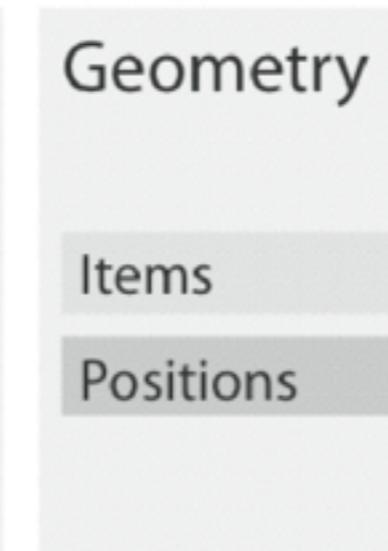
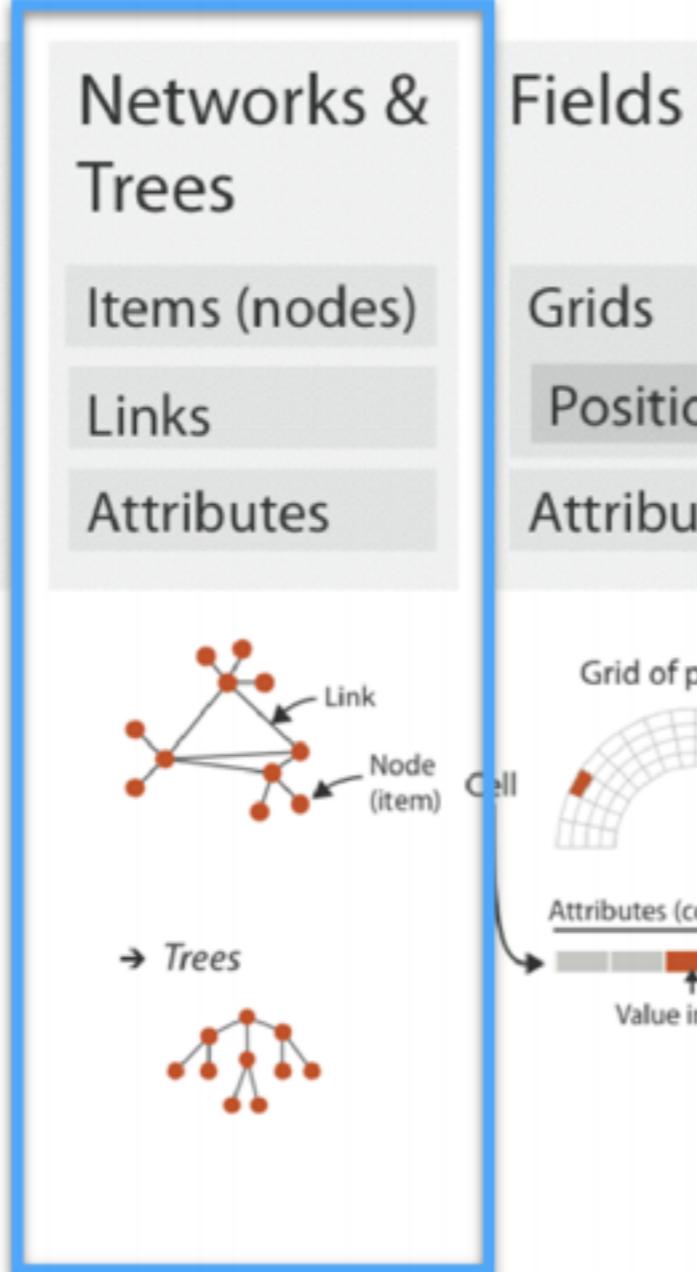
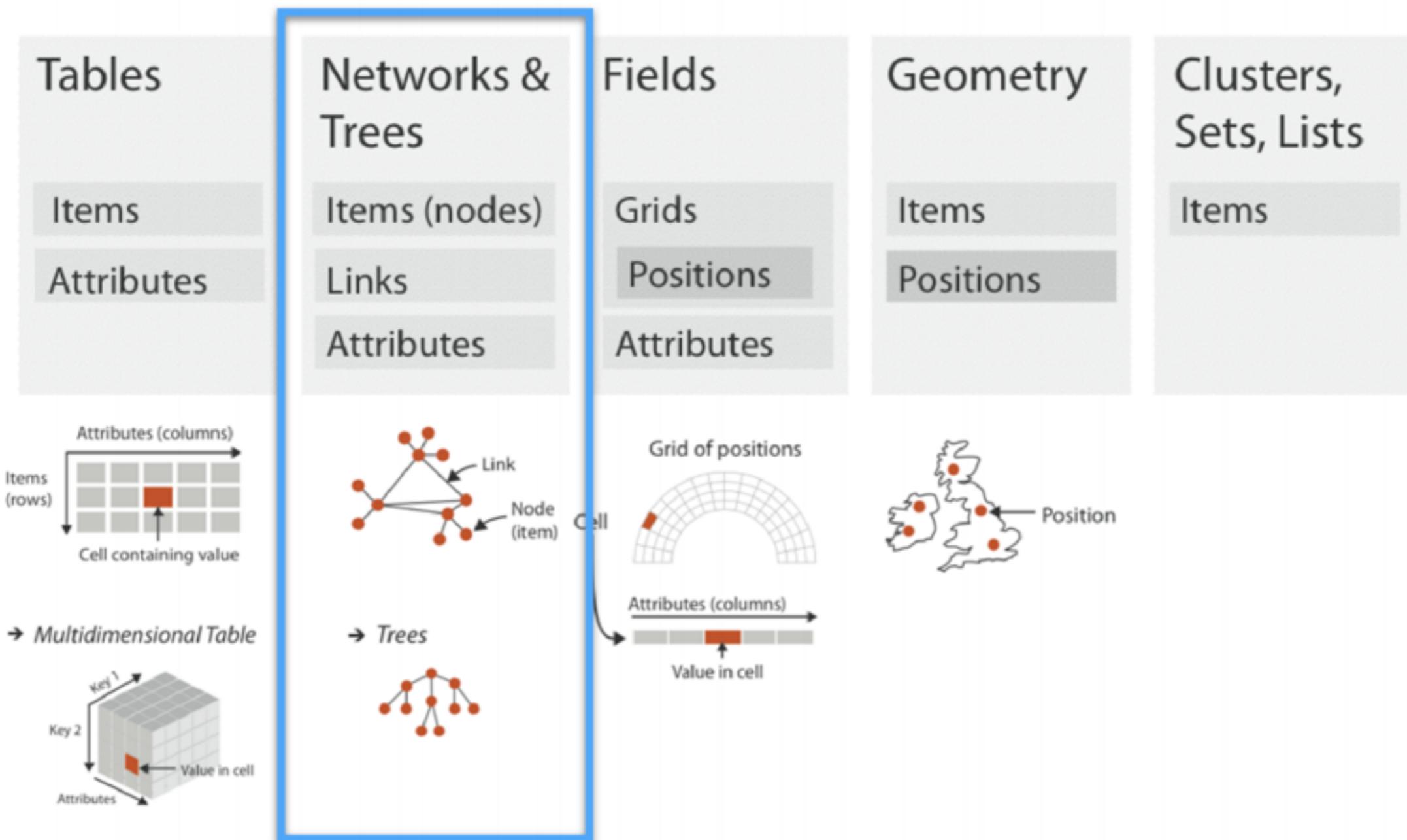


Relational Data

Hierarchies

CSC544

Why hierarchies?



Clusters, Sets, Lists

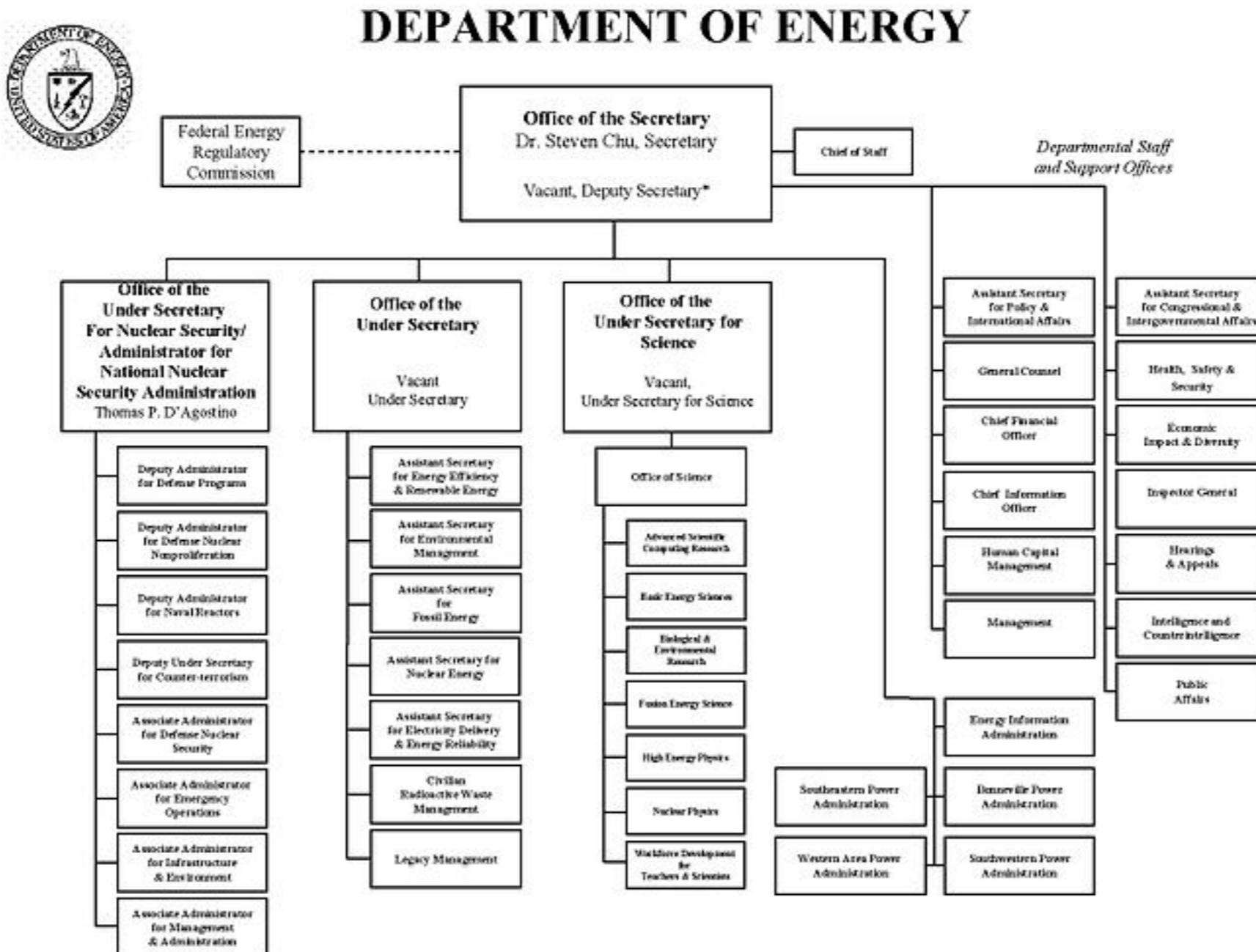
Items

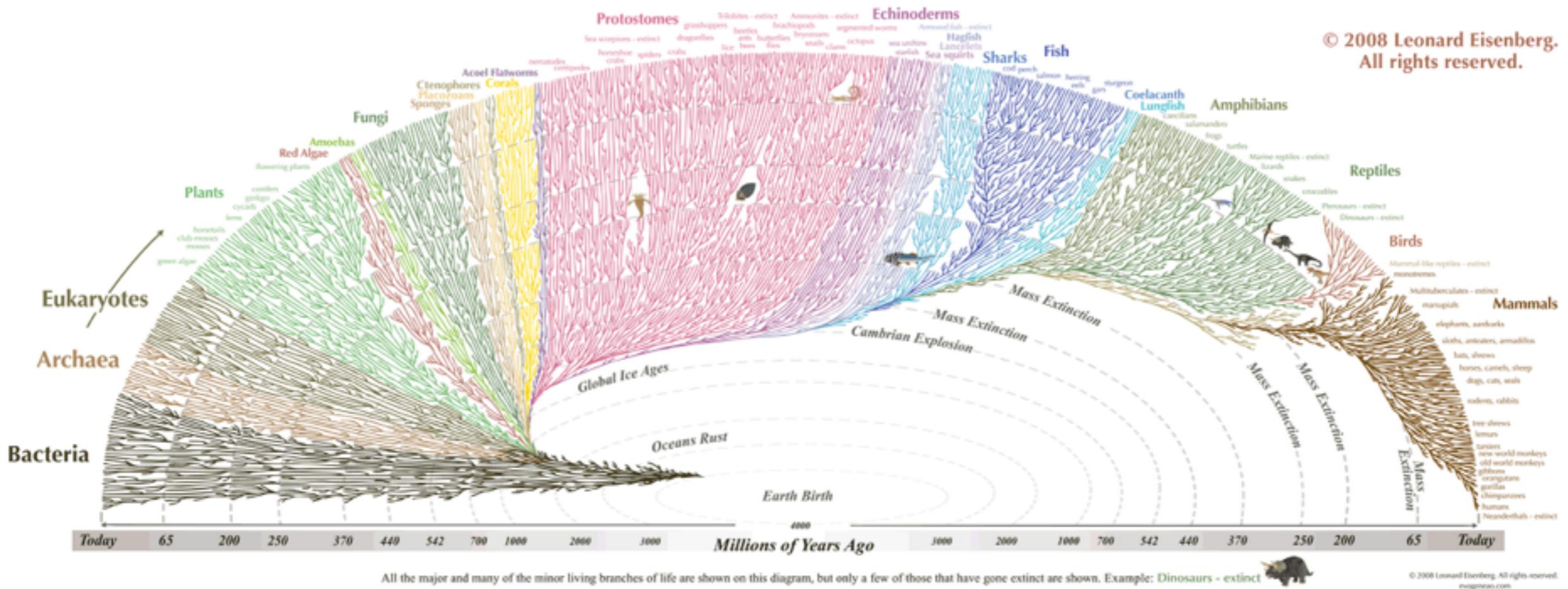
Scatterplots; dot plots; line charts, etc.

Until now, our data points were “independent of one another”

In “relational data”, it’s the
relationship between
points that matters

- The reports-to relationship in an organization





- The “tree of life”
 - evolution of species creates branching mechanism and “ancestor-of” relationship

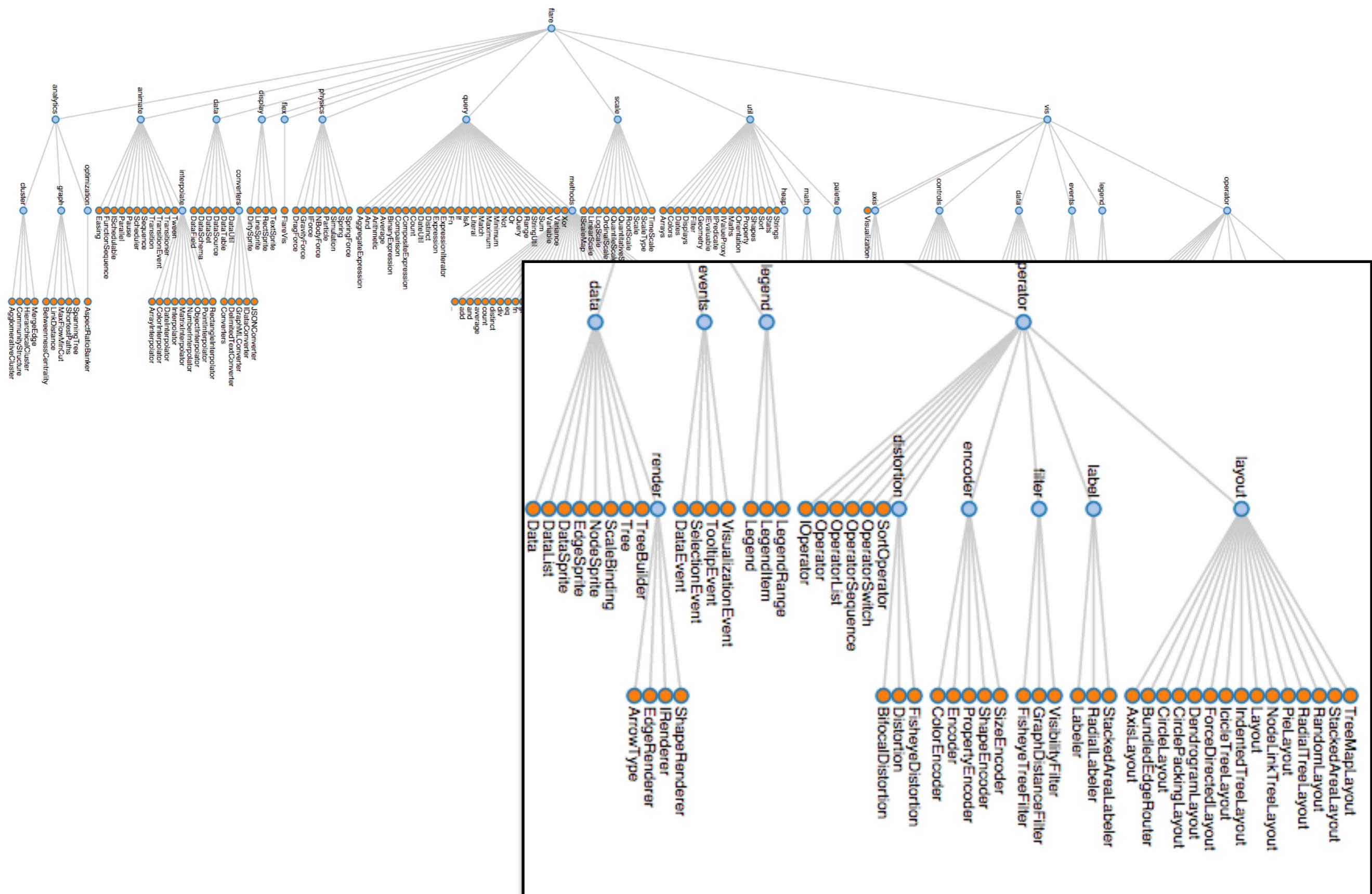
Tree Hierarchy

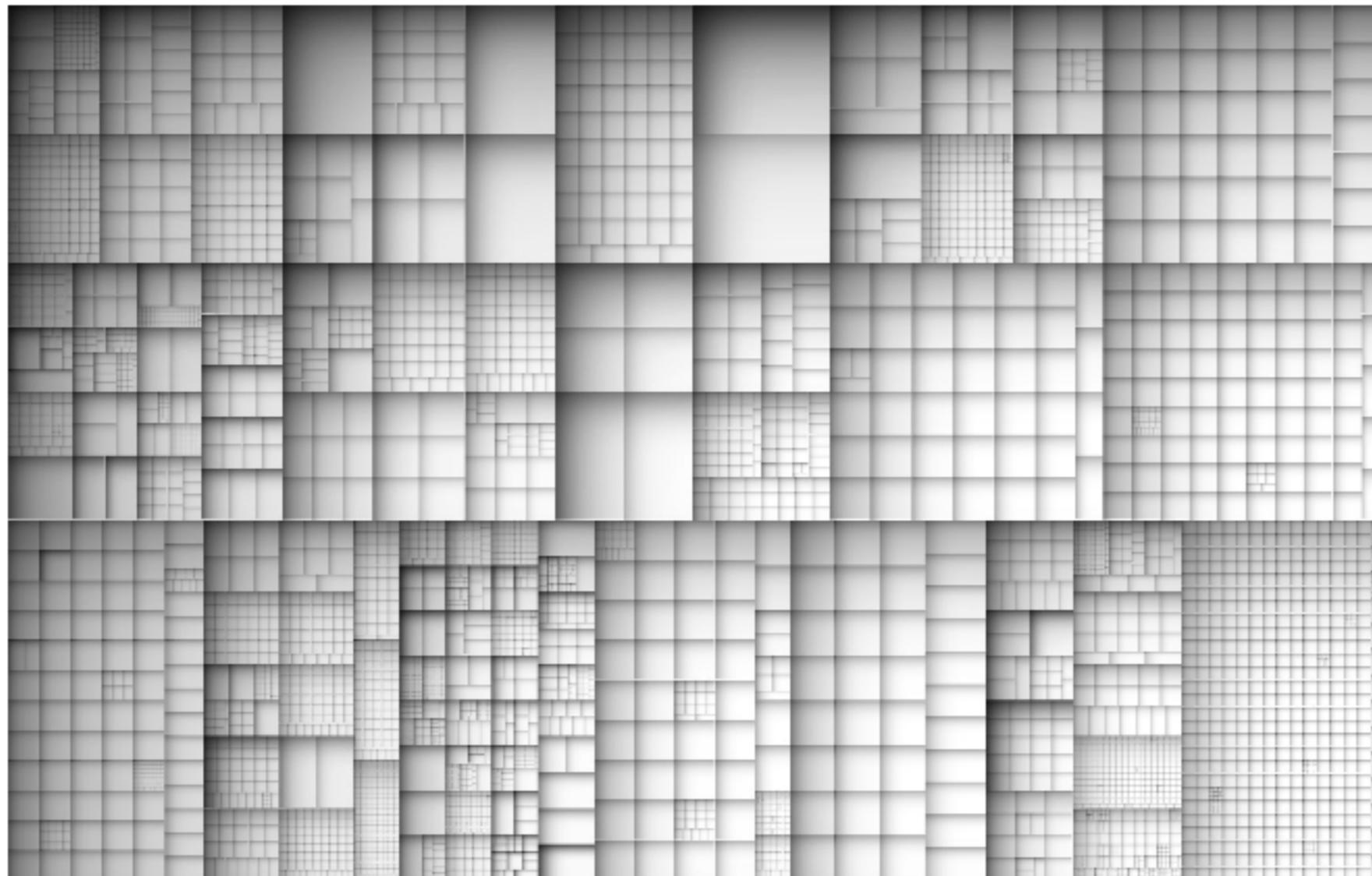
- “Rooted tree”: Every node has exactly one “parent” node, except for the **root**, which has none

What do we want our drawings to show?

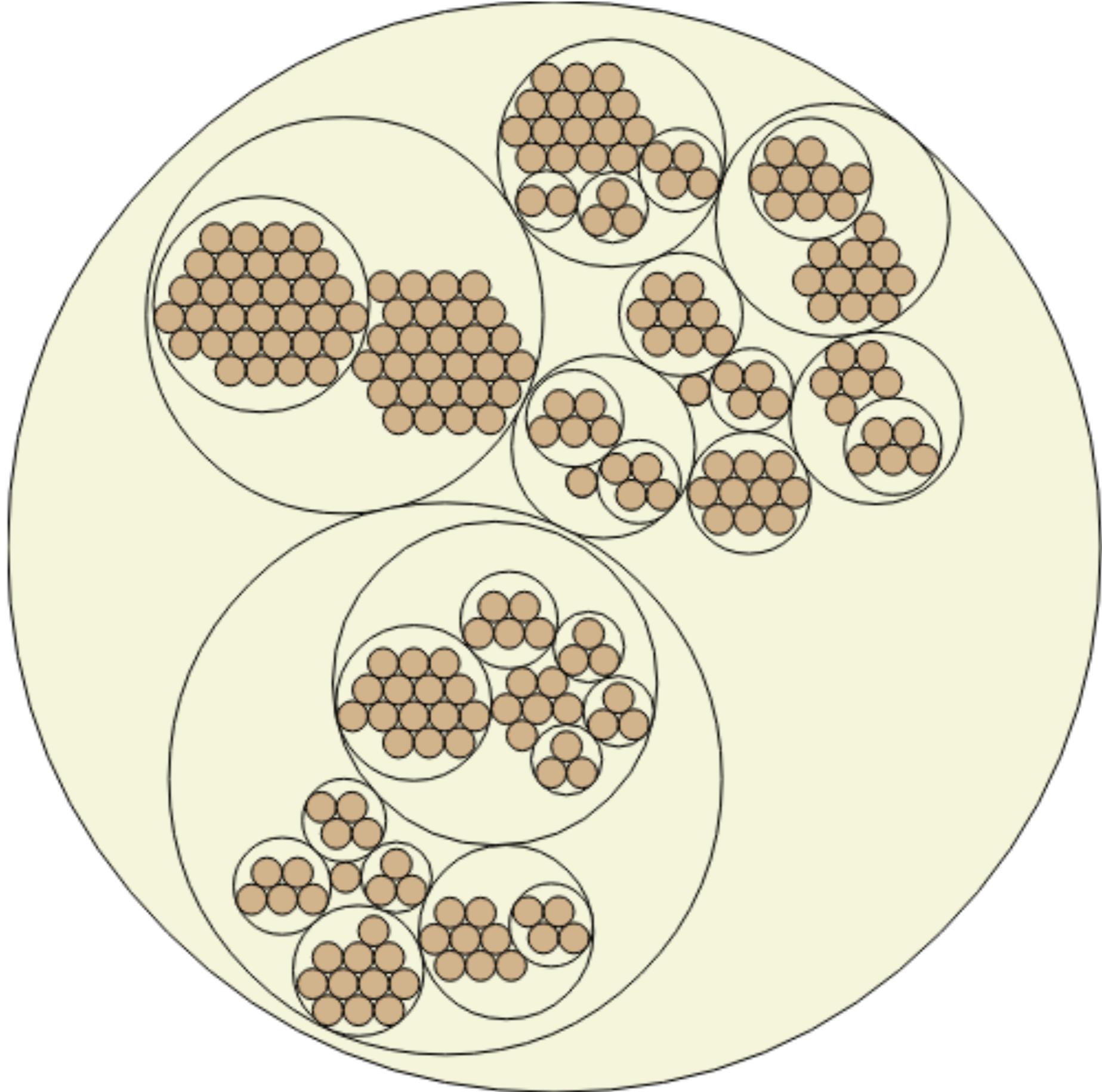
- Who reports to whom
 - ... and **who doesn't**
- How big are sub-trees
 - etc

Many different ways to
visualize trees





<http://www.cs.rug.nl/svcg/SoftVis/ViewFusion>



Reingold-Tilford binary tree drawing

- **aesthetics:** properties which we believe are responsible for good drawings
 - 1) nodes at same level of tree should lie along a horizontal line
 - 2) left child should be positioned to the left of parent; same with right child
 - 3) parent should be centered over children
 - 4) subtree drawing should be independent of subtree position on general drawing, and tree and “its mirror” should produce mirror drawings of one another

[http://hci.stanford.edu/courses/cs448b/f11/lectures/
CS448B-20111110-GraphsAndTrees.pdf](http://hci.stanford.edu/courses/cs448b/f11/lectures/CS448B-20111110-GraphsAndTrees.pdf)

Reingold-Tilford tree drawing

<http://www.reingold.co/tidier-drawings.pdf>

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-7, NO. 2, MARCH 1981

223

Tidier Drawings of Trees

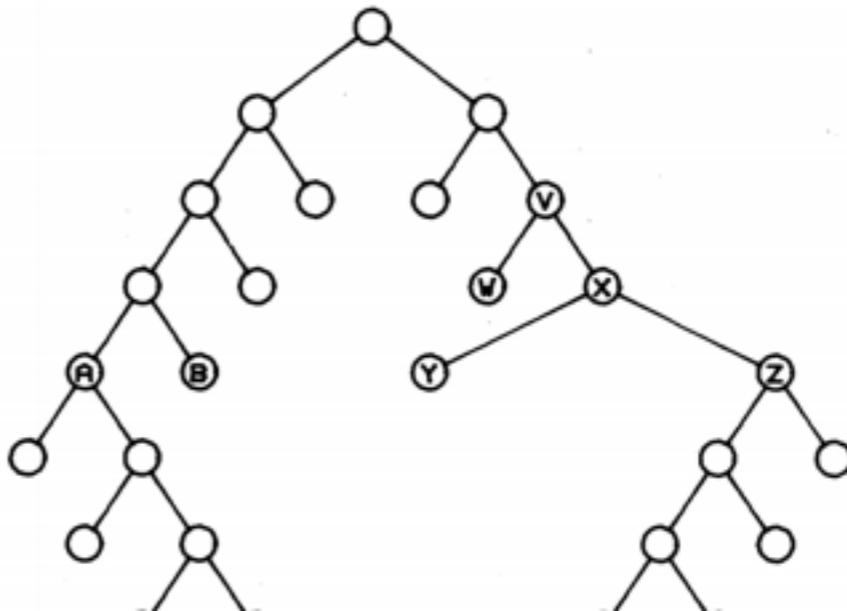
EDWARD M. REINGOLD AND JOHN S. TILFORD

Abstract—Various algorithms have been proposed for producing tidy drawings of trees—drawings that are aesthetically pleasing and use minimum drawing space. We show that these algorithms contain some difficulties that lead to aesthetically unpleasing, wider than necessary drawings. We then present a new algorithm with comparable time and storage requirements that produces tidier drawings. Generalizations to forests and m-ary trees are discussed, as are some problems in discretization when alphanumeric output devices are used.

Index Terms—Data structures, trees, tree structures.

INTRODUCTION

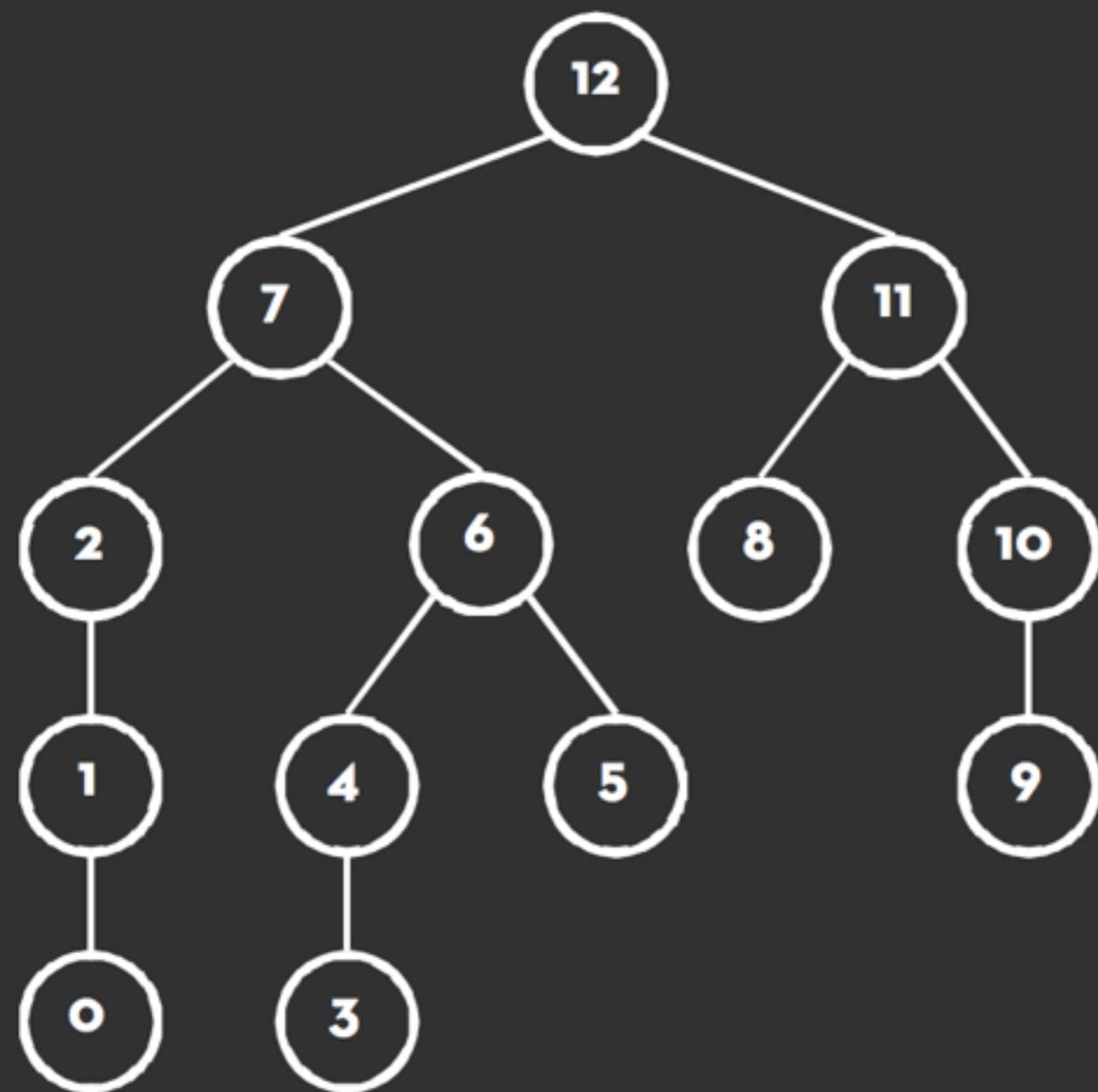
In a recent article [6], Wetherell and Shannon presented algo-



Reingold-Tilford Algorithm

- Bottom-up tree traversal
- y-coord is the depth of the node, x-coords are “locally defined” (so first is arbitrary)
- merge trees
 - push right tree as close as possible to left tree (this is where the contour comes in)
 - position **shifts** saved at each node
 - parent nodes are centered above direct children
- Final top-down pass to convert shifts to positions

Reingold-Tilford Algorithm



Reingold-Tilford Algorithm



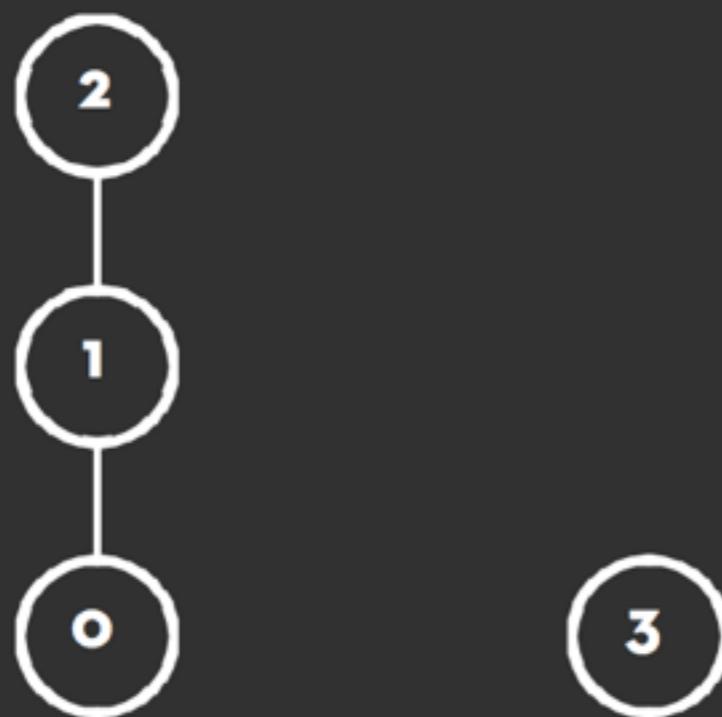
Reingold-Tilford Algorithm



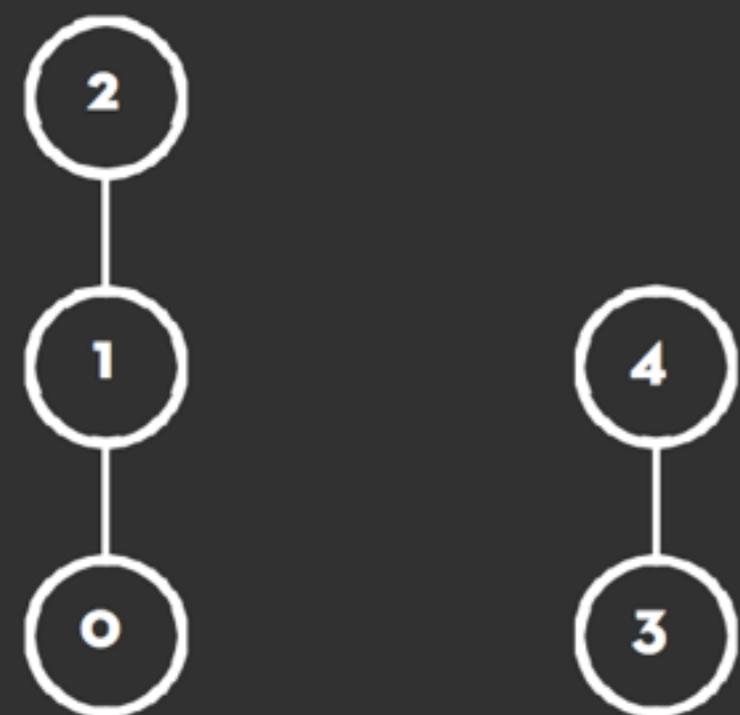
Reingold-Tilford Algorithm



Reingold-Tilford Algorithm



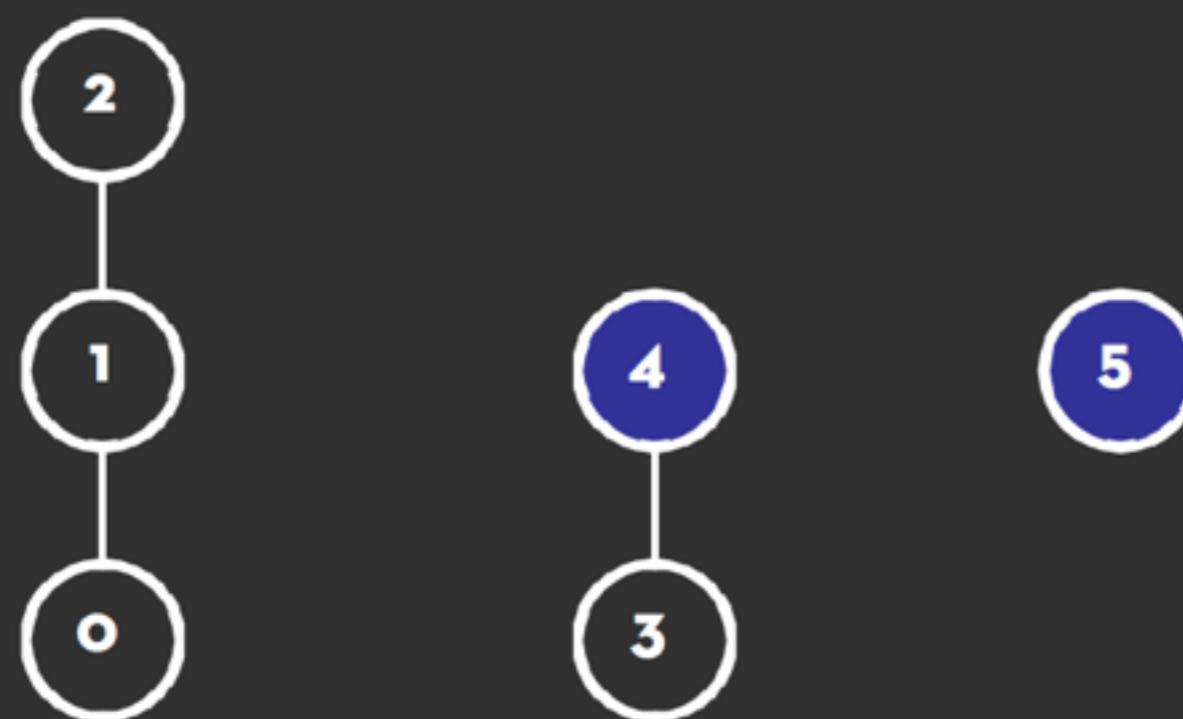
Reingold-Tilford Algorithm



Reingold-Tilford Algorithm



Reingold-Tilford Algorithm



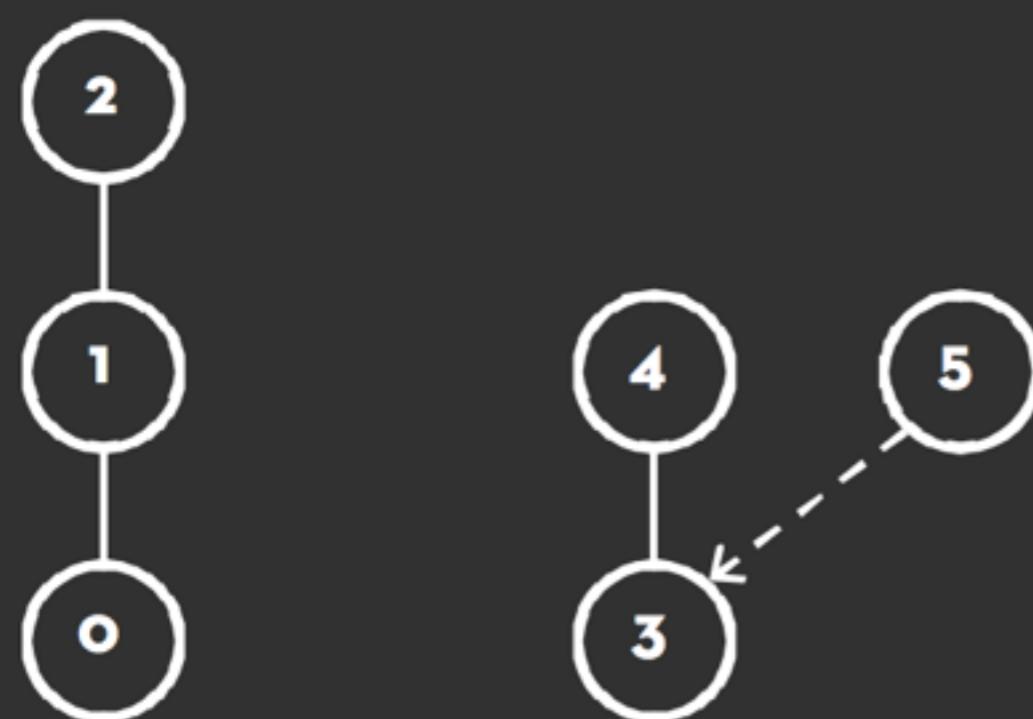
Reingold-Tilford Algorithm



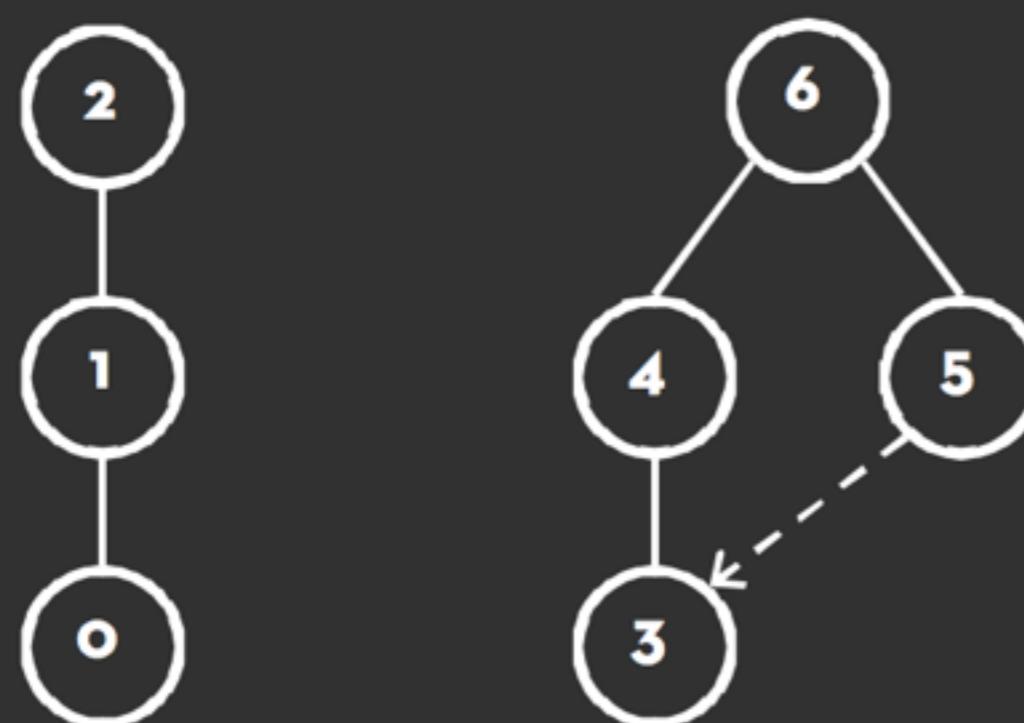
Reingold-Tilford Algorithm



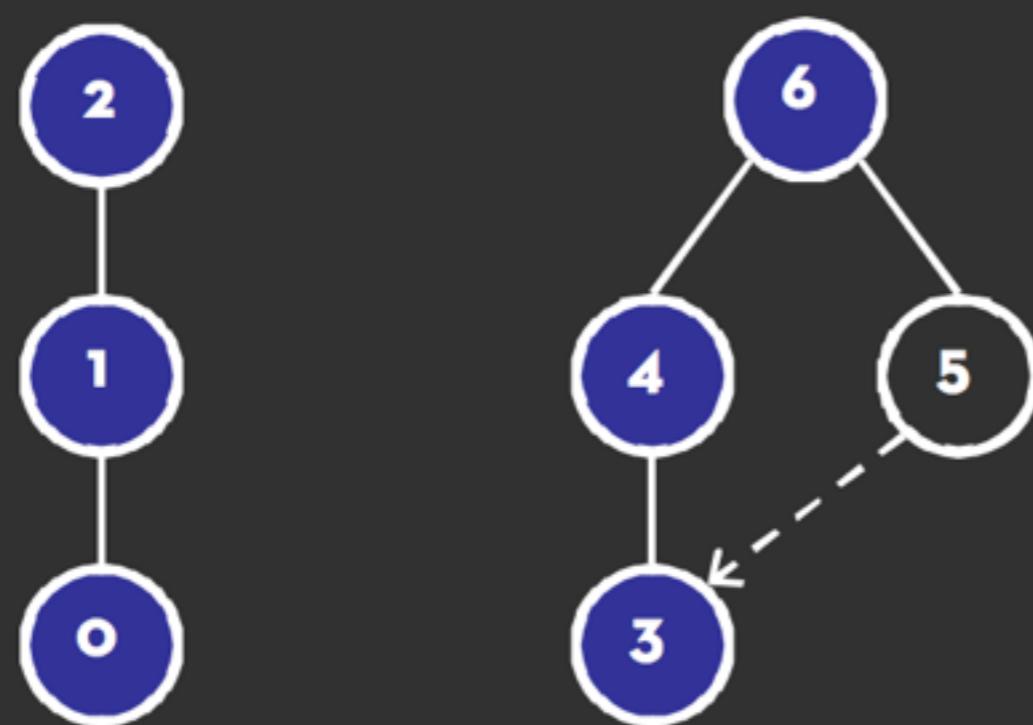
Reingold-Tilford Algorithm



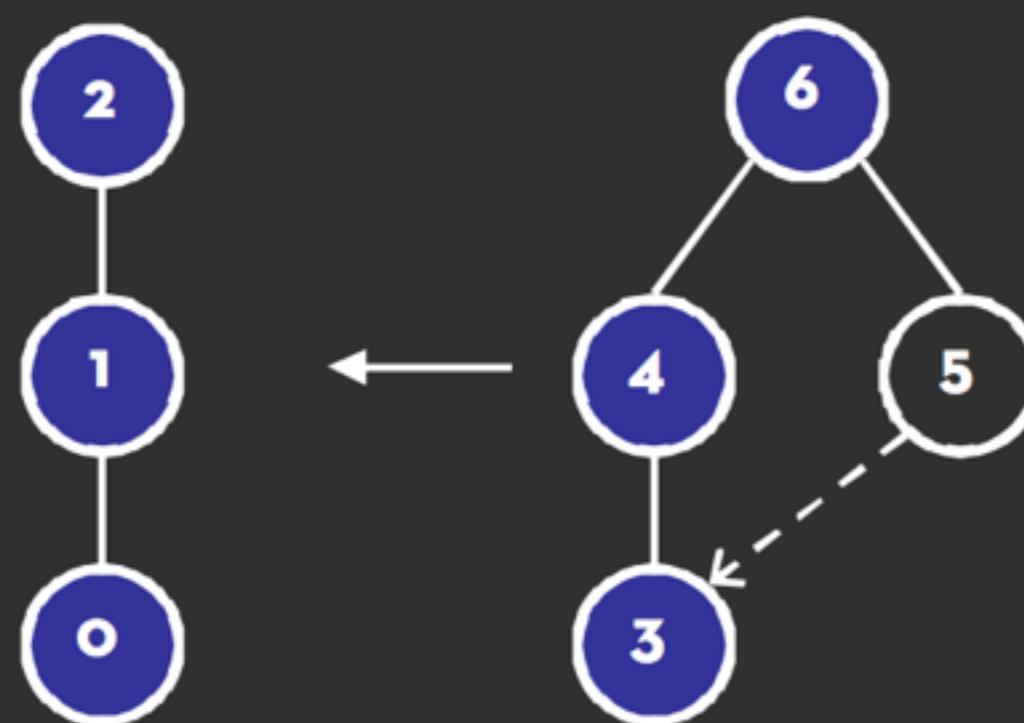
Reingold-Tilford Algorithm



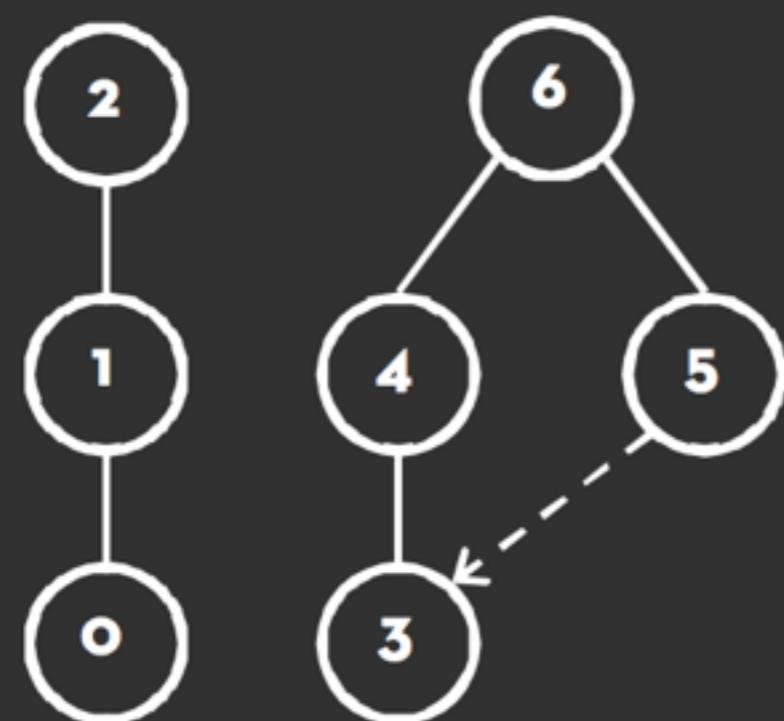
Reingold-Tilford Algorithm



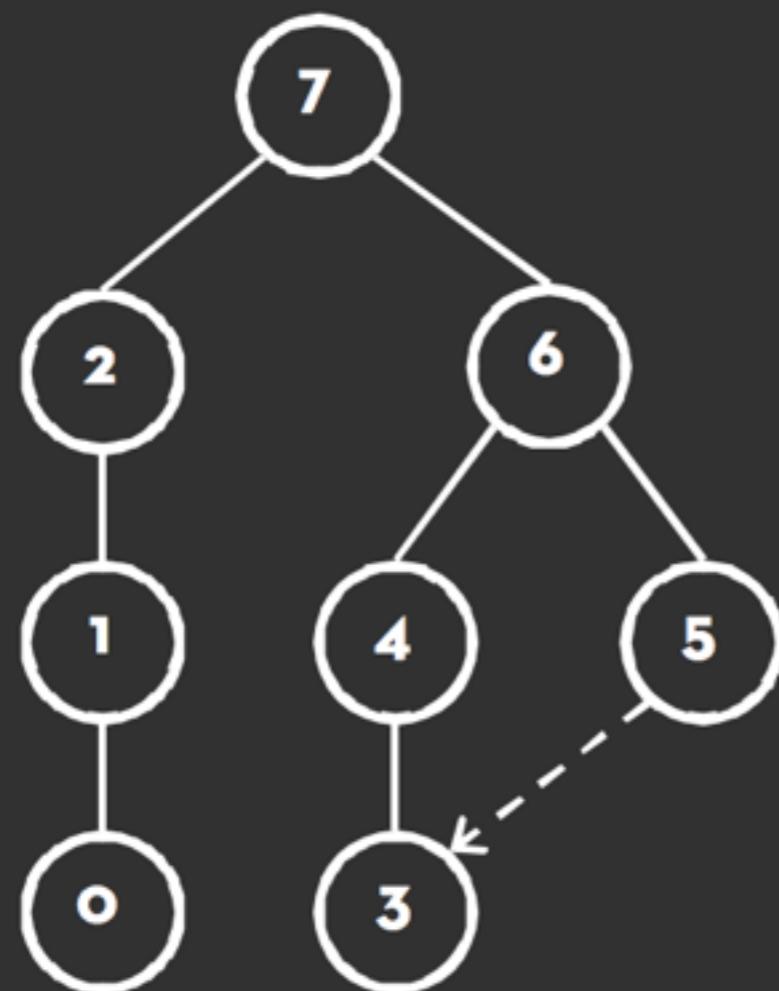
Reingold-Tilford Algorithm



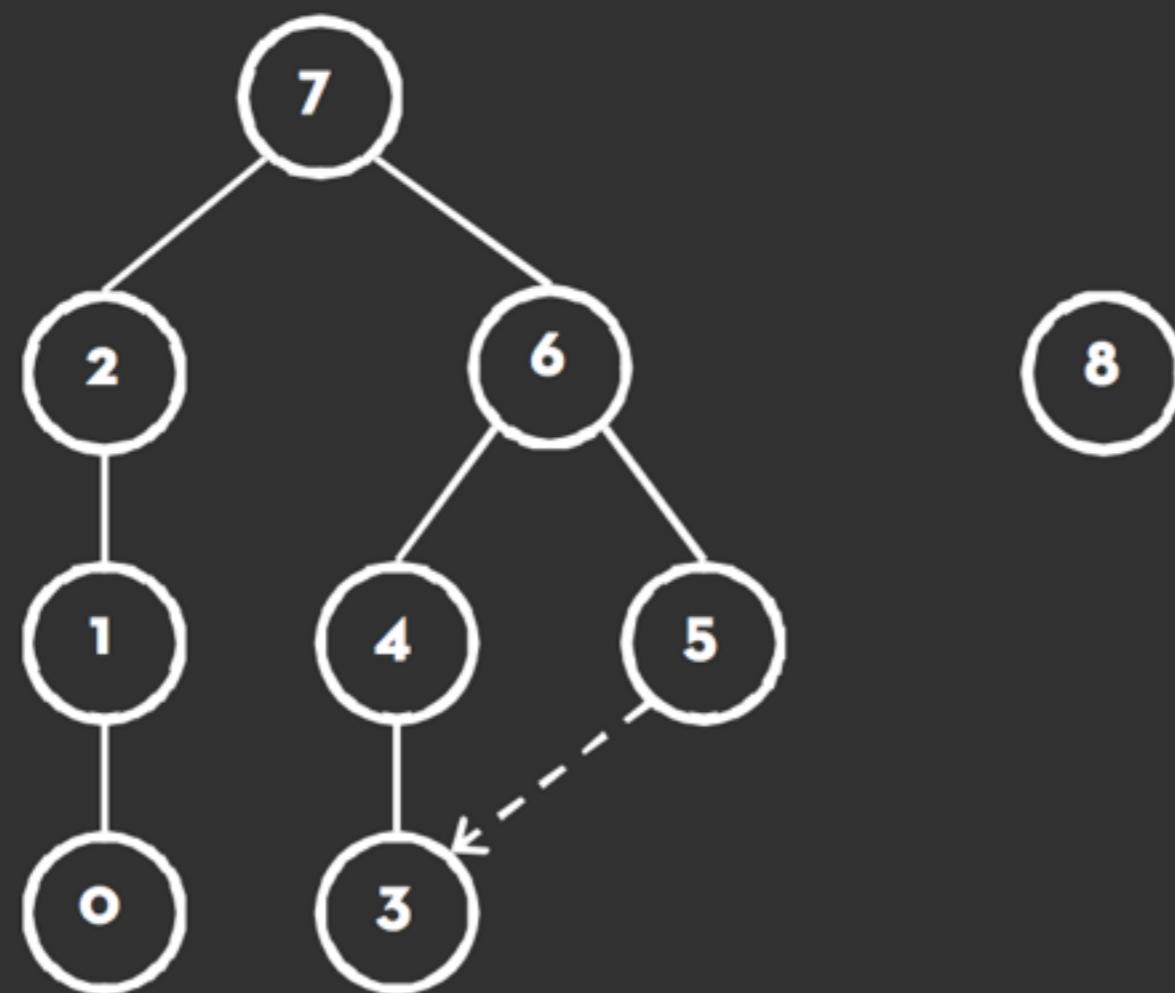
Reingold-Tilford Algorithm



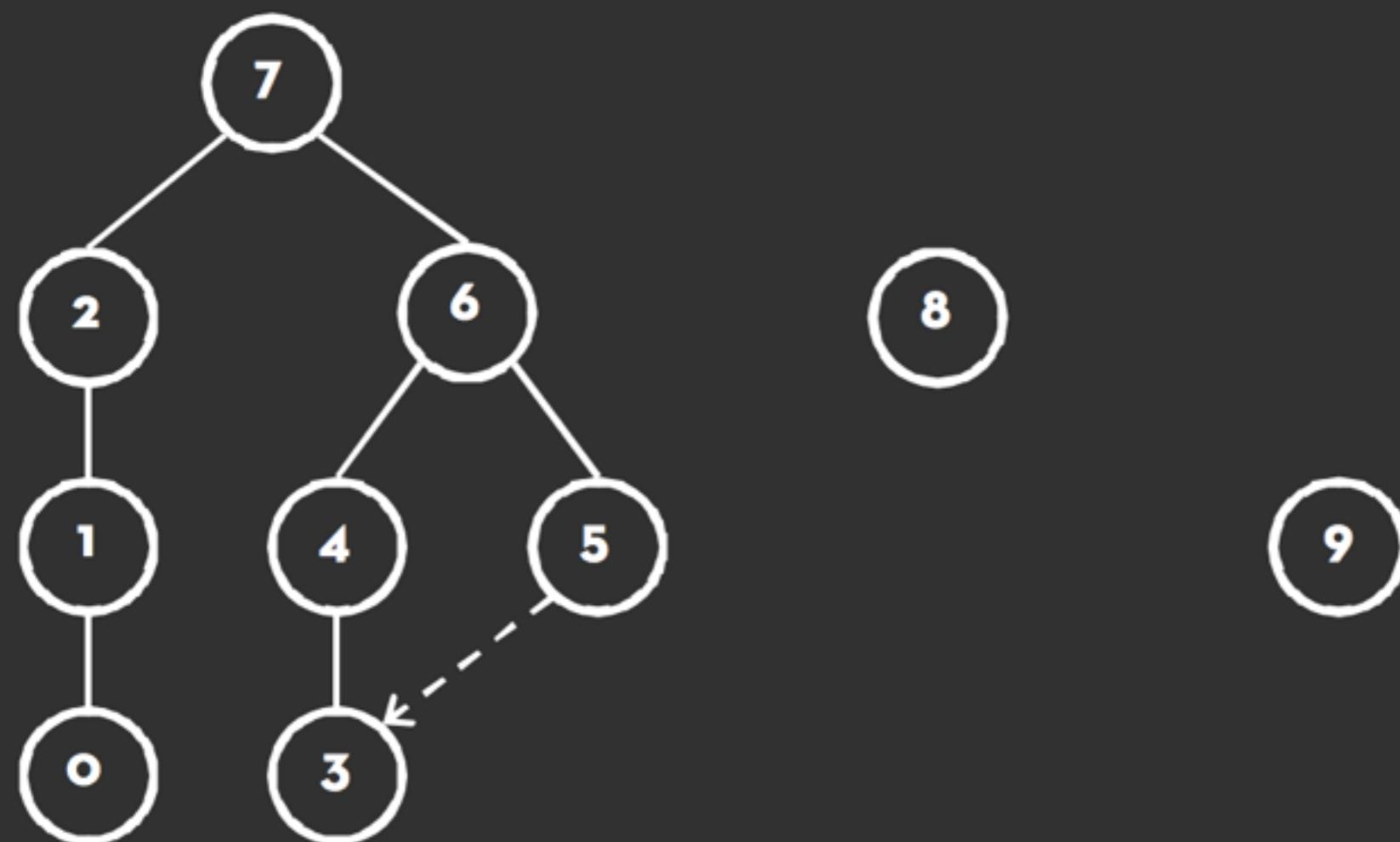
Reingold-Tilford Algorithm



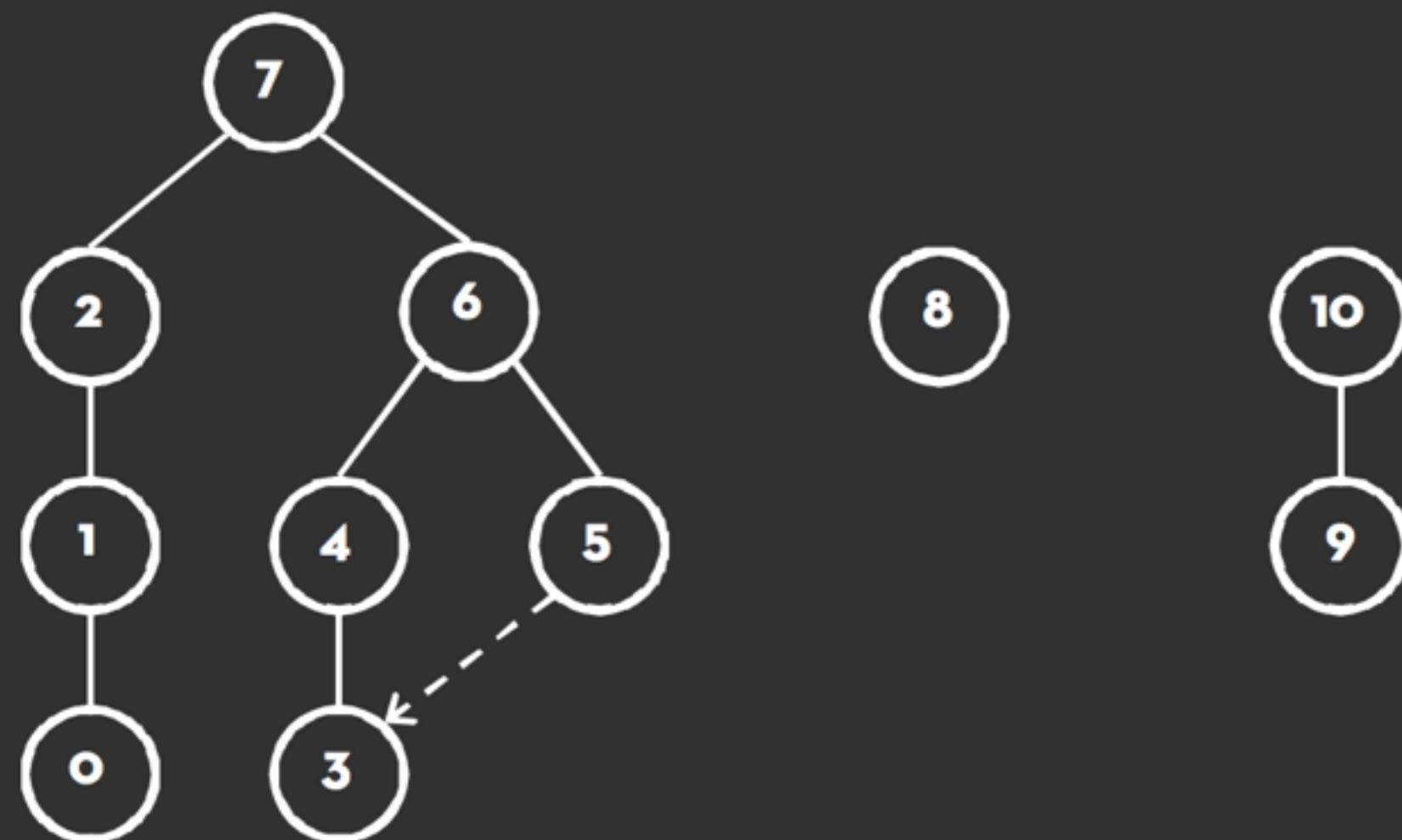
Reingold-Tilford Algorithm



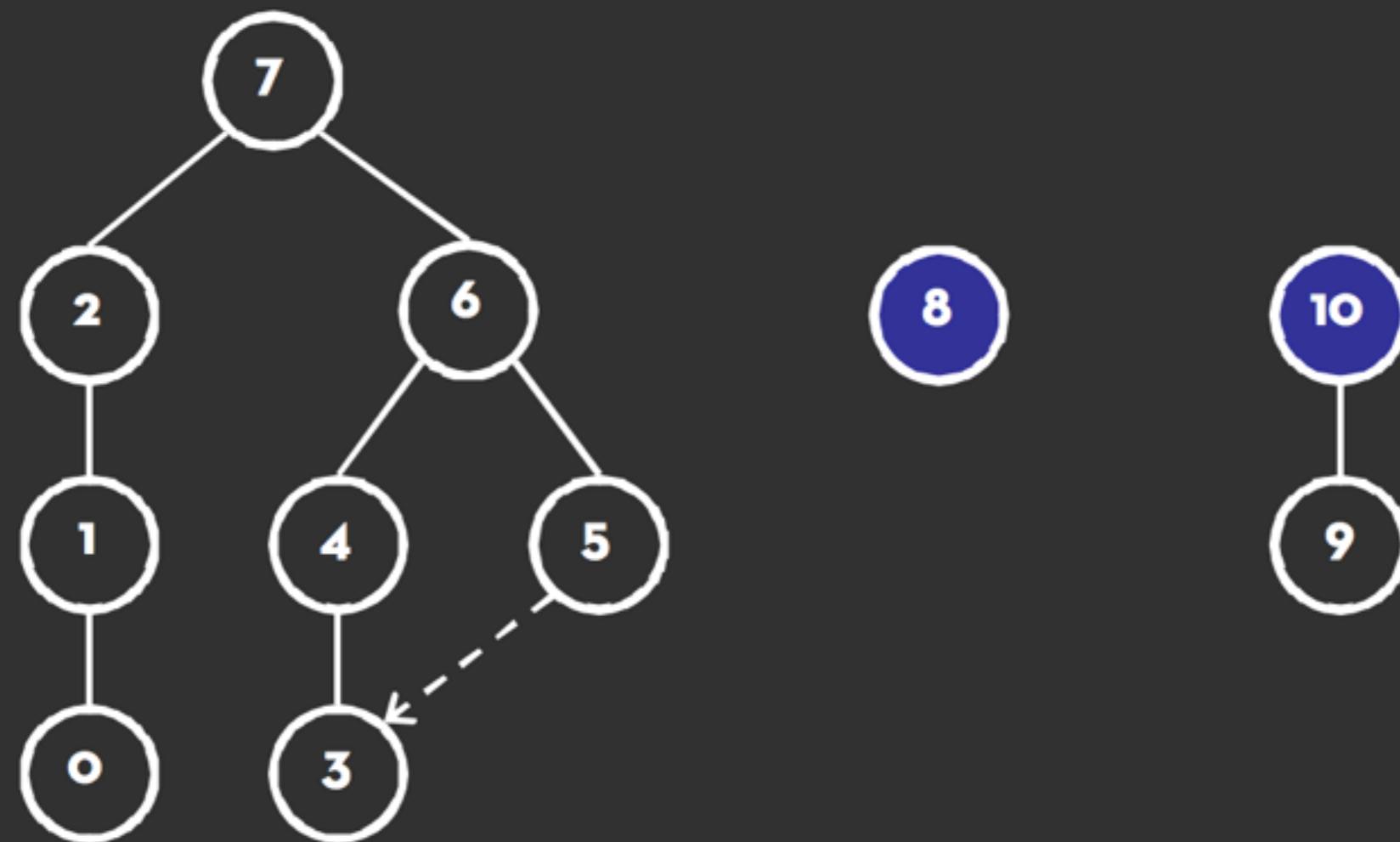
Reingold-Tilford Algorithm



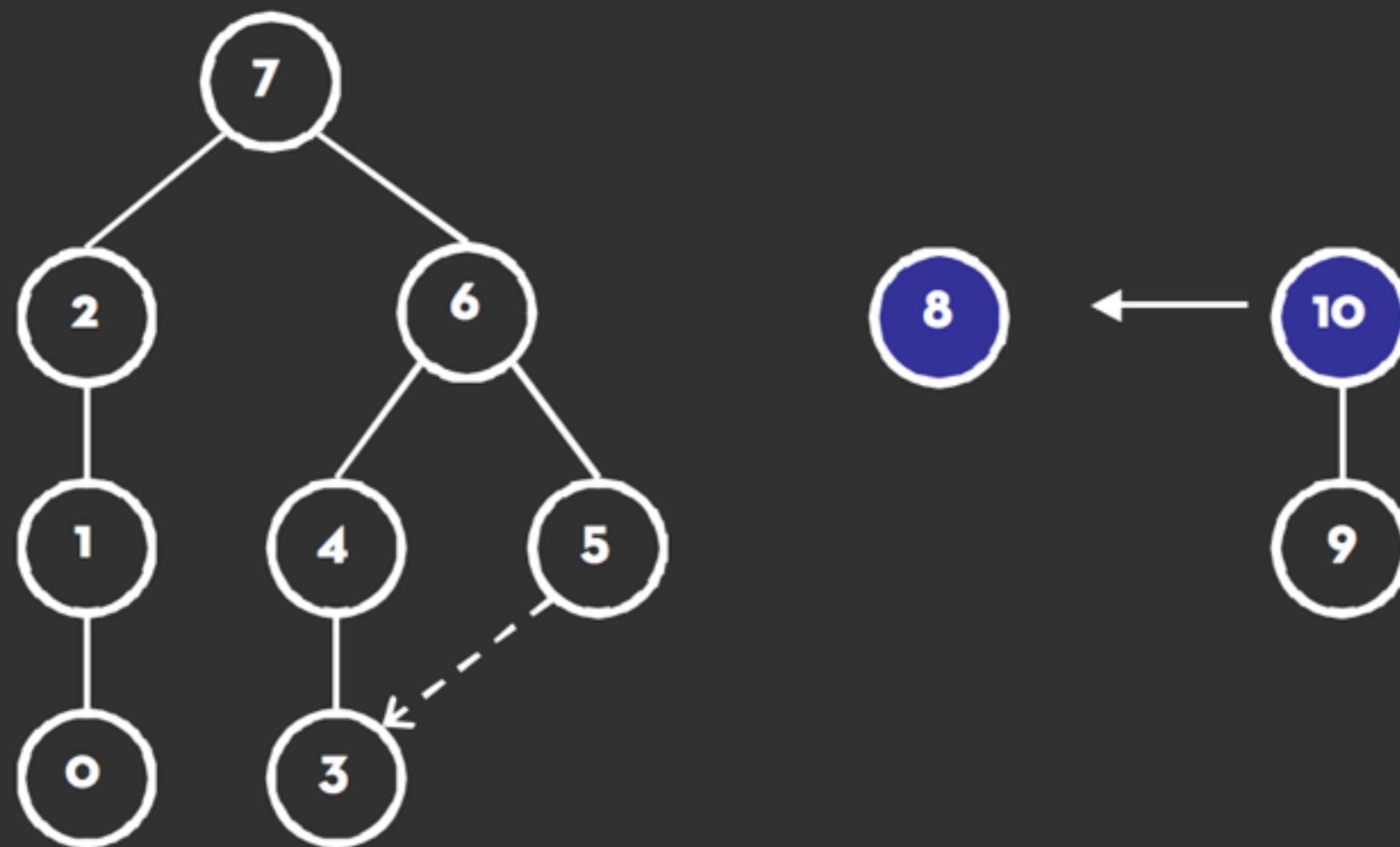
Reingold-Tilford Algorithm



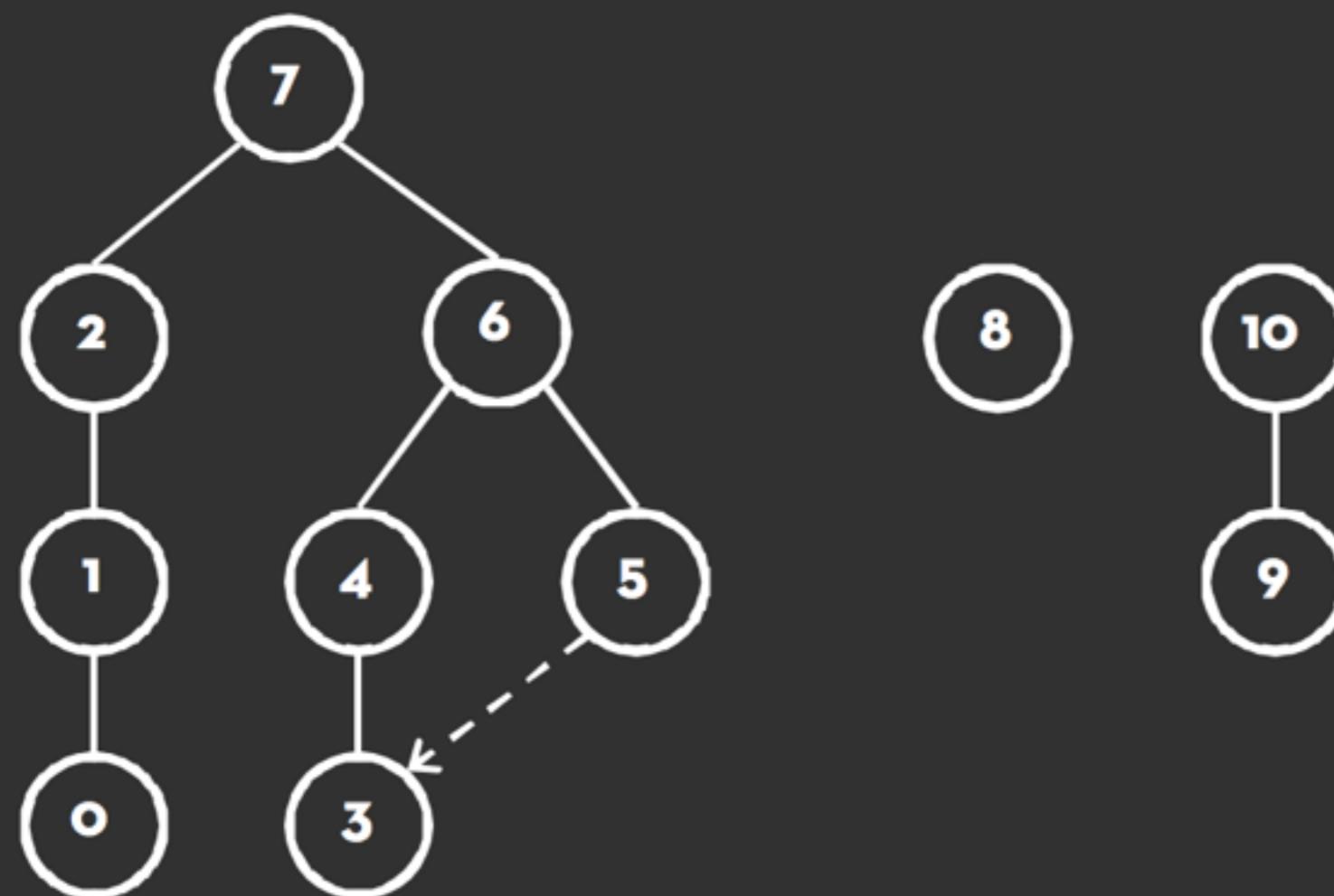
Reingold-Tilford Algorithm



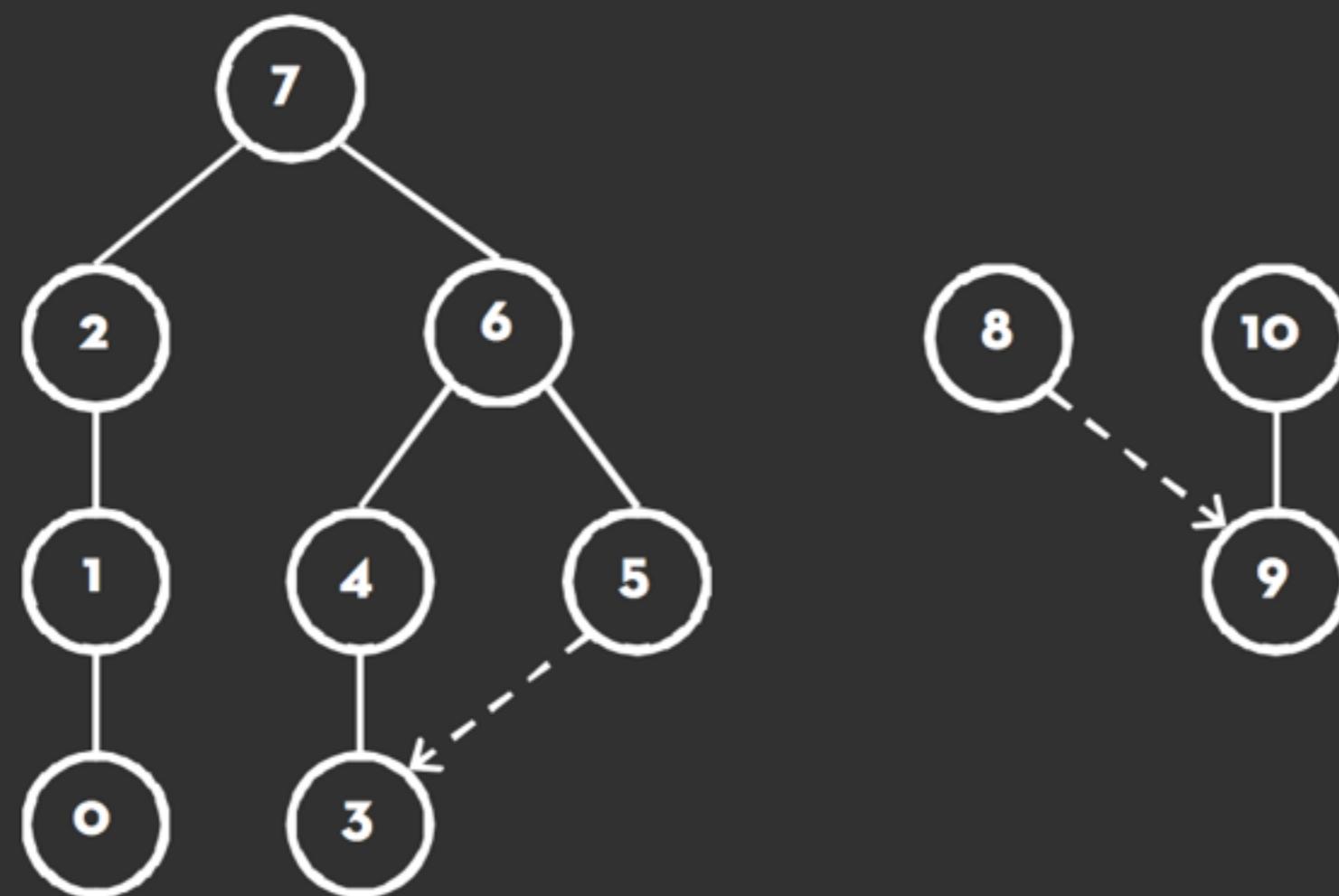
Reingold-Tilford Algorithm



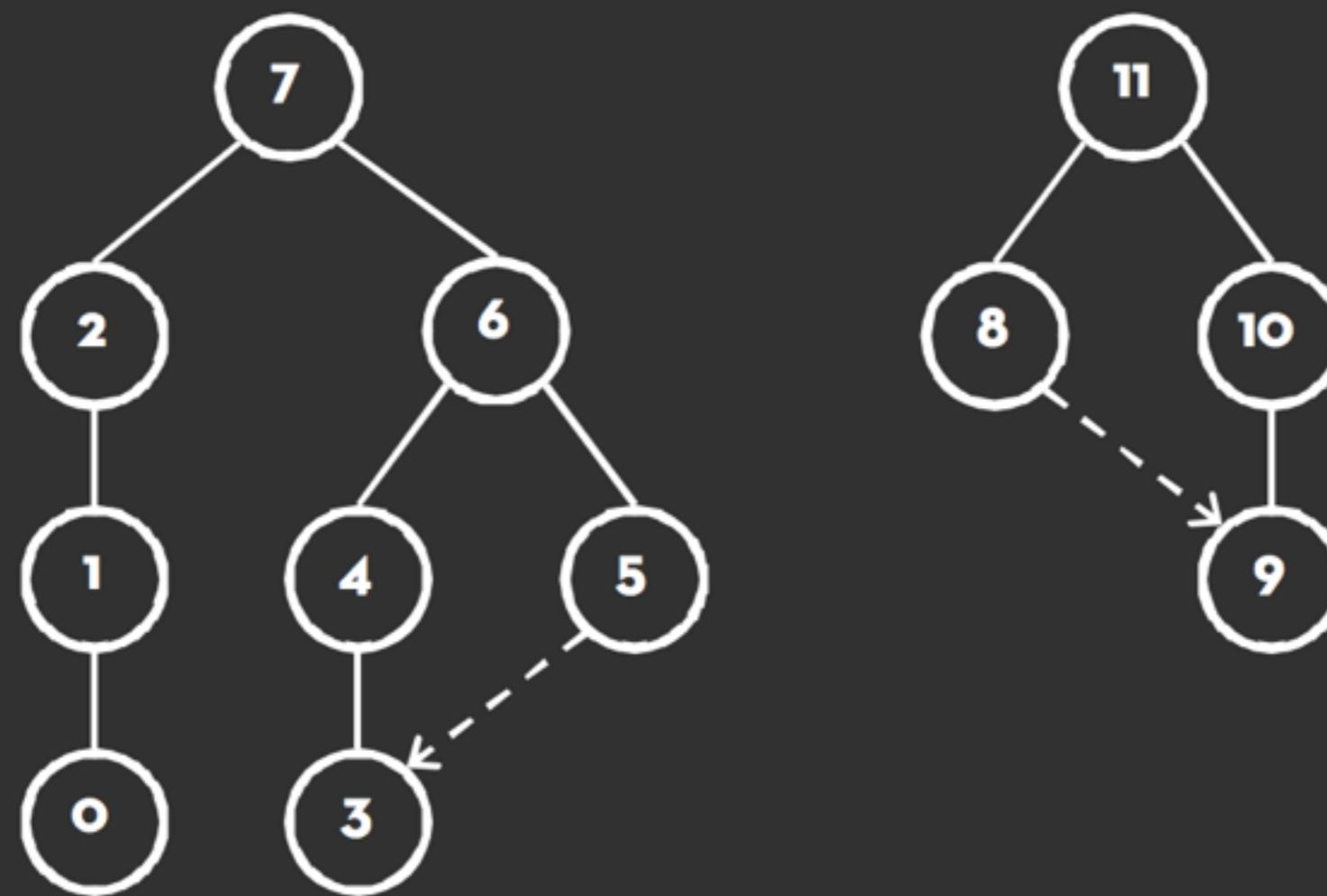
Reingold-Tilford Algorithm



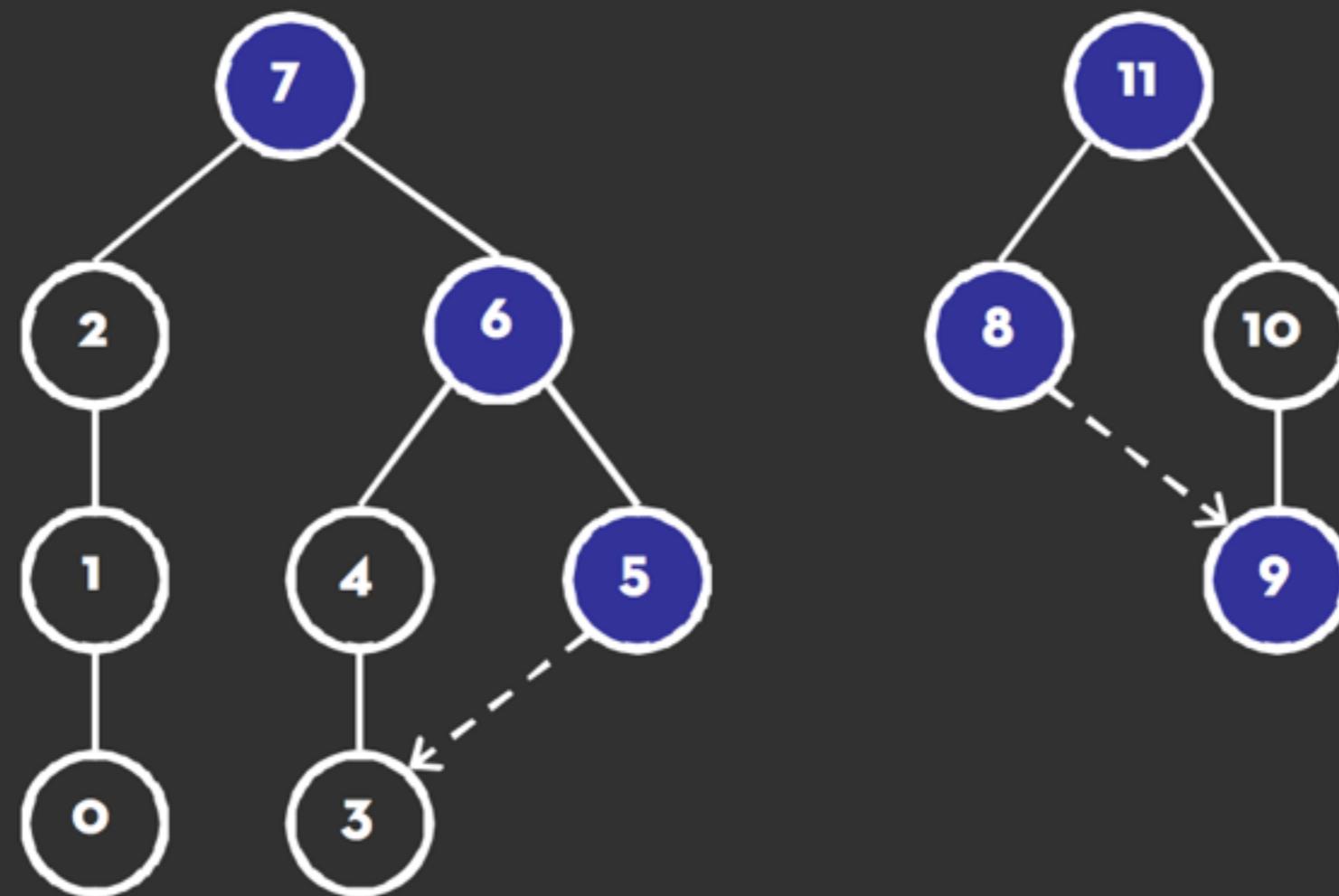
Reingold-Tilford Algorithm



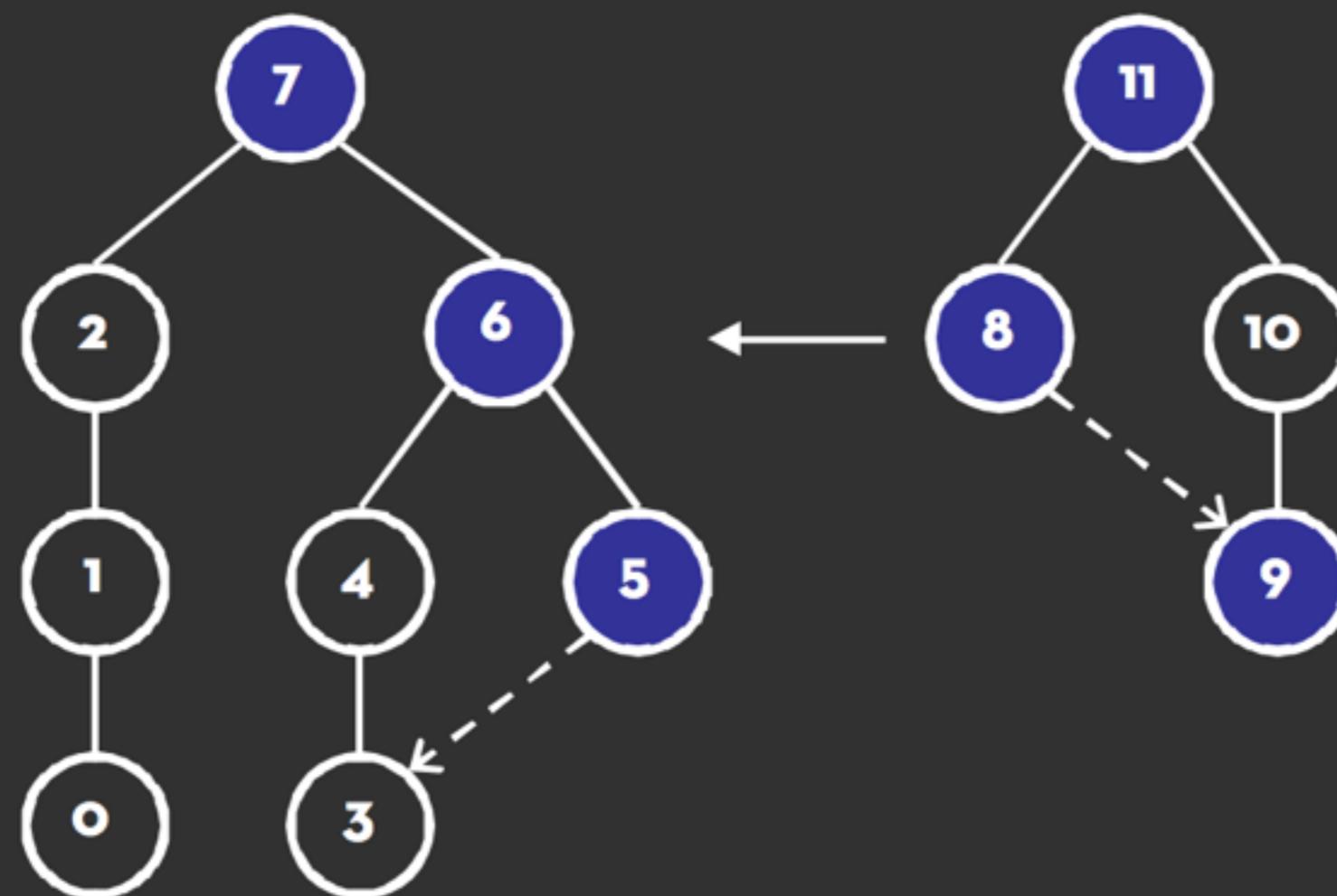
Reingold-Tilford Algorithm



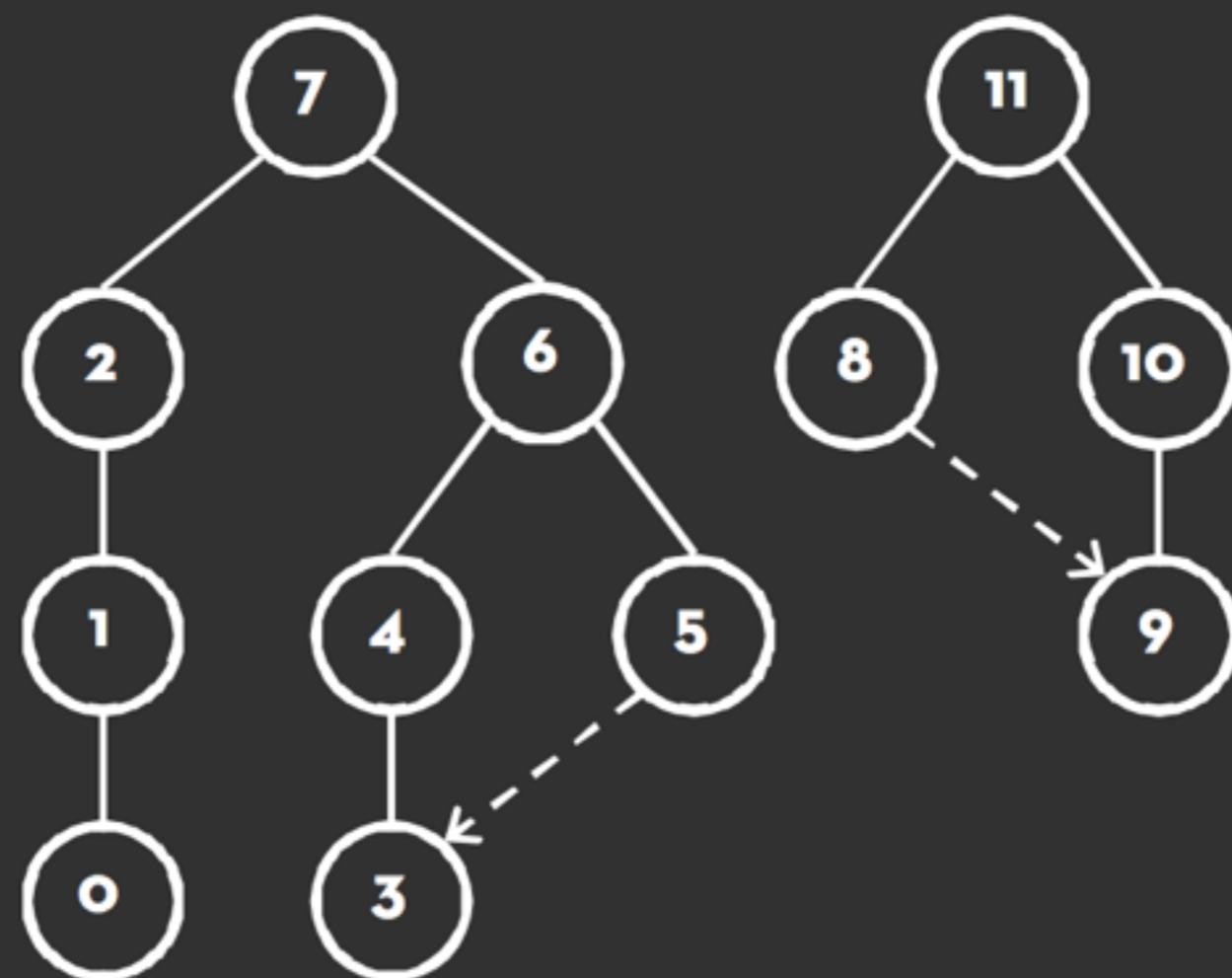
Reingold-Tilford Algorithm



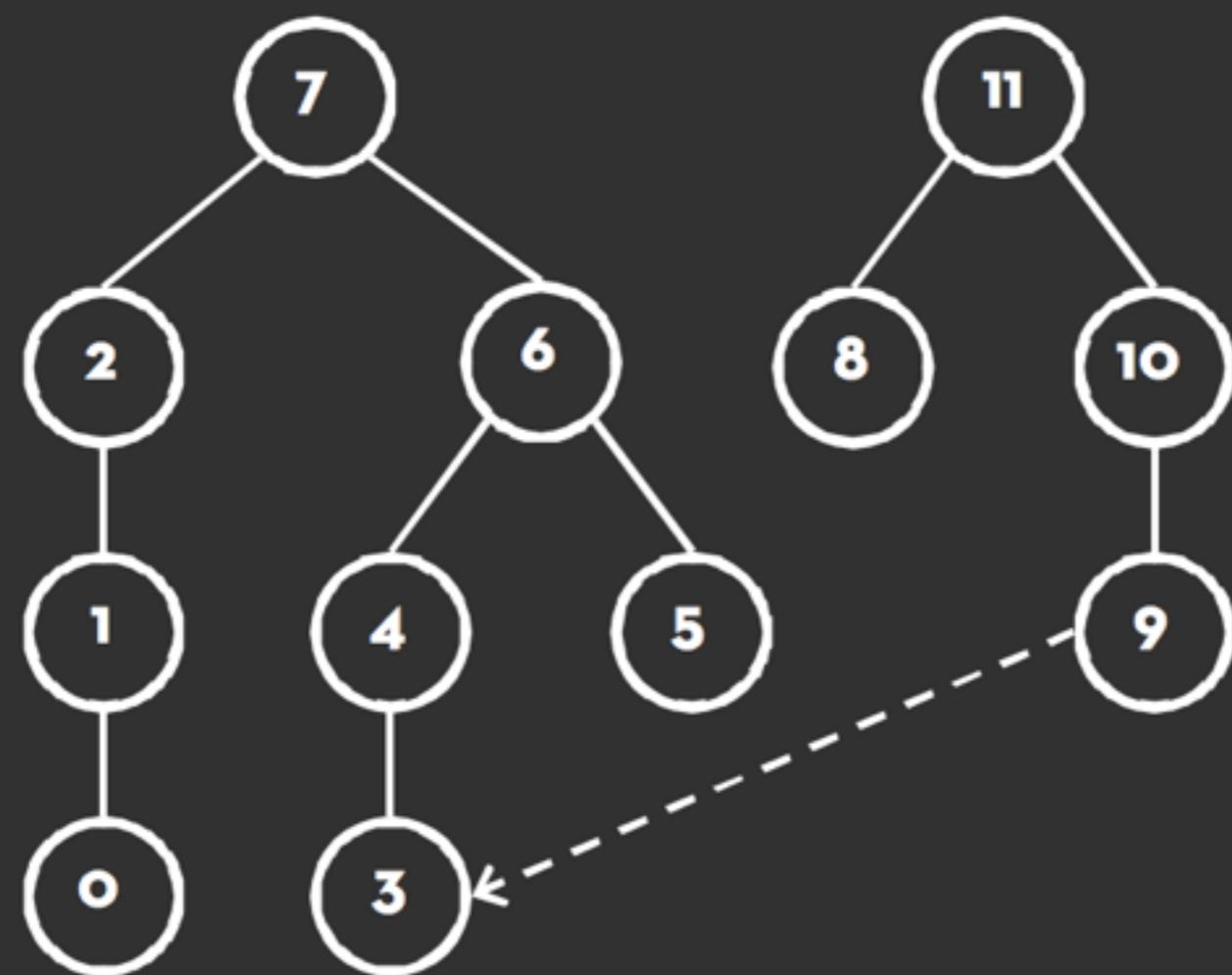
Reingold-Tilford Algorithm



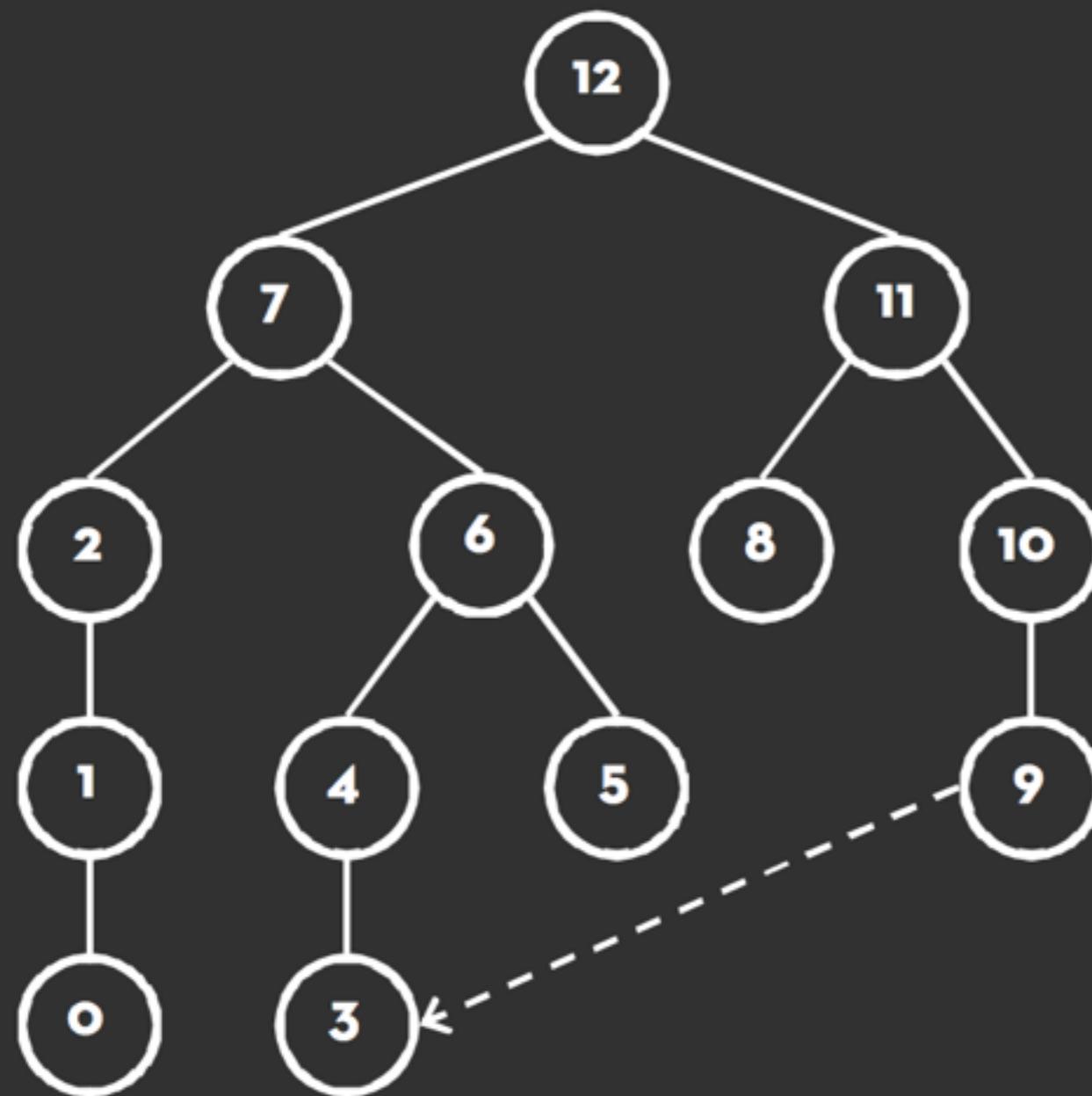
Reingold-Tilford Algorithm



Reingold-Tilford Algorithm



Reingold-Tilford Algorithm

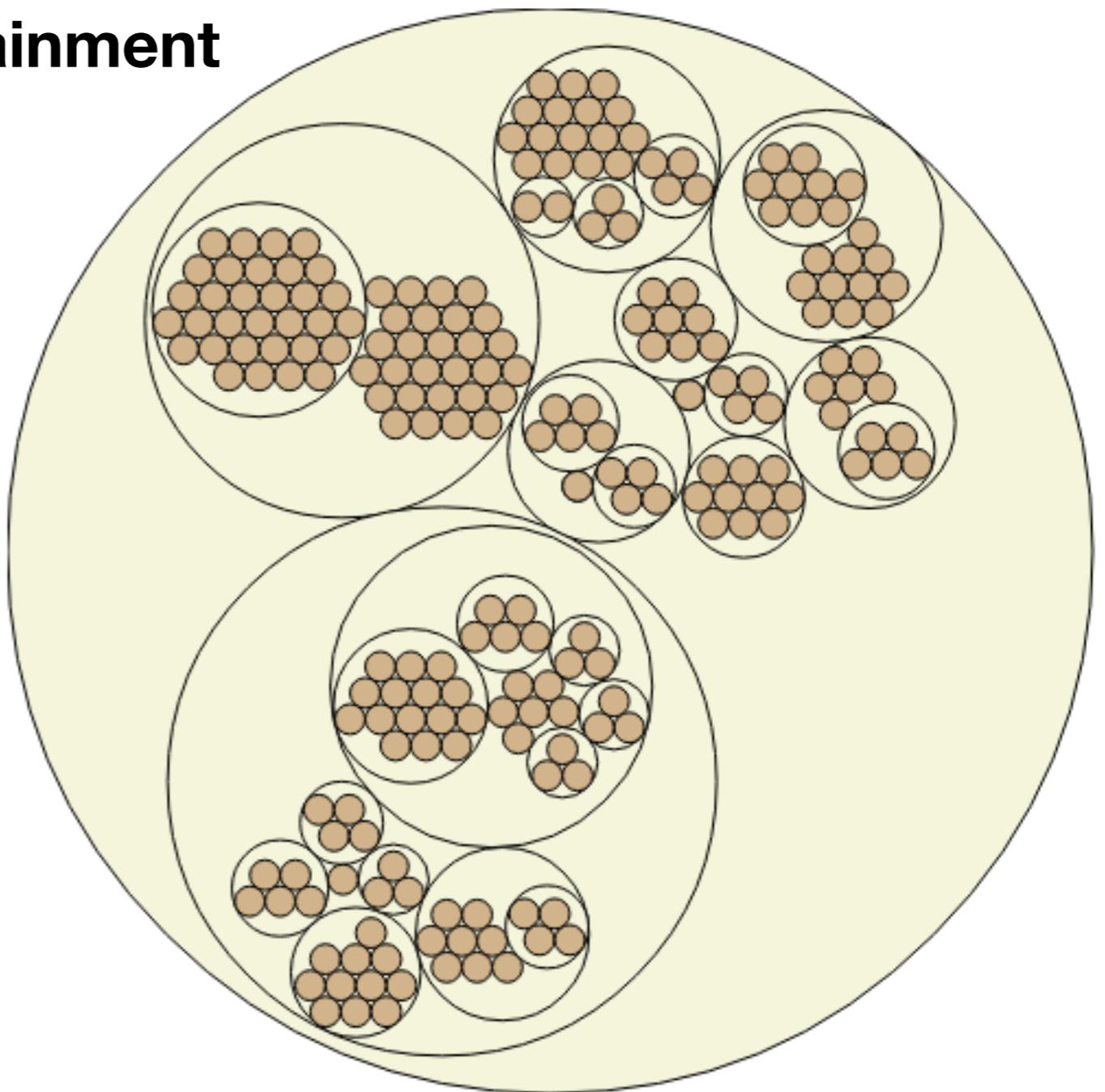


Reingold-Tilford Algorithm, once again

- Bottom-up tree traversal
- y-coord is the depth of the node, x-coords are “locally defined” (so first is arbitrary)
- merge trees
 - push right tree as close as possible to left tree (this is where the contour comes in)
 - position **shifts** saved at each node
 - parent nodes are centered above direct children
- Final top-down pass to convert shifts to positions

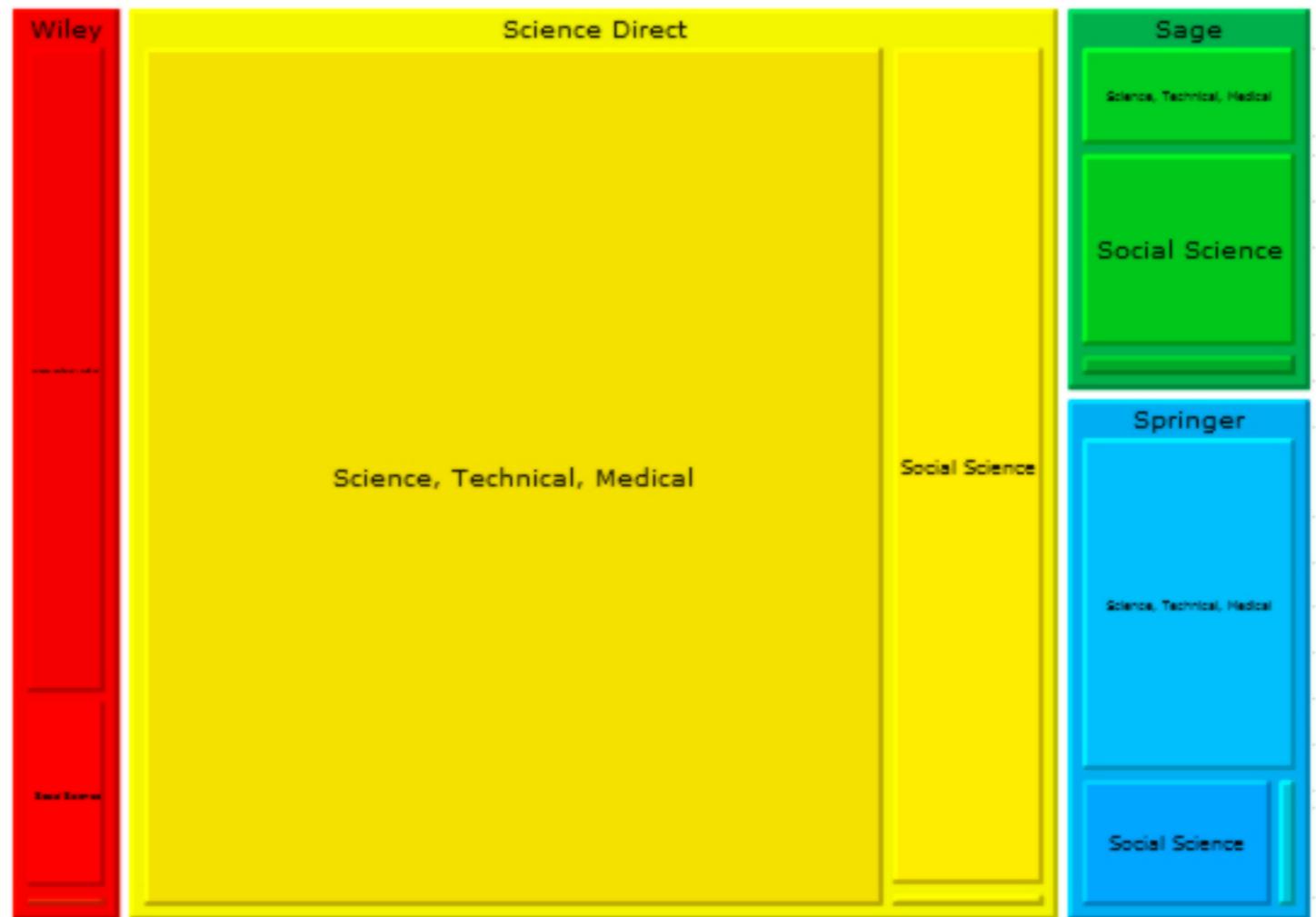
Bubble Charts

- Represent **hierarchy** by **containment**
- Let's work out a simple algo!



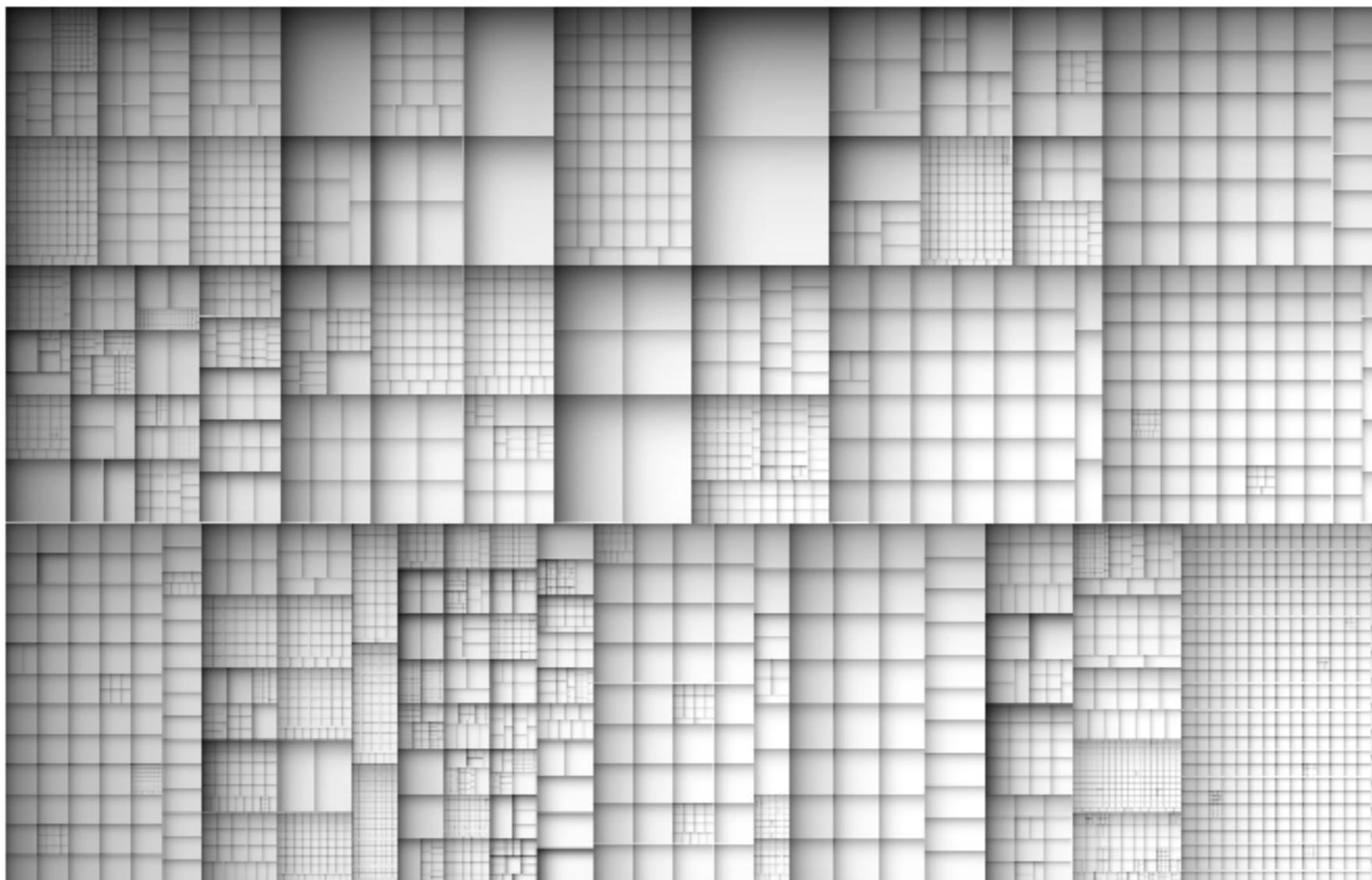
Treemaps

- Represent **hierarchy** by **containment**,
 - ... and **sizes** by **areas**
- Let's work out a simple algo!



Squarified Treemaps

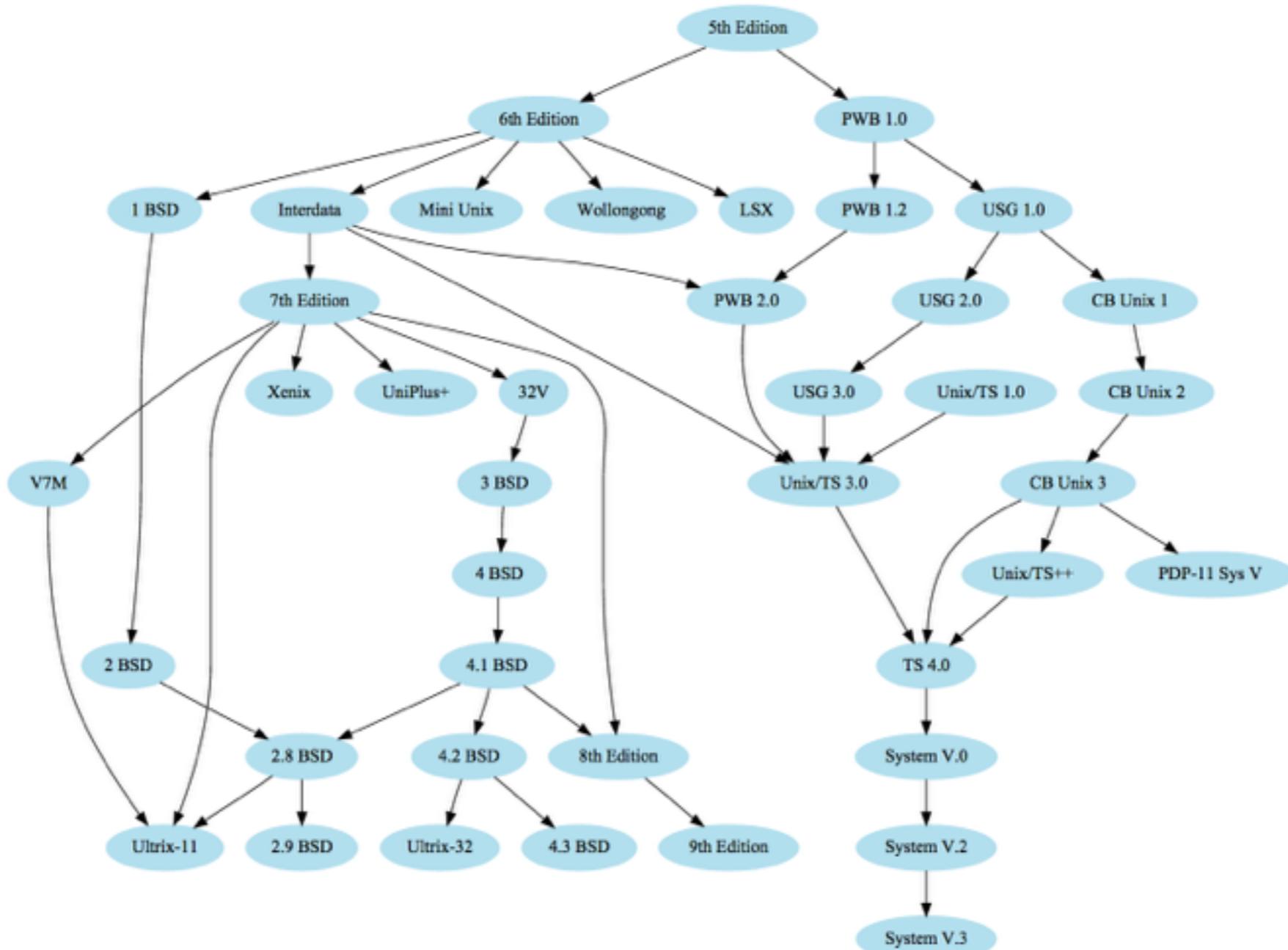
- A little harder, tries to make square shapes



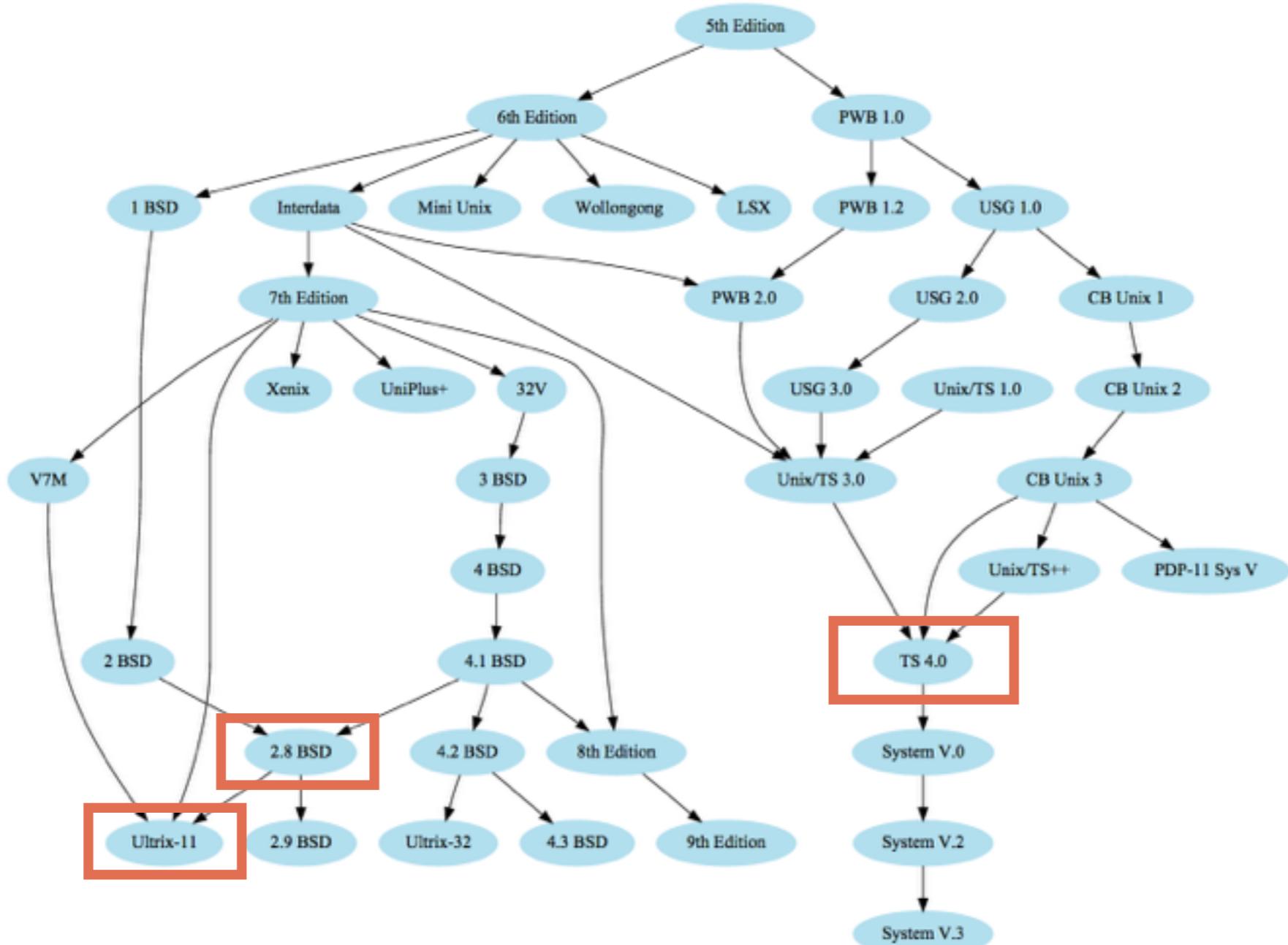
Not all Hierarchies are
Trees

Given what we know
about tree drawing, how
do we draw a DAG?

The evolution of UNIX

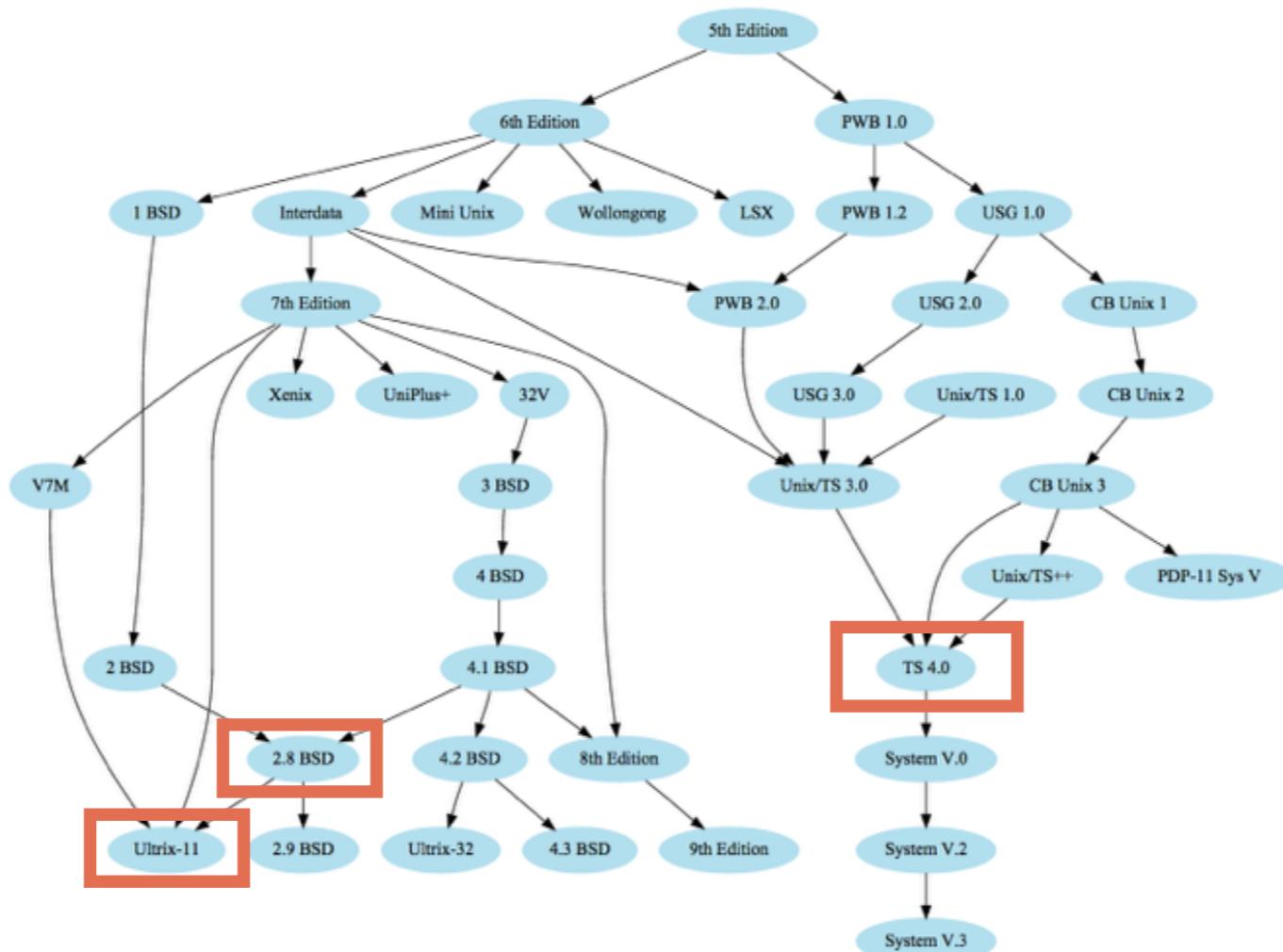


The evolution of UNIX



Directed, Acyclic Graphs

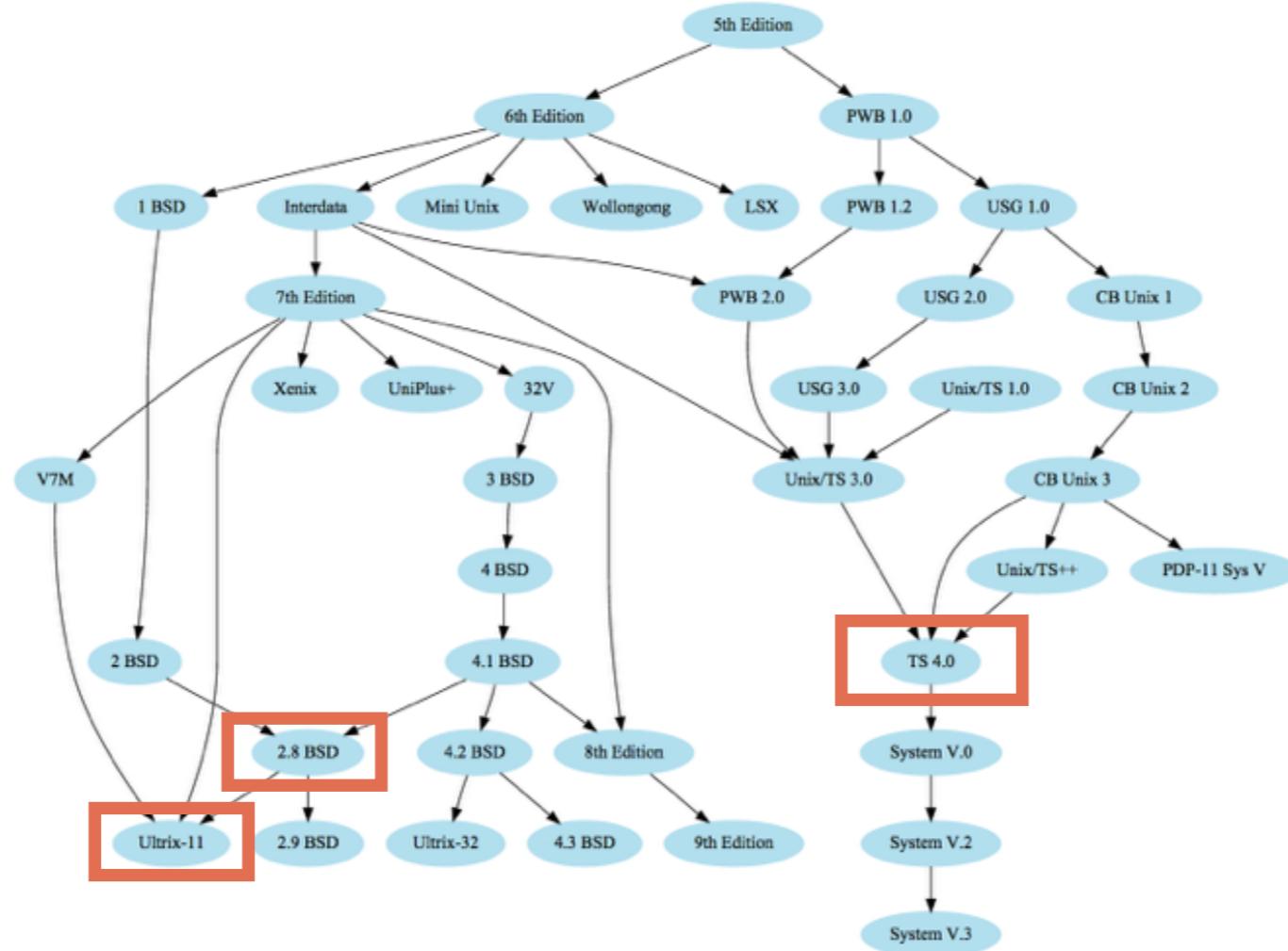
- Like a hierarchy, but “direct ancestor” is not unique



Let's draw a DAG

- Compute **rank**: height of node
 - Requirement: if aRb , $\text{height}(a) > \text{height}(b)$
- Order nodes of same rank to minimize crossings
- This is known as a “Sugiyama layout” for its inventor
- Gansner et al., *A Technique for Drawing Directed Graphs.* <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=221135>

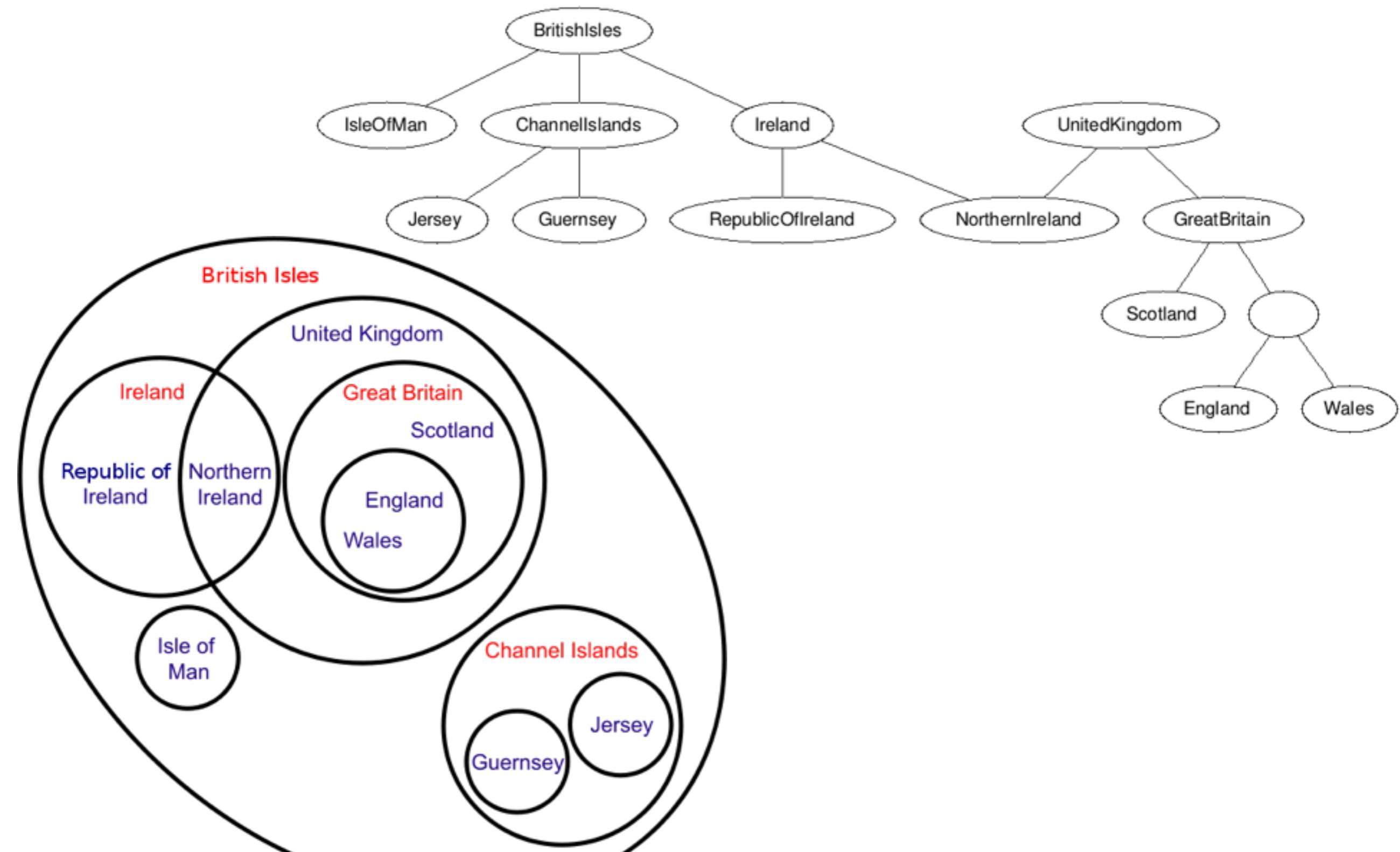
Let's draw a DAG



- Gansner et al., *A Technique for Drawing Directed Graphs.* <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=221135>

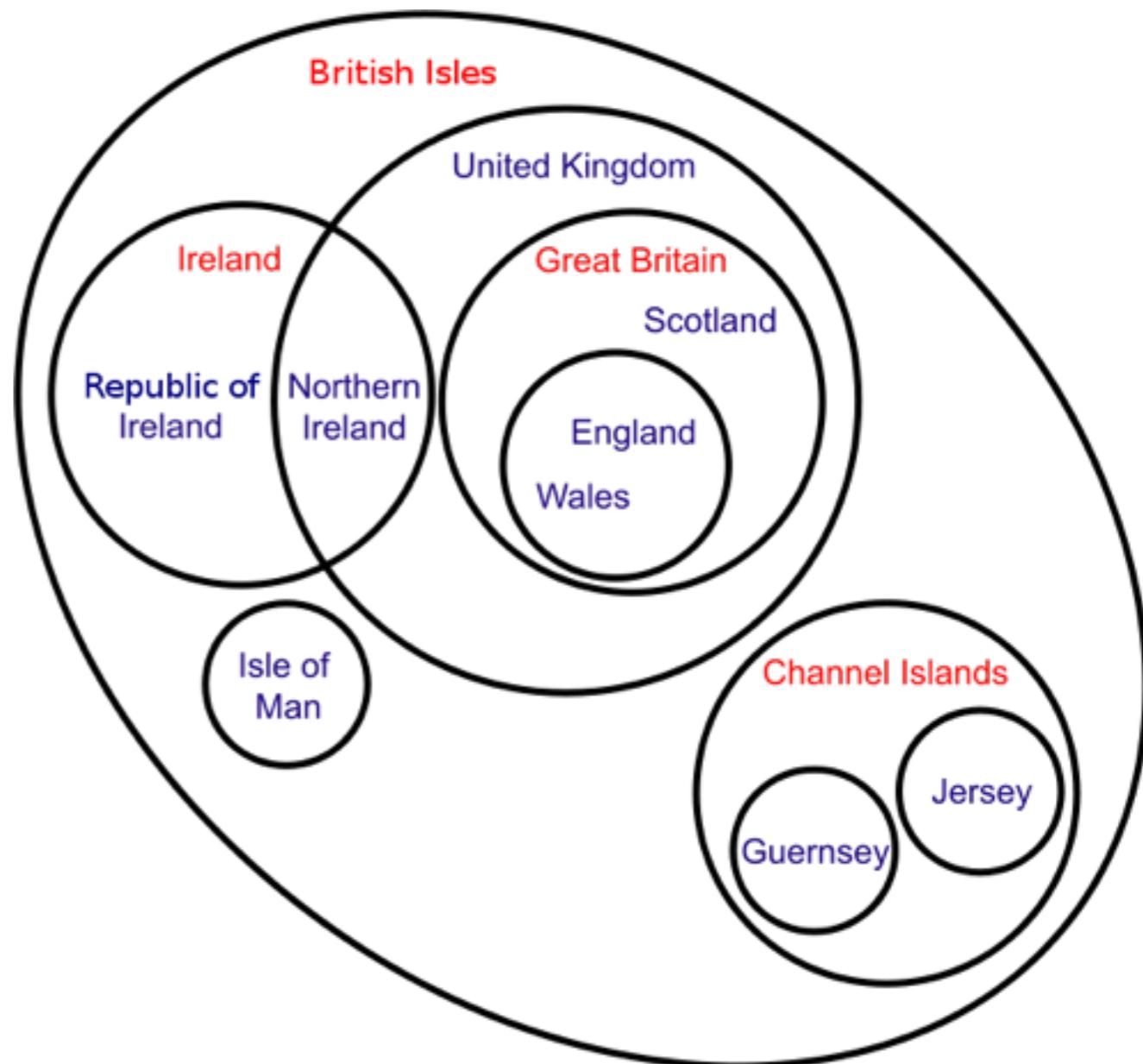
Given what we know
about treemaps, can we
draw a DAG?

Euler Diagrams (Venn Diagrams)



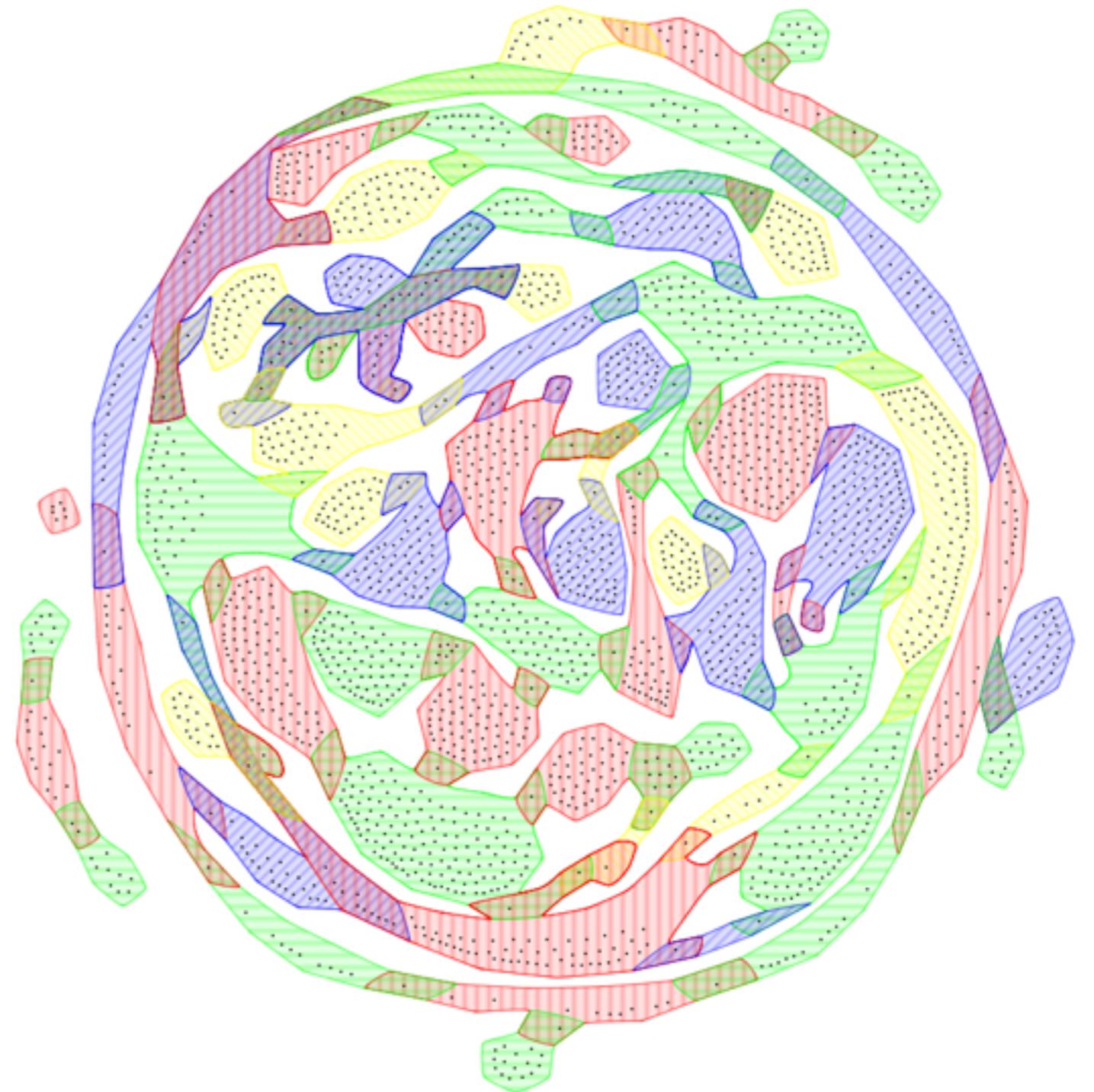
Euler Diagrams

- Represent relationship by containment
- Algorithms are very complicated, tend to produce bad shapes

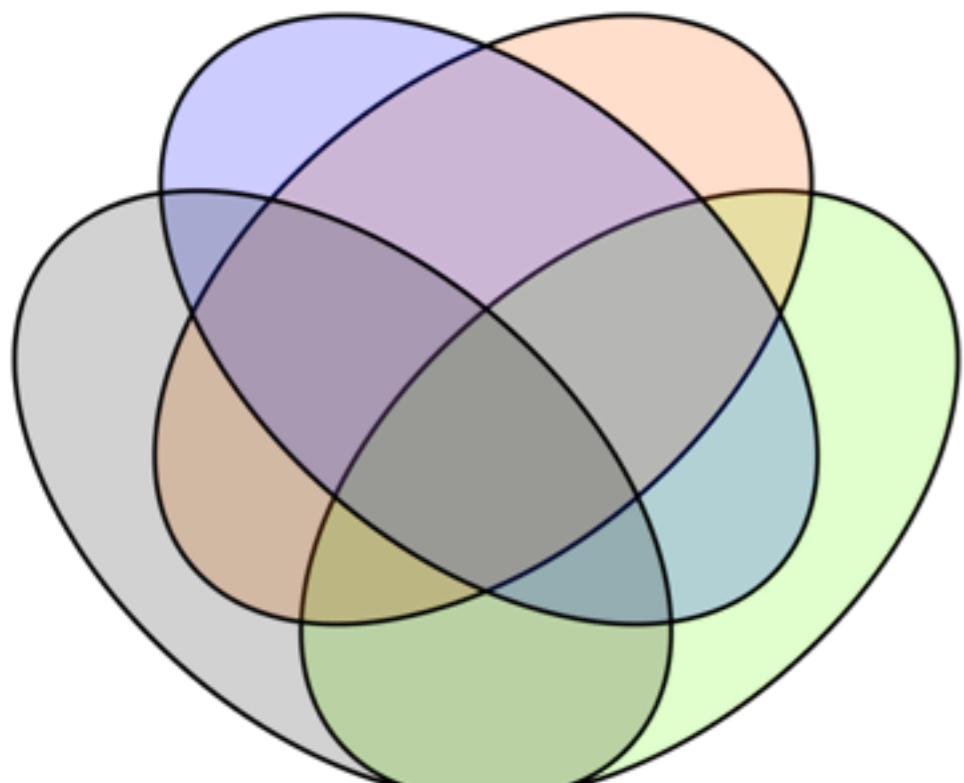


Euler Diagrams

- Doesn't scale to large diagrams



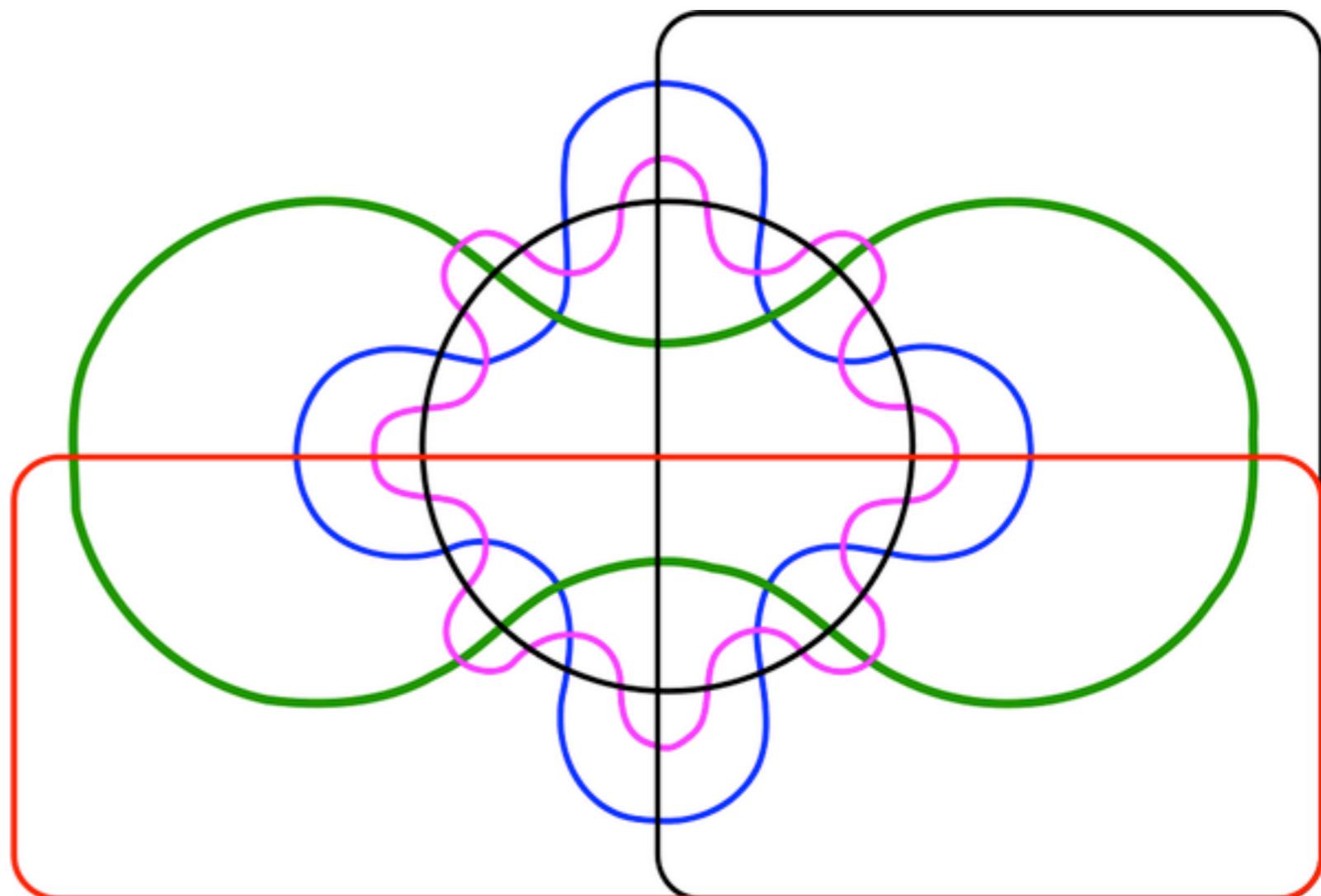
Euler Diagrams



16 regions

- Doesn't scale to “large” diagrams

64 regions



Recap

	Not a Tree	NEXT	Not a Hierarchy	Hierarchy
Not a Tree				Sugiyama's algorithm Euler Diagrams
A Tree		NEXT		Reingold-Tilford Treemaps