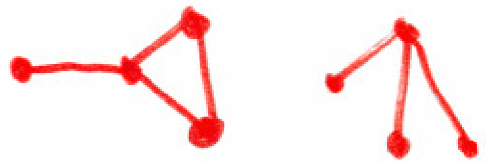
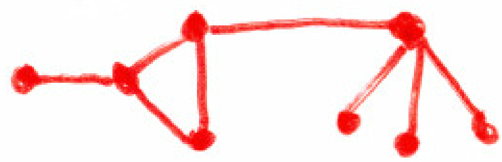
Recall:

def A **tree** is a connected, acyclic graph.

Ex:

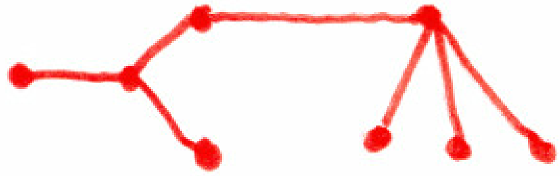


neither connected nor acyclic



Connected but not acyclic

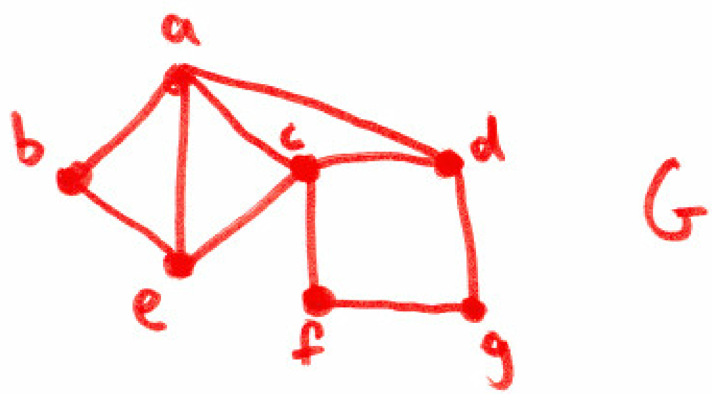
acyclic but not connected



a tree

def A **dominating set** in a graph G is a set $S \subseteq V(G)$ of vertices such that every vertex is either in S or adjacent to a vertex in S.

Ex:



- $\{a,d\}$ is not a dominating set because $f \notin \{a,d\}$ and none of the neighbors $N(f)=\{c,g\}$ are in $\{a,d\}$.
- $\{a,c,d\}$ is a dominating set
- Can you find a dominating set of size 2?

<u>WARNING!!</u> The following "proof" has an error. Can you find it?

"<u>Thm</u>" If T is an n-vertex tree, then T contains a dominating set of size at most $\frac{n+2}{3}$.

"<u>Pf</u>": 1. By induction on n.

2. <u>Base cases</u>: if $n=1$, $T = \bullet$. If $n=2$, then $T =$ . If $n=3$, then $T =$ . In each case, T has a dominating set of size $1 \leq \frac{n+2}{3}$, so the statement holds.

3. <u>Inductive Step</u>: Let $n \geq 4$.

4. Because T is a tree on at least 4 vertices, T contains a vertex u with $d(u) \geq 2$.

5. If $N(u) = V(T) - \{u\}$, then $S = \{u\}$ is a dominating set of size $1 \leq \frac{n+2}{3}$.

6. Otherwise T contains a vertex $v \neq u$ that is not adjacent to u.

7. Let $T' = T - u - N(u)$. That is, T' is the tree obtained from T by deleting u and all of u's neighbors.

8. Note that T' has $n-(1+d(u))$ vertices.

9. Because $d(u) \geq 2$, $|V(T')| \leq n-3$.

10. Because $v \in V(T')$, $|V(T')| \geq 1$.

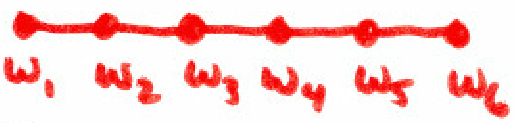
11. Therefore the inductive hypothesis implies that T' has a dominating set S' of size at most $\frac{|V(T')|+2}{3} \leq \frac{(n-3)+2}{3} = \frac{n+2}{3} - 1$.

12. Now $S = S' \cup \{u\}$ is a dominating set of size at most $\frac{n+2}{3}$:

12(a). • Vertices in T' are taken care of by S'.

12(b). • All other vertices are either neighbors of u or u itself, and are taken care of by u. ∎

Did you spot the error? Let's try an example:

<u>Ex</u> $T = P_6 =$ 

The ~~theorem~~ proof claims to show us how to find a dominating set in P_6 of size at most $\frac{6+2}{3} = \frac{8}{3} = 2.666\cdots$, so it must find a dominating set of size at most 2.

Let's run our proof on $T = P_6$.

Because P_6 has $6 \geq 4$ vertices, the inductive step applies.

In (4), the proof asserts that P_6 contains a vertex of degree at least two and gives it a name u. The proof does not assert any other properties about u, so our proof must work if we set u to be any vertex of degree at least 2.

Let us try $u = w_3$. Next, w_3 is not dominating, so (5) ~~does~~ does not apply.

Similarly, in (6) our proof must work if we choose v to be any vertex that is not u or adjacent to u; let us pick $v = w_6$.

On to step (7). We set T' to be the graph obtained from P_6 by deleting $u = w_3$ and its neighbors $N(u) = \{w_2, w_4\}$.

So:

$T' =$ 

__AHA!!__ A problem: T' is __not__ a tree! So step (7) is wrong to assert that T' is a tree.

But what if in step (7) we demote T' from a tree to a plain old graph? Would our proof then be correct?

Steps (8)–(10) still go through OK:

$$1 \le |V(T')| \le n-3$$

But now, step (11) presents a major problem: it invokes the inductive hypothesis on T' to obtain a dominating set S' of size at most $\frac{3+2}{3} = 1.666\cdots$. The smallest dom. sets in T' all have size 2.

Even though T' is a <u>graph</u> with fewer vertices than T, we cannot invoke the inductive hypothesis on T' because T' is <u>not</u> a smaller instance/input to our theorem: all inputs to our theorem are <u>trees</u>.

Our theorem does not claim to hold for T' and, in fact it does not.

But wait. Maybe we can pick the vertex u more carefully, back in step (4). (If we pick $u = w_2$, we're OK.)

Our proof would work if it was the case that we could find a vertex u with $d(u) \geq 2$ such that

$$T' = T - u - N(u)$$

was still a tree. Because T' would necessarily be acyclic, all we have to do is worry about whether T' is connected. Does such a vertex always exist? If so, we can fix our proof.

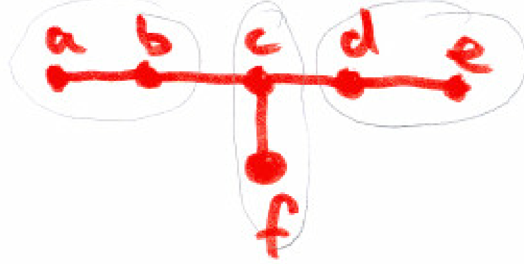
This also tells us that a minimal counterexample to the "theorem" must have the property that $\forall u \; d(u) \geq 2$, $T - u - N(u)$ is disconnected.

This is a common situation:

⇒ Understanding where a "proof" of a statement fails gives you information about how to find a counterexample.

⇒ Understanding why the statement is true of some examples gives you information about how to find a proof.

In our case, the statement of the "theorem" is <u>false</u>:

$T =$ 

is a tree whose minimum dominating sets have size at least $3 > \frac{n+2}{3}$.

Recall: A rooted tree is a tree with a distinguished vertex, called the root.

We can also define a kind of rooted tree recursively:

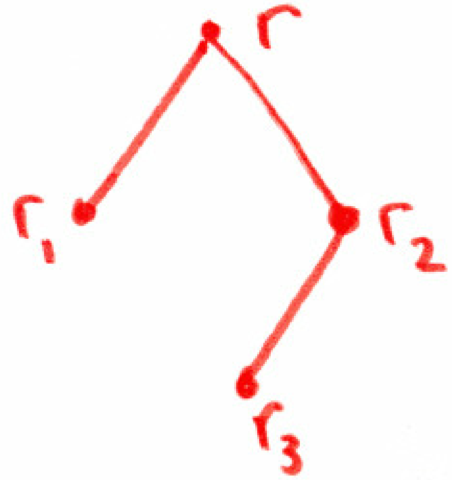
def A k-ary tree T is either

- Empty $T = ()$ (The **null tree**), or
- an ~~vertex~~ ordered list $T = (r, T_1, T_2, \ldots, T_k)$ where r is a vertex/node, called the **root**, and each T_j is a k-ary tree. Each T_j is a **subtree** of T.

A 2-ary tree is called a **binary tree**.

Remark: Trees are very important; many data structures are trees with added structure.

Ex: $T = (r, \underbrace{(r_1, (), ())}_{T_1}, \underbrace{(r_2, (r_3, (), ()), ())}_{T_2})$



- The parent of r_3 is r_2.
- The children of r are $\{r_1, r_2\}$.
- The ancestors of r_3 are $\{r, r_2, r_3\}$
- The descendants of r_2 are $\{r_2, r_3\}$.

More Tree Terminology:

- If $T = (r, T_1, T_2, \ldots, T_k)$ and the root of T_j is r_j, then we say r is the **parent** of r_j and $r_1, r_2, \ldots, r_k$ are **children** of r.
- An **ancestor** of a vertex u is either u or an ancestor of the parent of u. A ~~pro~~ **proper ancestor** of u is an ancestor that is not u.

- A **decendant** of a vertex u is either u or a descendant of a child of u.

  A **proper descendant** of u is a decendant that is not u.

Remark: Recursive structures (like trees) lend themselves to inductive proofs.

def The **depth** of a k-ary tree T is defined recursively:

$$\text{depth}(T) = \begin{cases} -1 & T = () \text{ the null tree} \\ 1 + \max\limits_{T_j}\{\text{depth}(T_j)\} & \text{otherwise} \end{cases}$$

$T = (r, T_1, T_2, \ldots, T_k)$

Note: $\text{depth}(T)$ is the maximum distance of a leaf in T to the root of T.

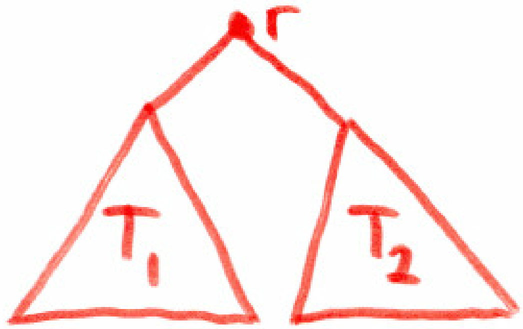
Exercise: Prove this by induction.

__Thm__ If T is a binary tree and $d \geq \text{depth}(T)$, then T has at most $2^{d+1}-1$ vertices.

__Pf:__ By induction on d.

If $d=-1$, then T is the null tree, which has $0 \leq 2^{d+1}-1$ vertices.

If $d \geq 0$, then $T = (r, T_1, T_2)$,



where T_1 and T_2 are binary trees of depth at most $d-1$. By the inductive hypothesis, T_1 has at most $2^{(d-1)+1}-1 = 2^d-1$ vertices. Also, the inductive hypothesis implies T_2 has at most 2^d-1 vertices. Therefore, T has at most $1 + 2^d-1 + 2^d-1 = 2 \cdot 2^d - 1 = 2^{d+1}-1$ vertices. ∎

__Cor__ If T is a binary tree with n vertices, then $\operatorname{depth}(T) \geq \lg(n+1) - 1$.

<u>Pf</u>: Because

$$n \leq 2^{\operatorname{depth}(T)+1} - 1,$$

we have

$$2^{\operatorname{depth}(T)+1} \geq n+1,$$

and therefore

$$\operatorname{depth}(T) + 1 \geq \lg(n+1)$$

so $\operatorname{depth}(T) \geq \lg(n+1) - 1$. ∎