ratio



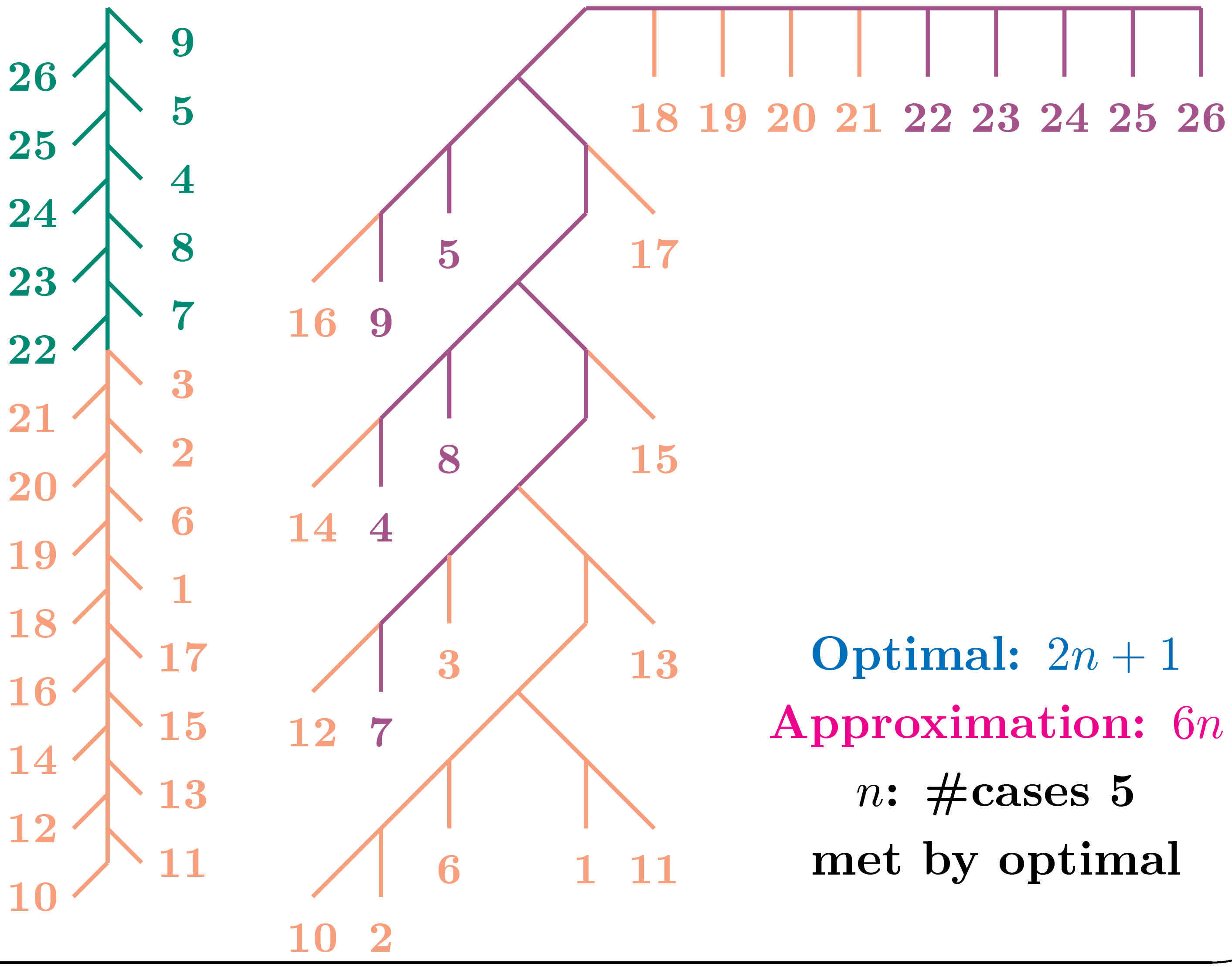
Optimal: $2n+1$

Approximation: $6n$

n**: #cases 5**

met by optimal

This is a tight ratio



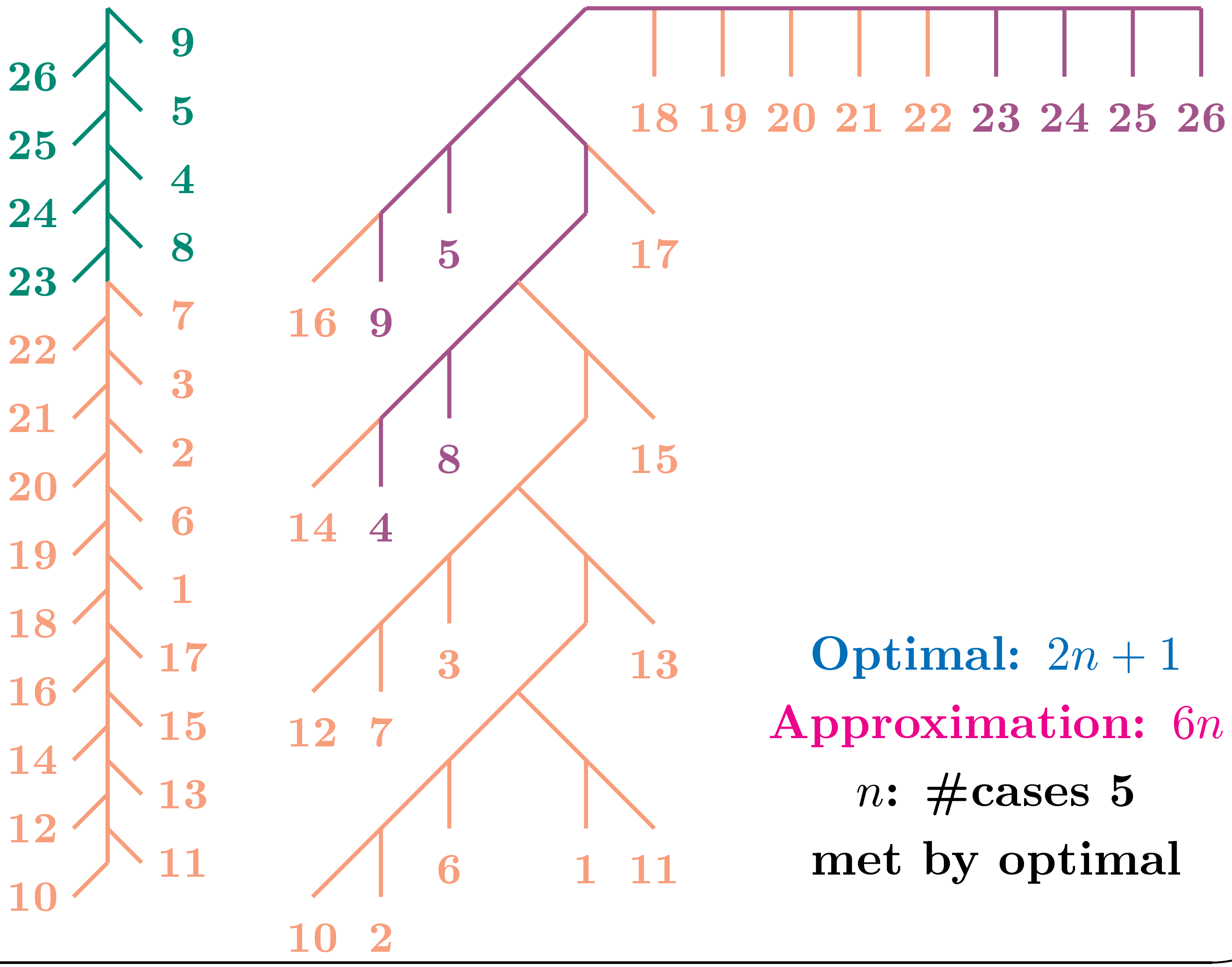
Optimal: $2n + 1$

Approximation: $6n$

n: **#cases 5**

met by optimal

This is a tight ratio



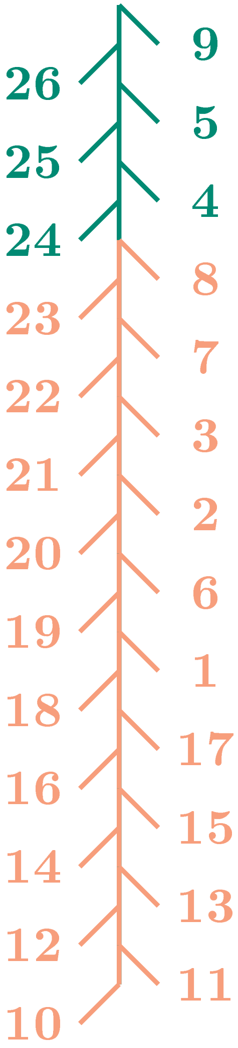
Optimal: $2n+1$

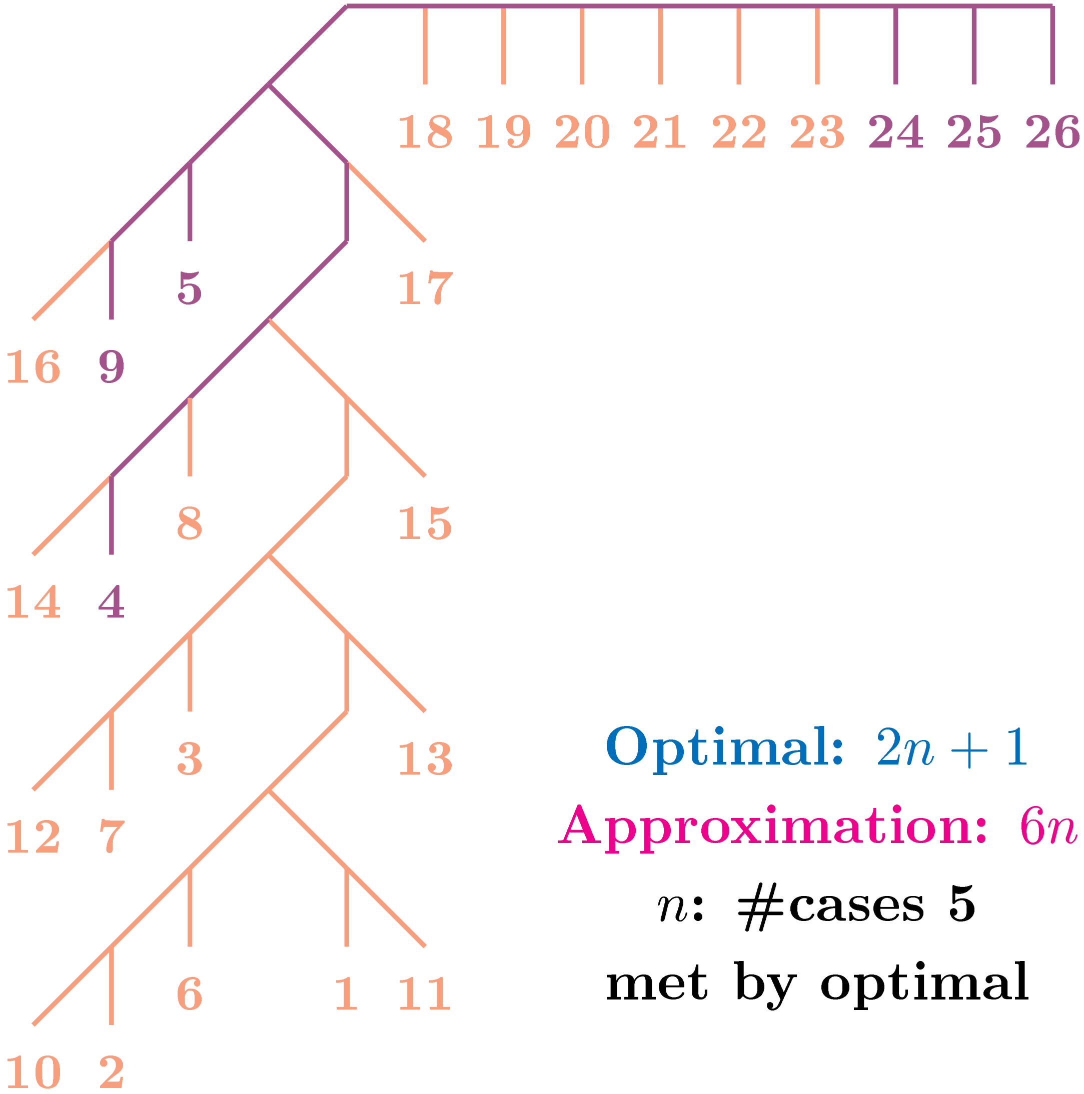
Approximation: $6n$

n**: #cases 5**

met by optimal

This is a tight ratio





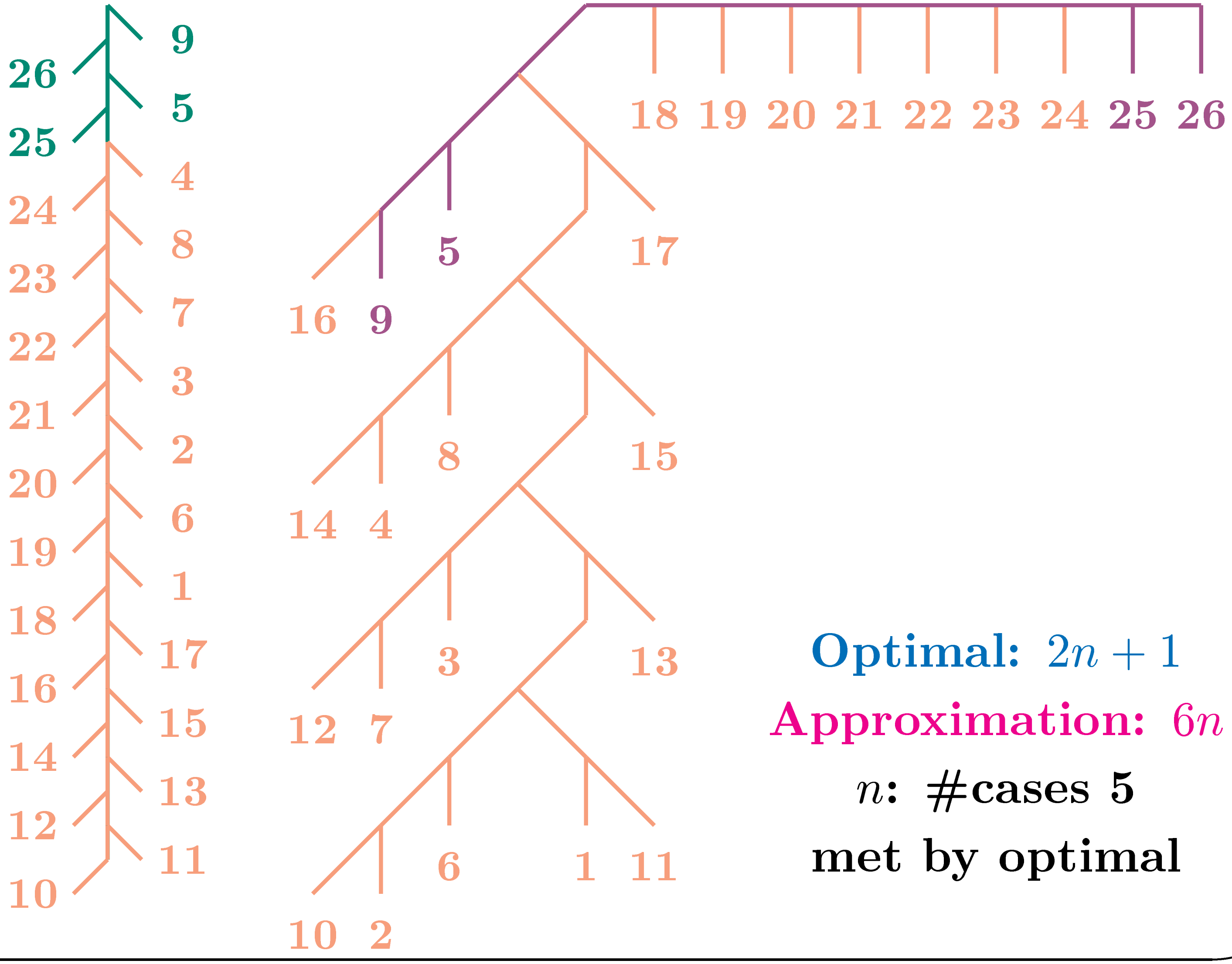
Optimal: $2n + 1$

Approximation: $6n$

n**: #cases 5**

met by optimal

This is a tight ratio



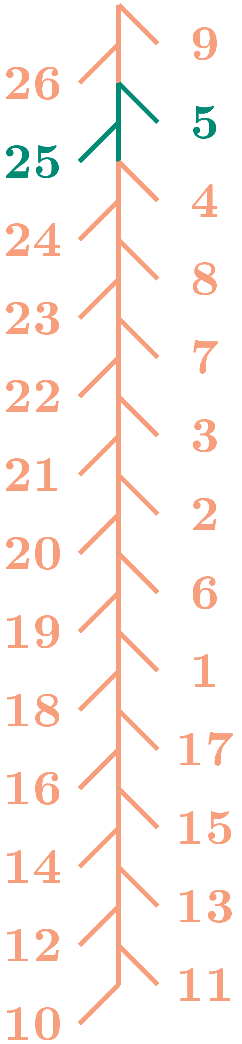
Optimal: $2n + 1$

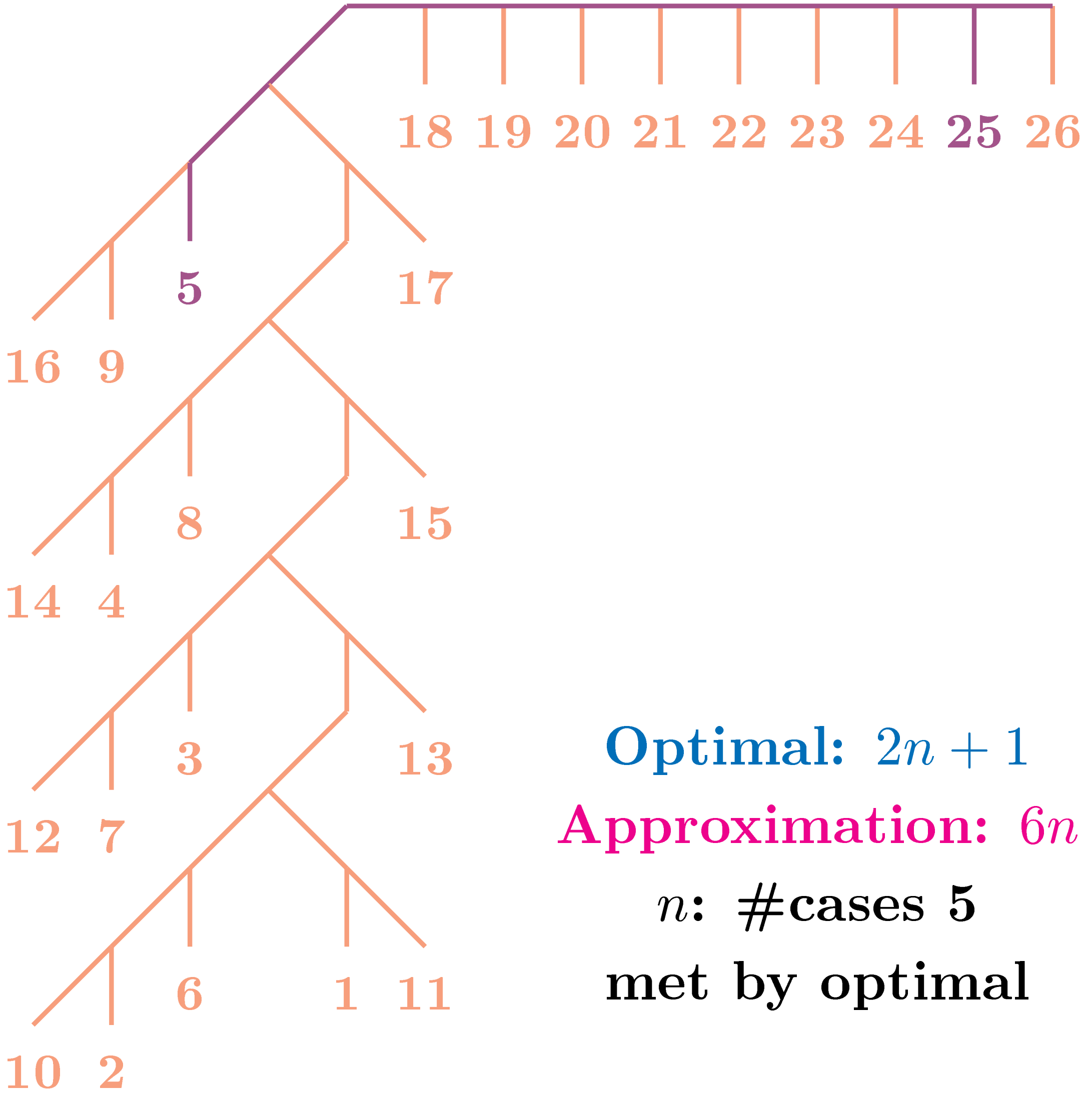
Approximation: $6n$

n**: #cases 5**

met by optimal

This is a tight ratio





Optimal: $2n+1$

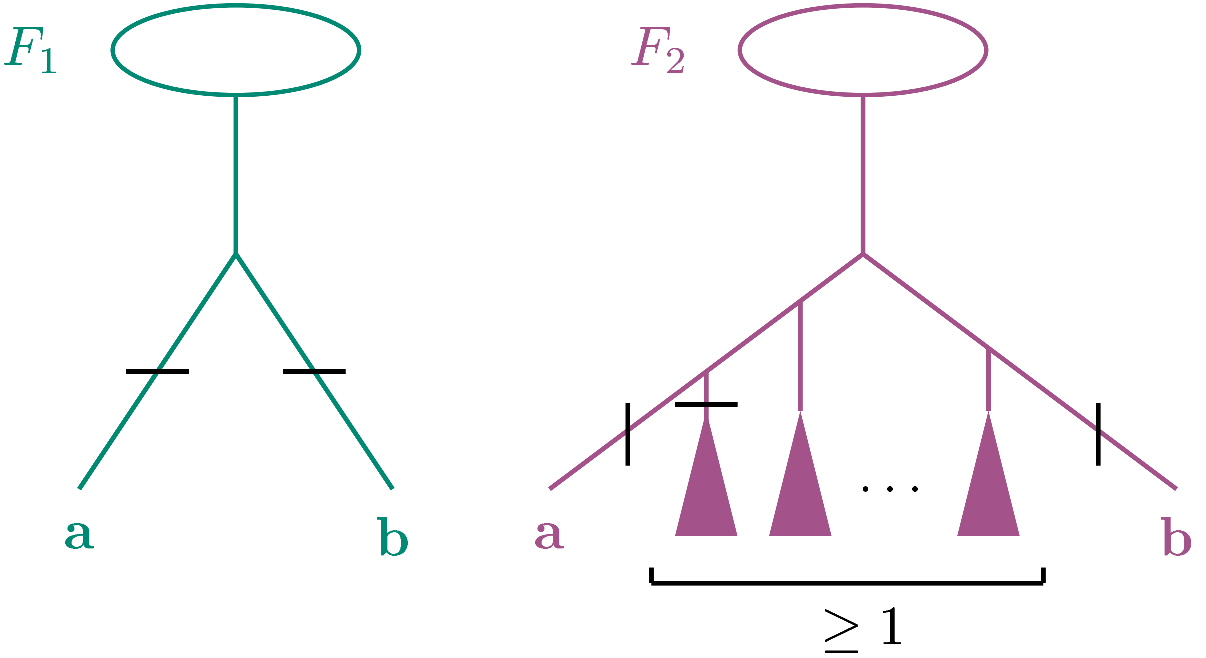
Approximation: $6n$

n**: #cases 5**

met by optimal

A simpler algorithm with the same ratio

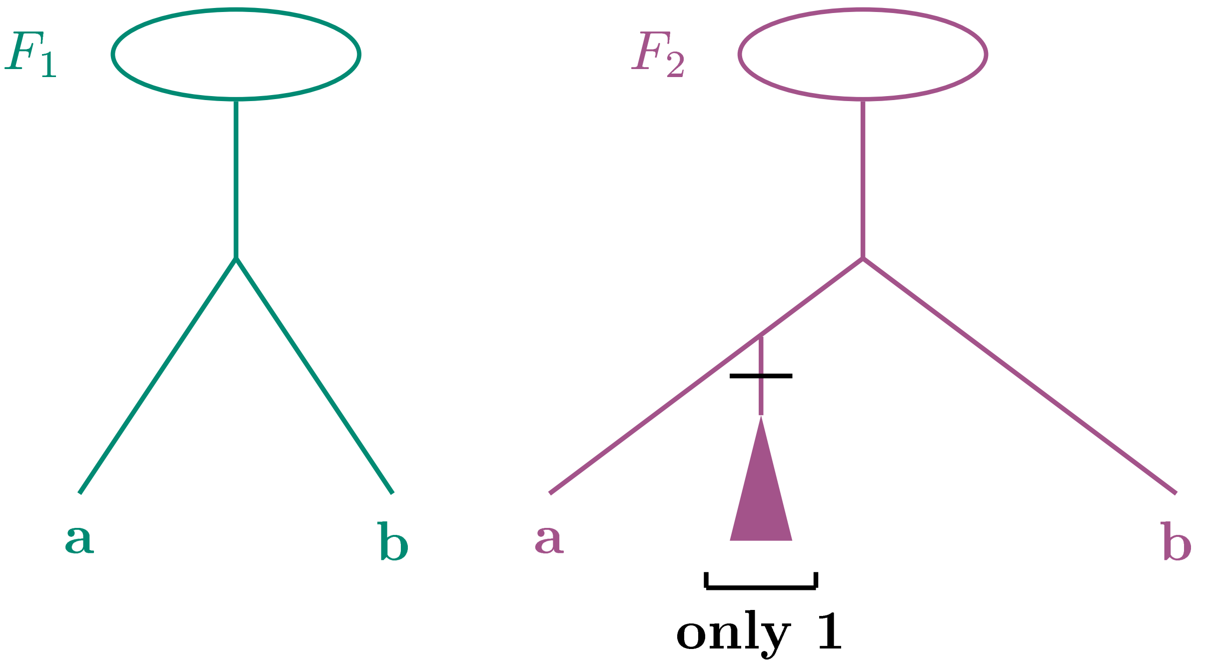
Cases 4 and 5:



cut the two edges in F_1 and F_2
leading to the leaves labelled a and b
plus one subtree (any one) hanging from the path from a to b

Why is Hein's a 4 instead of 3-approximation algorithm ?

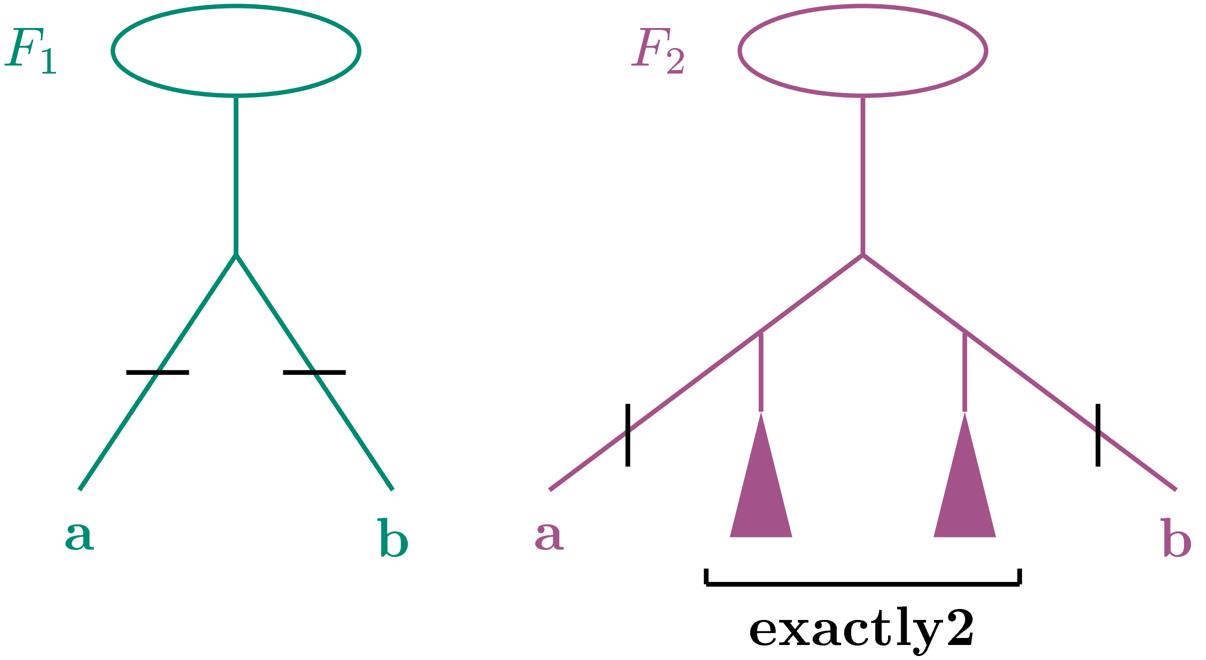
Case 4:



cut the subtree hanging from the path from a to b in F_2

The two more delicate cases

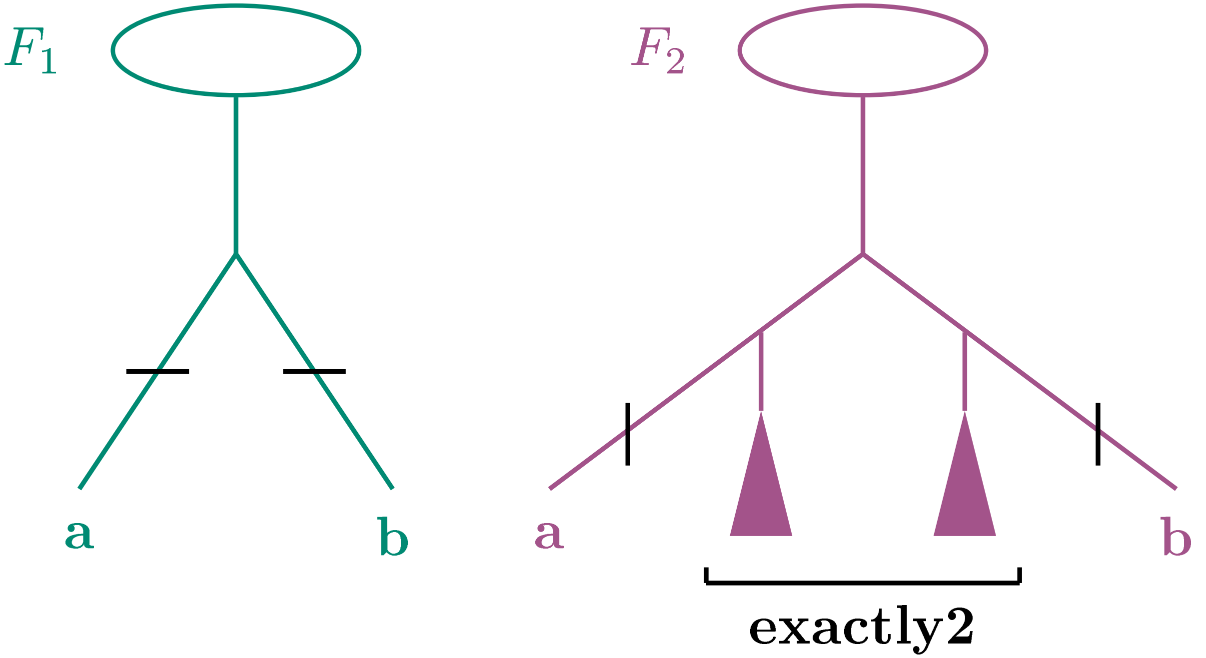
Case 4 which becomes part of case 5:



cut edges leading to the leaves labelled a and b and contract in F_1

What happens in this case

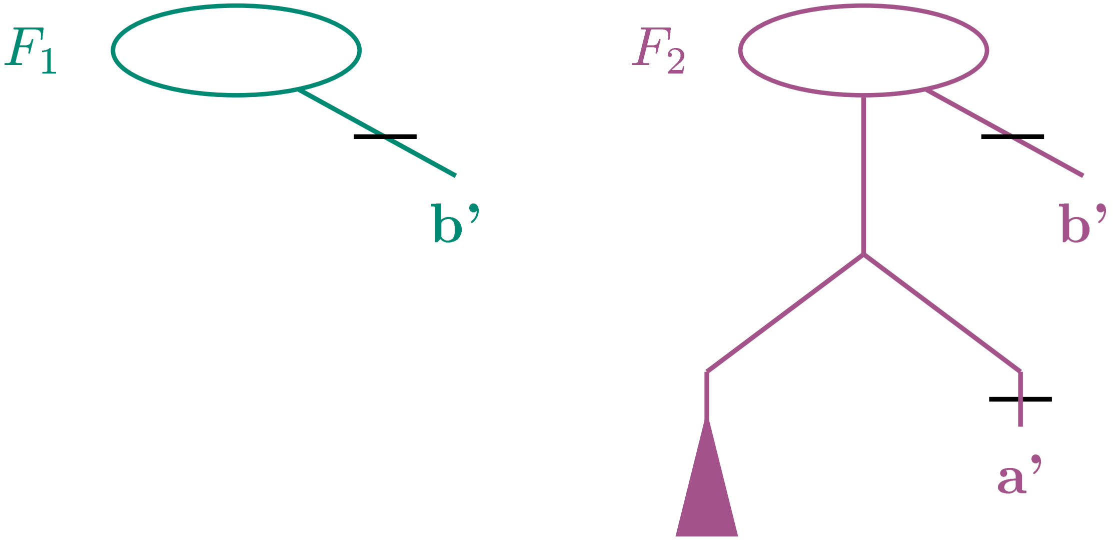
Case 5: **Iteration i**



possibly both are bad eliminations AND...

The two more delicate cases

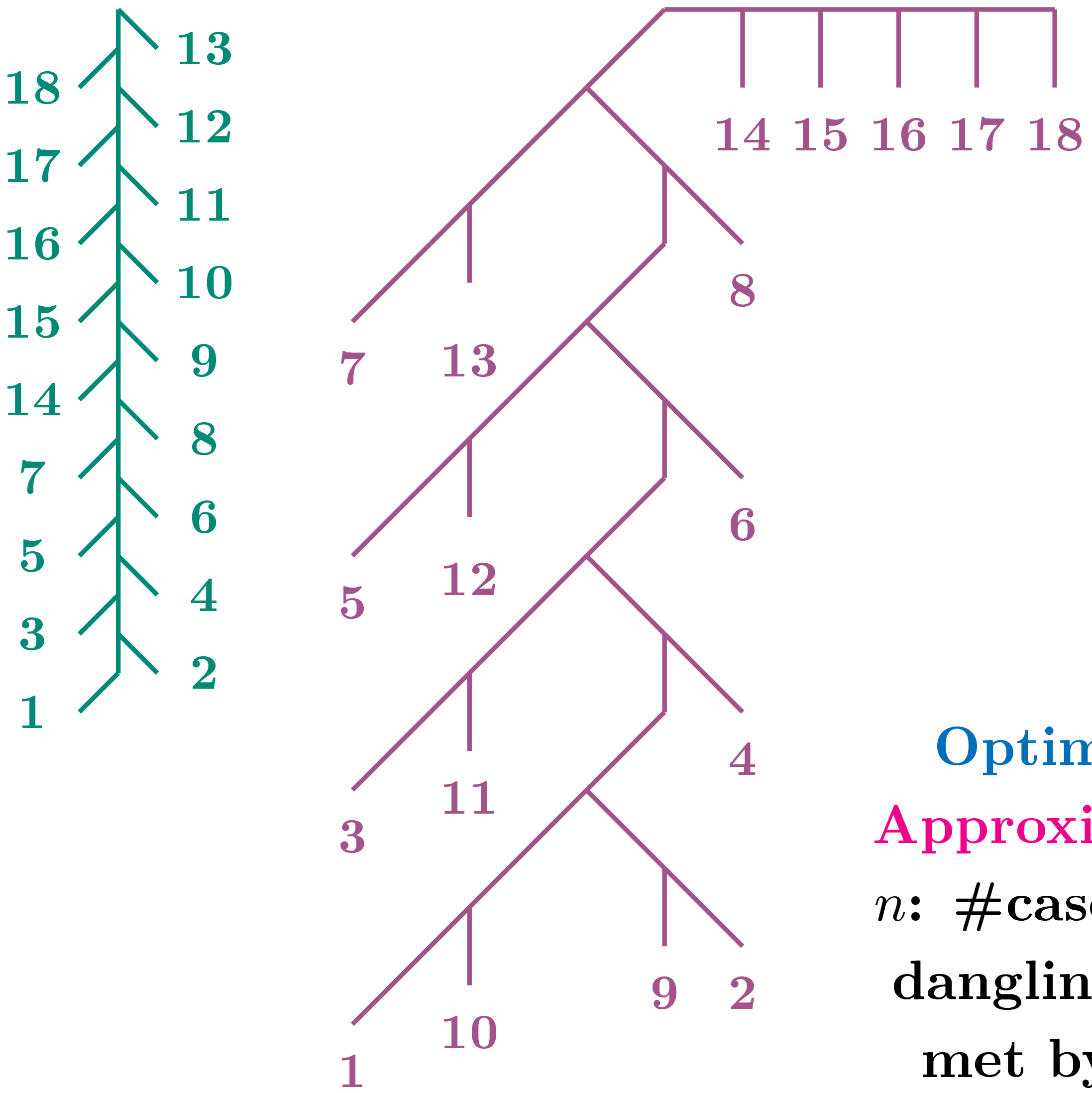
Case 5: **Iteration $i' > i$**



there is one good eliminination which compensates for the other, possibly bad elimination, in the same move and for one of the two bad eliminations at step i

but this may now be the only such possible good elimination

Again, Hein's ratio (of 4) is tight



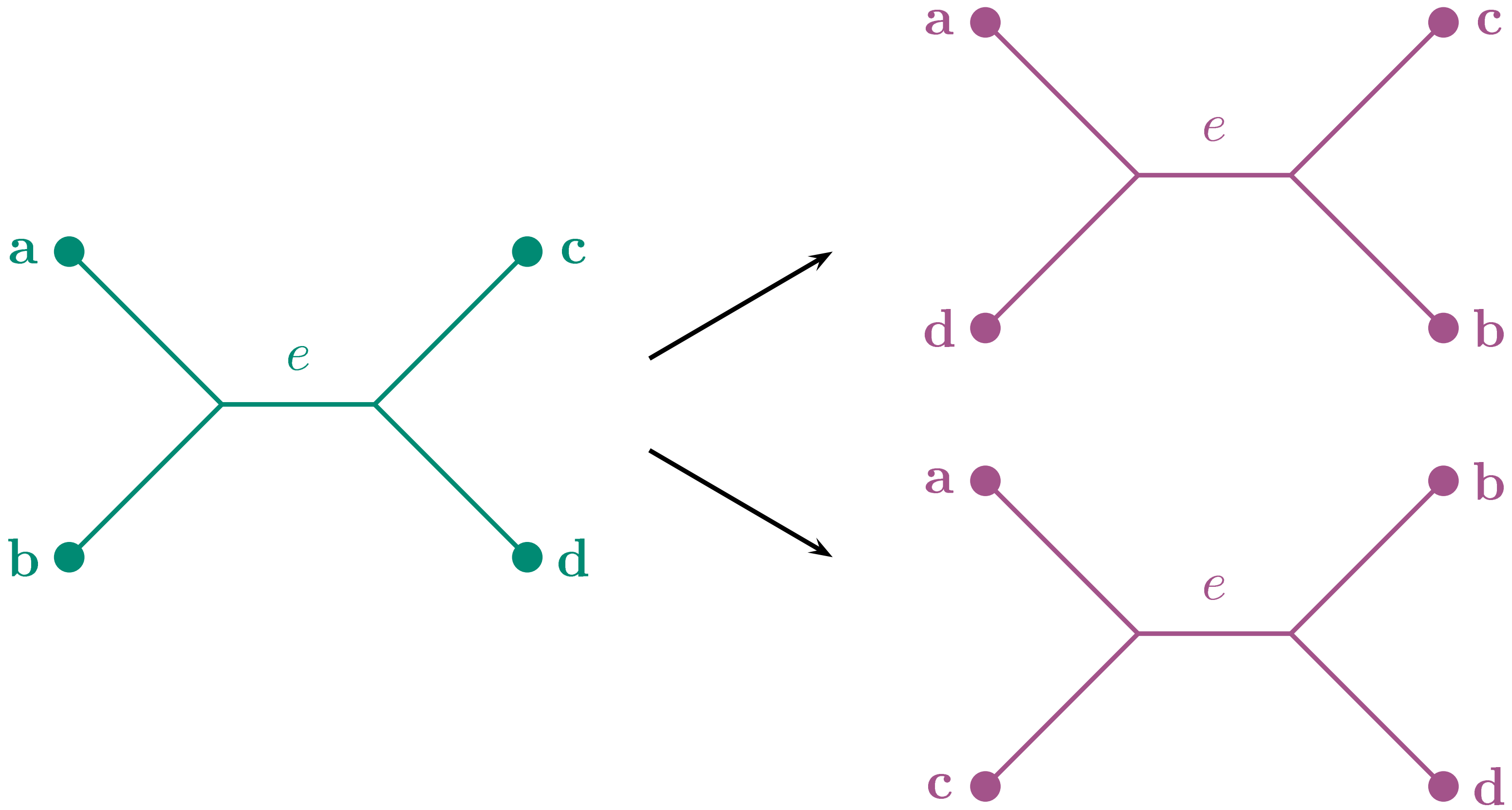
Optimal: $n+1$

Approximation: $4n$

n: **#cases 5 with 2 dangling subtrees met by optimal**

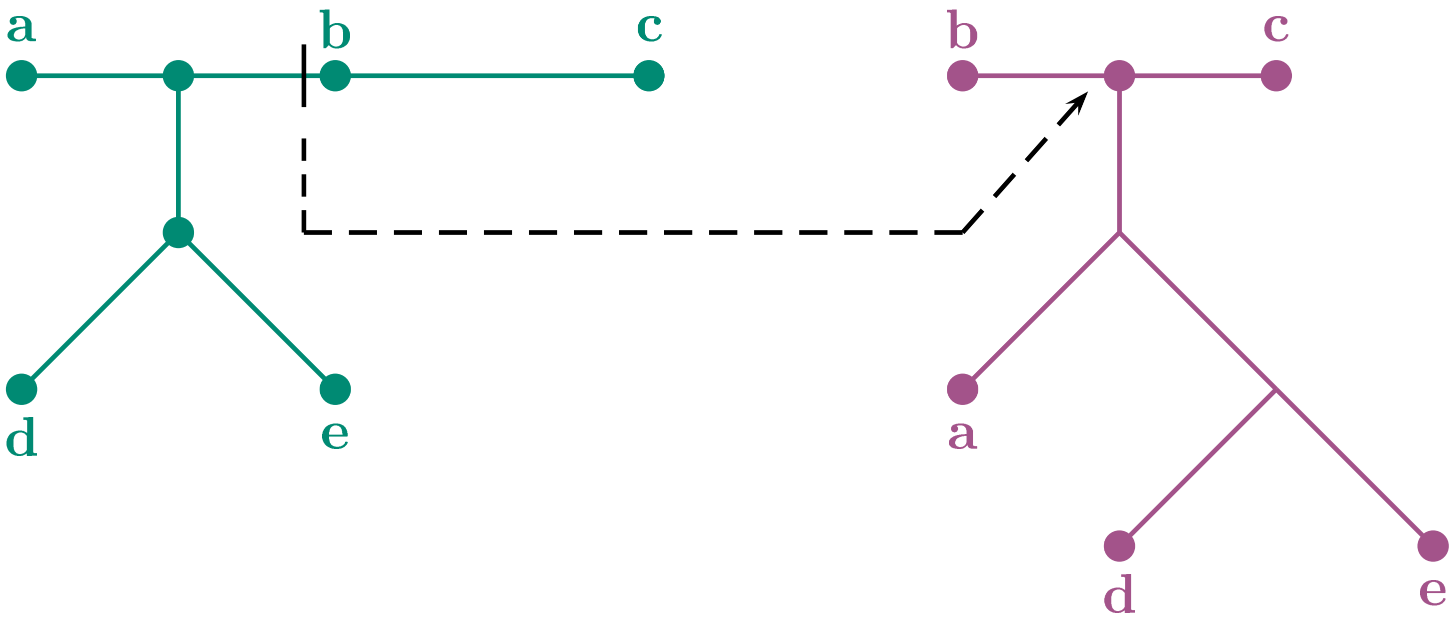
Other possible distances

Nearest Neighbour Interchange (NNI)



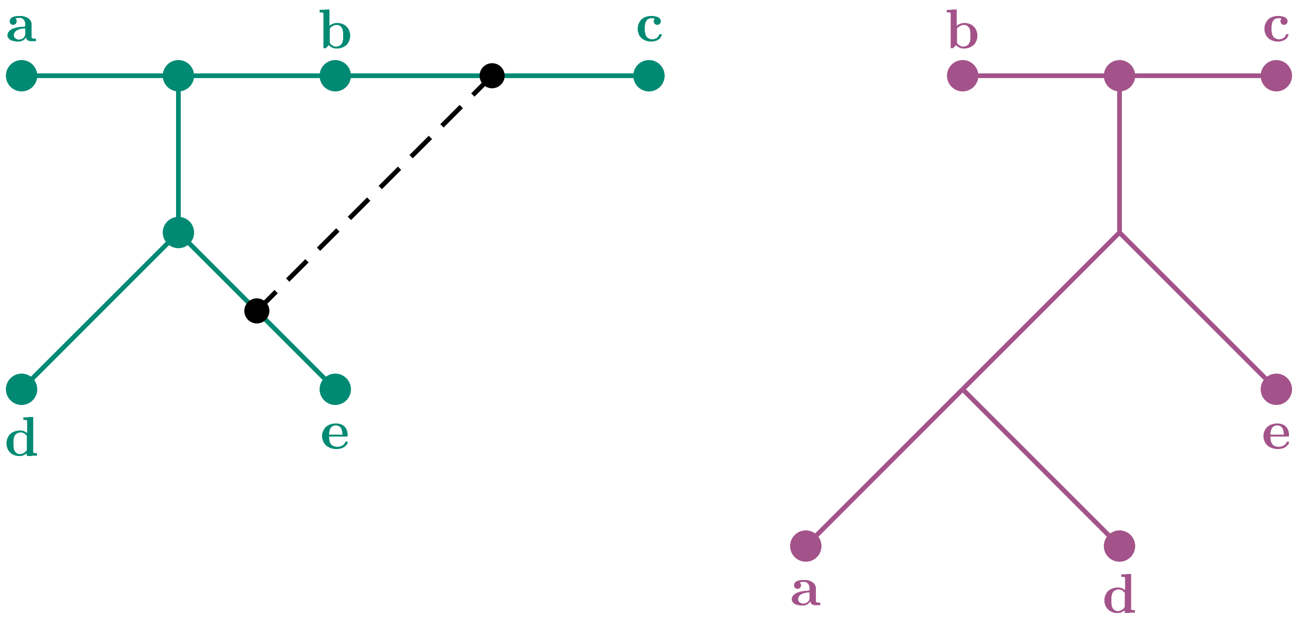
Other possible distances

Subtree Prune and Regraft (SPR)



Other possible distances

Tree Bisection and Reconnection (TBR)



a SPR operation is always a TBR but not conversely

Relation between the various distances

(Allen and Steel, 2001) For any two binary trees T_1 and T_2 labelled at the leaves:

$$D_{TBR}(T_1, T_2) \leq D_{SPR}(T_1, T_2) \leq D_{NNI}(T_1, T_2)$$

$$D_{SPR}(T_1, T_2) \leq 2 \times D_{TBR}(T_1, T_2)$$

$$MAF(T_1, T_2) - 1 = D_{TBR}(T_1, T_2)$$

Our conjecture: $D_{SPR}(T_1, T_2) = \ MAF(T_1, T_2)$

or $MAF(T_1, T_2) - 1$

Some of our perspectives concerning this work

Prove the conjecture and improve the bounds

Obtain a better than 3-approximation for the MAF

Explore relation between MAF and MAST

Examine the "fixed parameter tractability" of each distance (already shown for TBR by Allen and Steel)

> **Ex: When the maximum distance between the two trees is (small) d, can we devise an algorithm whose complexity depends on d and only linearly on the number of leaves ?**

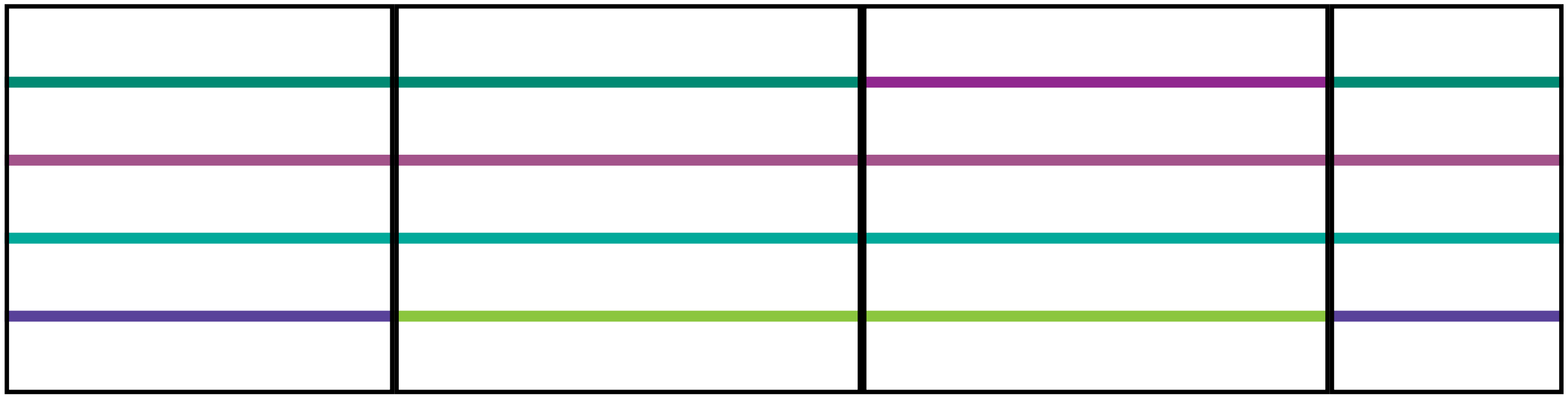
Examine further what is (perhaps ?) the most interesting genetic transfer distance: weighted SPR

The most difficult ? go back to where we came from

Genetic transfer detection and localization as an optimisation problem

The (ideal ?) result

alignement



tree 1 **tree 2** **tree 3** **tree 4**

“optimal” partition: sum of lengths of trees + penalty for genetic transfers is MINIMUM