Dichotomies for Tree Minor Containment with Structural Parameters

Tatsuya Gima[1] **Soh Kumabe**[2]

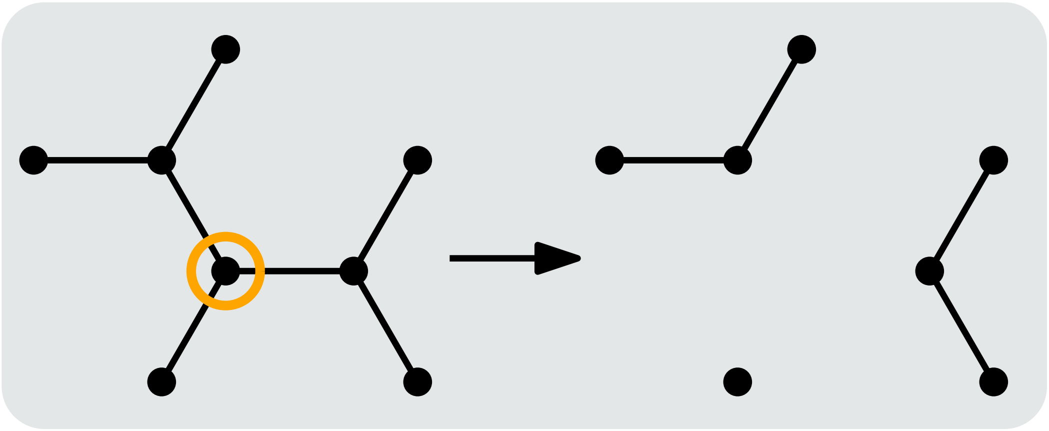
Kazuhiro Kurita[1] **Yuto Okada**[1] **Yota Otachi**[1]

[1]Nagoya University [2]The University of Tokyo

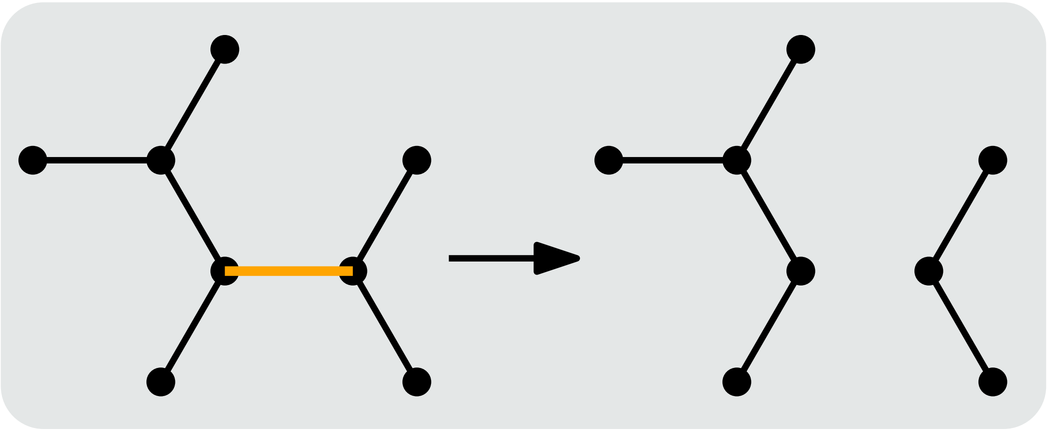
2024-03-20 @ WALCOM2024

Introduction: What is Graph Minor?

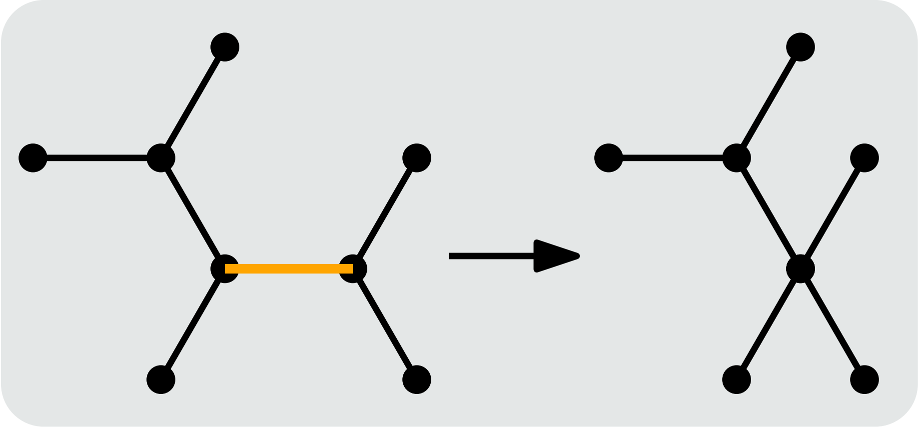
A graph G contains a graph H as a minor
If we can obtain H by repeating these operations to G.



(1) Vertex Deletion



(2) Edge Deletion



(3) Edge Contraction (merging the end vertices)

Abstract

In this talk, we consider the following problem.

Tree Minor Containment (TMC)

Input: Trees T, P.
Question: Does T contain P as a minor?

This problem is known to be NP-complete.

What if we bound the following parameters of T, P?

- diameter
- path eccentricity
- pathwidth

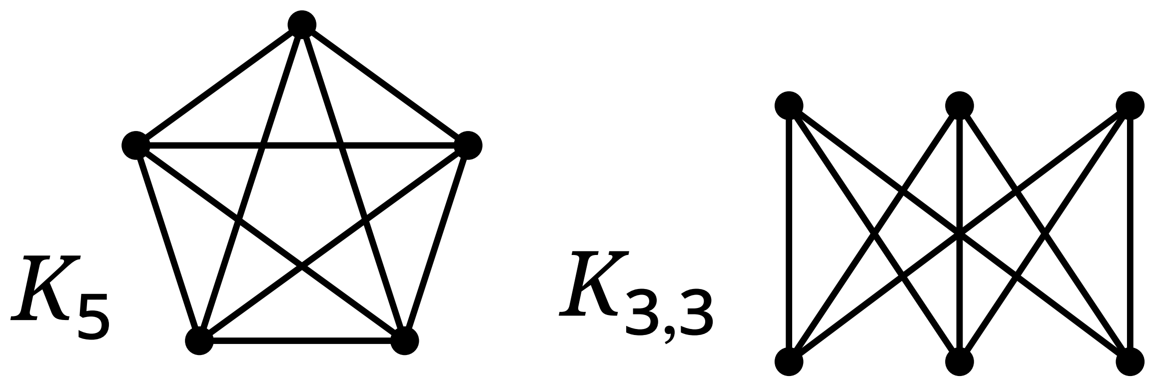
For each parameter, there exists a tractability border from which the problem becomes NP-complete.

We give precise borders for the parameters.

Motivations: The Importance of Problem

Graph minor has been a main topic in graph theory.

Kuratowski (1930): G is planar ⇔
G does not contain $K_5, K_{3,3}$ as **topological minor**.



Wagner (1937): G is planar ⇔
G does not contain $K_5, K_{3,3}$ as **minor**.

≠ generalization

Robertson & Seymor: (1983〜2004) **Graph Minor Theorem**

Minor Containment is also an important problem.
It is worth studying the tractability borders.

Motivations: Contrast with Similar Problems

There are some “containment” problems.

Subgraph Isomorphism

Does G contain H as a **subgraph**?

Topological Minor Containment

Does G contain H as a **topological minor**?

Minor Containment

Does G contain H as a **minor**?

The above problems are all NP-complete.
But on trees they are tractable, except minor.

- How much restrictions to make it tractable?
- What makes Minor Containment so difficult?

Abstract

For those motivations,
we use three structural parameters and fill these table.

Diameter

P \ T	3	4	5	6	7	8
3						
4						
5						
6						
7						
8						

Path Eccentricity

P \ T	1	2	3
1			
2			
3			

Pathwidth

P \ T	1	2	3
1			
2			
3			

Trivial No   P   NP-Complete

Known Results: Degree and Diameter

TMC is FPT w.r.t. d (the **maximum degree** of P).

Kilpeläinen and Mannila, 1995

TMC can be solved in $O(4^d \cdot \text{poly}(|T| + |P|))$ time.

Akutsu et al., 2021

TMC can be solved in $O(2^d \cdot \text{poly}(|T| + |P|))$ time.

On the other hand, TMC remains hard
even if the **diameters** are bounded by a constant.

Matoušek and Thomas, 1992

TMC is NP-complete even if $\text{diam}(T), \text{diam}(P) \leq 8$.

Known Results: Caterpillar

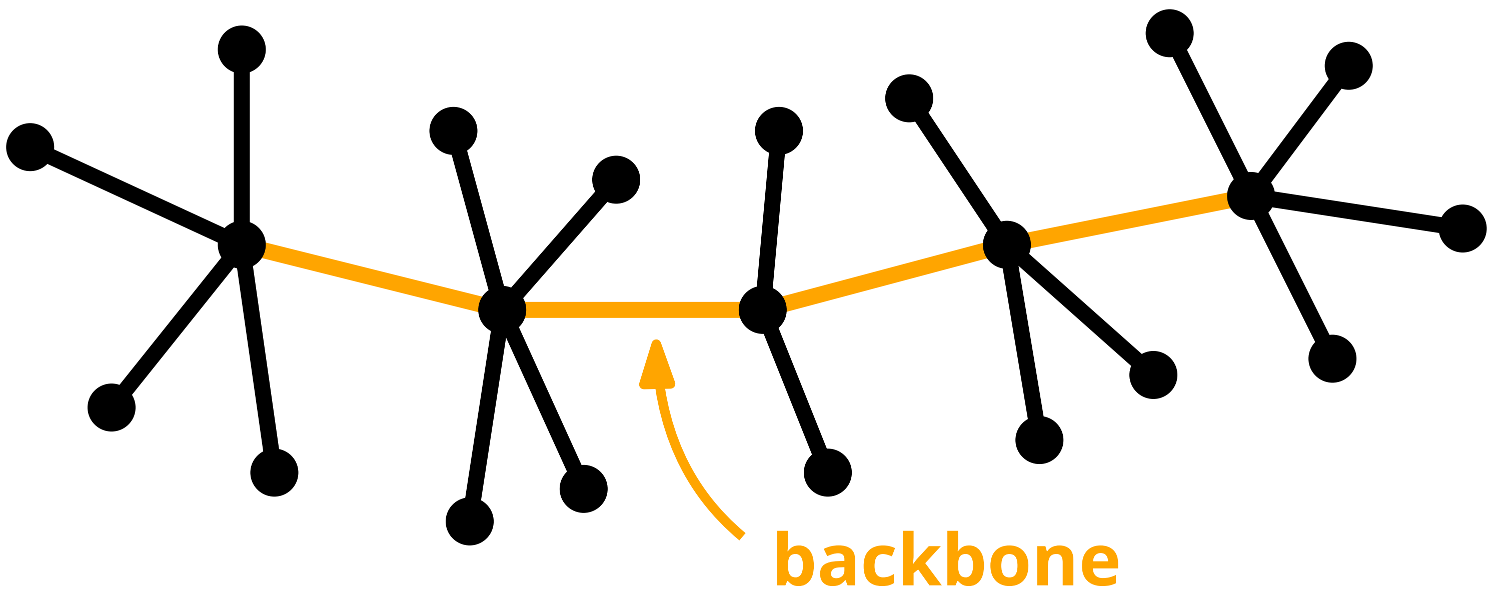
TMC is tractable on **caterpillars**.

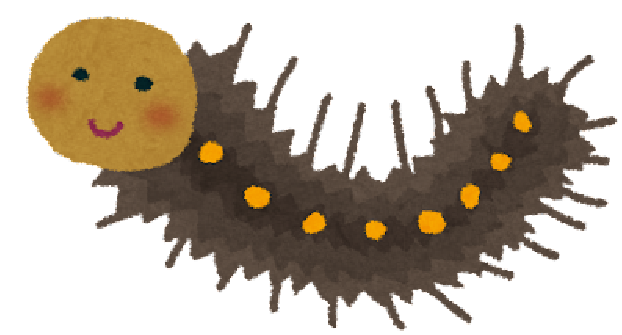
Gupta et al., 2005

If T, P be caterpillars, TMC can be solved in polynomial time.

caterpillar

There is a path s.t. the distance from every vertex is at most 1.





We extend this result with two parameters.

Generalization of Caterpillar

We generalize caterpillar with these parameters.

- **Path Eccentricity** $\mathrm{pe}(T)$

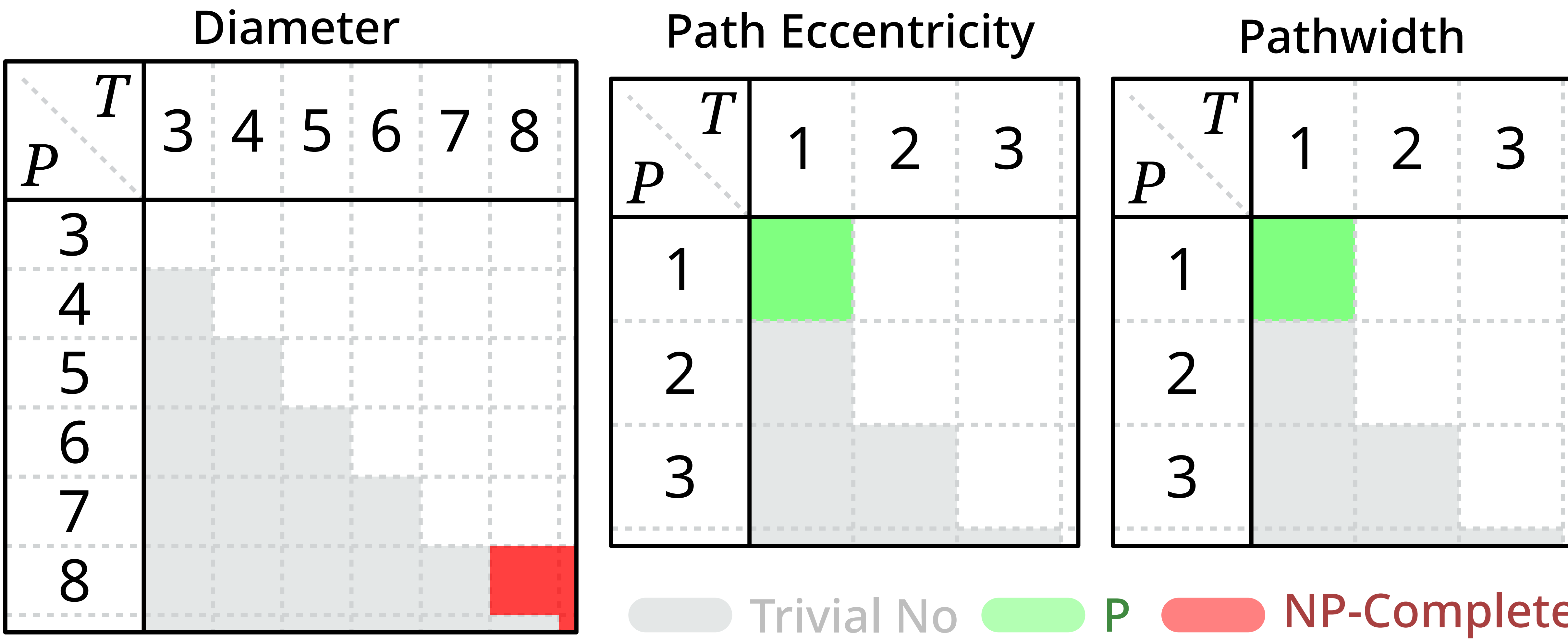
  The minimum integer k such that
  there is a path s.t. the distance from every vertex is at most k.

- **Pathwidth** $\mathrm{pw}(T)$

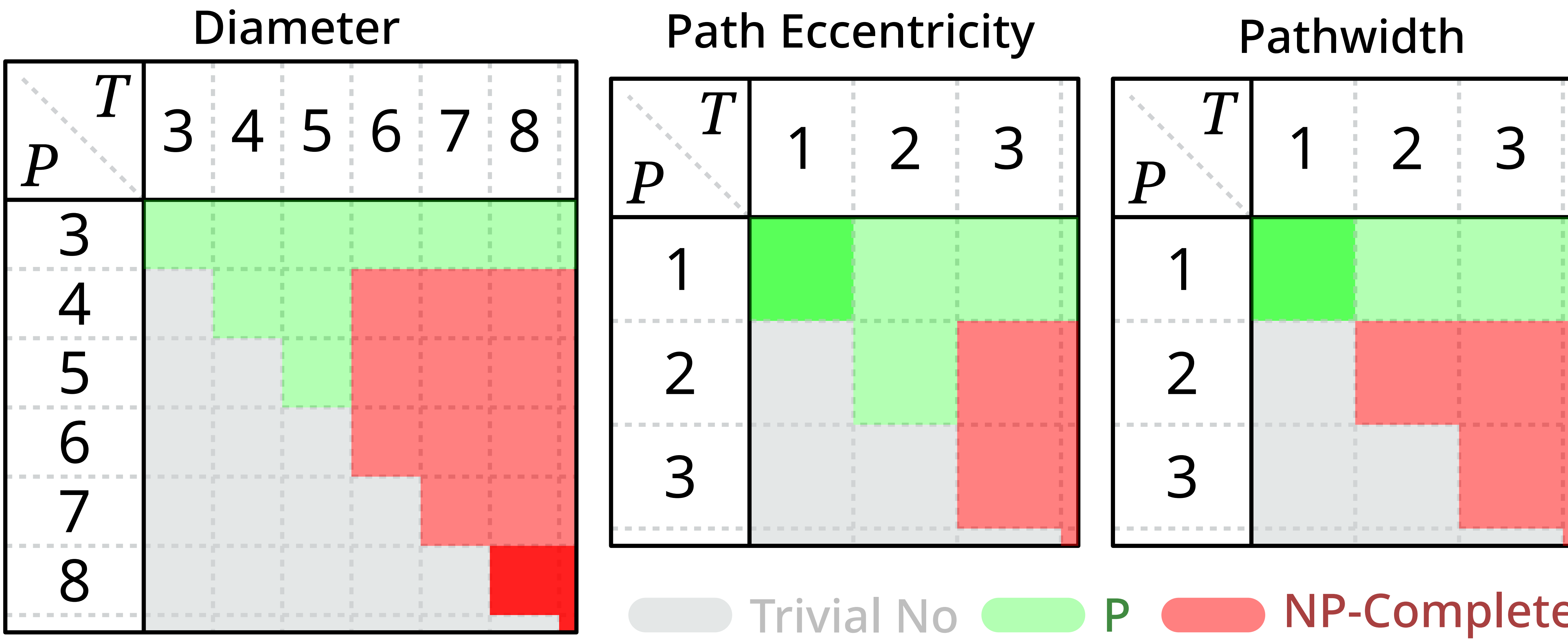
  Path version of treewidth (details omitted).

It is known that
caterpillar $\Leftrightarrow$ path eccentricity ≤ 1 $\Leftrightarrow$ pathwidth ≤ 1.

Our Results


Diameter
T
P
3 4 5 6 7 8
3
4
5
6
7
8
Path Eccentricity
T
P
1 2 3
1
2
3
Pathwidth
T
P
1 2 3
1
2
3
Trivial No
P
NP-Complete

Our Results



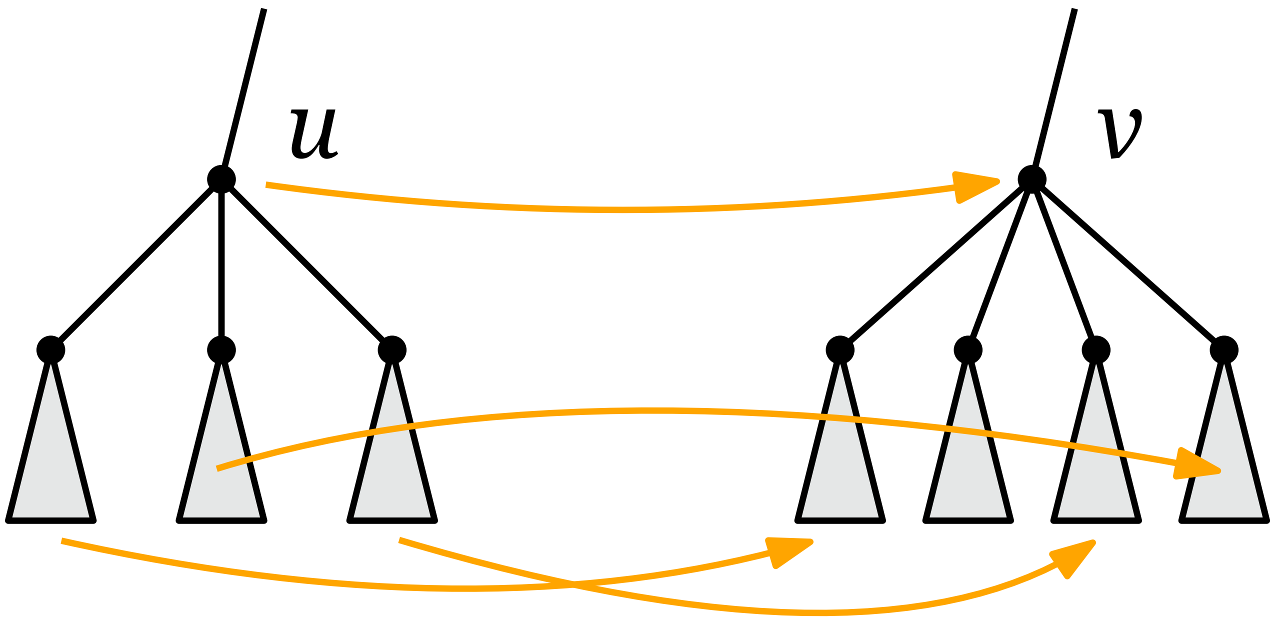
Our Contribution

- NP-completeness with best possible combinations.
- Polynomial-time algorithms for the remaining.

Dynamic Programming on Other Problems

We can solve these problems on trees by DP.

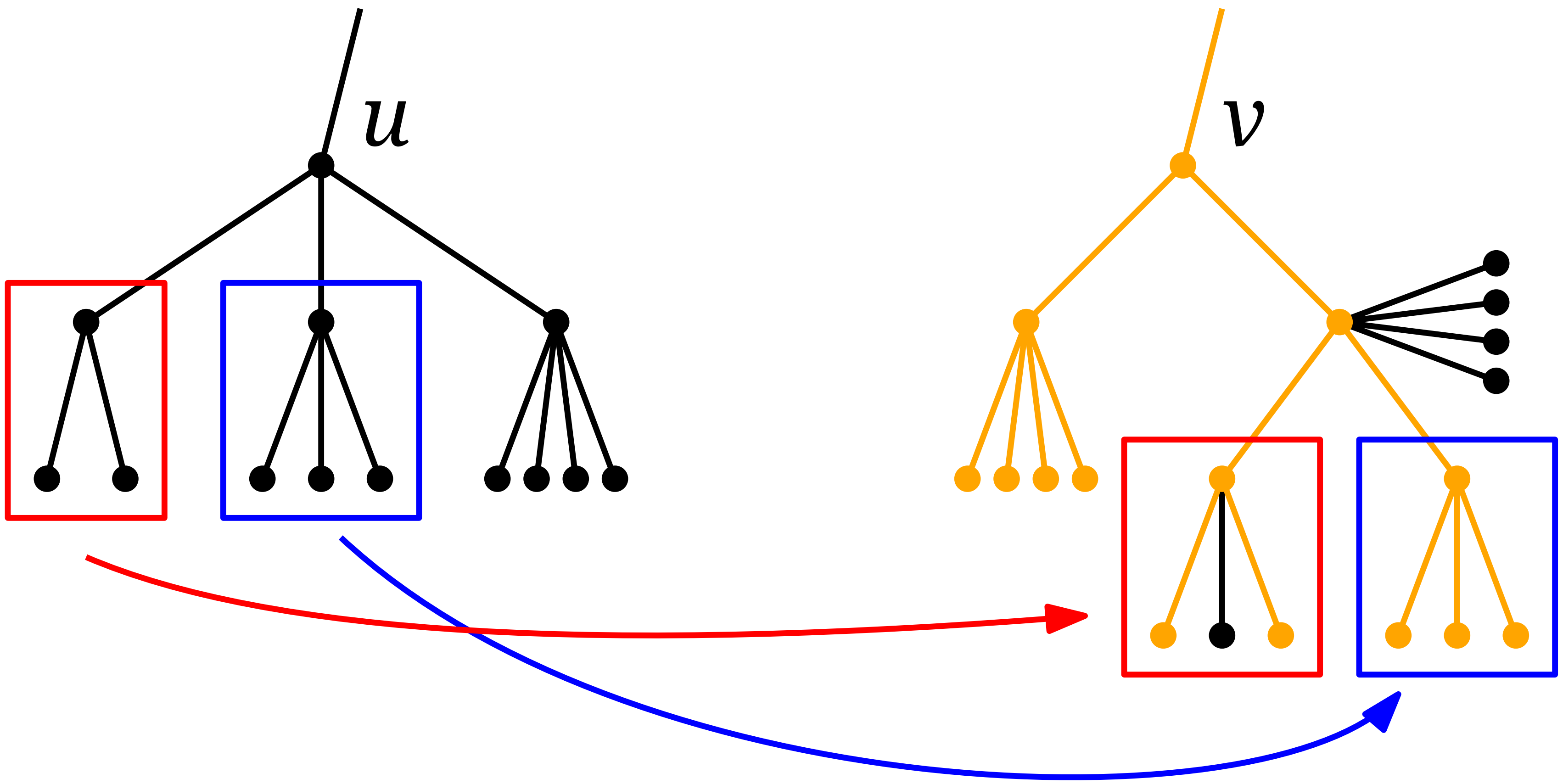
- Subgraph Isomorphism
- Topological Minor Containment



1. Assume T, P are rooted.
2. For each $u \in P$ and $v \in T$, Compute if we can "embed" subtree rooted at u into subtree rooted at v.
3. Bipartite matching between the subtrees.

DP fails on Minor Containment

On Minor Containment,
we can embed multiple subtrees in one subtree!



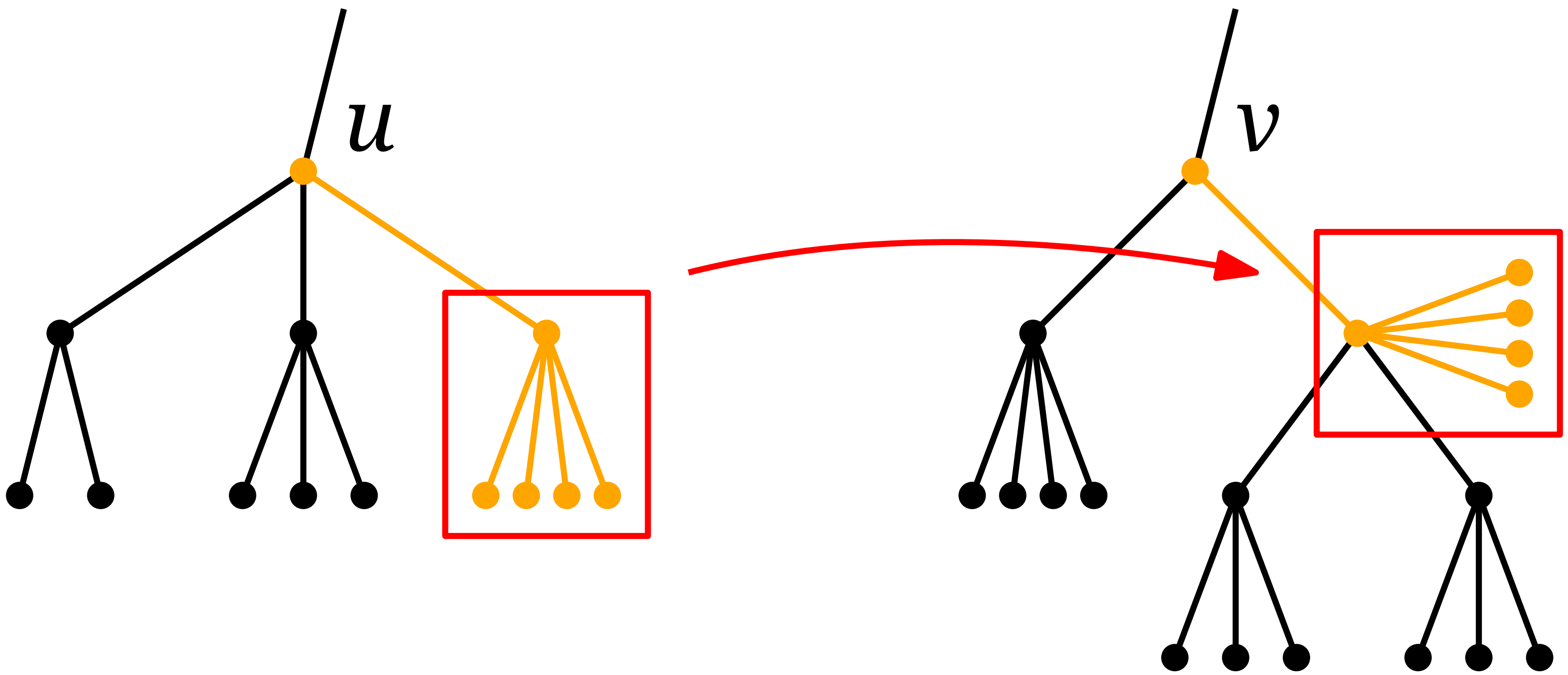
This makes things complicated:

- Guess the grouping of subtrees.
- Embed each group into one subtree.

Another Observation

Observation

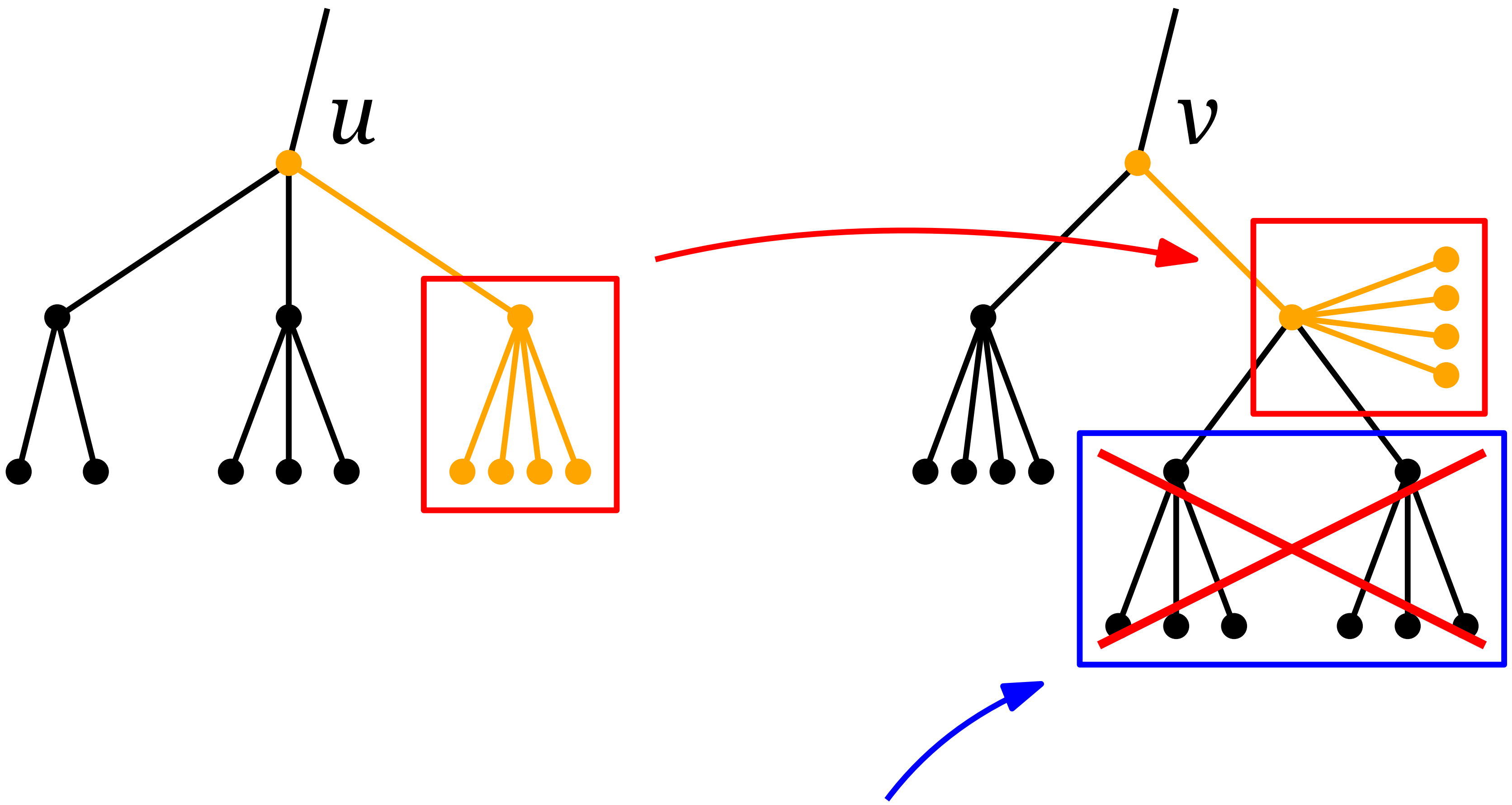
On this configuration, we can not embed like this.



Another Observation

Observation

On this configuration, we can not embed like this.

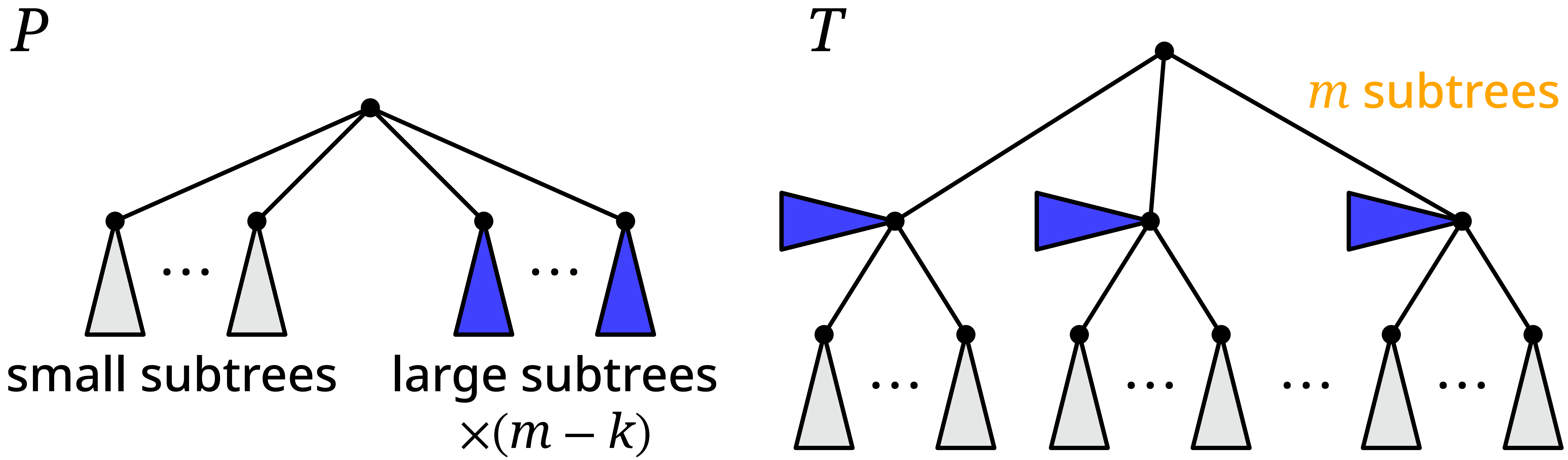


If we do this, we can't use this parts anymore.

We use these structures to show the NP-completeness.

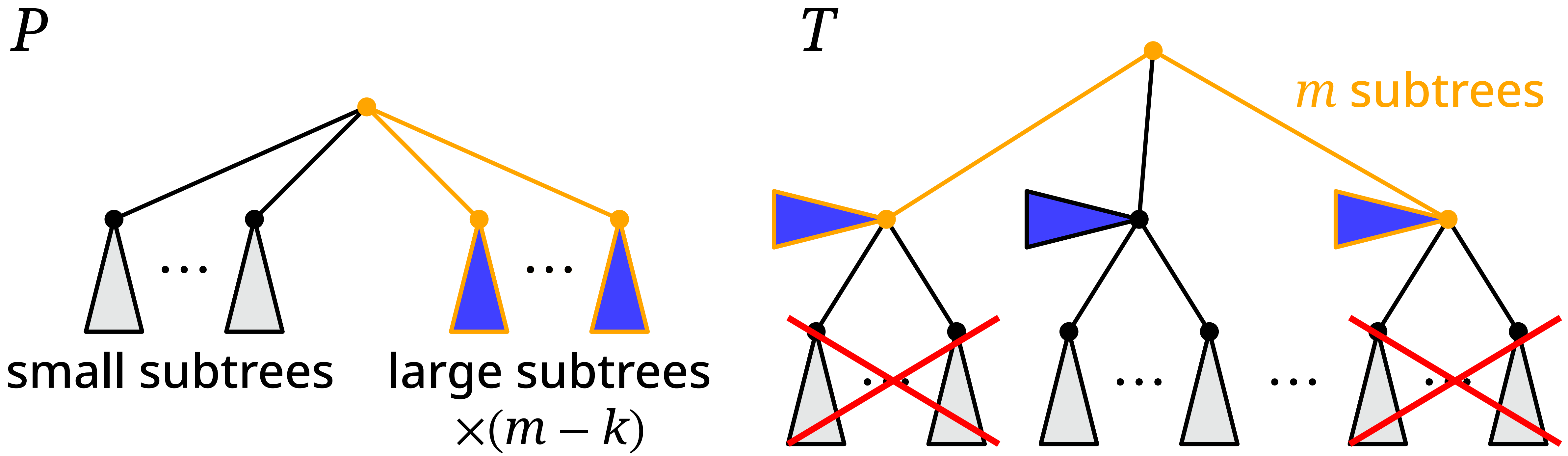
Sketch of the Reduction

We create an instance like this:



Sketch of the Reduction

We create an instance like this:

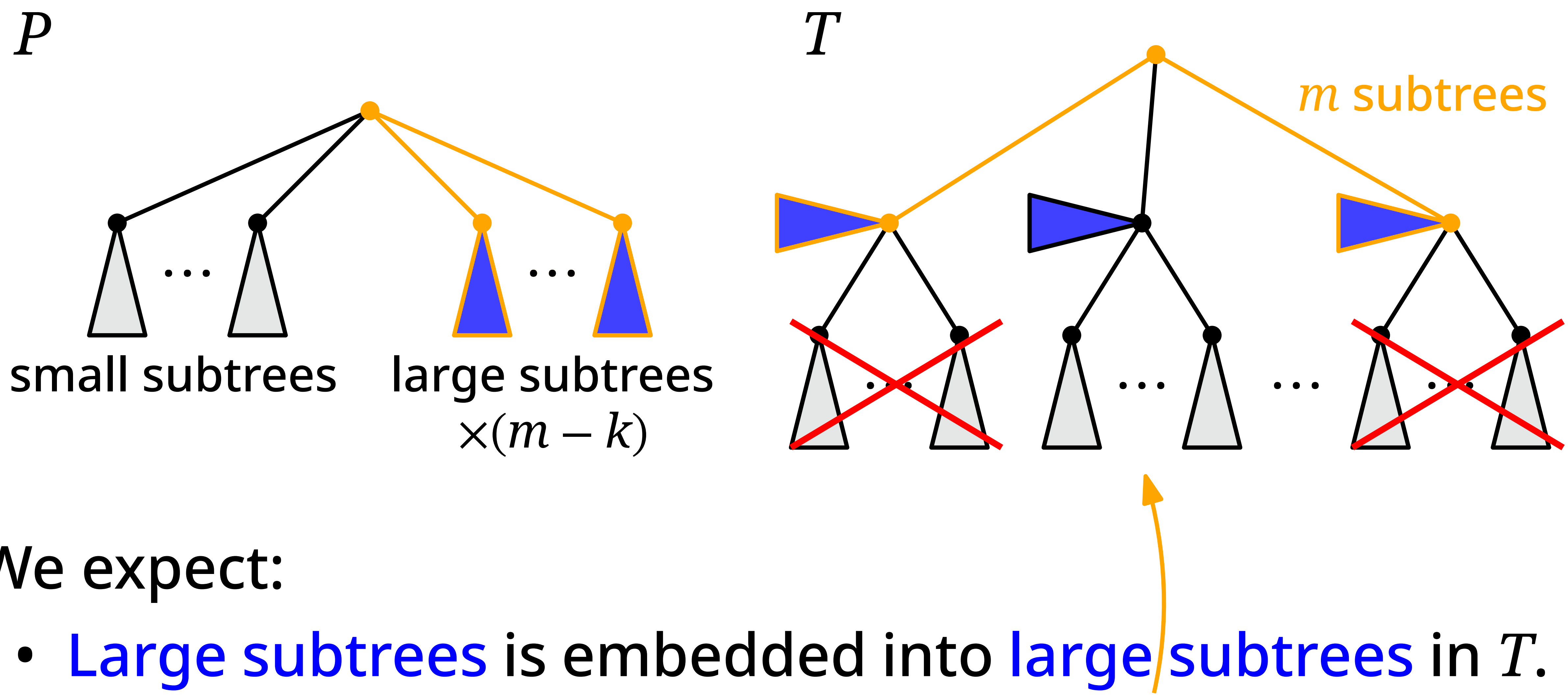


We expect:

- Large subtrees is embedded into large subtrees in T.

Sketch of the Reduction

We create an instance like this:

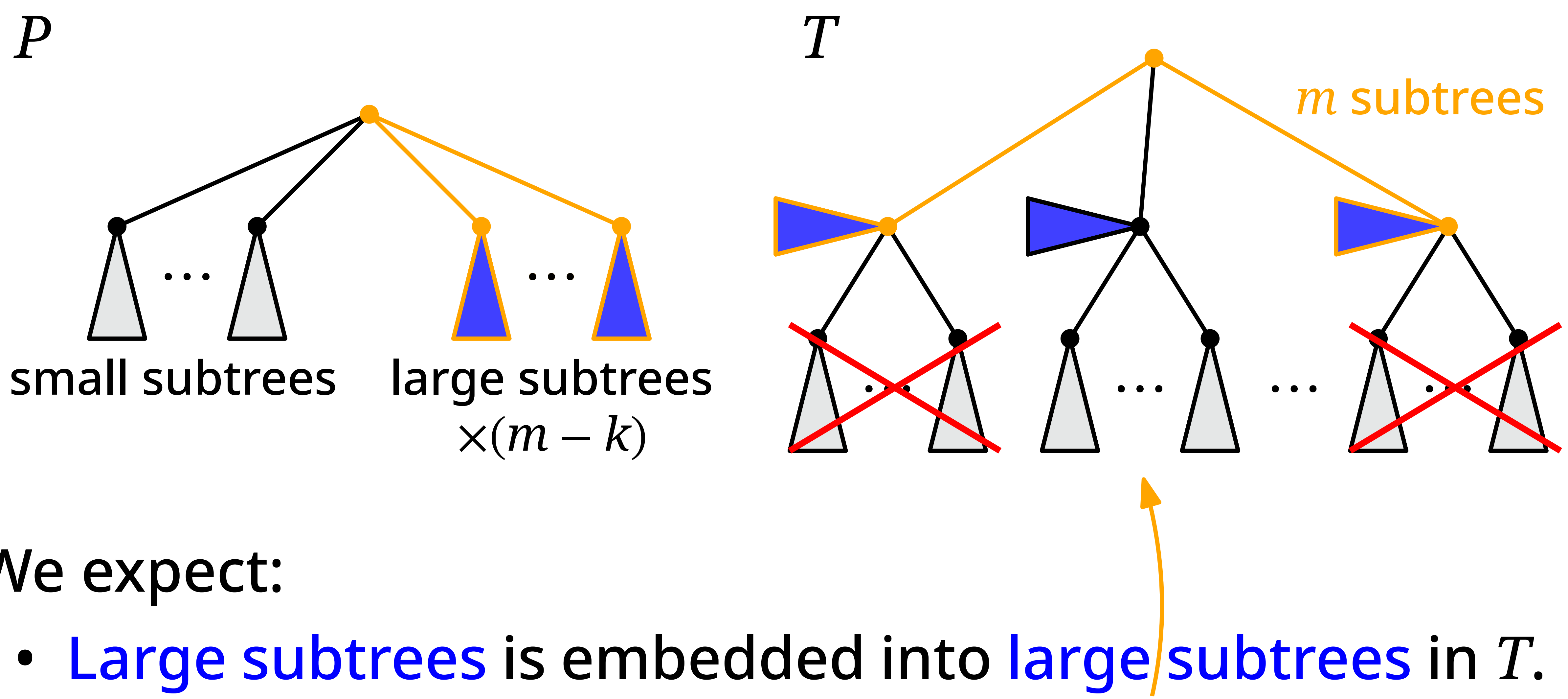


We expect:

- Large subtrees is embedded into large subtrees in T.
- small subtrees is embedded into k residual subtrees.

Sketch of the Reduction

We create an instance like this:



We expect:

- Large subtrees is embedded into large subtrees in T.
- small subtrees is embedded into k residual subtrees.

Choose k subtrees to cover small small subtrees.
$\rightarrow$ Seems like **Set Cover**.

Inclusive Set Cover

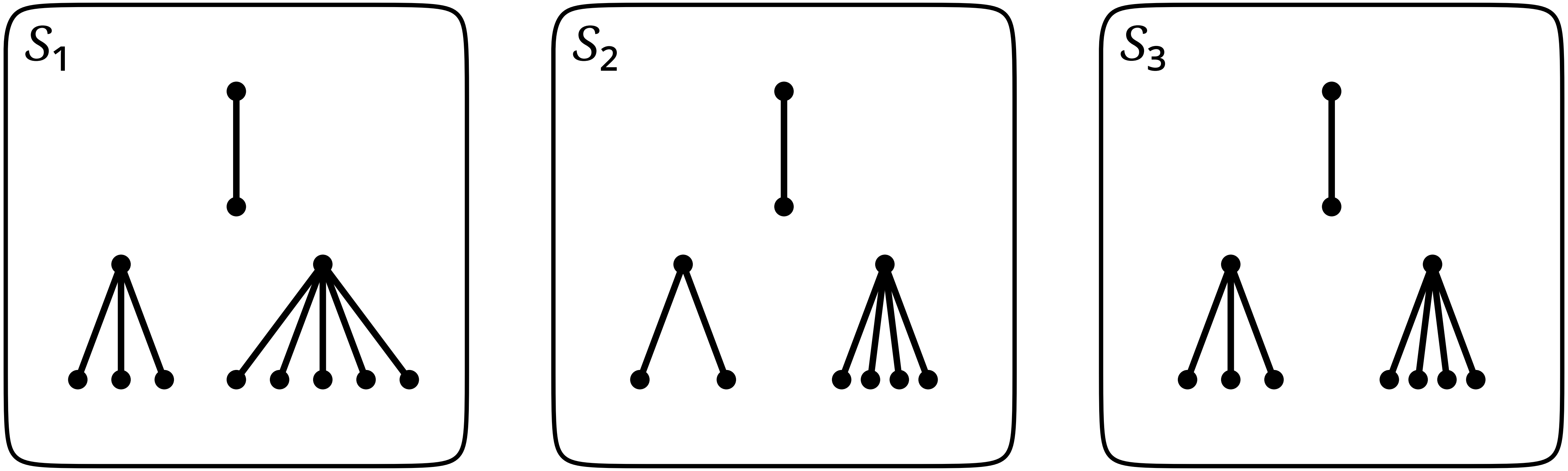
We first show that this problem is NP-complete.

Inclusive Set Cover

Input: sets of stars $S_1, S_2, \ldots, S_m$, an integer k.

Question: Can we pick k sets from $S_1, S_2, \ldots, S_m$ so that we can embed stars with $1, 2, \ldots n$ leaves into them.

example) $n = 5, m = 3, k = 2$



Inclusive Set Cover

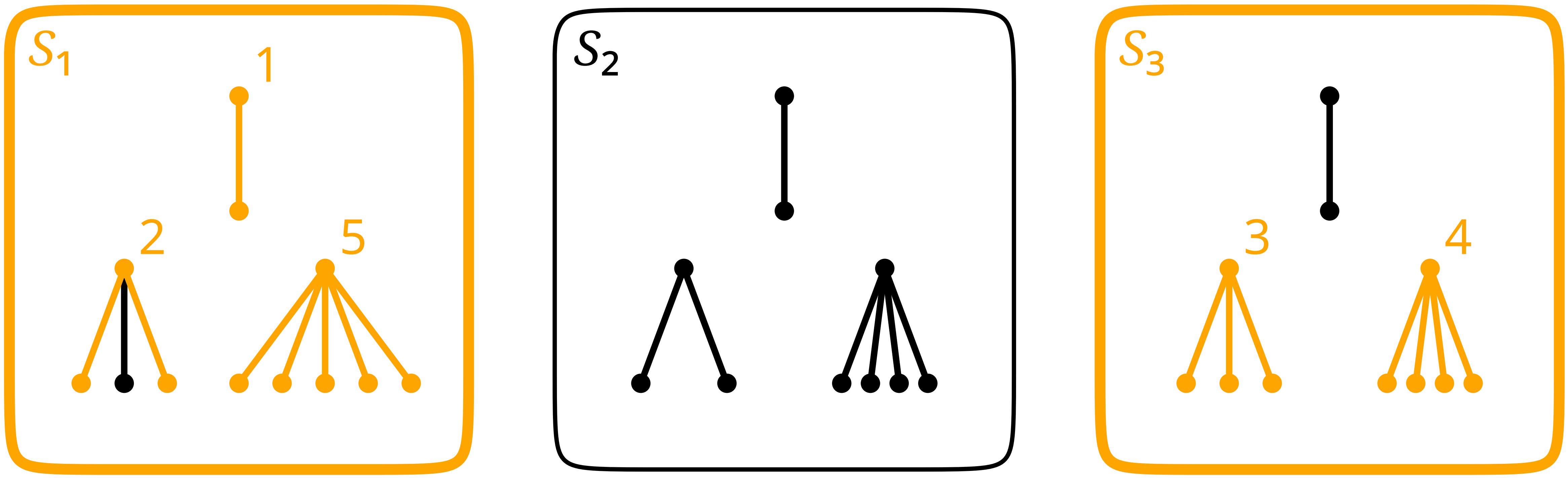
We first show that this problem is NP-complete.

Inclusive Set Cover

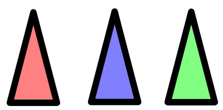
Input: sets of stars $S_1, S_2, \ldots, S_m$, an integer k.

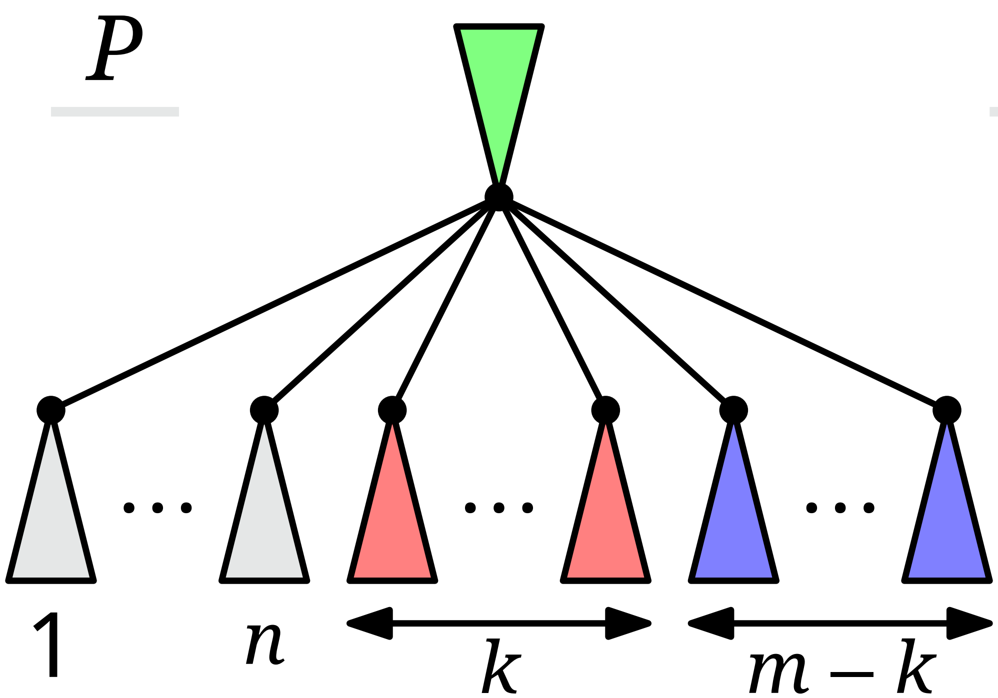
Question: Can we pick k sets from $S_1, S_2, \ldots, S_m$ so that we can embed stars with $1, 2, \ldots n$ leaves into them.

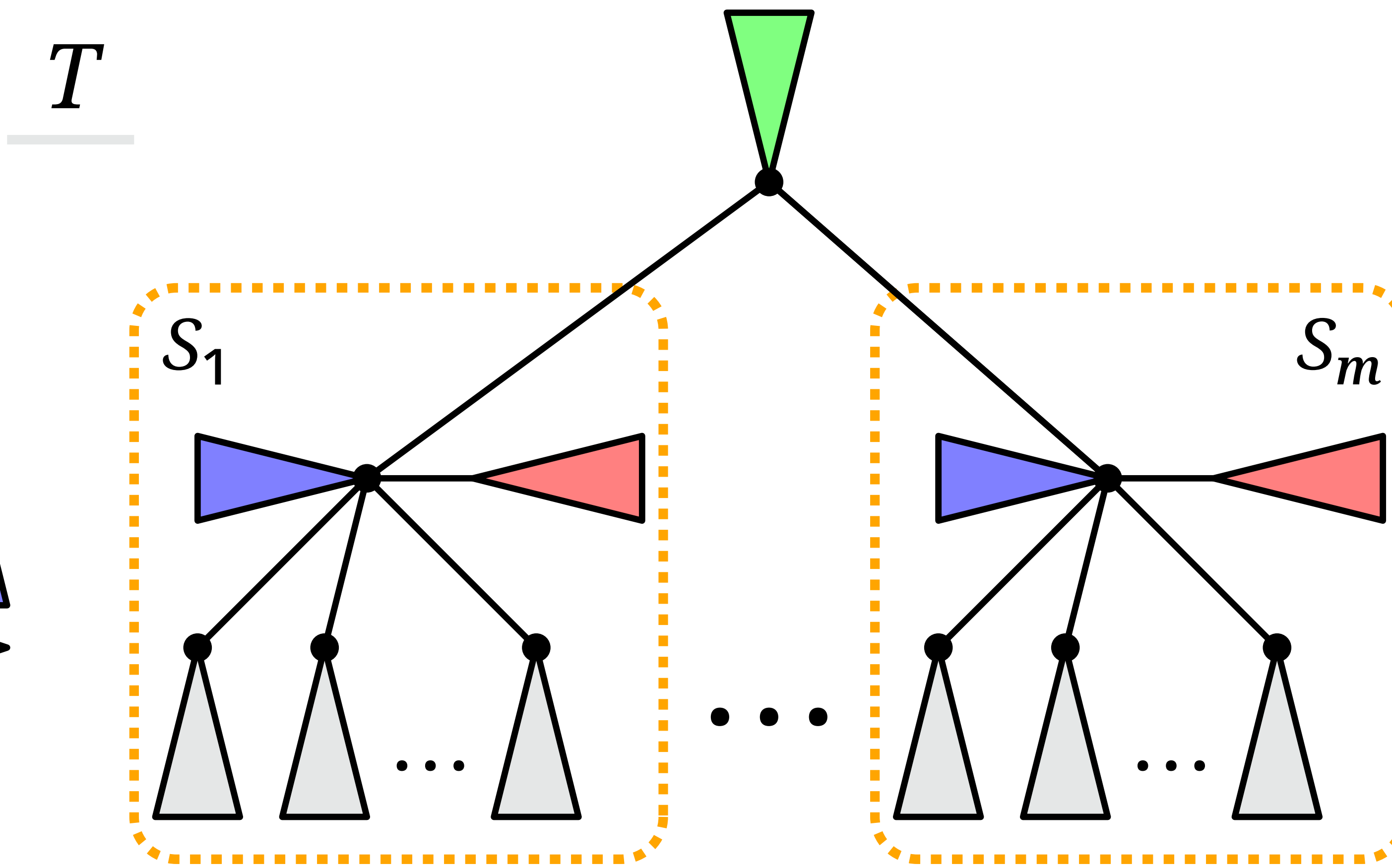
example) $n = 5, m = 3, k = 2$



Evaluating the Reduction

This is the complete looking of the reduction
where  are large enough stars.





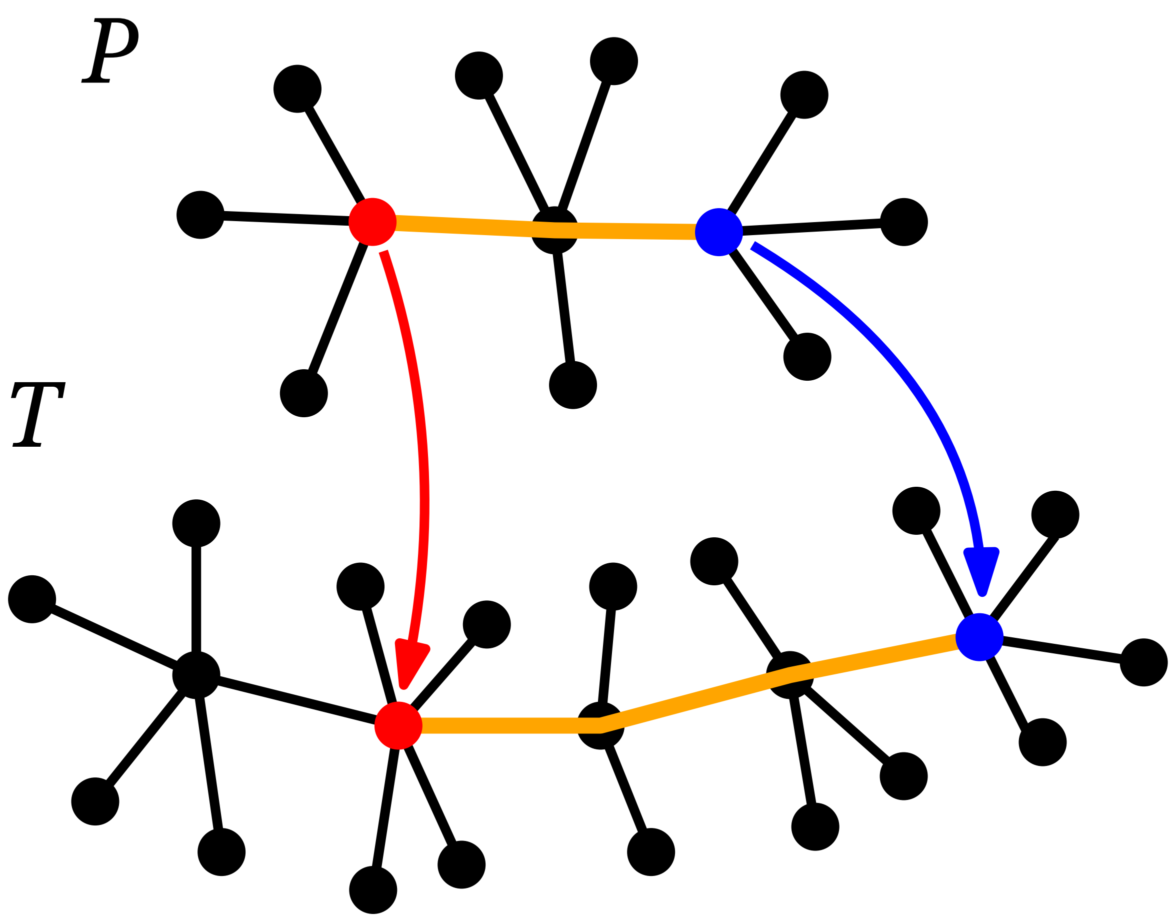
This reduction satisfies:

- $\text{diam}(P) \leq 4, \text{diam}(T) \leq 6$
- $\text{pe}(P) \leq 2, \text{pe}(T) \leq 3$ (by $\text{pe}(G) \leq \text{diam}(G)/2$)

Similar idea can be used for pathwidth.

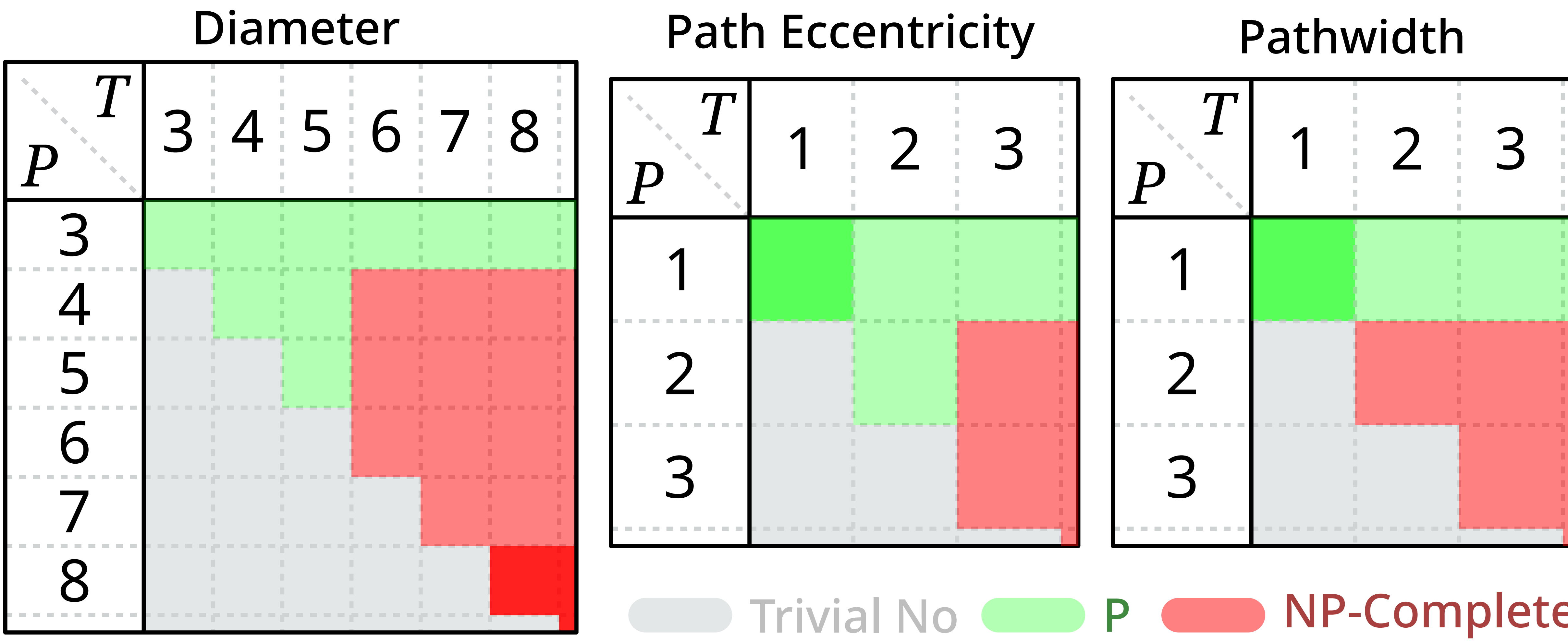
Algorithms for the Remaining Cases

On the remaining cases, P is like a caterpillar.



- Guess where to embed the **backbone** of P.
- Greedy (polynomial-time) algorithm to check if there is a such embedding.

Summary and Open Problems



- TMC is tractable only on very restricted instances.
- The border seems to be around “P is a caterpillar”.

Open Problems

- How about the maximum degree d ? (Known: $O^*(2^d)$)
- Another generalization of caterpillar?