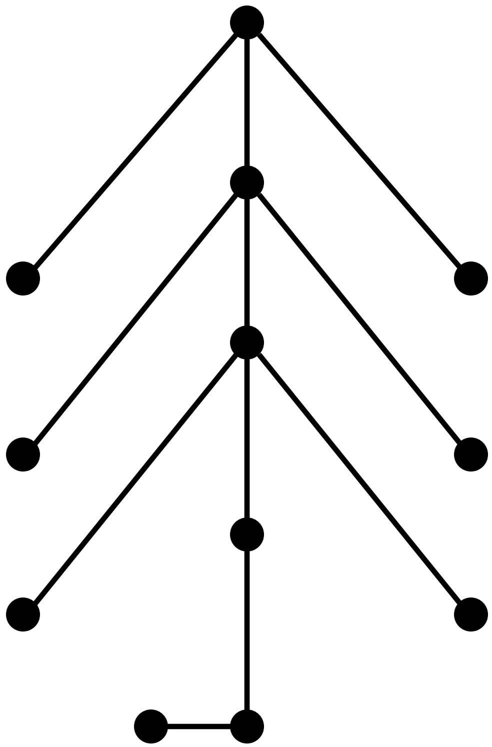
Trees

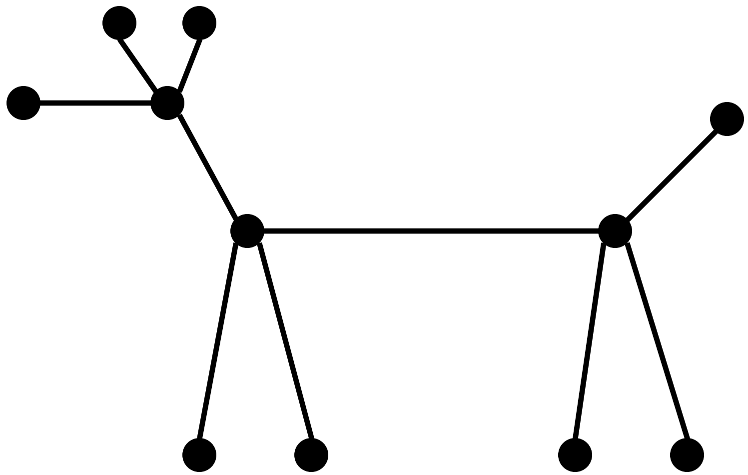
Discrete Mathematics

Courtesy of Andrei Bulatov

Trees

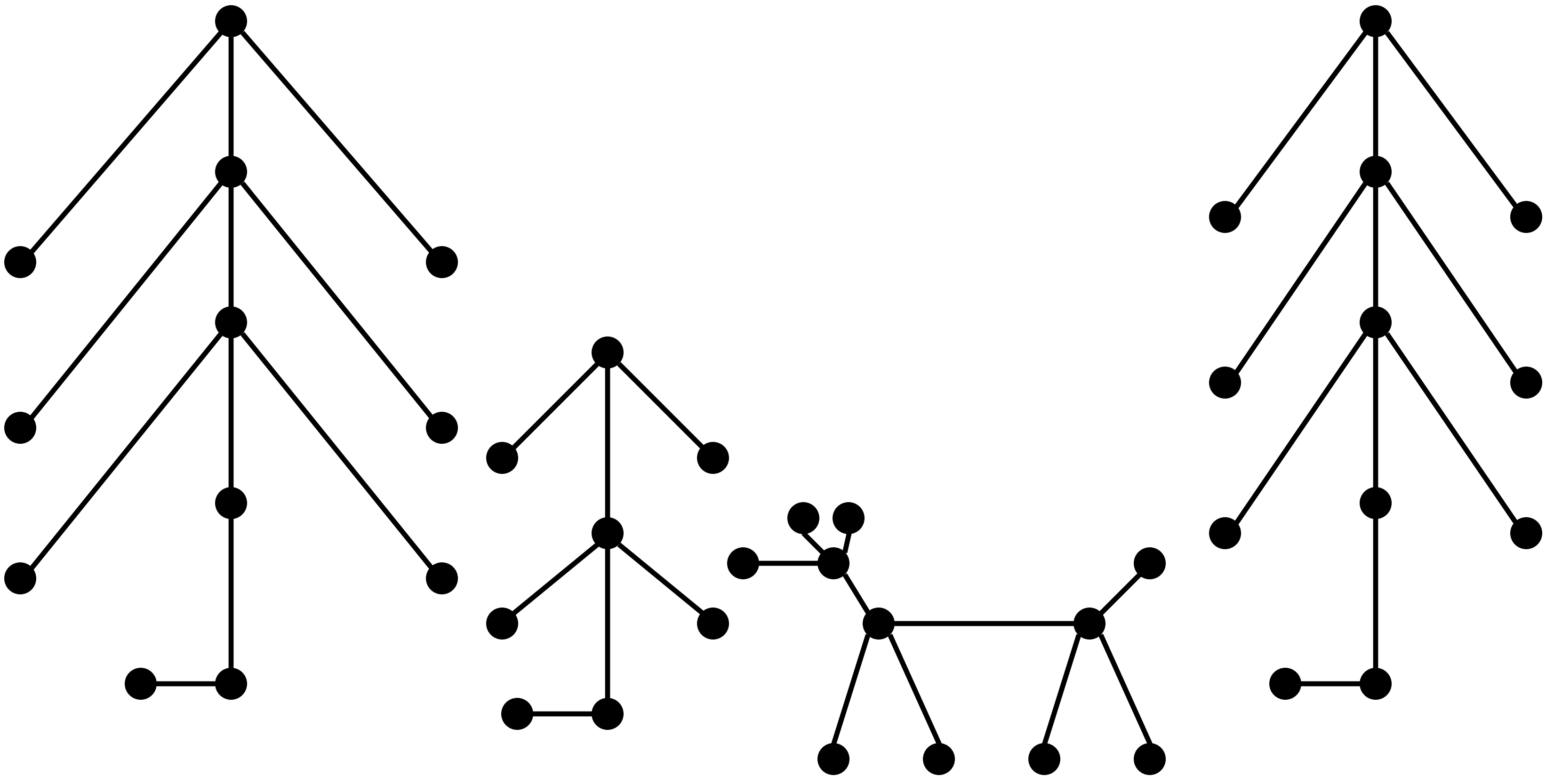
- *A tree is a simple connected graph that contains no cycle*





Forests

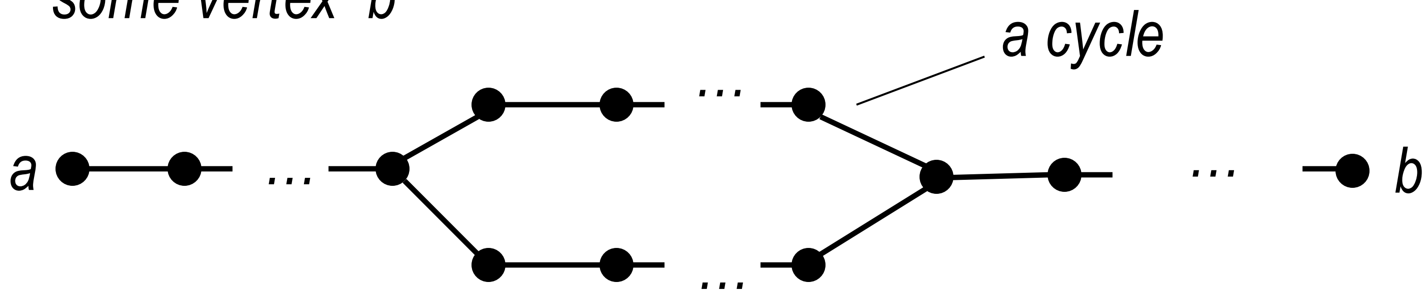
- *A* ***forest*** *is a simple graph where each connected component is a tree*



Properties of Trees: uniqueness of paths

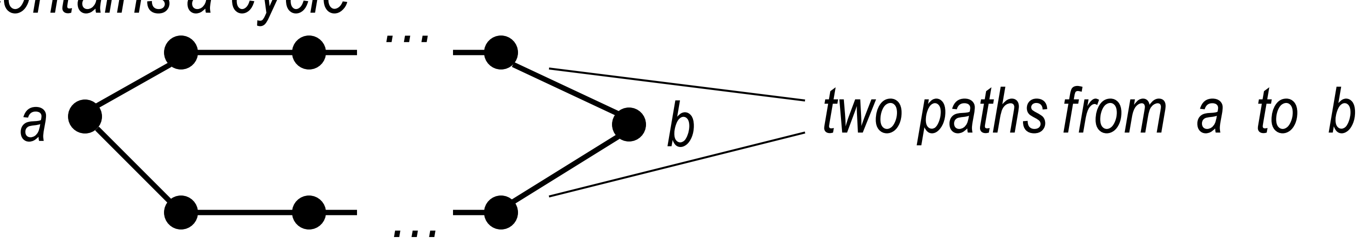
- **Property:** *A connected graph G is a tree if and only if for any vertices a and b in G there is exactly one path from a to b.*
- **Proof**

$\Rightarrow$ *Suppose G has two different paths from some vertex a to some vertex b*



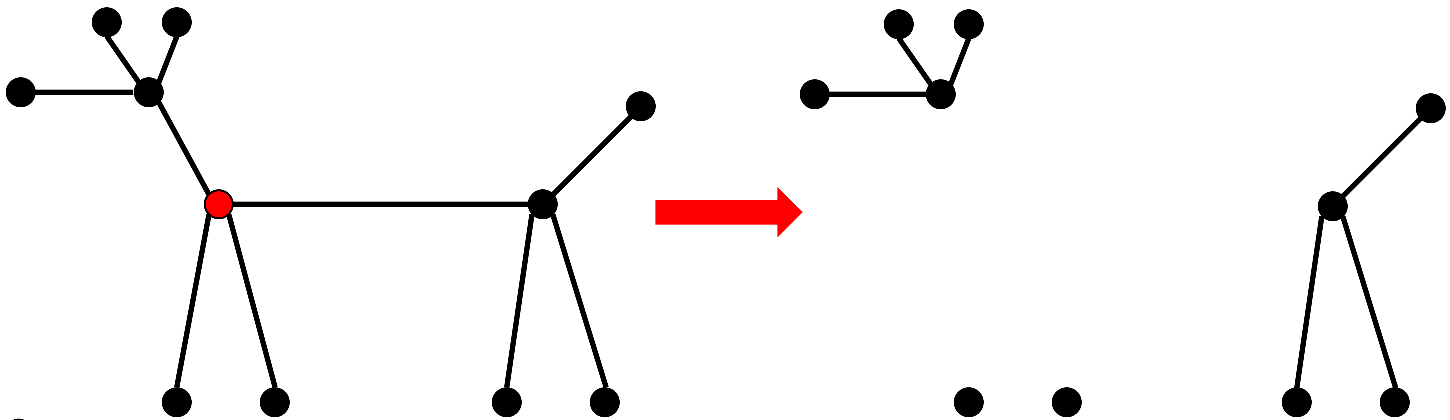
$\Leftarrow$ *Suppose G is connected but not a tree.*

Then it contains a cycle



Properties of Trees: subtree property

- **Property:** *Let v be a vertex in a tree G = (V,E). Then the induced subgraph G' on V - {v} is a forest, and each neighbour of v in G is in a different connected component (subtree) of G'*



- **Proof:**

  (a) G' is a forest because G contains no cycles

  (b) Let u, w be neighbours of v in G and suppose there is a path $u=u_0e_1u_1\ldots e_ku_k=w$ between then in G'.

  Then $u=u_0e_1u_1\ldots e_kwe_{k+1}ve_{k+2}u_0=u$ is a cycle in G

Sub-tree property

- *Recall that structural induction on a recursively defined binary tree has the following structure:*
  - *Base case: P is true for a leaf (tree with one vertex)*
  - *Inductive step: Show that if P holds for two trees T1, T2, P holds for a new tree T with a root vertex v connected to the roots of T1 and T2*

- *The previous property allows us to construct structurally inductive proofs by mathematical induction on the number of vertices:*
- *To prove property P(n) on trees with n vertices*

  Base case: 0, 1, or sometimes 2 vertices

  Inductive case: Assume P(i) for all I up to k

  Consider a tree with k+1 vertices and take one vertex v

  Its neighbours are all trees with at most k vertices!

Properties of Trees: pendants

- **Property**: *A tree G with at least 2 vertices contains at least two pendant vertices*
- **Proof**

By (strong) induction on the number of vertices n

Base case: n=2

Then the tree is  *which has 2 pendants*

Inductive case:

Assume that every tree on up to k vertices has 2 pendants

Pick an arbitrary vertex v. Since G is connected, deg(v) = m > 0

Let $u_1 \ldots u_m$ *be the neighbours of v in G. By the previous property, each* u_i *is in a separate component of G – {v}*

By the inductive hypothesis, each component has at least one pendant vertex w_i *which is not equal to* u_i

If m = 1 then w_1 *and v are pendant, otherwise* w_1 *and* w_2 *are pendant*

Properties of Trees: number of edges

- **Property**: *A tree with n vertices has $n - 1$ edges*
- **Proof (***By strong induction on n)*

  Base case: n=1

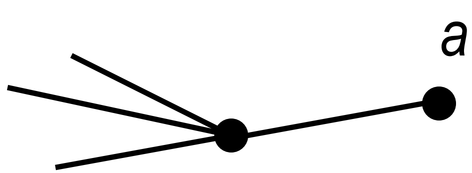
  0 edges, so the property holds

  Inductive case.

  Suppose the claim holds for all trees with at most k vertices.

  Let G be a tree with k+1 vertices. Then G has a pendant "a"

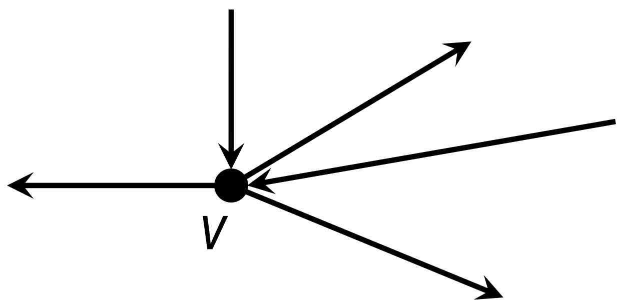
  G:

  Let G' be the tree with a removed. By the inductive hypothesis, G' has k-1 edges, so G has k edges.

Indegree and Outdegree

- Let $G = (V, E)$ be a directed graph, and v its vertex.
- The outdegree of v, denoted by $outdeg(v)$, is the number of edges e of G such that v is the source vertex of e
- The indegree of v, denoted by $indeg(v)$, is the number of edges e such that v is the terminal vertex of e

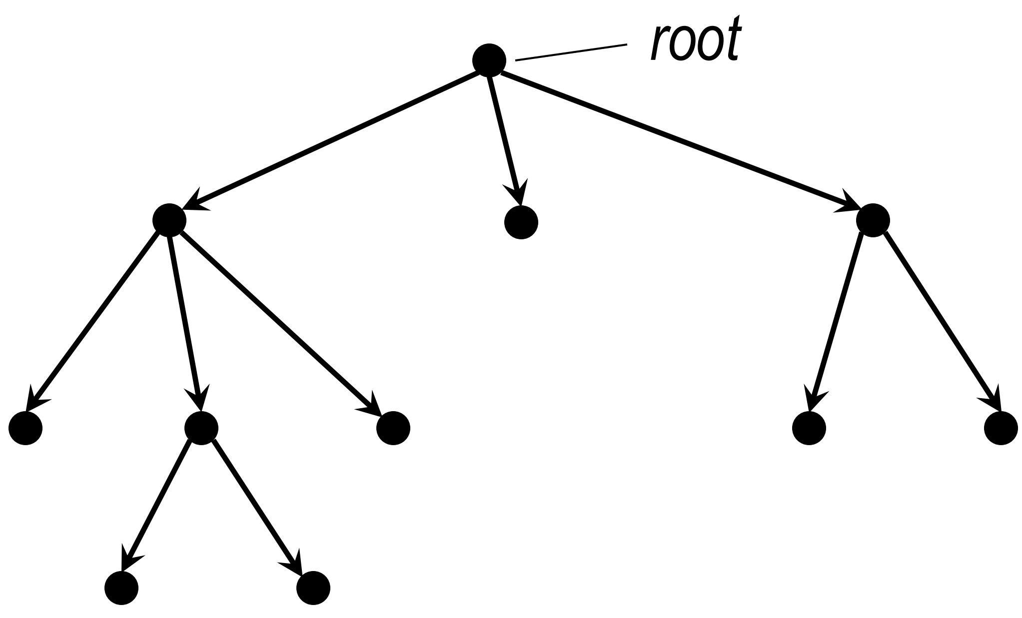


$outdeg(v) = 3$
$indeg(v) = 2$

- Clearly, $outdeg(v) + indeg(v) = deg(v)$

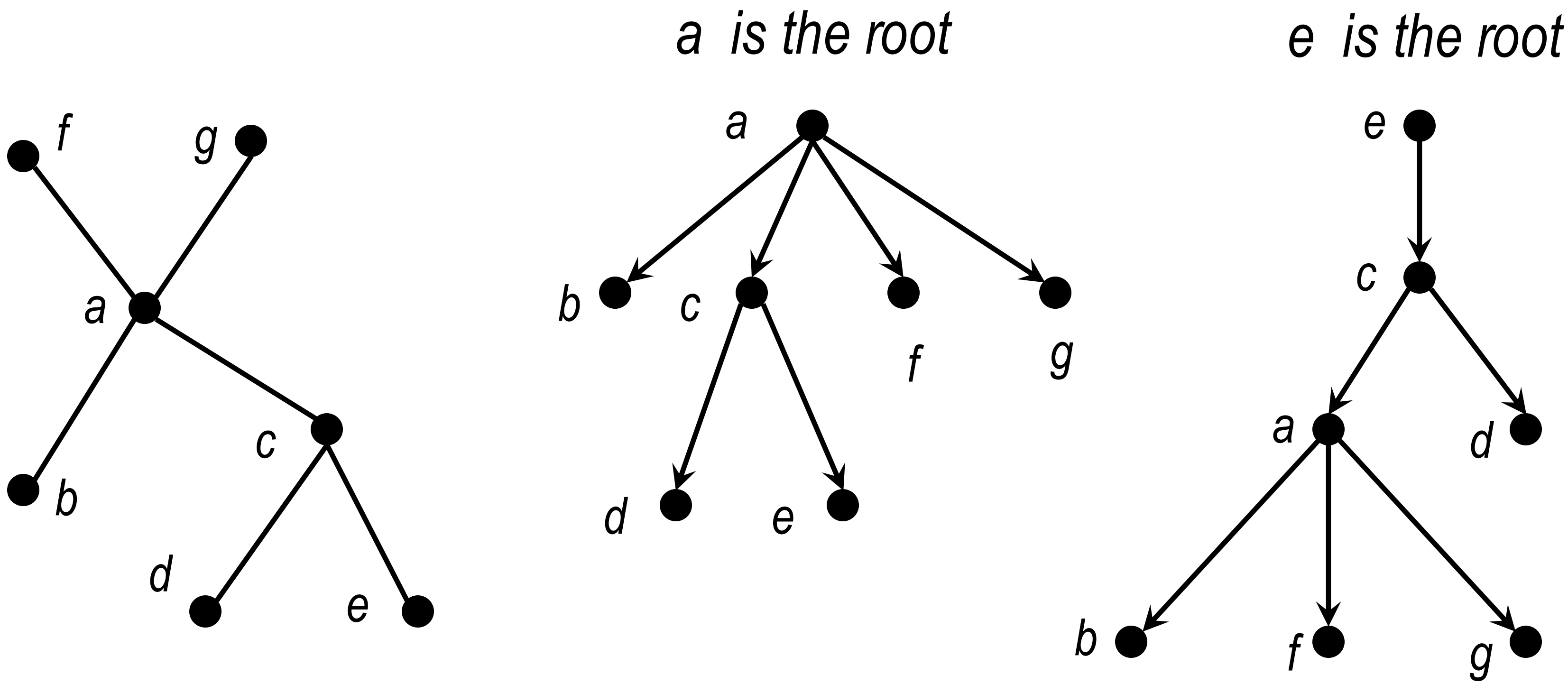
Rooted Trees

- *A directed graph G is called a rooted tree if*
  - *it is a tree if the directions of the edges are ignored*
  - *there is a unique vertex r such that $indeg(r) = 0$; this vertex is called the root of G*
  - *the indegree of any other vertex is equal to 1*

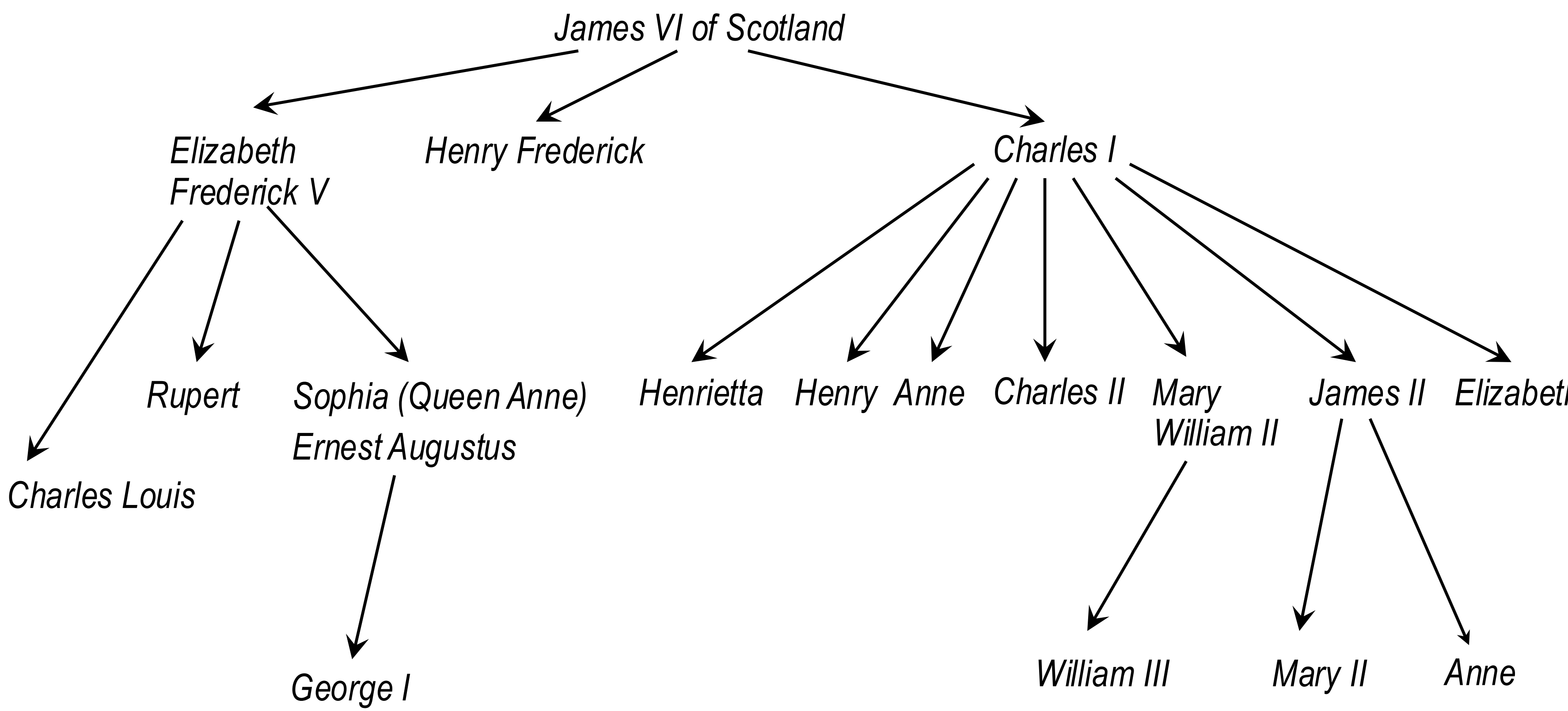


Rooted from Unrooted

- *Any ordinary tree can be converted into a rooted by simply by choosing a root*



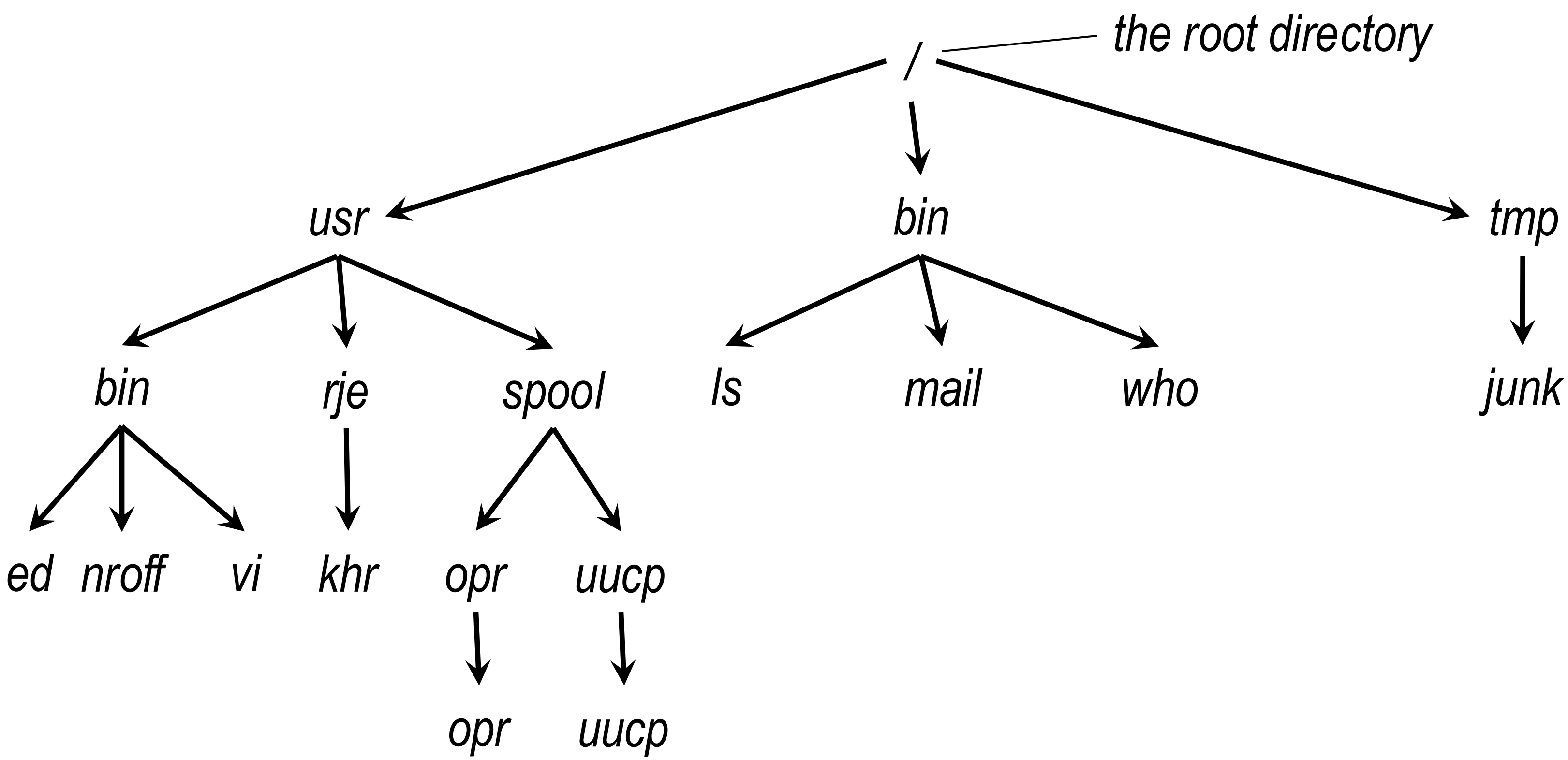
Examples: Family Tree



(http://www.royal.gov.uk)

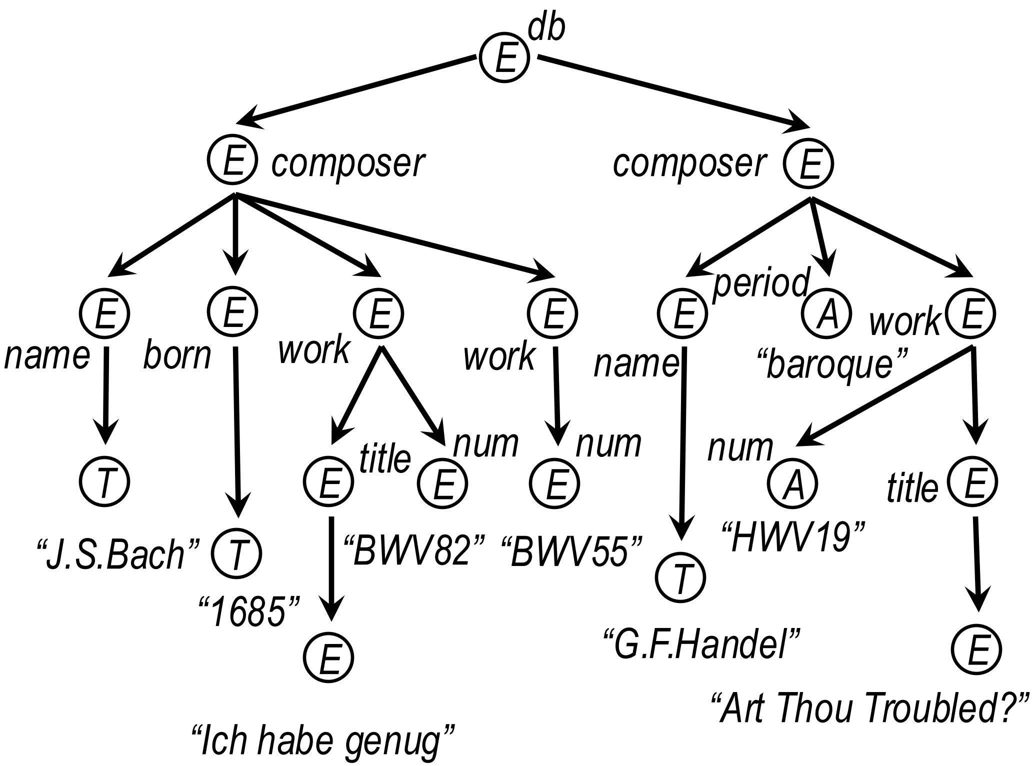
Examples: File System

- *Organization of the disk space.*
- *Every file or directory is represented by a vertex*
- *A directory is connected to each of its subdirs and each file in it*



Examples: XML Databases

- *XML (Extensible Markup Language) is a tool to support hierarchical information. In a nutshell this is a language to represent rooted trees in text form*



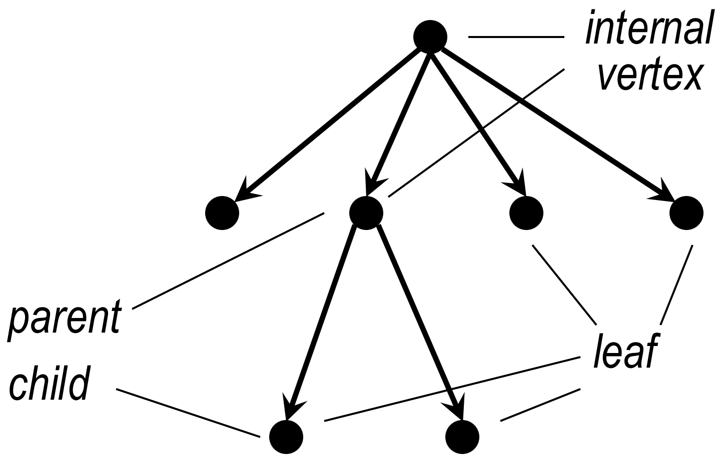

```
<db>
   <composer>
      <name>  J.S.Bach </name>
      <born>   1685  </born>
      <work num=“BWV82”>
         <title>  Ich habe genug  </title>
      </work>
      <work num=“BWV55”>      </work>
   </composer>
   <composer period=“baroque”>
      <name>  G.F.Handel  </name>
      <work num=“HWV19”>
         <title>  Art Thou Troubled?  </title>
      </work>
   </composer>
</db>
```

Terminology

- *For a directed edge from vertex u to vertex v:*

  u is a parent of v.

  v is a child of u.



- *For a vertex u:*

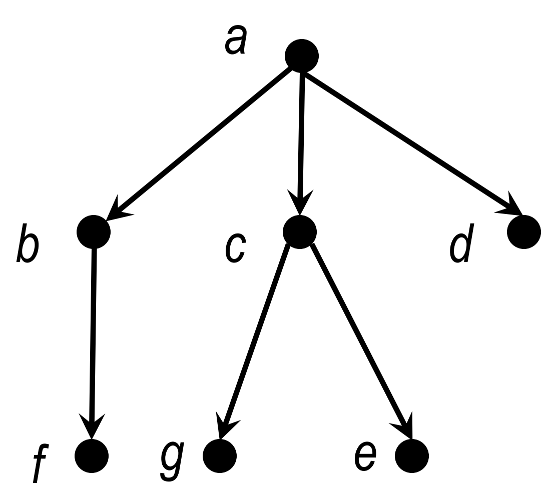
  Any vertices (excluding vertex u) on the path from the root to the vertex u are the ancestors of u.

  Vertex u is the descendant of these vertices.

- *A terminal vertex or a leaf is a vertex that has no children. (There must be at least one leaf.)*
- *An internal vertex is one that has children.*
- *A subtree is a subgraph of a tree induced by a vertex and all its descendants.*

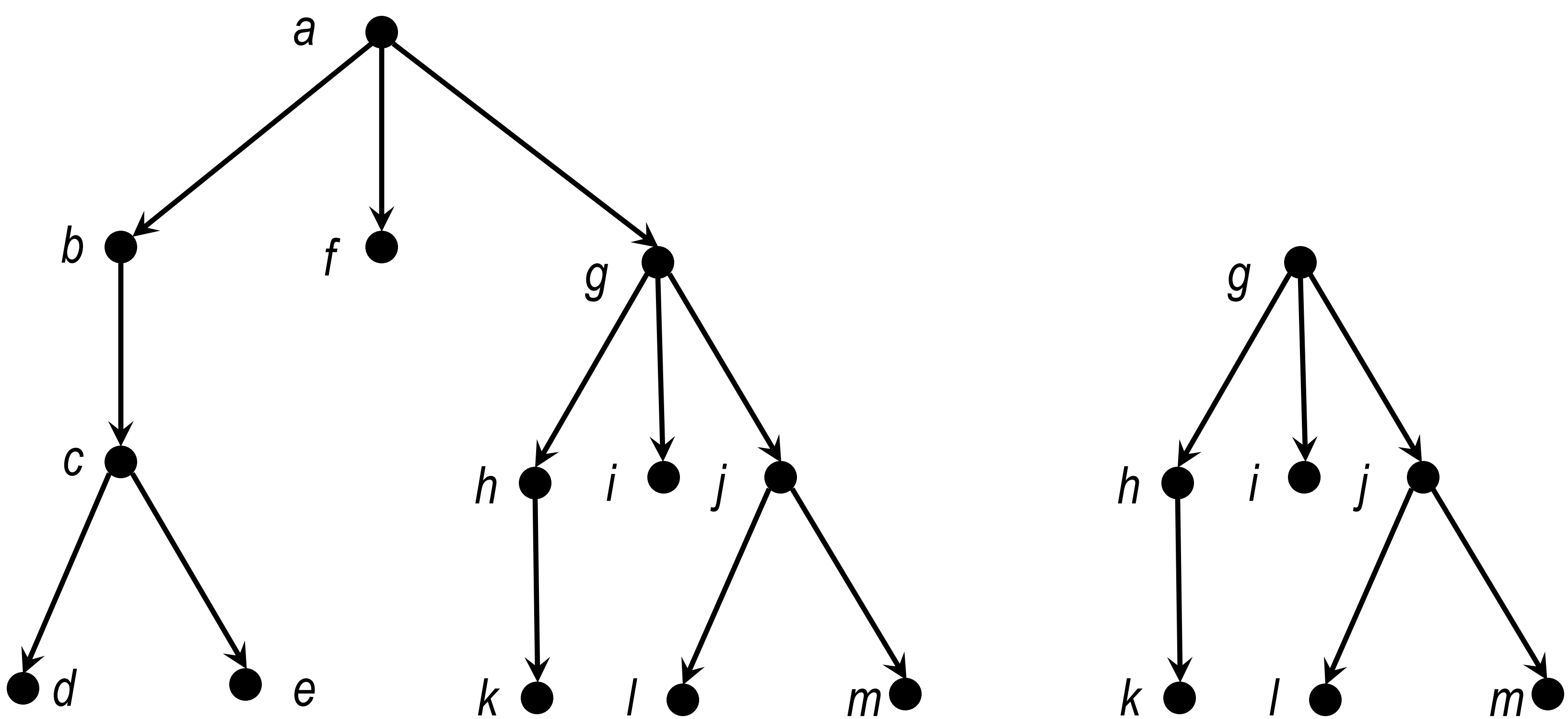
Examples

- *For the following rooted tree find*
  - *Children of b?*
  - *Parent of b?*
  - *Ancestors of f?*
  - *Descendant of a?*
  - *Terminal vertices (leaves)?*
  - *Internal vertices?*



Examples

- *A tree T and its subtree rooted at g.*



Practice

Exercises from the Book:

7th edition: 1, 3, 9, 11, 17, 31 (page 755 – 756

8th edition: 1, 3, 9, 11, 17, 31 (page 791 – 792)