RECOGNIZING K-LEAF POWERS IN POLYNOMIAL TIME, FOR CONSTANT K

Manuel Lafond, Université de Sherbrooke, Canada

UNIVERSITÉ DE SHERBROOKE

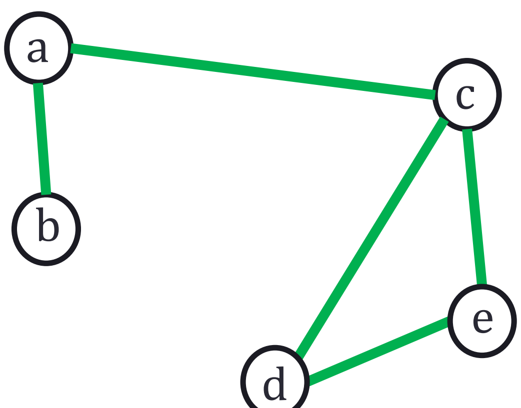
Definition

A graph G is a **k-leaf power** if there exists a tree T such that:

- $L(T) = V(G)$, where $L(T)$ is the set of leaves of T
- $uv \in E(G) \Leftrightarrow dist_T(u, v) \leq k$

$3 - leaf\ power\ ?$

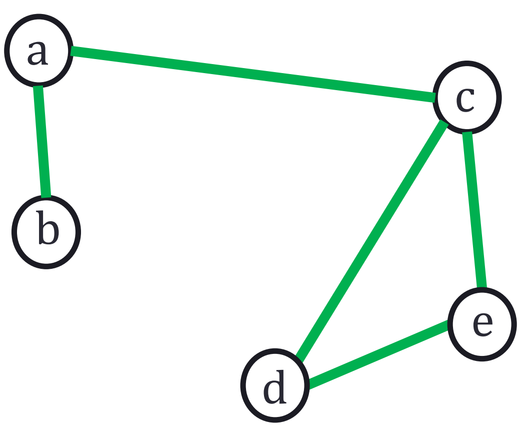
G



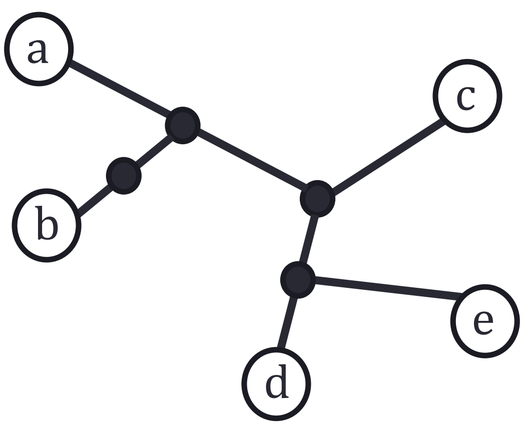
Definition
A graph G is a **k-leaf power** if there exists a tree T such that:
- $L(T) = V(G)$, where $L(T)$ is the set of leaves of T
- $uv \in E(G) \Leftrightarrow dist_T(u, v) \leq k$

$3 - leaf\ power$

G



$T:$



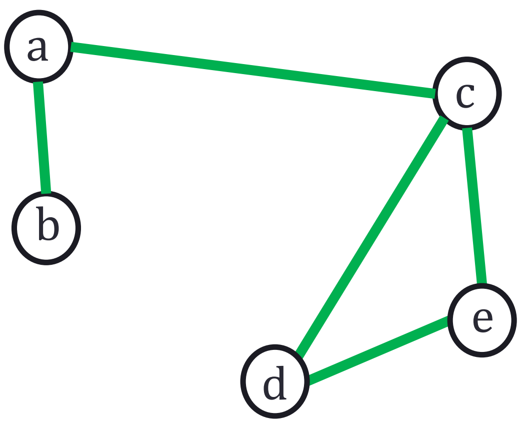
Definition
A graph G is a ***k*-leaf power** if there exists a tree T such that:

- $\boldsymbol{L(T) = V(G)}$, where $L(T)$ is the set of leaves of T
- $\boldsymbol{uv \in E(G) \Leftrightarrow dist_T(u,v) \leq k}$

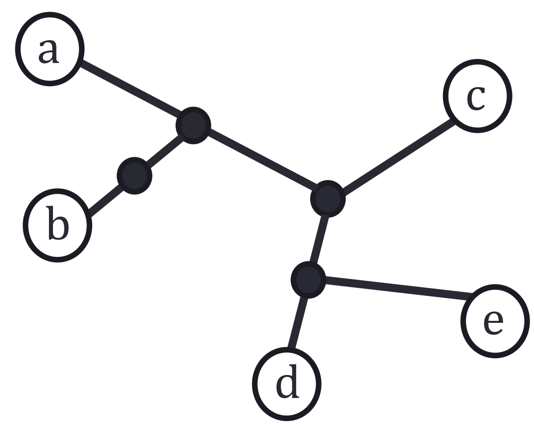
Equivalently, G is a k-leaf power if it can be obtained by taking the k-th power of a tree, and taking the subgraph induced by the leaves of the tree.

$3 - leaf\ power$

G



T :



Definition

A graph G is a ***k*-leaf power** if there exists a tree T such that:

- $\boldsymbol{L(T) = V(G)}$, where $L(T)$ is the set of leaves of T
- $\boldsymbol{uv \in E(G) \Leftrightarrow dist_T(u,v) \leq k}$

Open problems [Nishimura, Ragde, Thilikos, 2002]

- Characterize k-leaf powers, for every k.
- Characterize leaf powers, the union of k-leaf powers for all k.
- Is recognizing leaf powers in P?
- For fixed k, is recognizing k-leaf powers in P?

Definition

A graph G is a ***k*-leaf power** if there exists a tree T such that:

- $\boldsymbol{L(T) = V(G)}$, where $L(T)$ is the set of leaves of T
- $\boldsymbol{uv \in E(G) \Leftrightarrow dist_T(u,v) \leq k}$

Open problems [Nishimura, Ragde, Thilikos, 2002]

- Characterize k-leaf powers, for every k. **OPEN**
- Characterize leaf powers, the union of k-leaf powers for all k. **OPEN**
- Is recognizing leaf powers in P? **OPEN**
- For fixed k, is recognizing k-leaf powers in P? **YES, THIS TALK**

Theorem
There is an algorithm that, given a graph G, decides whether G is a k-leaf power in time $O(n^{f(k)})$, where $n = |V(G)|$ and f is a function that depends only on k.

Theorem
There is an algorithm that, given a graph G, decides whether G is a k-leaf power in time $O(n^{f(k)})$, where $n = |V(G)|$ and f is a function that depends only on k.

$$f(k) \simeq 2^{3k^{3k^{3k^{\cdot^{\cdot^{\cdot^{3k}}}}}}} \quad \left.\right\} k \text{ times}$$

Theorem

There is an algorithm that, given a graph G, decides whether G is a k-leaf power in time $O(n^{f(k)})$, where $n = |V(G)|$ and f is a function that depends only on k.

$$f(k) \simeq 2^{3k^{3k^{3k^{\cdot^{\cdot^{\cdot^{3k}}}}}}} \quad \Big\} \; k \text{ times}$$

Relevance

- Many papers on leaf powers, slow progress. Few results apply to all k.
- Several similar tree-definable graph classes. Techniques developed here might be applicable to them.

Known results

- **2-leaf powers** = P3-free graphs *[folklore]*
- **3-leaf powers** = chordal + (bull, gem, dart)-free graphs *[Rautenbach, Disc Maths 2006]*
- **4-leaf powers** = chordal + X-free, where X is a finite set of forbidden subgraphs *[Brandstädt et al., TALG 2008]*
- **5-leaf powers** recognition in P *[Chang & Ko, WG 2007]*
- **6-leaf powers** recognition in P *[Ducoffe, WG 2019]*
- Recognizing k-leaf powers is FPT in k + degeneracy(G), and **FPT in k + treewidth(G)**. *[Eppstein & Havvaei, IPEC 2018]*

Known results

- Leaf power = graphs that are k-leaf powers for some k.
- All leaf powers are **chordal**, and also **strongly chordal**
- Converse **not true** *[L, WG2017; Jaffke & al., TCS2019]*
- **Subclasses** of strongly chordal (interval, rooted directed, ptolemaic) graphs are **easy to recognize** *[Brandstädt et al., LATIN2008 & DiscMath2010]*
- Leaf powers have **mim-width 1** *[Jaffke & al., TCS2019]*
- Leaf powers with **star NeS models** in P *[Bergougnoux, 2021]*

Other tree-definable classes

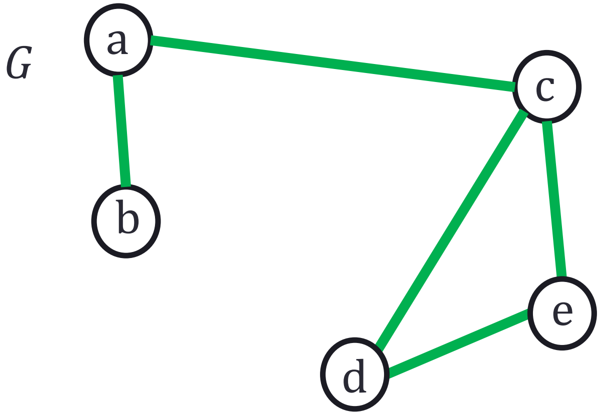
- Many other tree-to-graph representations, all with similar open problems
  - Pairwise compatiblity graphs (PCG)
    - uv edge iff distance in interval $[l, h]$
  - k-interval PCGs, OR-PCGs and AND-PCGs
    - Allow k-intervals, union/intersection of PCGs
  - Orthology graphs
    - uv edge iff lca has label 1
  - Fitch graphs
    - uv edge iff some edge on $u - v$ path has label 1
  - Best match graphs
  - …

Theorem

There is an algorithm that, given a graph G, decides whether G is a k-leaf power in time $O(n^{f(k)})$, where $n = |V(G)|$ and f is a function that depends only on k.

High-level overview

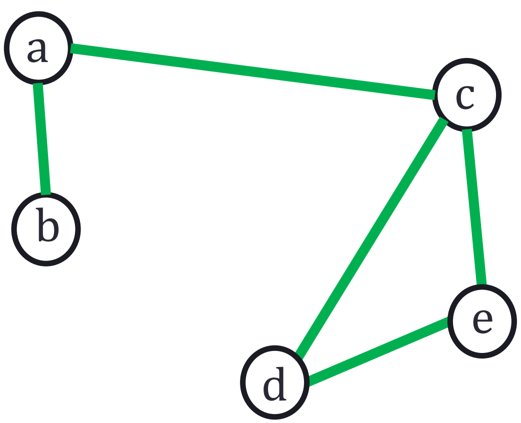
- Given a graph G, we must decide whether G is a k-leaf power (assume that k is fixed).



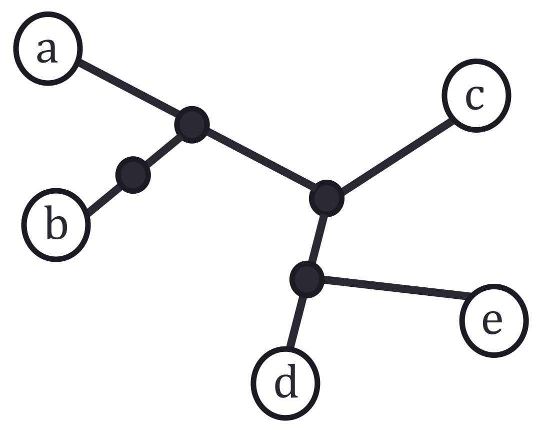
High-level overview

For G a k-leaf power, a ***k*-leaf root of *G*** is a tree with $\boldsymbol{L(T) = V(G)}$ satisfying $\boldsymbol{uv \in E(G) \Leftrightarrow dist_T(u,v) \leq k}$.

G



$3 - leaf\ root$

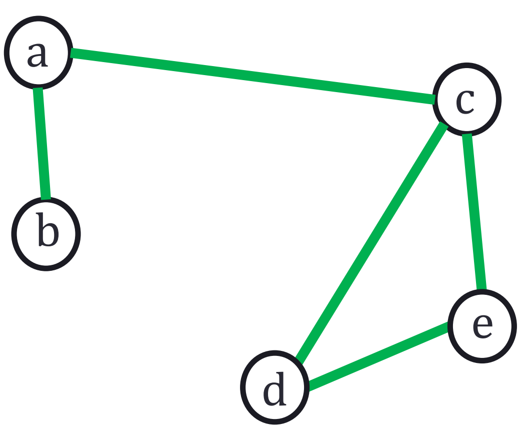


High-level overview

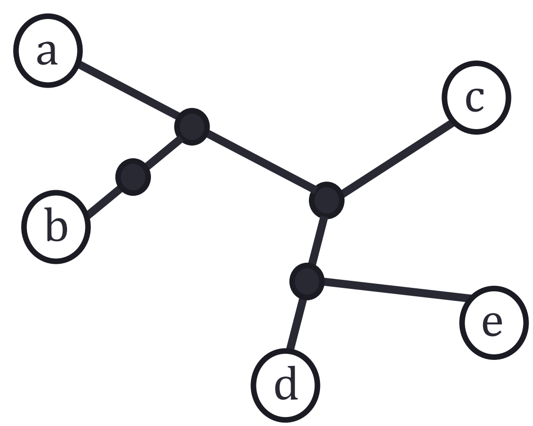
For G a k-leaf power, a ***k*-leaf root of *G*** is a tree with $\boldsymbol{L(T) = V(G)}$ satisfying $\boldsymbol{uv \in E(G) \Leftrightarrow dist_T(u,v) \leq k}$.

Theorem (from Eppstein & Havvaei, 2019)
There is a function g such that one can decide in time $O(g(tw(G),k)n)$ whether G is a k-leaf power, where $tw(G)$ is the treewidth of G.

G



$3 - leaf\ root$

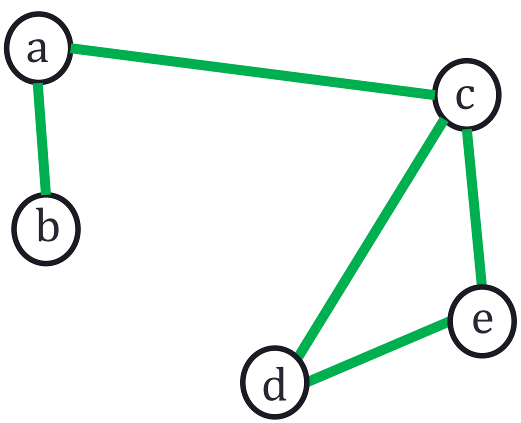


High-level overview

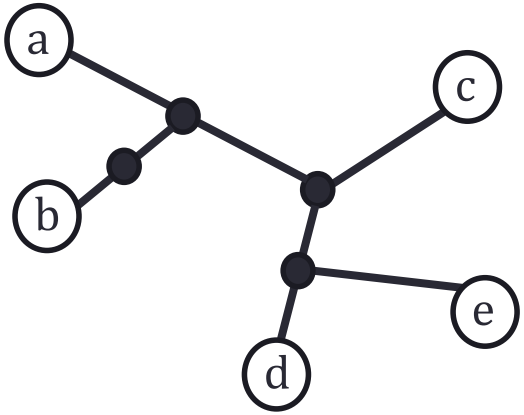
For G a k-leaf power, a **k-leaf root of G** is a tree with $\boldsymbol{L(T) = V(G)}$ satisfying $\boldsymbol{uv \in E(G) \Leftrightarrow dist_T(u,v) \leq k}$.

Theorem
Let d, k be integers. Then one can decide in time $O(g(d^k, k)n)$ whether a graph G admits a k-leaf root **of maximum degree d**.

G

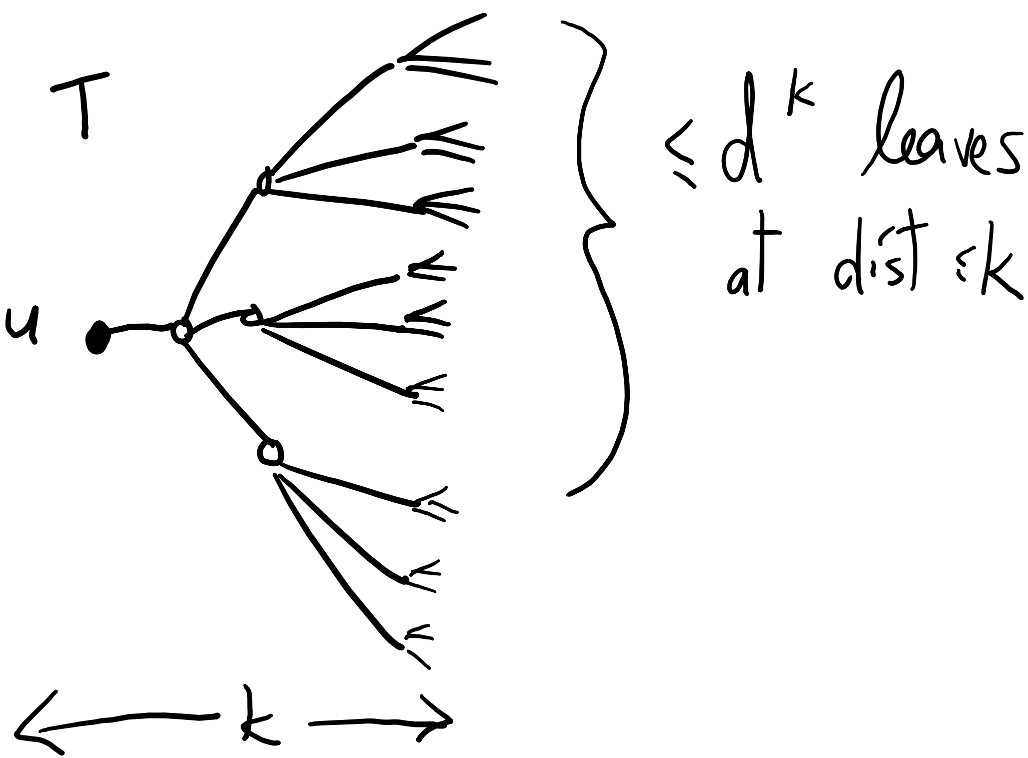


$3 - leaf\ root$



> **Theorem**
> Let d, k be integers. Then one can decide in time $O(g(d^k, k)n)$ whether a graph G admits a k-leaf root **of maximum degree d**.

- Proof idea.
- If G admits a k-leaf root of max degree d, then G has maximum degree d^k.



Theorem
Let d, k be integers. Then one can decide in time $\mathrm{O}(g(d^k, k)n)$ whether a graph G admits a k-leaf root **of maximum degree d**.

- Proof idea.
- If G admits a k-leaf root of max degree d, then G has maximum degree d^k.
- In chordal graphs, we have $tw(G) = w(G) - 1 \leq dk$.
  - tw(G) = treewidth, w(G) = clique number

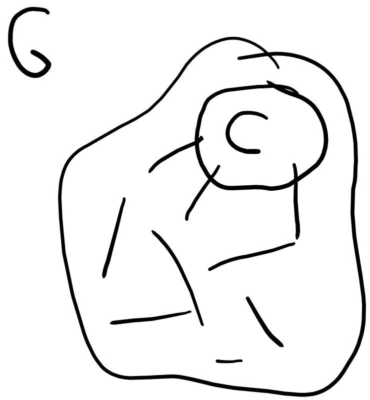
- Use Eppstein & Havvaei to decide in time $O(g(tw(G), k)n) = O(g(d^k, k)n)$ whether G is a k-leaf power.

Theorem
Let d, k be integers. Then one can decide in time $O(g(d^k, k)n)$ whether a graph G admits a k-leaf root **of maximum degree d**.

- If d is a function of k, problem solved.
- **Bottom-line** : the difficulty resides in k-leaf roots of high maximum degree.

k-leaf roots with high degree

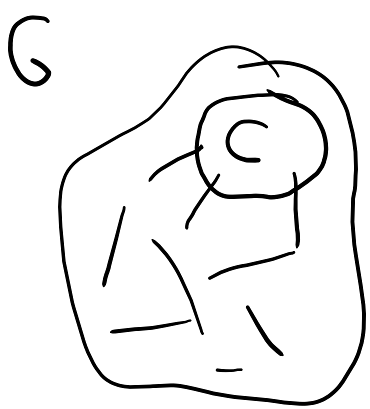
Theorem
There is f such that if G admits a k-leaf root of max degree $d > f(k)$, then G contains a subset C of vertices such that **G is a k-leaf power if and only if $G - C$ is a k-leaf power**.
Moreover, C can be found in time $O(n^{f(k)})$ if it exists.

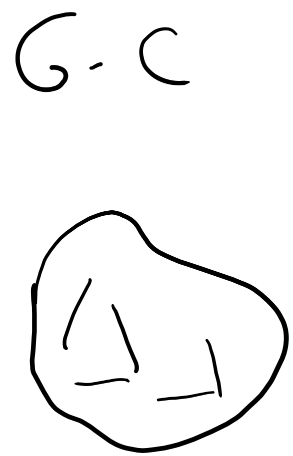


k-leaf roots with high degree

Theorem
There is f such that if G admits a k-leaf root of max degree $d > f(k)$, then G contains a subset C of vertices such that ***G* is a *k*-leaf power if and only if *$G - C$* is a *k*-leaf power**.
Moreover, C can be found in time $O(n^{f(k)})$ if it exists.

This says that if G has high-degree k-leaf roots, then G has a redundant subset of vertices C that can be found and pruned quickly.





k-leaf roots with high degree

Theorem
There is f such that if G admits a k-leaf root of max degree $d > f(k)$, then G contains a subset C of vertices such that ***G* is a *k*-leaf power if and only if *G* – *C* is a *k*-leaf power**.
Moreover, C can be found in time $O(n^{f(k)})$ if it exists.

The algorithm:
1) Check if G admits a k-leaf root of degree at most $d = f(k)$. If yes, return "yes".
2) Otherwise, check if G contains C as described above. If not, return "no".
3) Otherwise, repeat on $G - C$.

Finishes in polynomial time, since k is fixed and this is repeated at most n times.

k-leaf roots with high degree

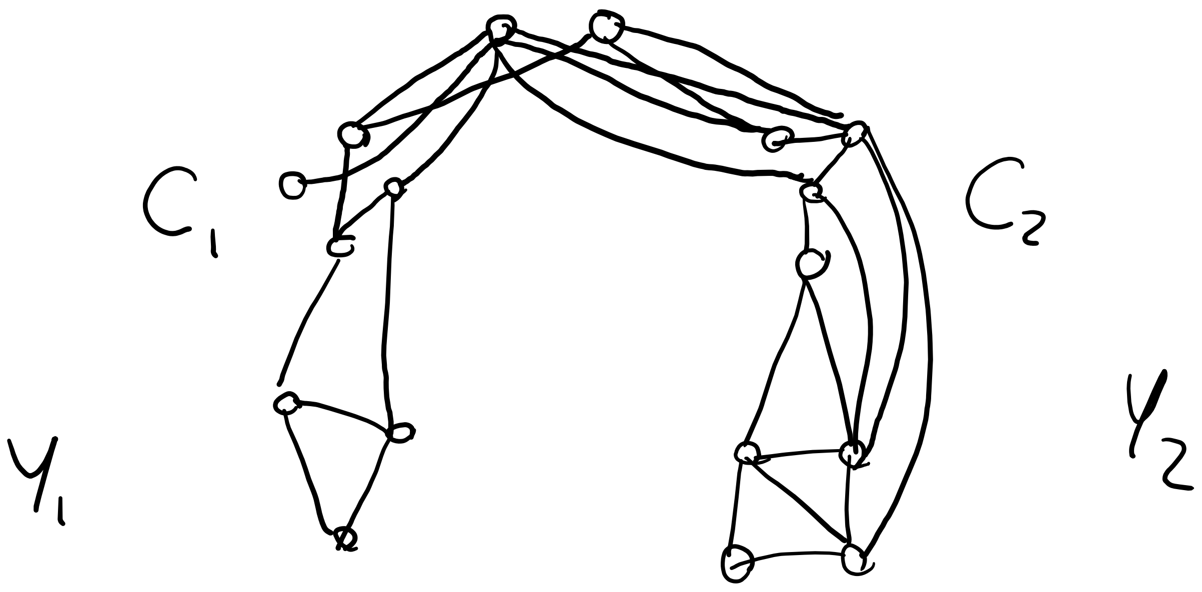
Theorem
There is f such that if G admits a k-leaf root of max degree $d > f(k)$, then G contains a subset C of vertices such that **G is a k-leaf power if and only if $G - C$ is a k-leaf power**.
Moreover, C can be found in time $O(n^{f(k)})$ if it exists.

This says that if G has high-degree k-leaf roots, then G has a redundant subset of vertices C that can be found and pruned quickly.

Step 1 : find lots of subsets $C_i \cup Y_i$ such that the C_i's are cutsets, and all have the same neighborhood structure.
Step 2 : argue that two of those $C_1 \cup Y_1$ and $C_2 \cup Y_2$ admits the "same" k-leaf roots.
Step 3 : argue that $C_1 \cup Y_1$ can be removed since it behaves like $C_2 \cup Y_2$

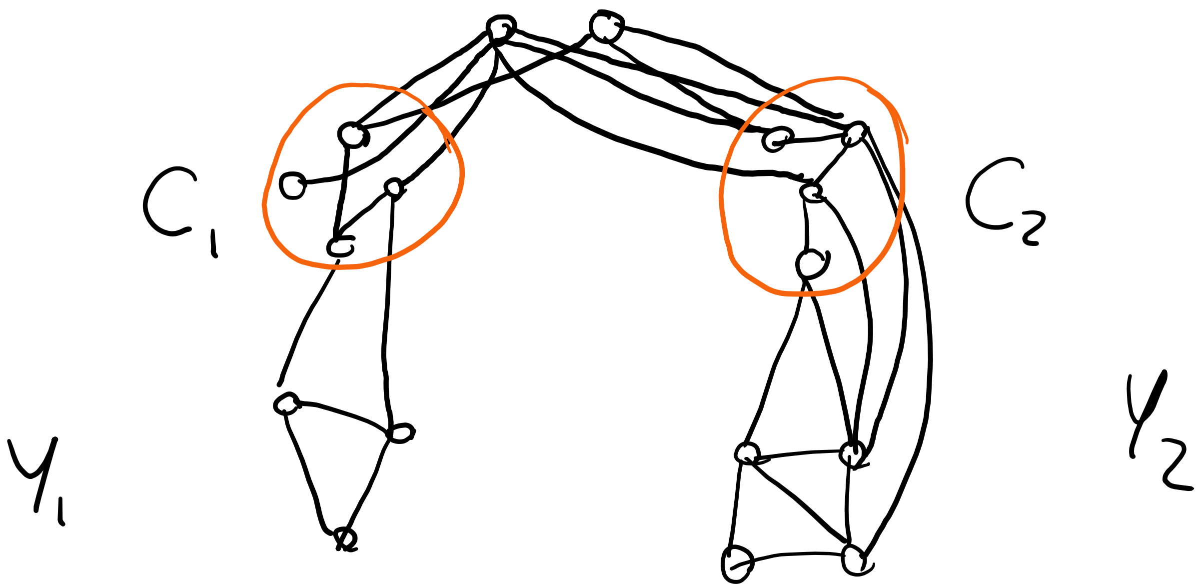
Similar sets of vertices

- We say that $C_1 \cup Y_1$ and $C_2 \cup Y_2 \subseteq V(G)$ are **similar** if
  - C_1 cuts Y_1 and C_2 cuts Y_2 from the rest of the graph
  - $C_1 \cup C_2$ can be partitioned into layers $L_1, \ldots, Lk$ such that vertices in the same layer have the same neighbors in $G - (C_1 \cup Y_1 \cup C_2 \cup Y_2)$.



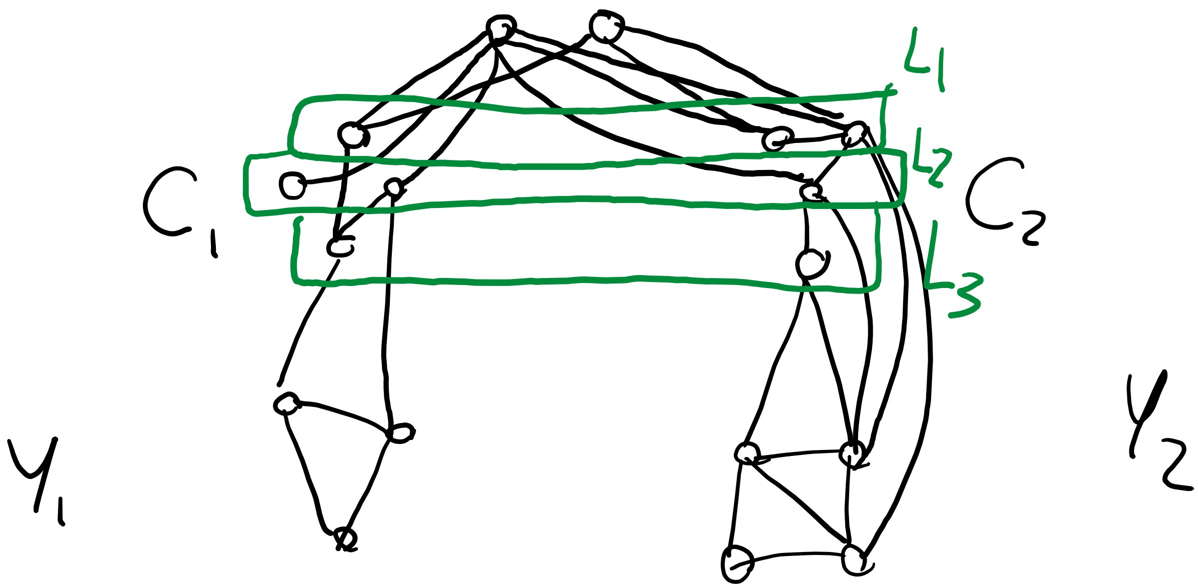
Similar sets of vertices

- We say that $C_1 \cup Y_1$ and $C_2 \cup Y_2 \subseteq V(G)$ are **similar** if
  - C_1 cuts Y_1 and C_2 cuts Y_2 from the rest of the graph
  - $C_1 \cup C_2$ can be partitioned into layers $L_1, \dots, Lk$ such that vertices in the same layer have the same neighbors in $G - (C_1 \cup Y_1 \cup C_2 \cup Y_2)$.



Similar sets of vertices

- We say that $C_1 \cup Y_1$ and $C_2 \cup Y_2 \subseteq V(G)$ are **similar** if
  - C_1 cuts Y_1 and C_2 cuts Y_2 from the rest of the graph
  - $C_1 \cup C_2$ can be partitioned into layers $L_1, \ldots, Lk$ such that vertices in the same layer have the same neighbors in $G - (C_1 \cup Y_1 \cup C_2 \cup Y_2)$.

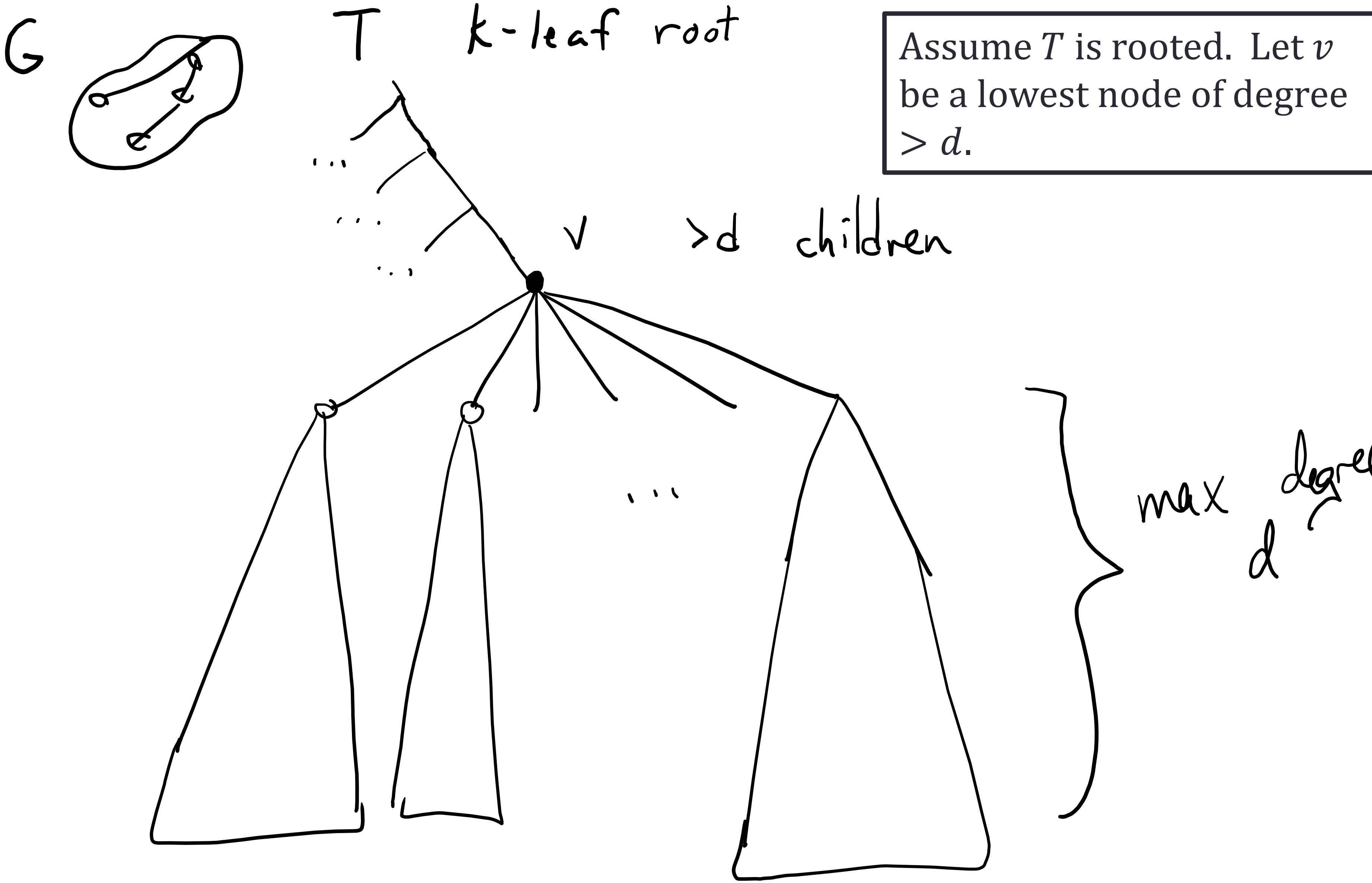


Similar sets of vertices

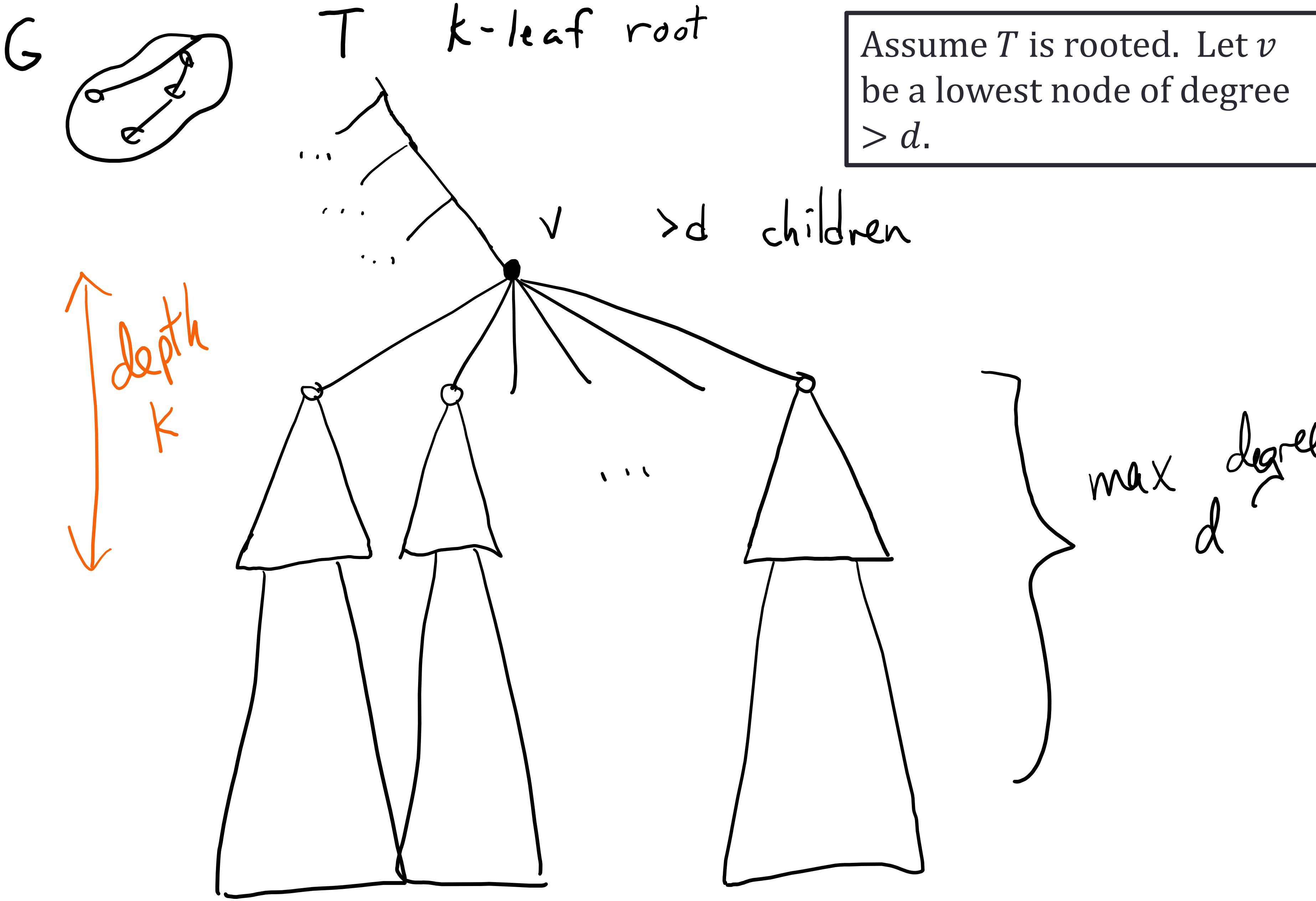
- We say that $C_1 \cup\ Y_1$ and $C_2 \cup\ Y_2\ \subseteq V(G)$ are **similar** if
  - C_1 cuts Y_1 and C_2 cuts Y_2 from the rest of the graph
  - $C_1 \cup C_2$ can be partitioned into layers $L_1, \ldots, Lk$ such that vertices in the same layer have the same neighbors in $G - \ (C_1 \cup Y_1 \cup C_2 \cup Y_2)$.

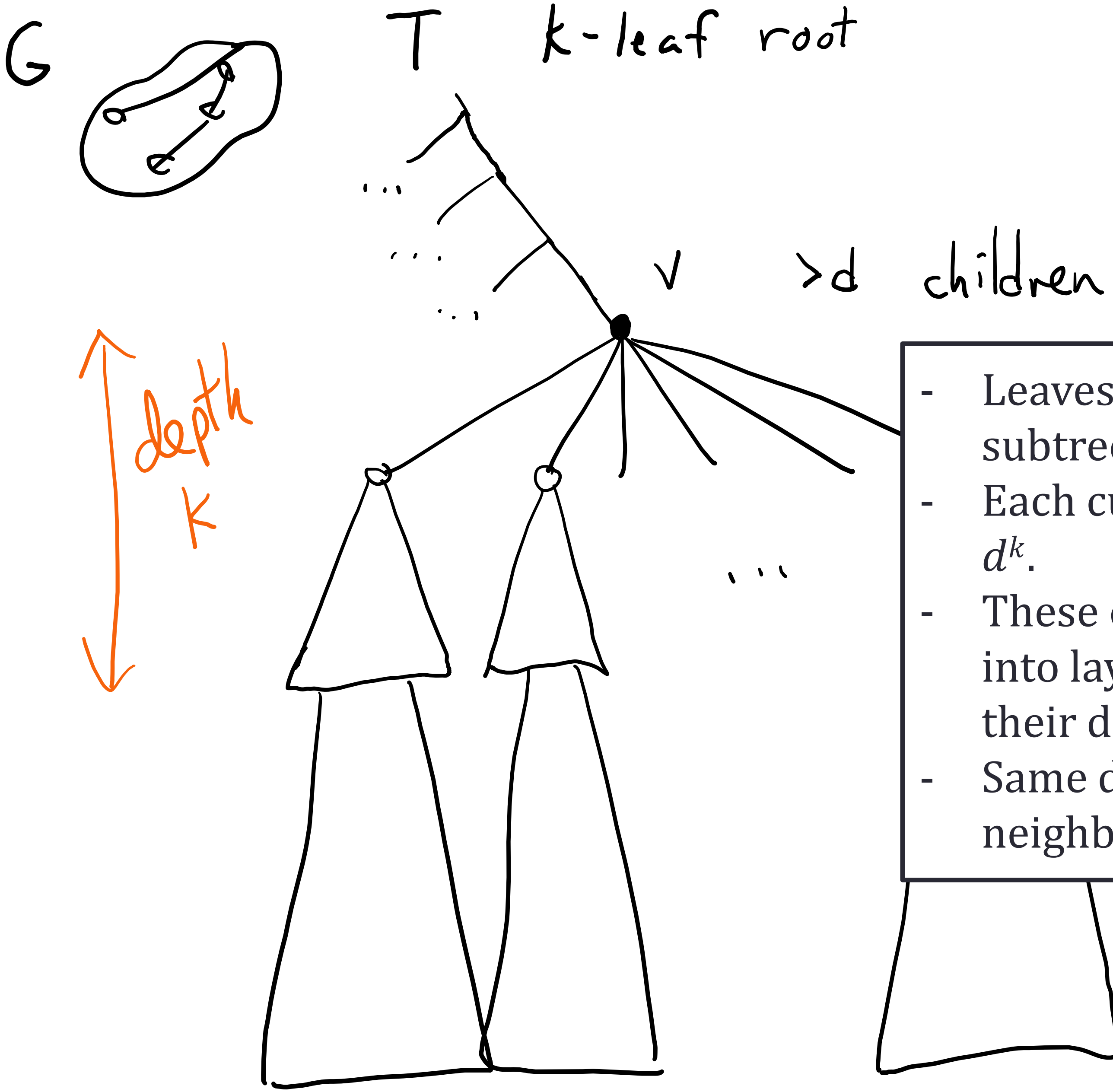
Lemma
If G has a k-leaf root of maximum degree $> d$, then there exist disjoint $C_1 \cup\ Y_1, \ldots, Cd \cup Yd$ pairwise-similar subsets. Also, each C_i has size $\leq dk$.

Assume T is rooted. Let v be a lowest node of degree $> d$.

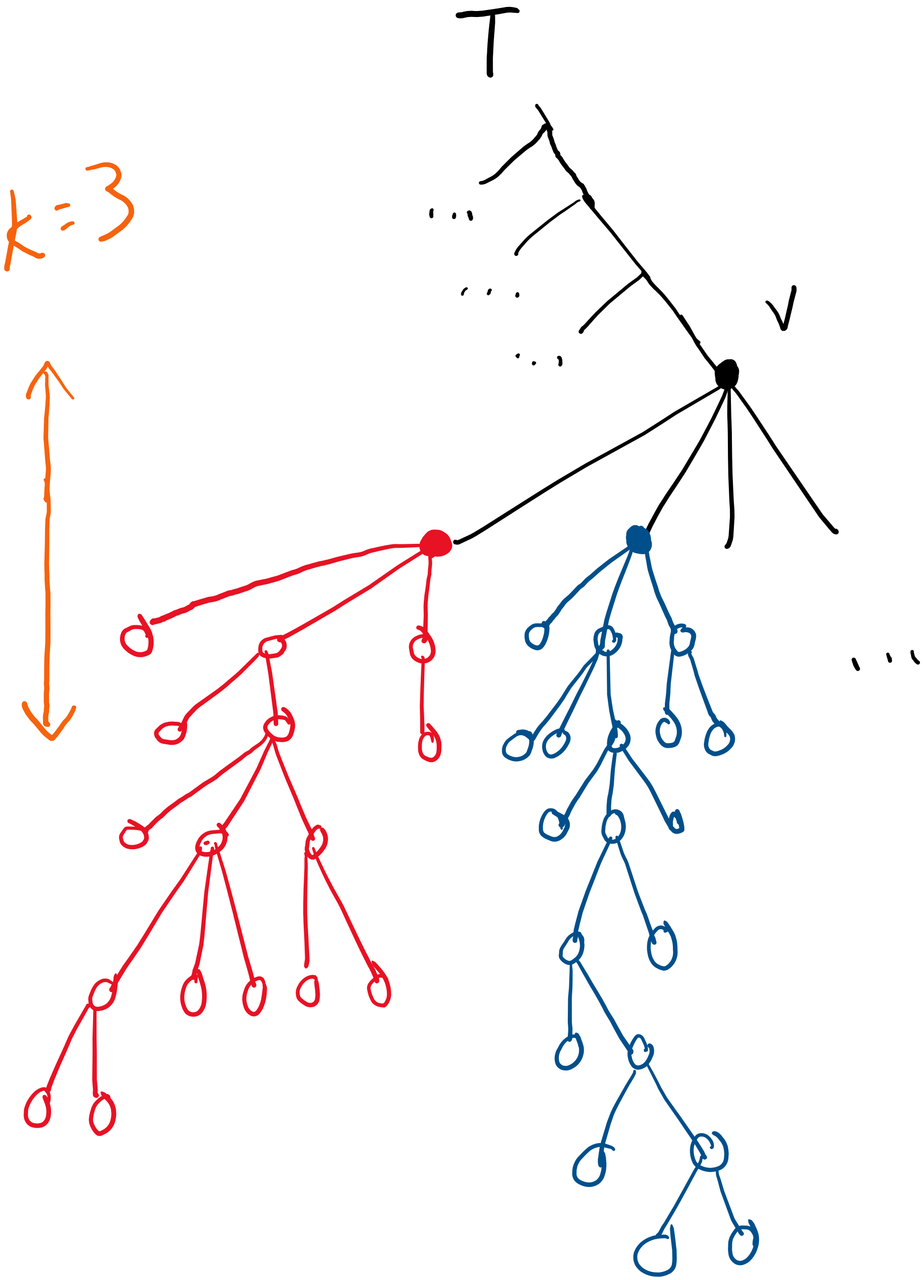


Assume T is rooted. Let v be a lowest node of degree $> d$.

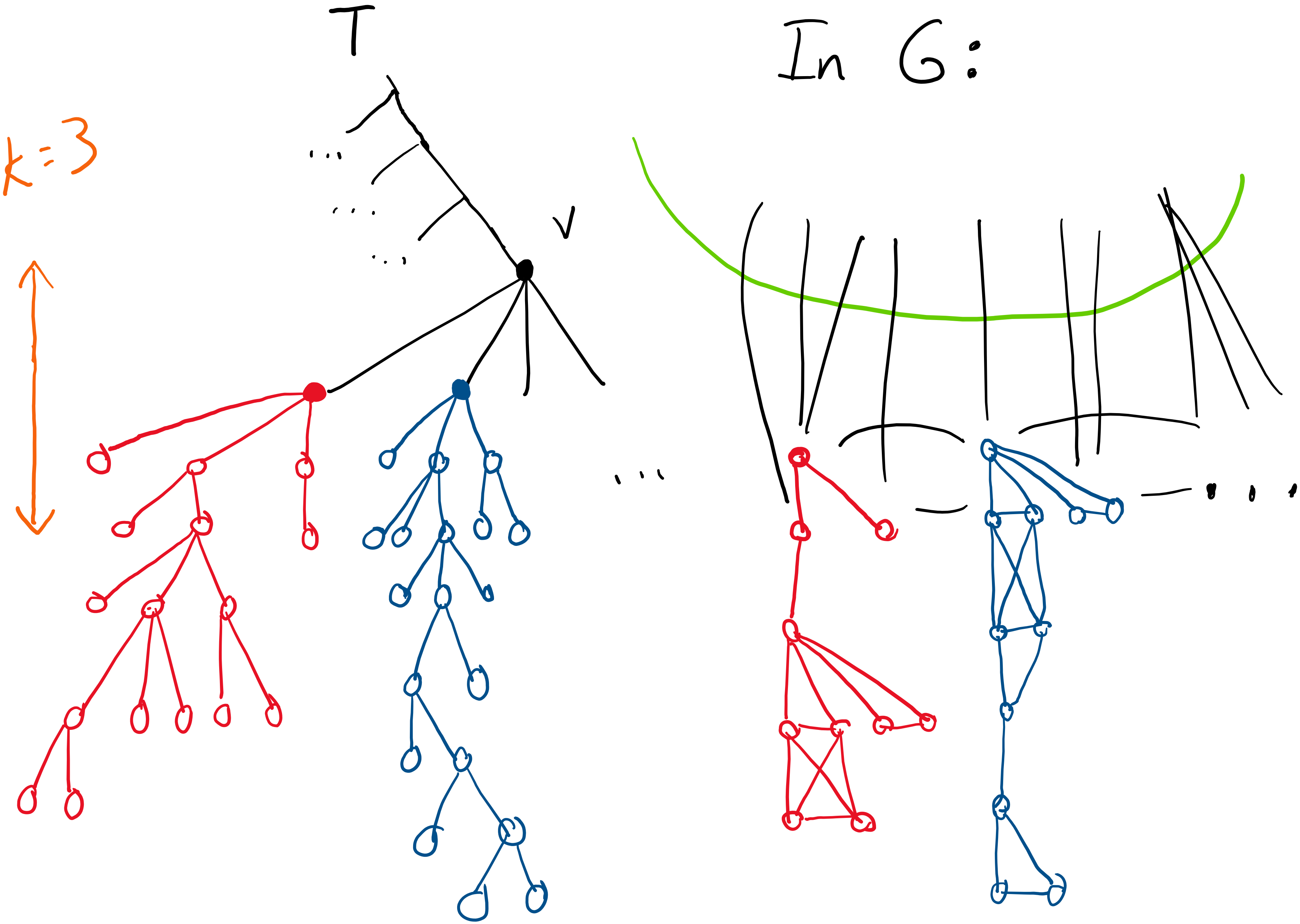


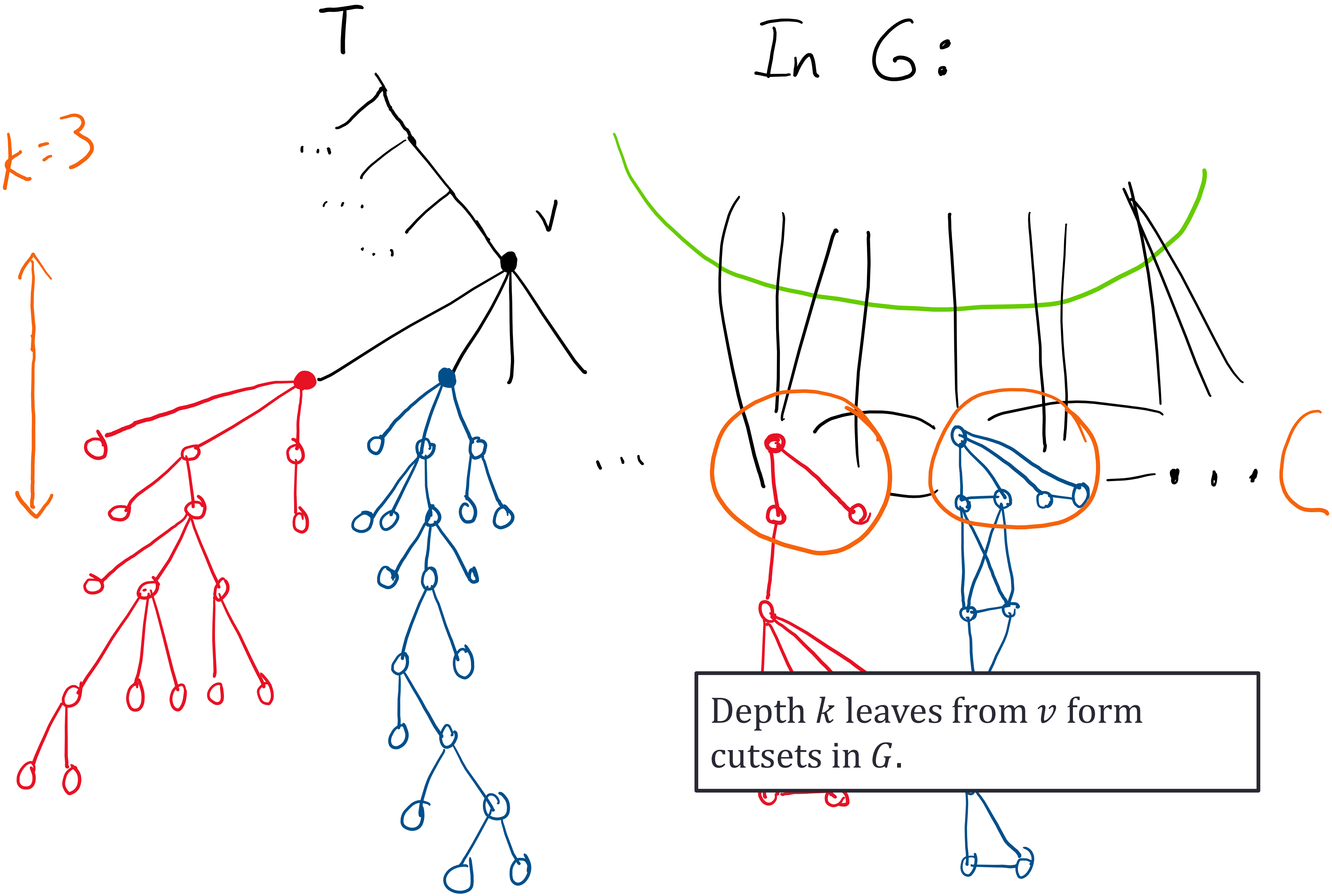


- Leaves in these depth k subtrees form cutsets in G.
- Each cutset has size at most d^k.
- These cutsets are organized into layers determined by their distance to v.
- Same distance = same neighbors "above" v.

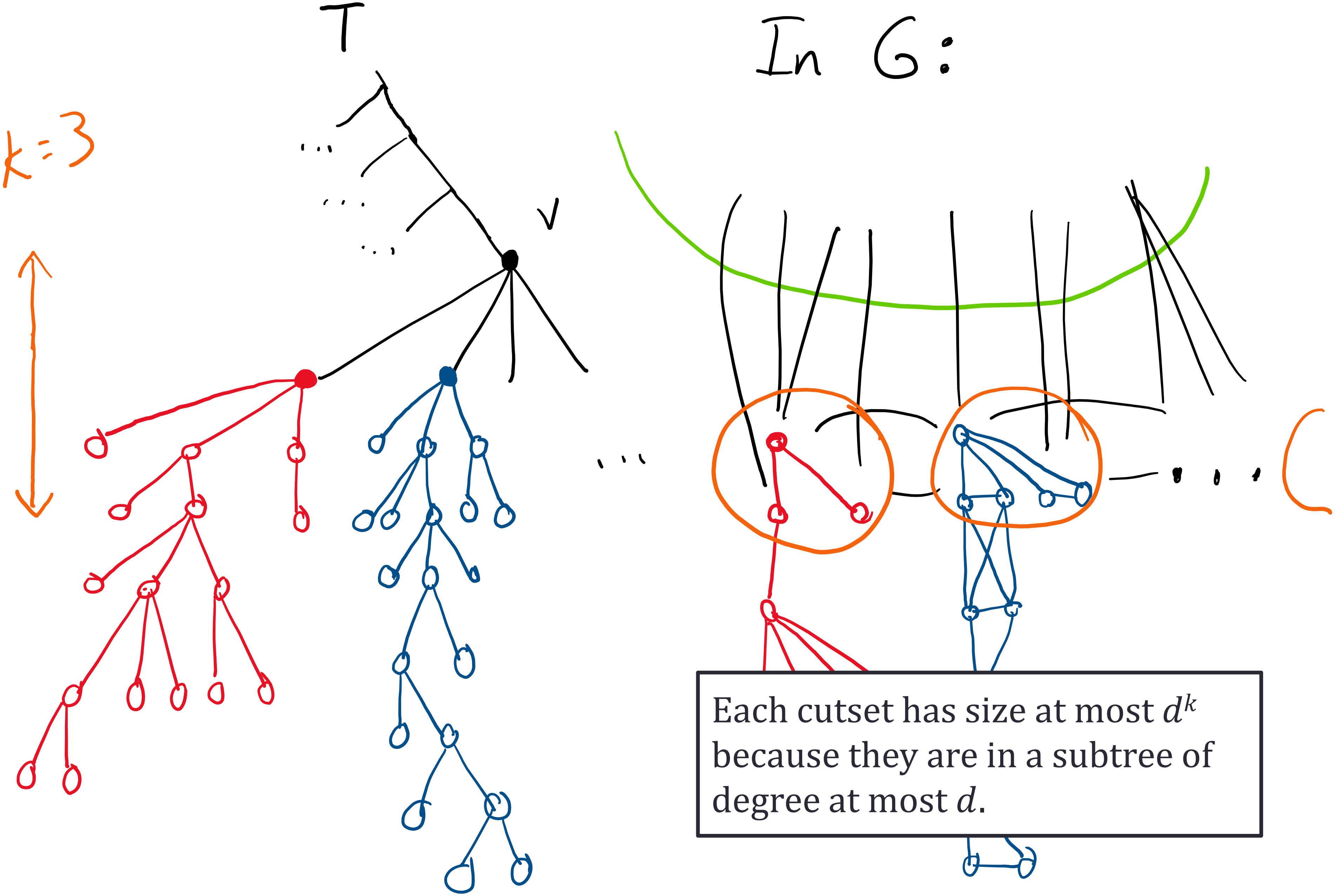


- Leaves in these depth k subtrees form cutsets in G.
- Each cutset has size at most d^k.
- These cutsets are organized into layers determined by their distance to v.
- Same distance = same neighbors “above” v.

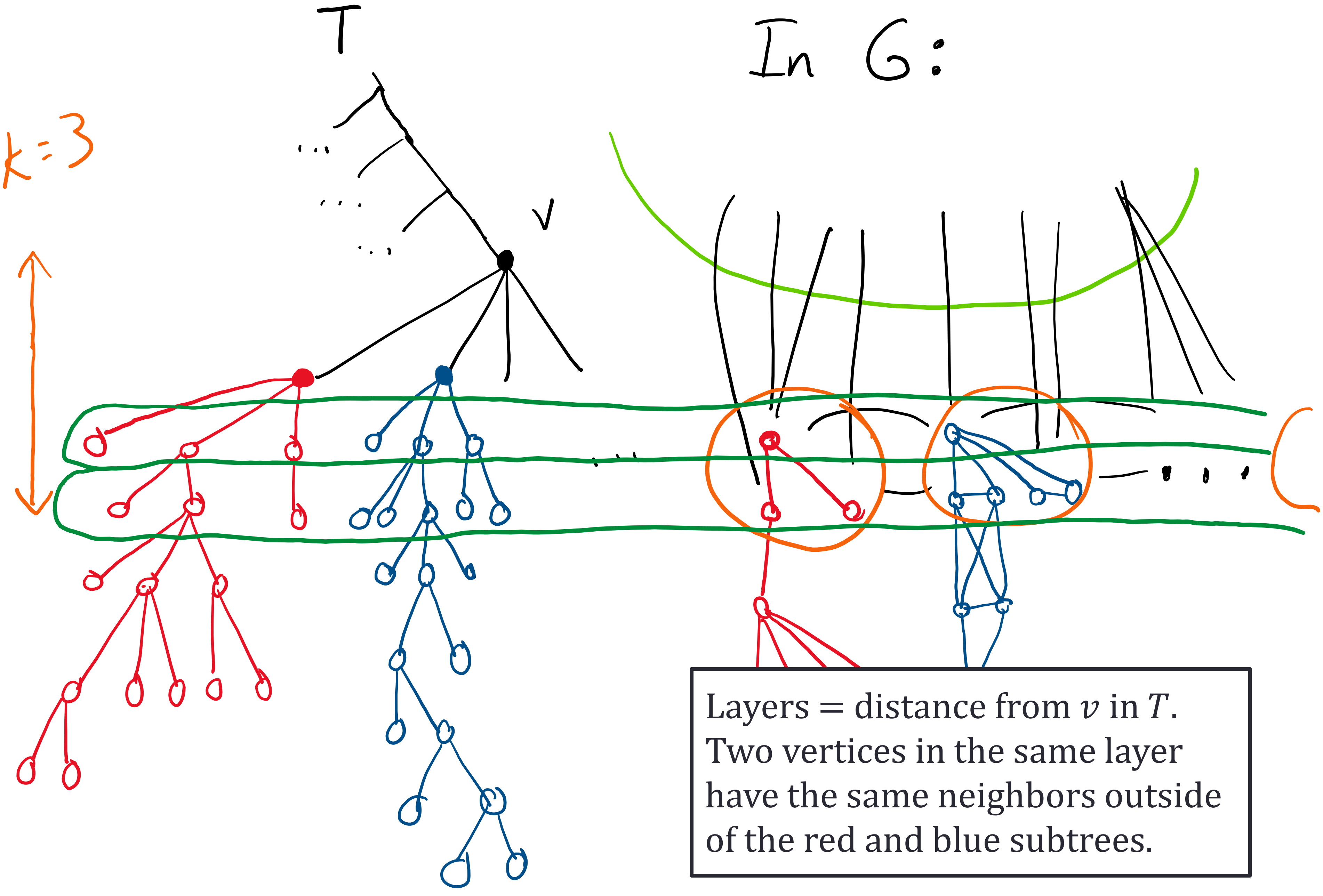

T
In G:
k=3
v



Depth k leaves from v form cutsets in G.



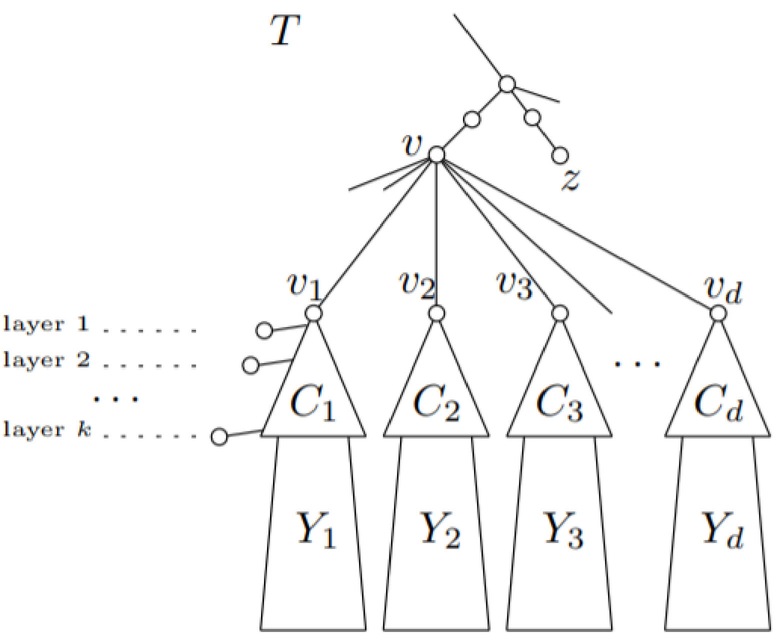
Each cutset has size at most d^k because they are in a subtree of degree at most d.

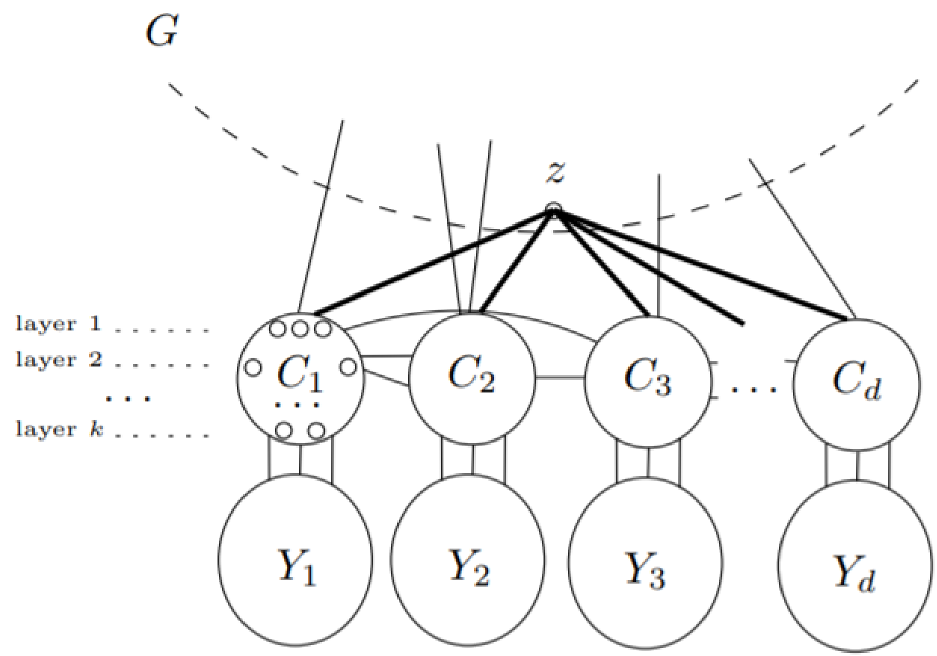


Layers = distance from v in T. Two vertices in the same layer have the same neighbors outside of the red and blue subtrees.

Lemma

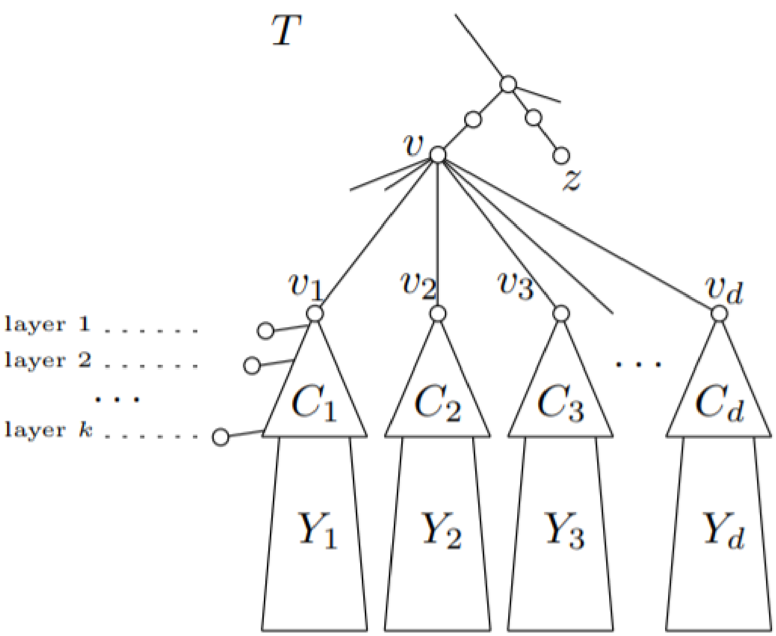
If G has a k-leaf root of maximum degree $> d$, then there exist disjoint $C_1 \cup \ Y_1, \dots, Cd \cup Yd$ pairwise-similar subsets. Also, each C_i has size $\leq dk$.

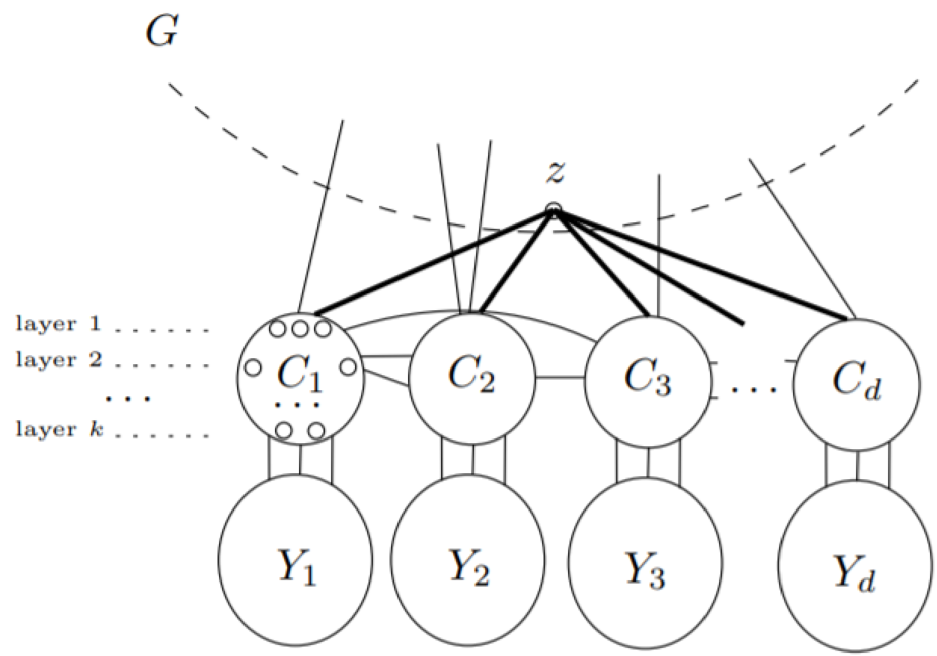




Lemma
If G has a k-leaf root of maximum degree $> d$, then there exist disjoint $C_1 \cup Y_1, \ldots, Cd \cup Yd$ pairwise-similar subsets. Also, each C_i has size $\leq dk$.

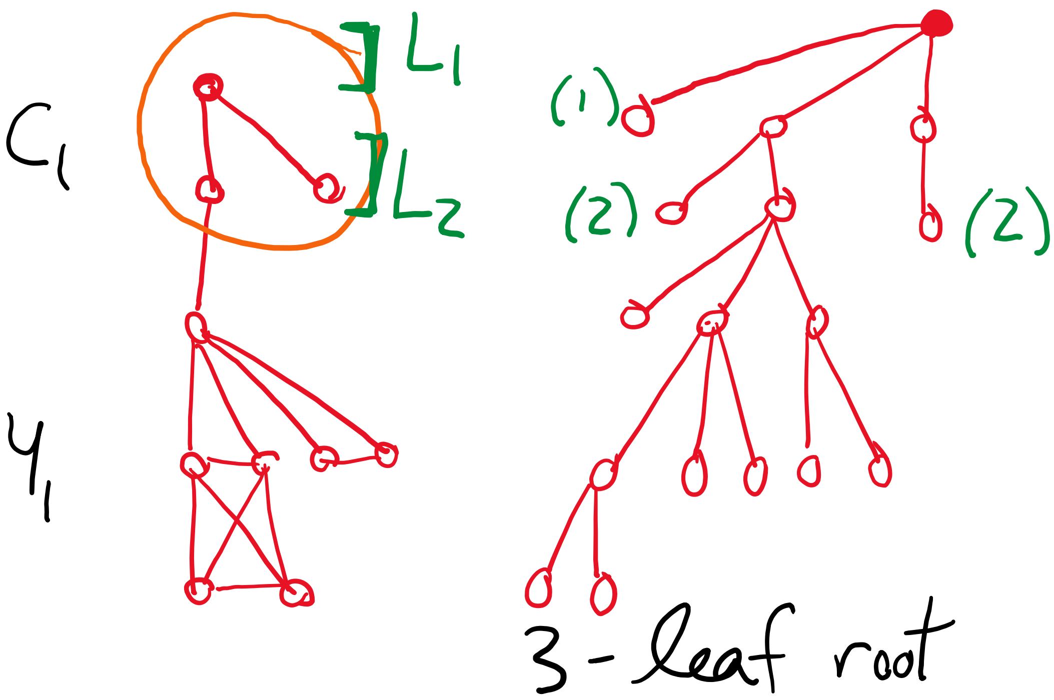
- So we can find many subsets with the same neighborhood structure.
- Next : find two among those that have the “same” k-leaf roots.





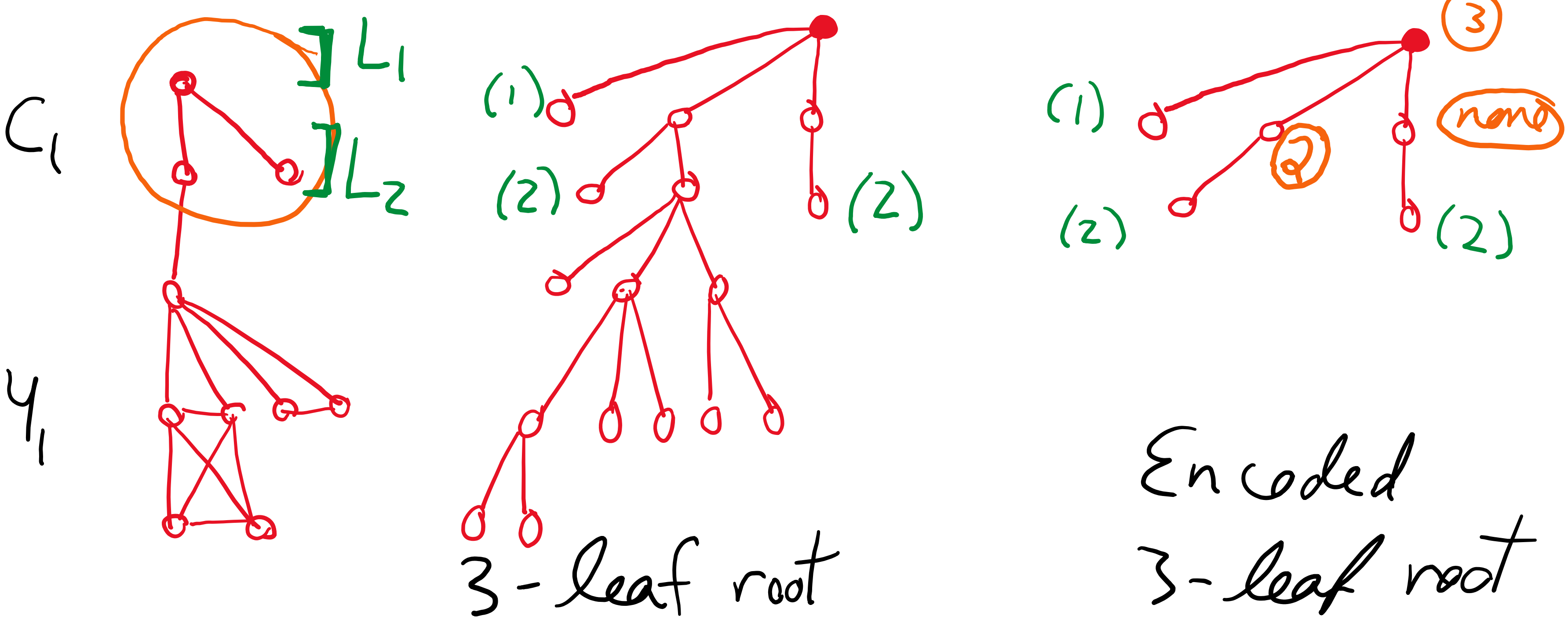
Similar sets with the same leaf roots

- Let $C_1 \cup Y_1$ be a set of vertices organized into layers $L_1, \ldots, Lk$.
- Let T_1 be a k-leaf root of G$[C_1 \cup Y_1]$. The **layer-encoding** of T_1 is obtained by
  - restricting T_1 to C_1 and their ancestors
  - replacing each leaf of C_1 by its layer number.
  - labeling internal nodes by the distance to its closest Y_1 leaf
  - keeping at most 2 copies of each identical child subtree (see paper)



Similar sets with the same leaf roots

- Let $C_1 \cup Y_1$ be a set of vertices organized into layers $L_1, \ldots, Lk$.
- Let T_1 be a k-leaf root of G$[C_1 \cup Y_1]$. The **layer-encoding** of T_1 is obtained by
  - restricting T_1 to C_1 and their ancestors
  - replacing each leaf of C_1 by its layer number.
  - labeling internal nodes by the distance to its closest Y_1 leaf
  - keeping at most 2 copies of each identical child subtree (see paper)



Similar sets with the same leaf roots

- Let $C_1 \cup Y_1$ be a set of vertices organized into layers $L_1, \ldots, Lk$.
- Let T_1 be a k-leaf root of G$[C_1 \cup Y_1]$. The **layer-encoding** of T_1 is obtained by
  - restricting T_1 to C_1 and their ancestors
  - replacing each leaf of C_1 by its layer number.
  - labeling internal nodes by the distance to its closest Y_1 leaf
  - keeping at most 2 copies of each identical child subtree (see paper)

Lemma
The number of possible layer-encoded k-leaf roots is at most $s(k)$, a function that depends only on k.

$$s(k) \simeq 3k^{3k^{3k^{3k^{\cdots^{3k}}}}} \quad \Big\} \; k \text{ times}$$

Similar sets with the same leaf roots

Lemma
The number of possible layer-encoded k-leaf roots is at most $s(k)$, a function that depends only on k.

Similar sets with the same leaf roots

Lemma
The number of possible layer-encoded k-leaf roots is at most $s(k)$, a function that depends only on k.

- So what?
- Let $C_1 \cup Y_1$ be a set of vertices organized into layers $L_1, \ldots, Lk$.
- Let $S(C_1 \cup Y_1)$ be the set of layer-encoded k-leaf roots that encode some k-leaf root of $\mathrm{G}[C_1 \cup Y_1]$.

Lemma
If G admits a k-leaf root of maximum degree $d > 2^{s(k)}$, then G contains two similar subsets $C_1 \cup Y_1, C_2 \cup Y_2$ such that $S(C_1 \cup Y_1) = S(C_2 \cup Y_2)$.

Similar sets with the same leaf roots

Lemma
If G admits a k-leaf root of maximum degree $d > 2^{s(k)}$, then G contains two similar subsets $C_1 \cup Y_1, C_2 \cup Y_2$ such that $S(C_1 \cup Y_1) = S(C_2 \cup Y_2)$.

Similar sets with the same leaf roots

Lemma
If G admits a k-leaf root of maximum degree $d > 2^{s(k)}$, then G contains two similar subsets $C_1 \cup Y_1, C_2 \cup Y_2$ such that $S(C_1 \cup Y_1) = S(C_2 \cup Y_2)$.

- Proof idea.
- There are $s(k)$ layer-encoded k-leaf roots, and so $2^{s(k)}$ possible values for $S(Ci \cup Yi)$.
- If G has a k-leaf root with $d > 2^{s(k)}$, we know that we can find $> 2^{s(k)}$ pairwise similar subsets.
- By the pigeonhle principle, $S(Ci \cup Y_i) = S(C_j \cup Y_j)$ holds for two of them.
  - (just reindex them so that $i = 1, j = 2$)

Similar sets with the same leaf roots

So far, we know that:

Lemma
If G admits a k-leaf root of maximum degree $d > 2^{s(k)}$, then G contains two similar subsets $C_1 \cup Y_1, C_2 \cup Y_2$ such that $S(C_1 \cup Y_1) = S(C_2 \cup Y_2)$.

This is useful because:

Lemma
Let $C_1 \cup Y_1, C_2 \cup Y_2$ be similar subsets and assume that $S(C_1 \cup Y_1) = S(C_2 \cup Y_2)$.
Then G is a k-leaf power if and only if $G - (C_1 \cup Y_1)$ is a k-leaf power.

Similar sets with the same leaf roots

Lemma
Let $C_1 \cup Y_1, C_2 \cup Y_2$ be similar subsets and assume that $S(C_1 \cup Y_1) = S(C_2 \cup Y_2)$.
Then G is a k-leaf power if and only if $G - (C_1 \cup Y_1)$ is a k-leaf power.

Similar sets with the same leaf roots

Lemma
Let $C_1 \cup Y_1, C_2 \cup Y_2$ be similar subsets and assume that $S(C_1 \cup Y_1) = S(C_2 \cup Y_2)$.
Then G is a k-leaf power if and only if $G - (C_1 \cup Y_1)$ is a k-leaf power.

- Proof idea.
- If $G - (C_1 \cup Y_1)$ is not a k-leaf power, then neither is G. So assume that $G - (C_1 \cup Y_1)$ has a k-leaf root T.
- Look at $T_2 = (T$ restricted to $C_2 \cup Y_2)$. Now, $C_1 \cup Y_1$ admits a k-leaf root T_1 with the same layer-encoding as T_2.
- Embed T_1 into T by mimicking T_2. The result is a k-leaf power of G.
- This works because $C_1 \cup Y_1$ and $C_2 \cup Y_2$ are layered similarly, and because layer-encoding capture all relevant neighborhood and distance information.

Similar sets with the same leaf roots

Lemma
If G admits a k-leaf root of maximum degree $d > 2^{s(k)}$, then G contains two similar subsets $C_1 \cup Y_1, C_2 \cup Y_2$ such that $S(C_1 \cup Y_1) = S(C_2 \cup Y_2)$.

Lemma
Let $C_1 \cup Y_1, C_2 \cup Y_2$ be similar subsets and assume that $S(C_1 \cup Y_1) = S(C_2 \cup Y_2)$.
Then G is a k-leaf power if and only if $G - (C_1 \cup Y_1)$ is a k-leaf power.

Similar sets with the same leaf roots

Lemma
If G admits a k-leaf root of maximum degree $d > 2^{s(k)}$, then G contains two similar subsets $C_1 \cup Y_1, C_2 \cup Y_2$ such that $S(C_1 \cup Y_1) = S(C_2 \cup Y_2)$.

Lemma
Let $C_1 \cup Y_1, C_2 \cup Y_2$ be similar subsets and assume that $S(C_1 \cup Y_1) = S(C_2 \cup Y_2)$.
Then G is a k-leaf power if and only if $G - (C_1 \cup Y_1)$ is a k-leaf power.

Theorem
There is f such that if G admits a k-leaf root of max degree $d > f(k)$, then G contains a subset C of vertices such that ***G* is a *k*-leaf power if and only if *$G - C$* is a *k*-leaf power**.
Moreover, C can be found in time $O(n^{f(k)})$ if it exists.

Finding the redundant C

- To find the redundant $C = C_1 \cup Y_1$:
  - Enumerate every subset C_1, …, C_d of size at most at most d^k each, where $d = 2^{s(k)}$. This most time-consuming part takes $O(n^{2^{s(k)}})$.
  - Find the Y_i's by looking at $G - Ci$.
  - Check if the $C_i \cup Y_i$ form pairwise-similar sets (brute force every layering).
  - For each $C_i \cup Y_i$, compute the set of layer-encoded k-leaf roots to obtain $S(Ci \cup Y_i)$. This can be done by DP on the tree decomposition.
  - Find two equal $S(Ci \cup Y_i)$ sets.

Wrapping it up

- If G admits a k-leaf root of low degree, “easy”.
- If G has a k-leaf root T of high degree d:
  - High degree node of T implies many similarly layered $C_i \cup Y_i$’s
  - We can layer-encode the k-leaf roots of each $C_i \cup Y_i$
  - There are $s(k)$ possible layer-encoded k-leaf roots.
  - If d is large enough, two $C_i \cup Y_i$ and $C_j \cup Y_j$ admit the same layer-encoded k-leaf roots.
  - If this is the case, $C_i \cup Y_i$ is redundant because it can mimick $C_j \cup Y_j$. We can remove it without losing information.

What's next?

- Can the ridiculous $n^{f(k)}$ complexity be improved? Or is the power tower behavior necessary?
- Is k-leaf power recognition FPT in k? i.e. $f(k) * poly(n)$ algorithm?

- Techniques applicable to leaf powers? (not sure)
- Techniques applicable to other tree-definable graph classes?
  - e.g. PCGs with bounded interval.

- Graph-theoretical characterization of k-leaf powers?
  - ad hoc analysis for low degree, higher degree = redundancy

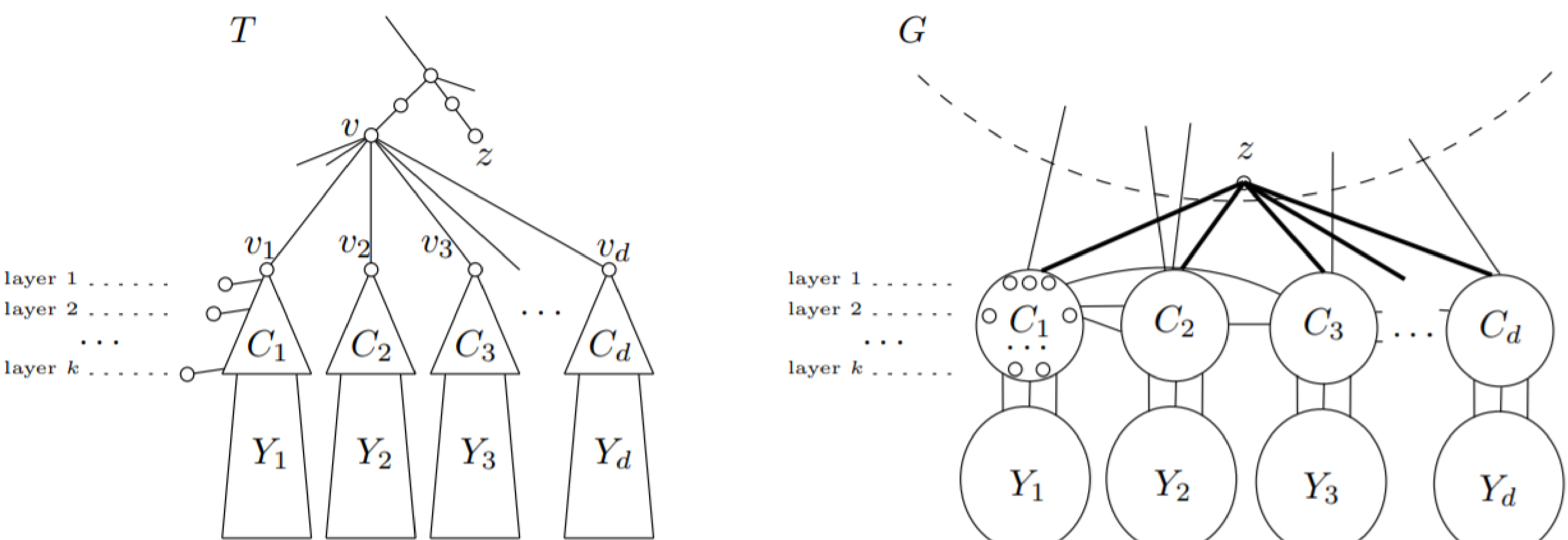
Theorem
There is f such that if G admits a k-leaf root of max degree $d > f(k)$, then G contains a subset C of vertices such that ***G* is a *k*-leaf power if and only if *$G - C$* is a *k*-leaf power**.
Moreover, C can be found in time $O(n^{f(k)})$ if it exists.

This is proved as follows:

1. Show that if a k-leaf root has degree $> d$, one can find subsets C1 U Y1, …, Cd U Yd, such that Ci cuts Yi from the rest of G.
2. Moreover, C1 U C2 U … U Cd can be partitioned into layers that have the same neighborhood in G – (C1 U Y1 U … U Cd U Yd).
3. Moreover again, G[C1 U Y1] admits the same set of encoded k-leaf roots as some G[Ci U Yi] (to be defined).
4. Find a k-leaf root T of G – (C1 U Y1). If none exists, we are done. Otherwise, look at how Ci U Yi is organized in T. By (3), C1 U Y1 allows the same k-leaf root organization. We embed C1 U Y1 into T by mimicking C2 U Y2. By (2), this works.

This is proved as follows:

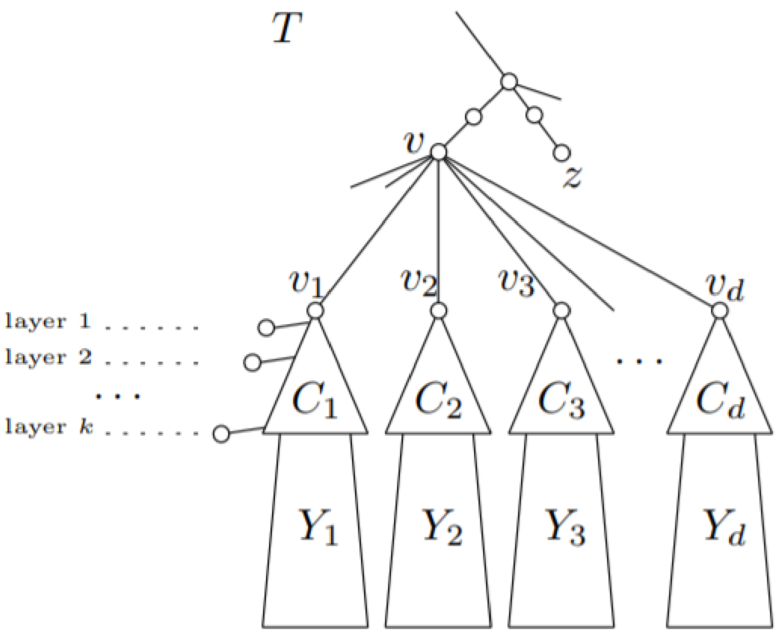
1. Show that if a k-leaf root has degree > d, one can find subsets C1 U Y1, …, Cd U Yd, such that Ci cuts Yi from the rest of G.
2. Moreover, C1 U C2 U … U Cd can be partitioned into layers that have the same neighborhood in G – (C1 U Y1 U … U Cd U Yd).
3. If d is large, some G[Ci U Yi] and G[Cj U Yj] admit the same set of encoded k-leaf roots (to be defined).
4. Find a k-leaf root T of G – (Ci U Yi). Look at how Cj U Yj is organized in T. By (3), Ci U Yi allows the same k-leaf root organization. We embed Ci U Yi into T by mimicking Cj U Yj. By (2), this works.

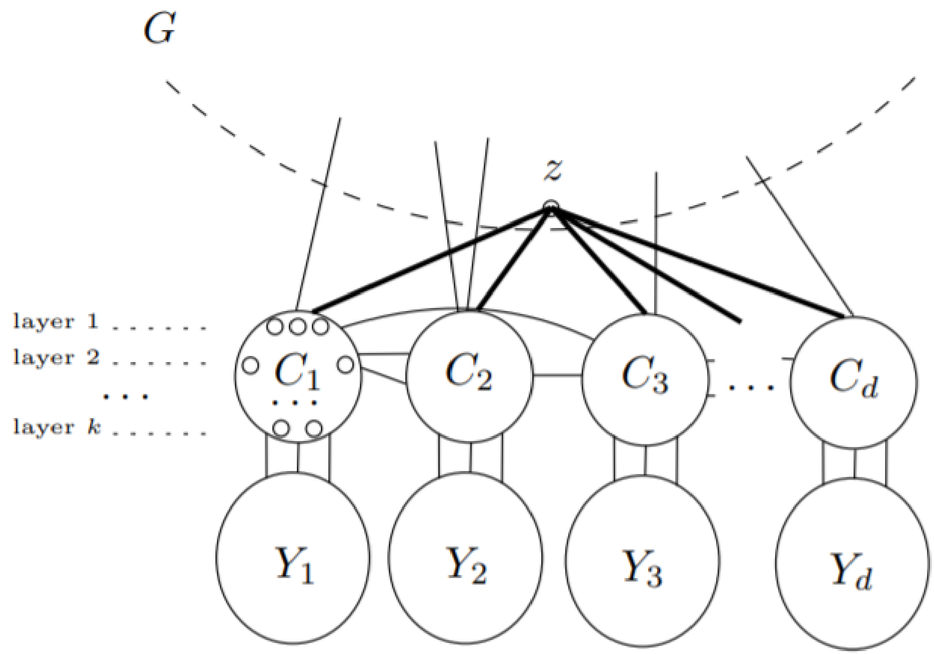


k-leaf roots with high degree

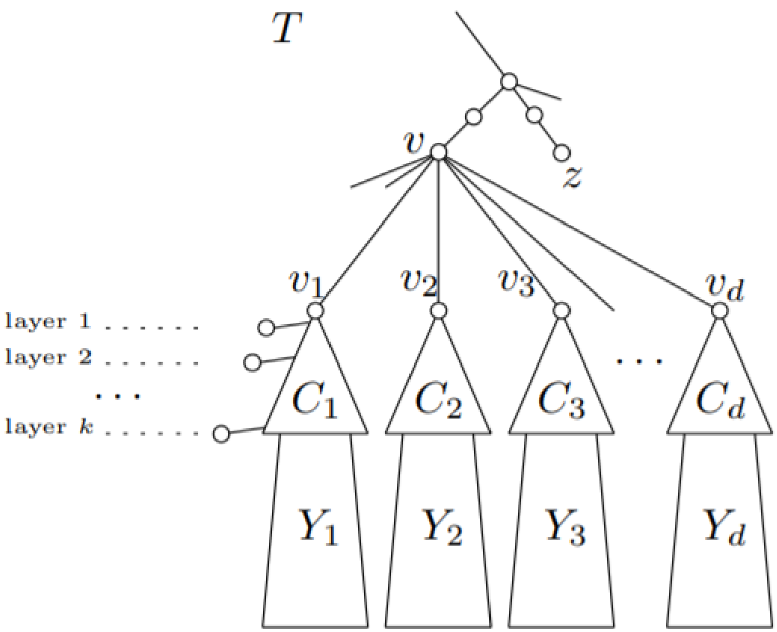
Theorem
There is f such that if G admits a k-leaf root of max degree $d > f(k)$, then G contains a subset C of vertices such that **G is a *k*-leaf power if and only if $G - C$ is a *k*-leaf power**.
Moreover, C can be found in time $O(n^{f(k)})$ if it exists.

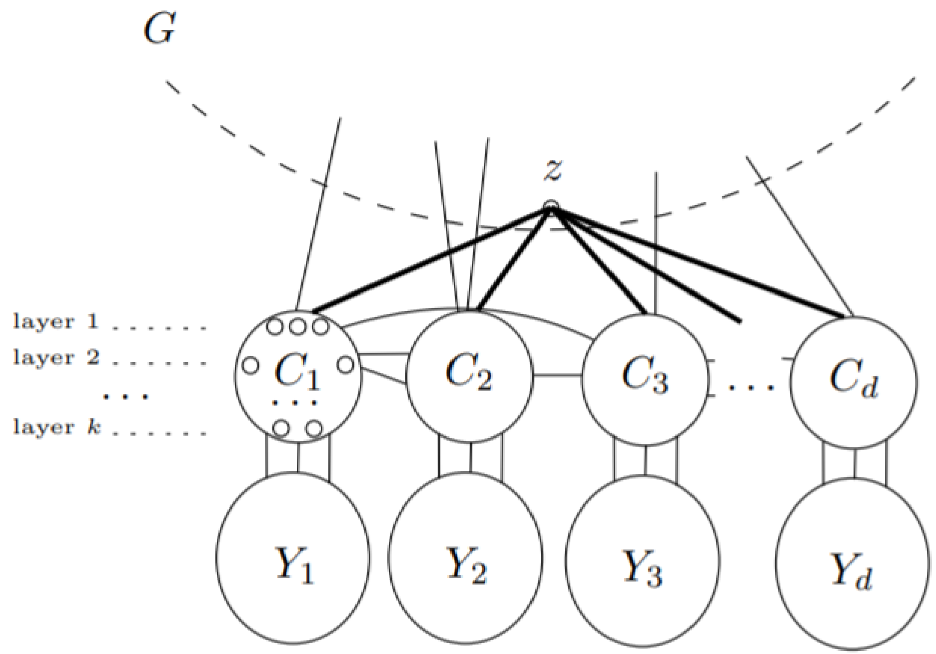
- T = leaf root of G
- v = lowest max of degree >d
- z = closest leaf to v
- Ci = subtrees at distance <= k from v
- Layer j = leaves at distance j from v





- Of course, we don't have T. Still, by brute-force we can find the C_i's and Y_i's that satisfy the cutset, size and layering properties. This is feasible since the C_i's have bounded size.





3.1 Similar structures A *similar structure* of a graph G is a tuple $\mathcal{S} = (\mathcal{C}, \mathcal{Y}, z, \mathcal{L})$ where:

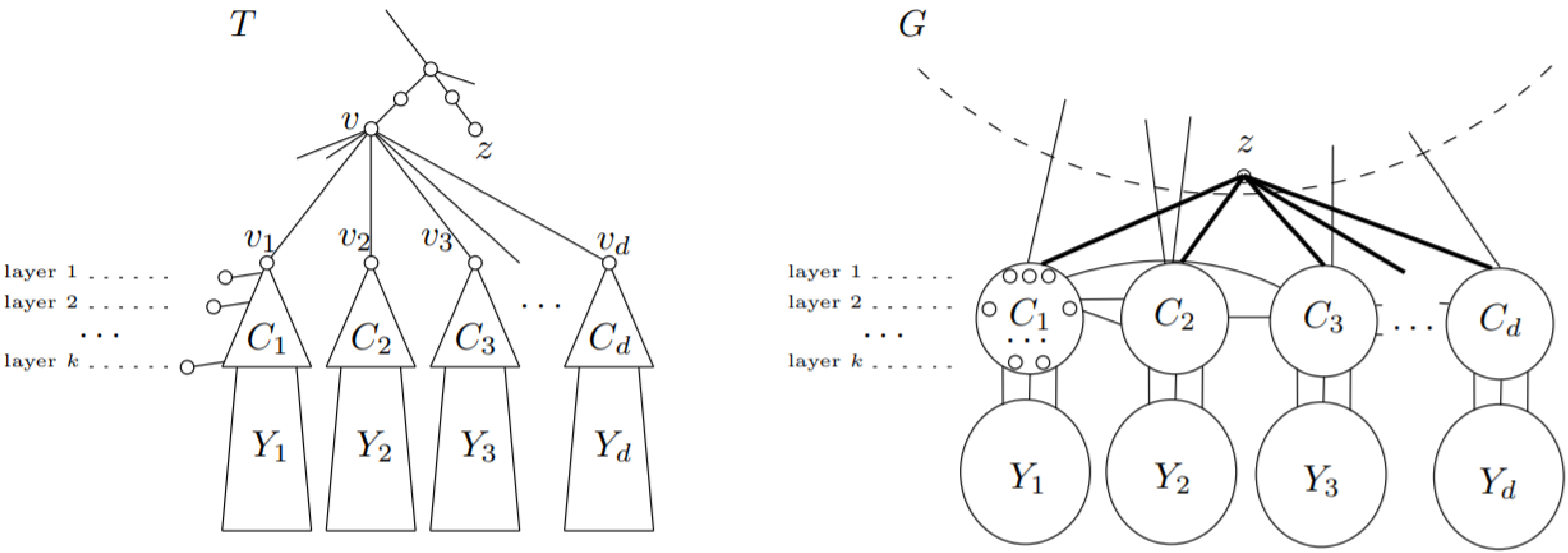
- $\mathcal{C} = \{C_1, \ldots, C_d\}$ is a collection of $d \geq 2$ pairwise disjoint, non-empty subsets of vertices of G;
- $\mathcal{Y} = \{Y_1, \ldots, Y_d\}$ is a collection of pairwise disjoint subsets of vertices of G, some of which are possibly empty. Also, $C_i \cap Y_j = \emptyset$ for any $i, j \in [d]$;
- $z \in V(G)$ and does not belong to any subset of $\mathcal{C}$ or $\mathcal{Y}$;
- $\mathcal{L} = \{\ell_1, \ldots, \ell_d\}$ is a set of functions where, for each $i \in [d]$, we have $\ell_i : C_i \cup \{z\} \to \{0, 1, \ldots, k\}$. The functions in $\mathcal{L}$ are called *layering functions.*

Additionally, $\mathcal{S}$ must satisfy several conditions. Let us denote $C^* = \bigcup_{i \in [d]} C_i$. Let $X = \{X_1, \ldots, X_t\}$ be the connected components of $G - C^*$. For each $i \in [d]$, denote $X^{(i)} = \{X_j \in X : N_G(X_j) \subseteq C_i\}$, i.e. the components that have neighbors only in C_i.

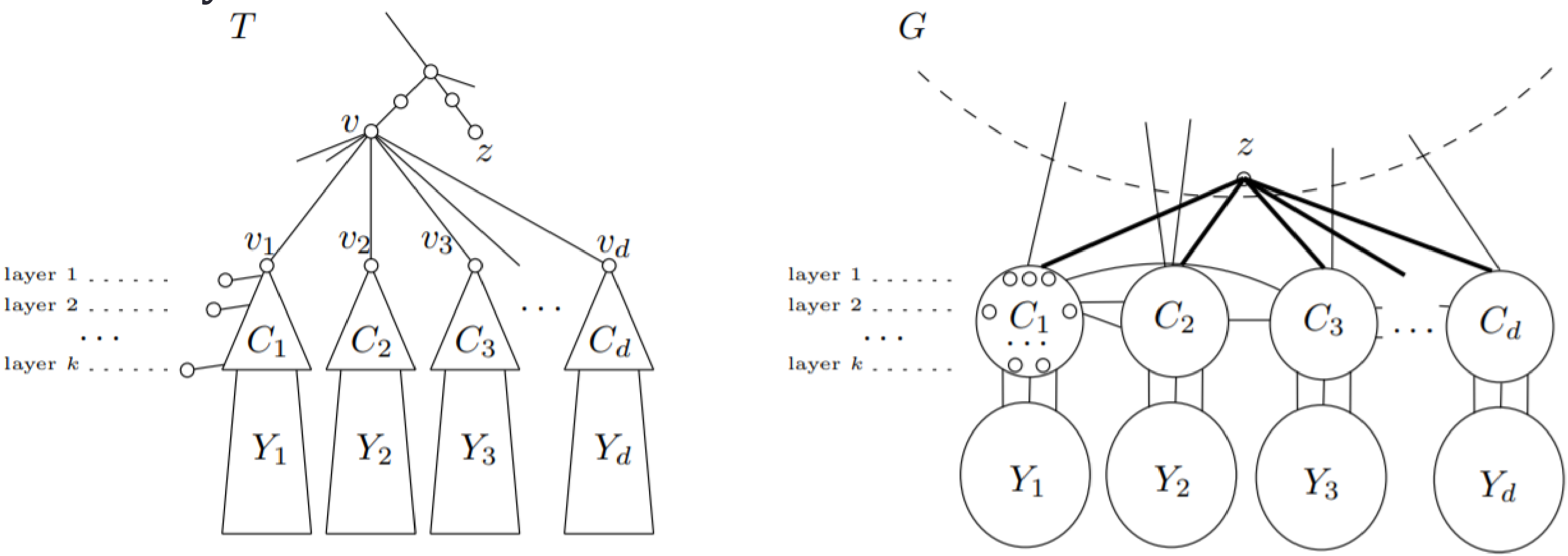
Then all the following conditions must hold:

1. for each $i \in [d]$, $Y_i = \bigcup_{X_j \in X^{(i)}} X_j$ ($Y_i = \emptyset$ is possible);
2. there is exactly one connected component $X_z \in X$ such that for all $i \in [d]$, $N_G(X_z) \cap C_i \neq \emptyset$. Moreover, $z \in X_z$ and $C^* \subseteq N_G(z)$;
3. for all $X_j \in X \setminus \{X_z\}$, $X_j \subseteq Y_i$ for some $i \in [d]$. In particular, X_z is the only connected component of $G - C^*$ with neighbors in two or more C_i's;
4. the layering functions $\mathcal{L}$ satisfy the following:
   (a) for each $i \in [d]$, $\ell_i(z) = 0$. Moreover, $\ell_i(x) > 0$ for any $x \in C_i$;
   (b) for any $i, j \in [d]$ and any $x \in C_i, y \in C_j$, $\ell_i(x) = \ell_j(y)$ implies $N_G(x) \setminus (C_i \cup Y_i \cup C_j \cup Y_j) = N_G(y) \setminus (C_i \cup Y_i \cup C_j \cup Y_j)$. Note that this includes the case $i = j$;
   (c) for any $i, j \in [d]$ and any $x \in C_i, y \in C_j$, $\ell_i(x) + \ell_j(y) \leq k$ implies $xy \in E(G)$. Note that this includes the case $i = j$.
   (d) for any *two distinct* $i, j \in [d]$ and any $x \in C_i, y \in C_j$, $\ell_i(x) + \ell_j(y) > k$ implies $xy \notin E(G)$. Note that this does *not* include the case $i = j$

- Of course, we don't have T. Still, by brute-force we can find the C_i's and Y_i's that satisfy the cutset, size and layering properties. This is feasible since the C_i's have bounded size.



- Of course, we don't have T. Still, by brute-force we can find the C_i's and Y_i's that satisfy the cutset, size and layering properties. This is feasible since the C_i's have bounded size.
- Look at the k-leaf roots of each G[Ci U Yi].
- WANT : two G[Ci U Yi] and G[Cj U Yj] that admit the same set of layer-encoded k-leaf roots.



- WANT : two G[Ci U Yi] and G[Cj U Yj] that admit the same set of layer-encoded k-leaf roots.