Middlebury

CSCI 200: Math Foundations of Computing

Spring 2026

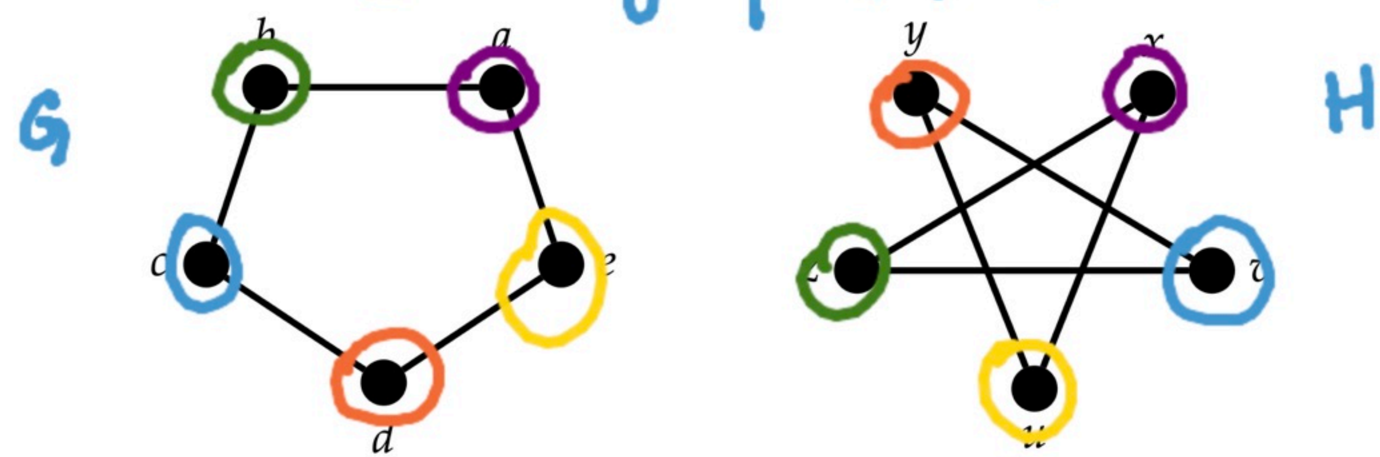
Lecture 8M: Trees

First, a connection betwen functions and graph theory: *isomorphism*.

Definition: an *isomorphism* between two graphs G and H is a bijection $f: V(G) \to V(H)$ such that

$$(u, v) \in E(G) \iff (f(u), f(v)) \in E(H)$$

In words: every edge in first graph (G) can be found in the second graph (H) with relabeled vertices



$f(a) = x$

$f(b) = z$

$f(c) = v$

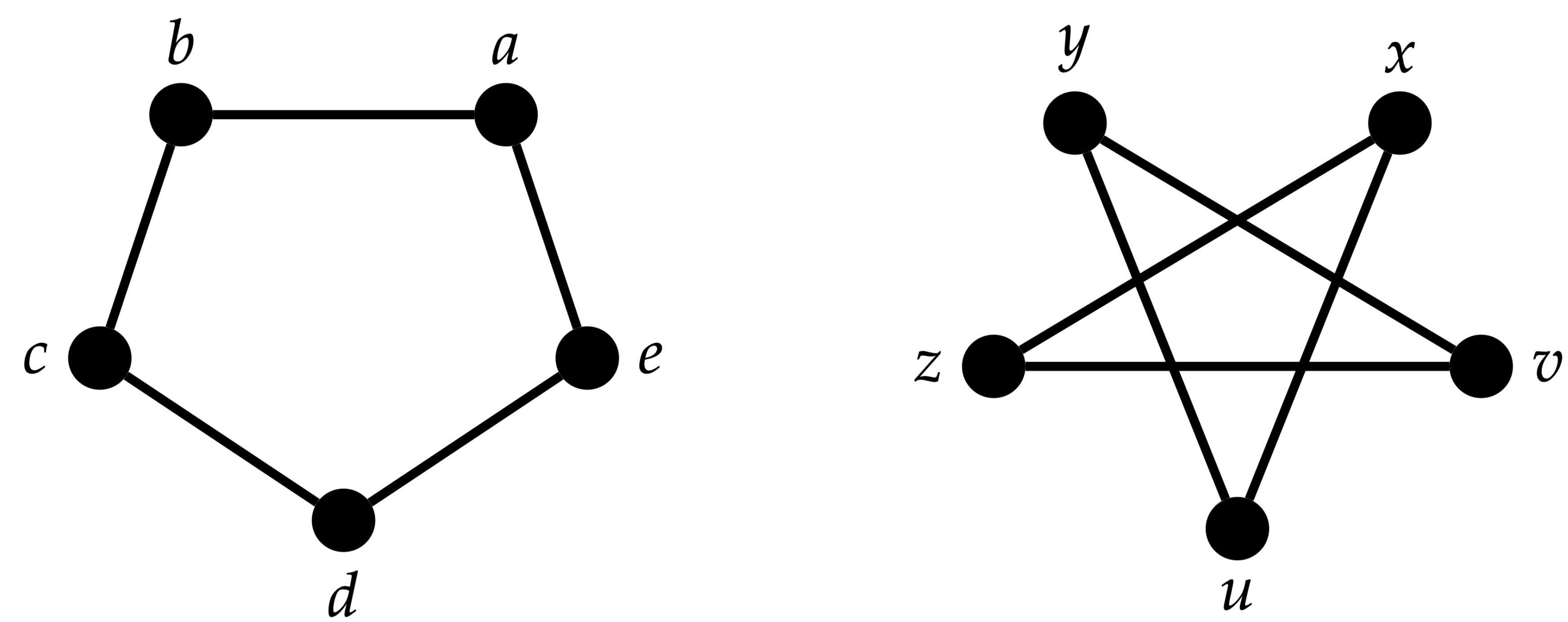
$f(d) = y$

$f(e) = u$

Are these graphs isomorphic? If so, what is a possible relabeling?

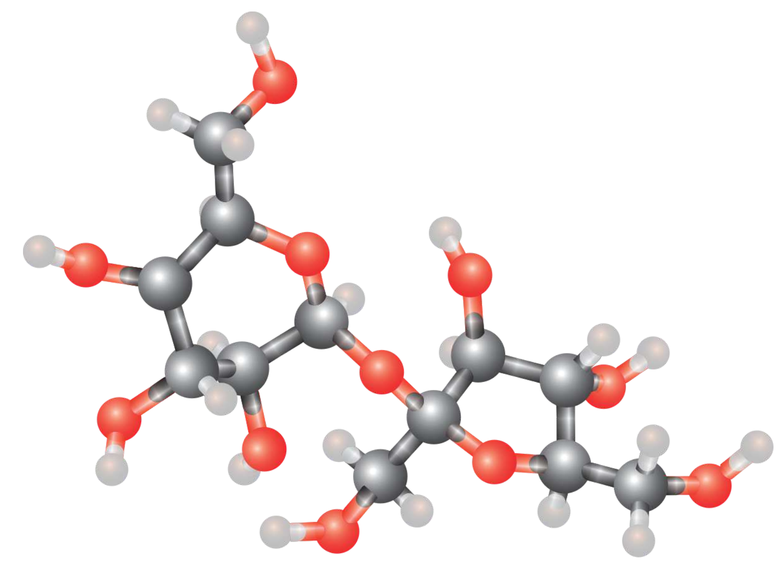
Testing for an isomorphism.

1. Check if the number of vertices and edges are the same.
2. Check all vertex degrees in G are found in H.
3. Check cycle lengths (we'll talk about cycles today) in G are found in H.

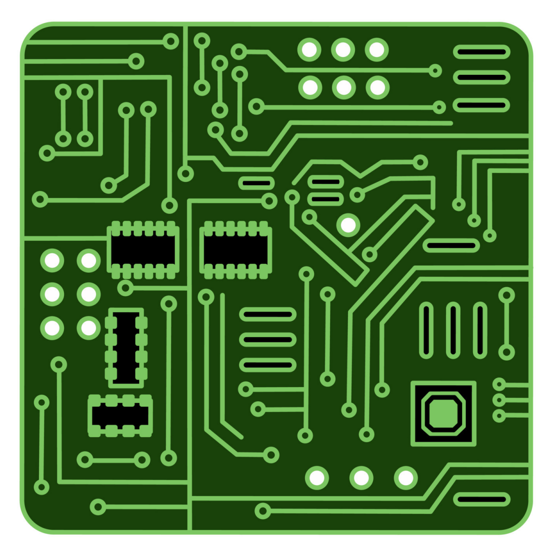


Applications of isomorphisms.

Chemistry: *does this chemical compound exist already?*



Circuits: *is this design protected by intellectual property?*



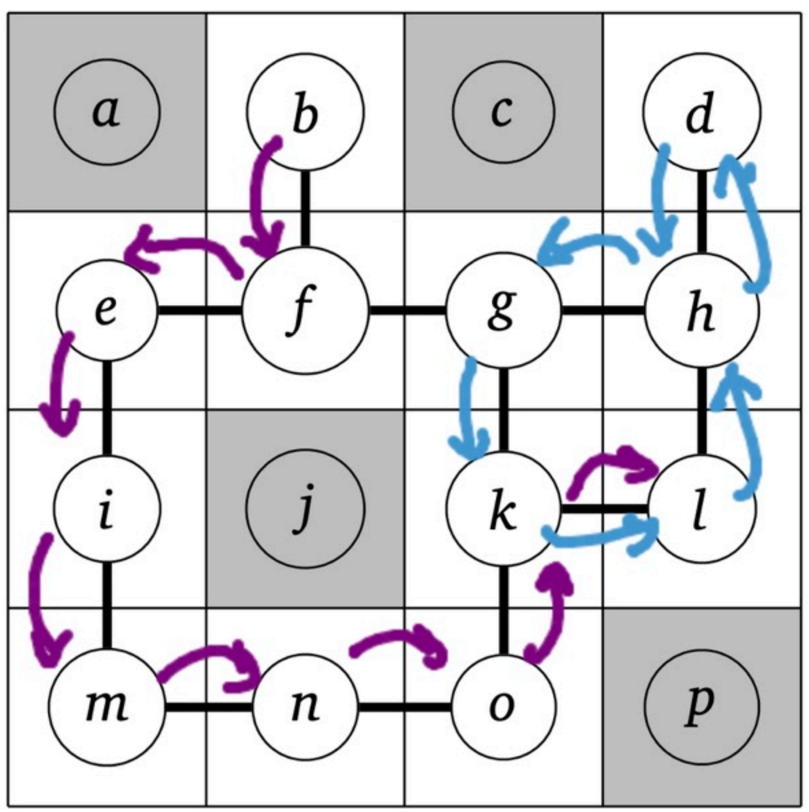
Goals for today:

1. Describe properties of graphs: walks, paths, cycles.
2. Prove the # edges in a tree is equal to # vertices - 1.
3. Build a spanning tree from an arbitrary connected graph.

Walks.

Definition 1: A walk (of length k) in a graph $G = (V, E)$ is a sequence of vertices that are connected by k edges.

Definition 2: A closed walk is a walk which starts and ends at the same vertex.



example:

walk of length 8

① b - f - e - i - m - n - o - k - l

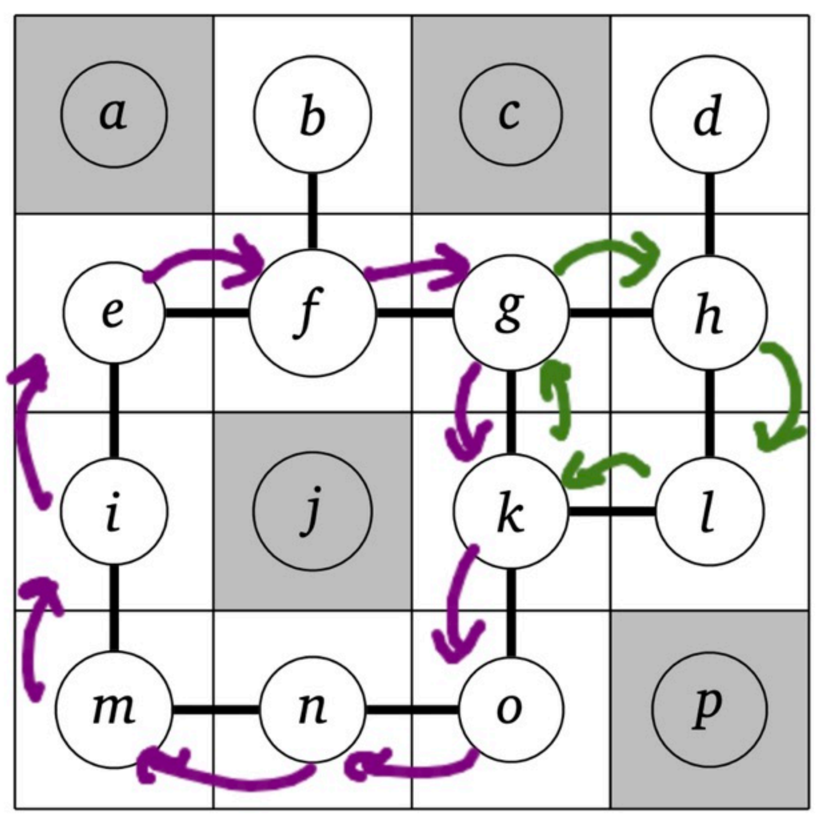
② g - k - l - h - d - h - g

closed walk of length 6.

Paths and Cycles.

Definition 3: A __path__ is a __walk__ where all the vertices are __different__.

Definition 4: A __cycle__ is a __closed__ walk of (length __>2__) in which all the vertices are __different.__.



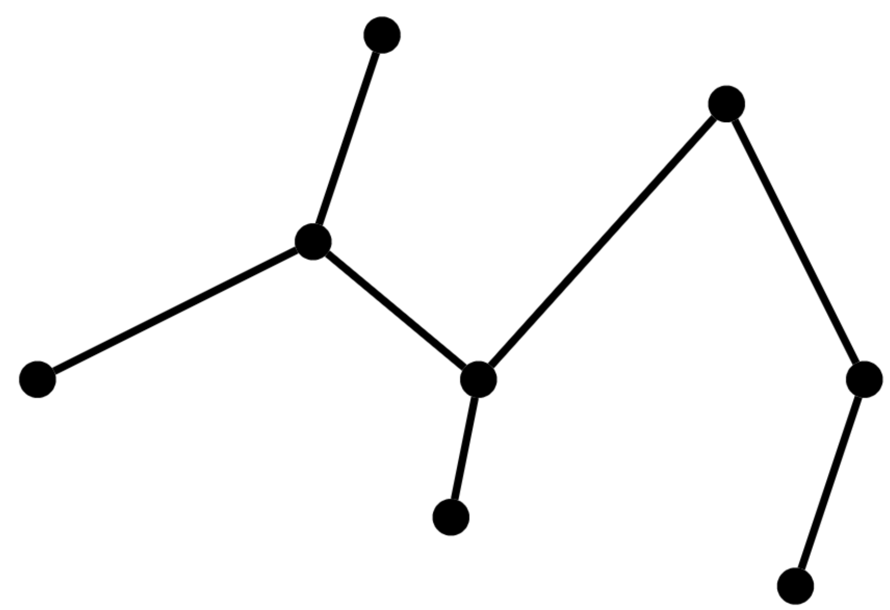
examples:

① e-f-g-k-o-n-m-i-e

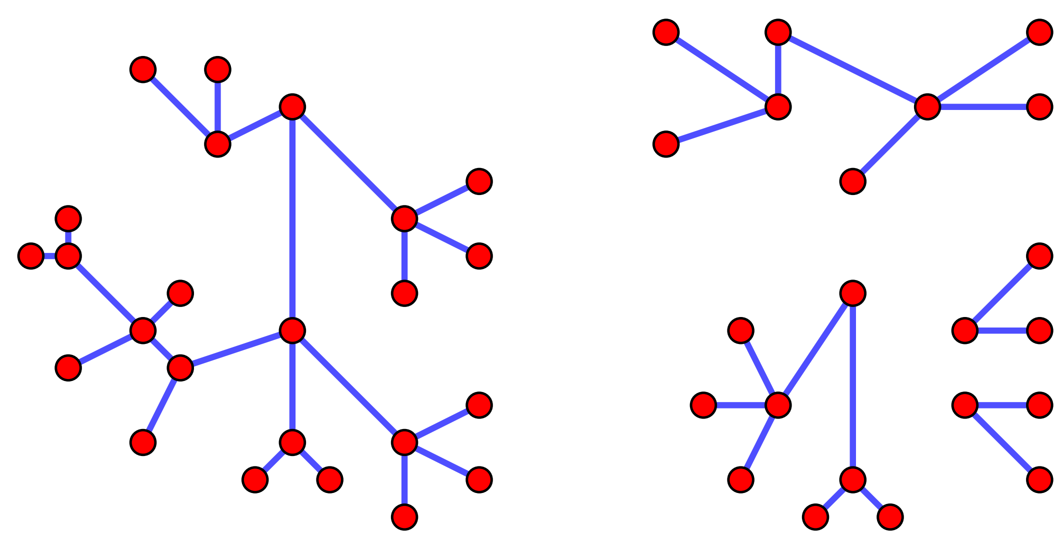
② g-h-l-k-g

Definition 5: Vertices u and v are said to be “connected” if there is a path between u and v.

Finally! A tree is a *connected* graph with no cycles (*acyclic*).



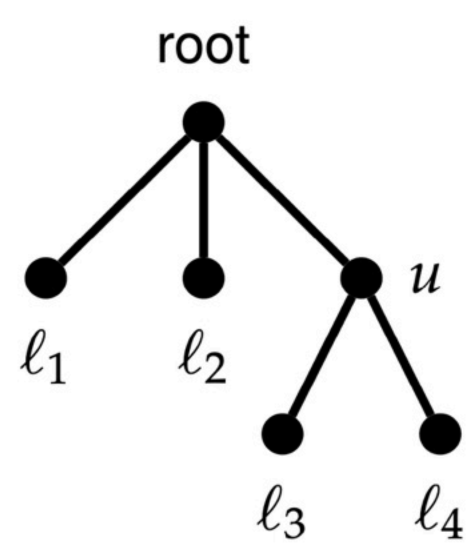
No cycles but not connected? $\longrightarrow$ **Forest.**



Tree Forest

Rooted and k-ary trees.

Definition 6: A ___rooted tree___ is a ___tree___ in which a single vertex is designated as the ___root___ and every edge is directed away from the ___root___.

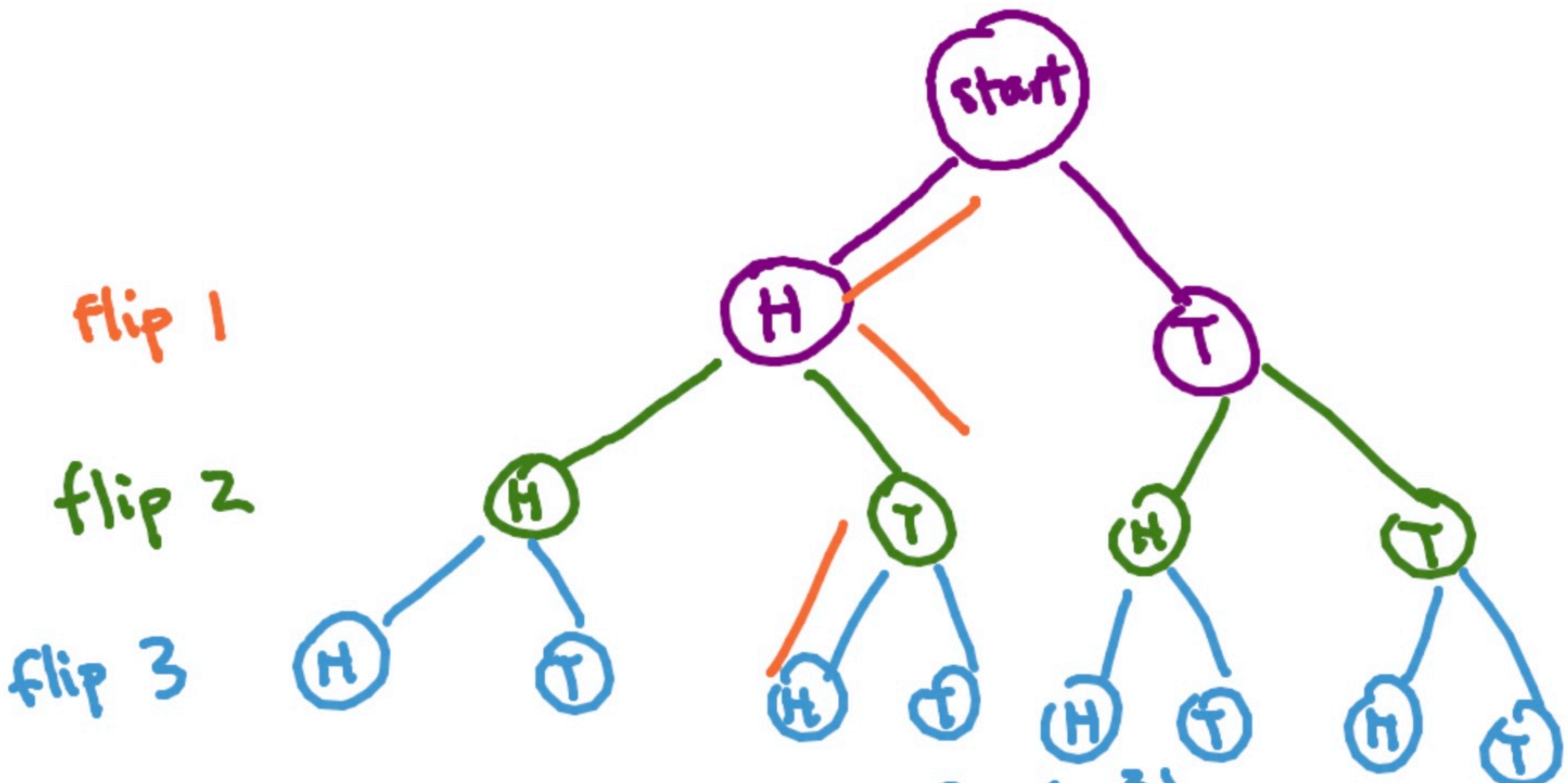


What is k for this tree?

A. 2
B. 3
C. 5
D. 6

Definition 7: A ___k-ary___ is a tree in which each node has ___$\leq k$___ children.

Exercise 1: draw the binary tree resulting from all possible outcomes of flipping a coin three times.

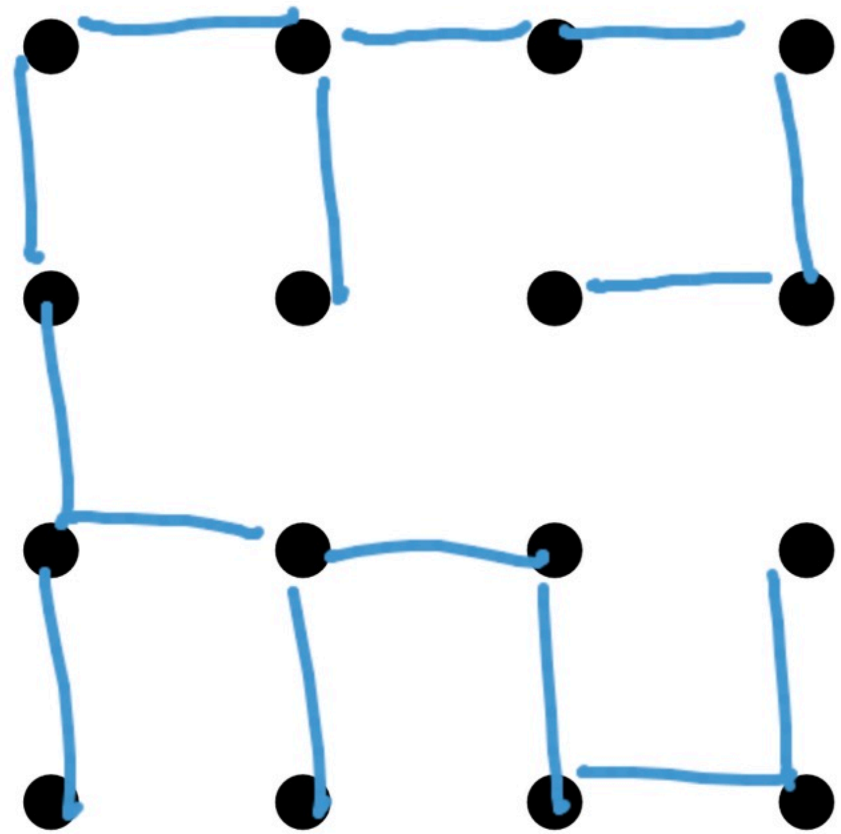


1. How many leaves are there? 8 (2^3)
2. What is the probability of getting the sequence H-T-H (3 flips)? $1/8$
3. What is the probability of getting the sequence H-T-T-T-H-H (6 flips)? $\frac{1}{2^6}$

more on this in a few weeks.

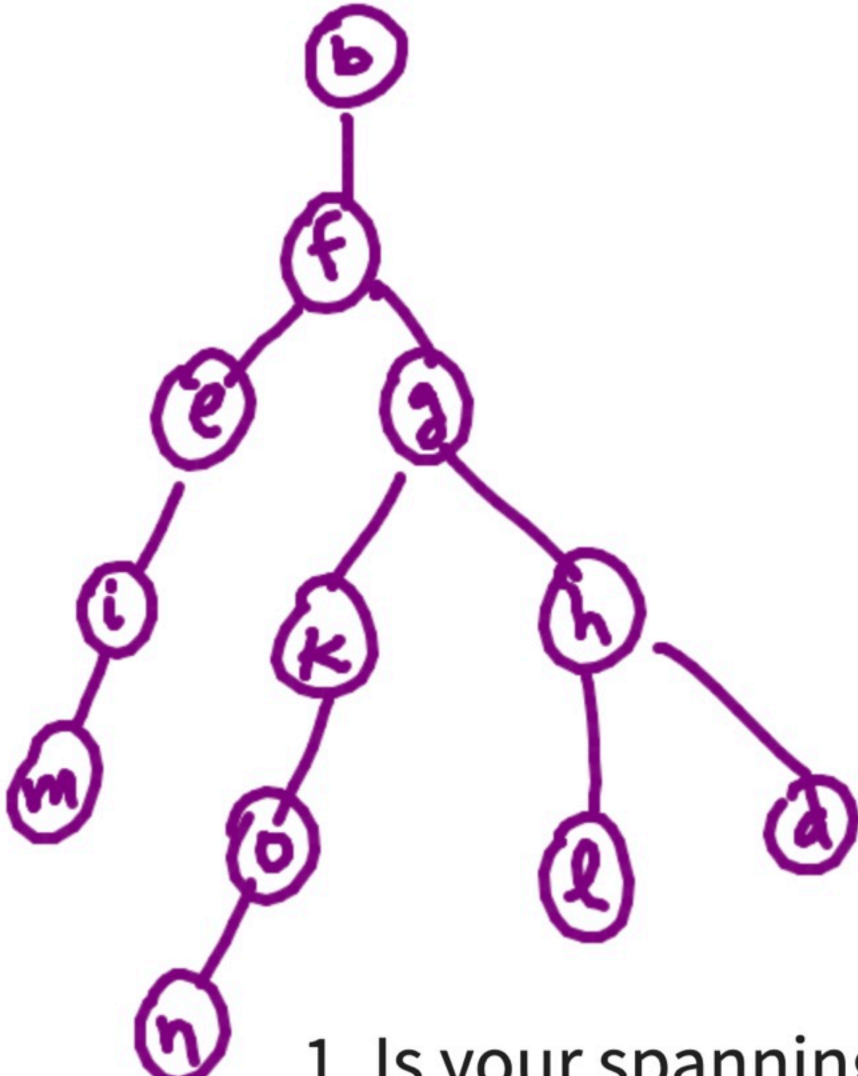
Exercise 2: make a tree that has all of these vertices!

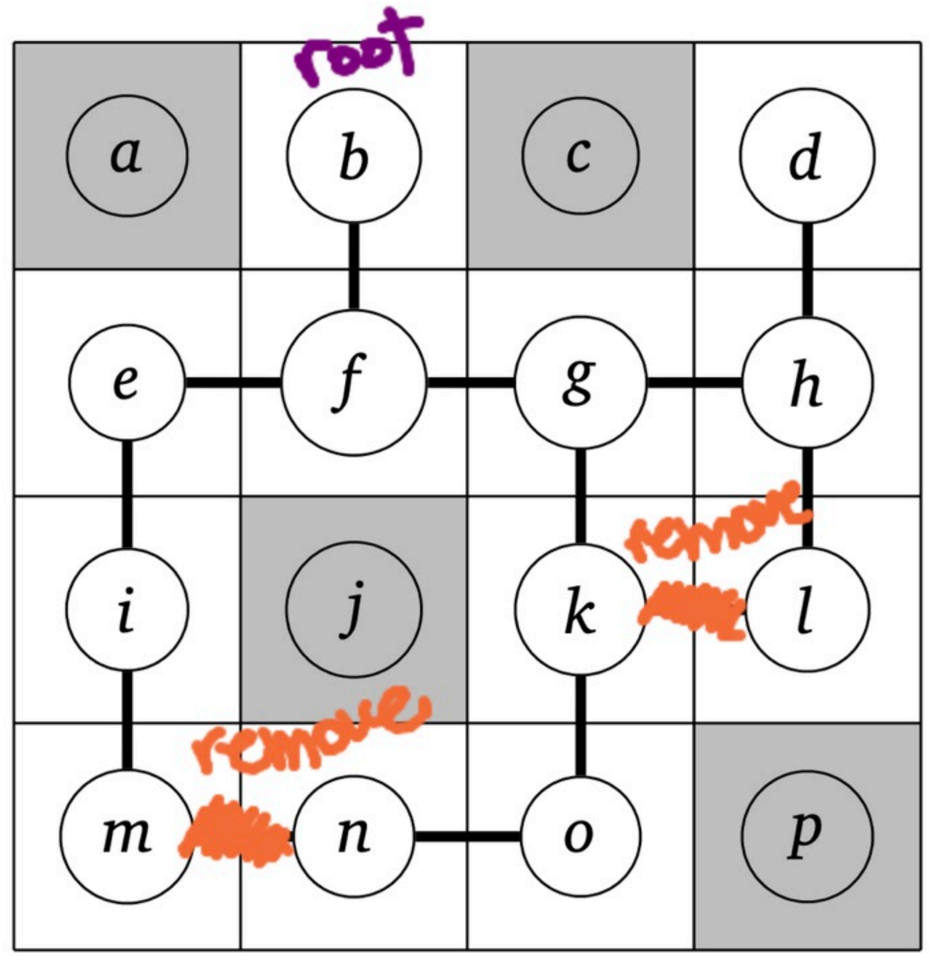
- connected
- no cycles



Definition 8: A spanning tree of a connected graph $G = (V, E)$ is a subgraph of G with the same vertices as G.

Exercise 3: make a spanning tree of this graph.





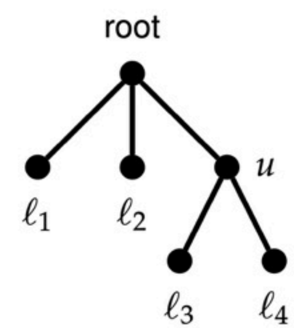
12 vertices

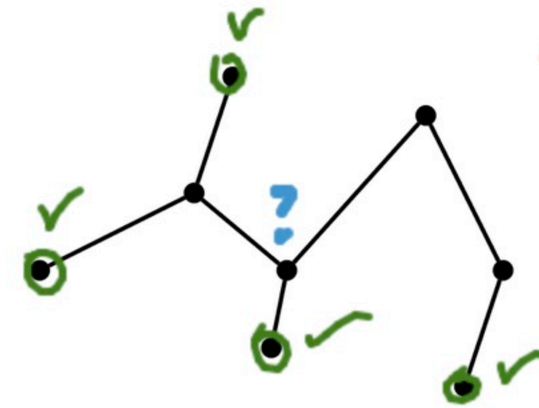
1. Is your spanning tree unique? NO
2. How many edges are there in your tree? 11 edges
3. How many edges in the trees of those around you? should be 11.

An important property of trees.

base case: •

Thm: # edges in a tree = # vertices − 1





say this is the (n+1)-vertex tree.

which vertex can be removed? (to get n-vertex).

We use a proof by induction on the number of vertices n. Let the induction hypothesis be $p(n)$: "there are $n-1$ edges in an n-vertex tree."

- **Base case:** A tree with a single vertex ($n=1$) has no edges, which is consistent with $1-1=0$, so $p(1)$ is true.
- **Inductive step:** Assume $p(n)$ is true. That is, assume a tree with n vertices has $n-1$ edges. We need to show a tree with $n+1$ vertices has n edges. Starting with a tree with $n+1$ vertices, ... remove one vertex to get an n-vertex tree. The vertex removed can only be a leaf, otherwise the resulting tree would not be connected. Removing this leaf u also removes 1 edge. By p(n), the n-vertex tree (from removing u) has n-1 edges. Now, add u back in.

... sketch ... add a leaf (anywhere), which adds 1 edge. maintain a tree

edges in (n+1)-vertex tree = $\underbrace{n-1}_{\text{by } p(n) \text{ n-vertex tree}} + 1 = n$ ✓ verifying p(n+1)

By induction, ... □