Trees
(Part 1, Theoretical)

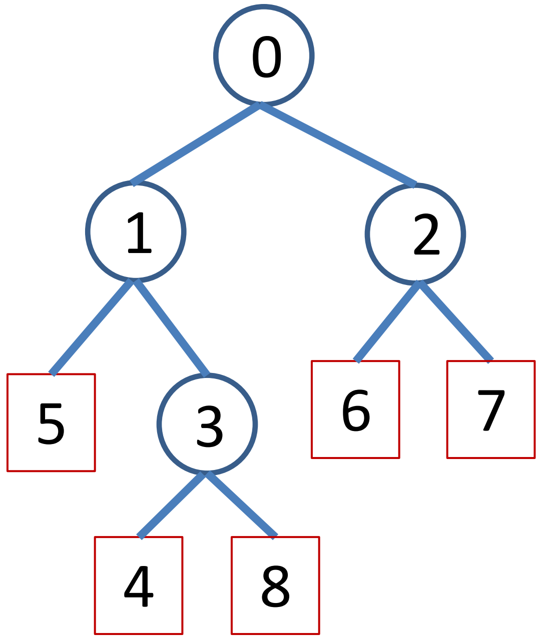
CSE 3318 – Algorithms and Data Structures
University of Texas at Arlington

Trees

- Trees are a natural data structure for representing specific data.
  - Family trees.
  - Organizational chart of a corporation, showing who supervises who.
  - Folder (directory) structure on a hard drive.
- Theoretical usage – analysis of recursive functions with 2 or more recursive calls

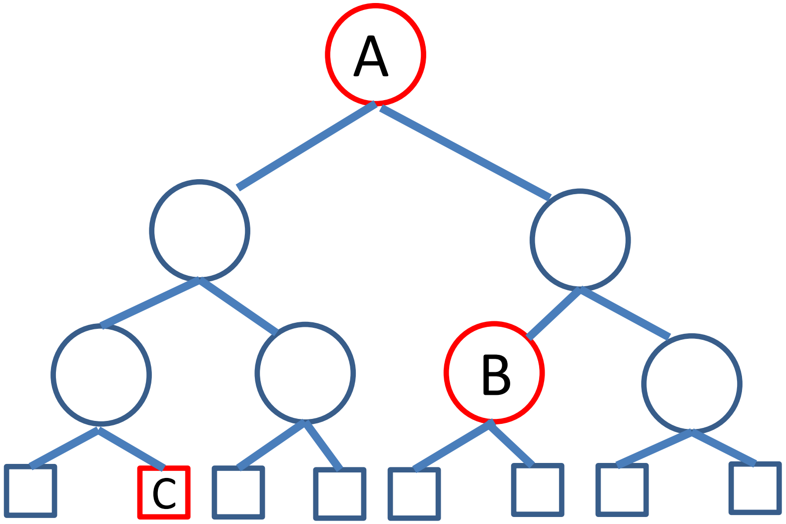
Terminology

- ***Root***: 0   (has no parent)
- ***Path***: 0-2-6, 1-3, 1-3-4
- ***Parent*** vs ***child***
- Ascendants vs descendants:
  - ascendants   of 3:   1, 0
  - descendants of 1:   5, 3, 4, 8
- ***Internal nodes***: 0, 1, 2, 3
  - Have 1 or more children
- ***Leaves***: 5,4,6,7,8
  - Have no children
- ***Subtree***



Terminology - Worksheet

- The ***level*** of the root is defined to be 0.
- The ***level*** of each node is defined to be 1+ the level of its parent.
- The ***depth*** of a node is the number of edges from the root to the node.
  (It is equal to the level of that node.)
- The ***height*** of a node is the number of edges *from the node to the deepest leaf*.
  (Treat that node as the root of a small tree)

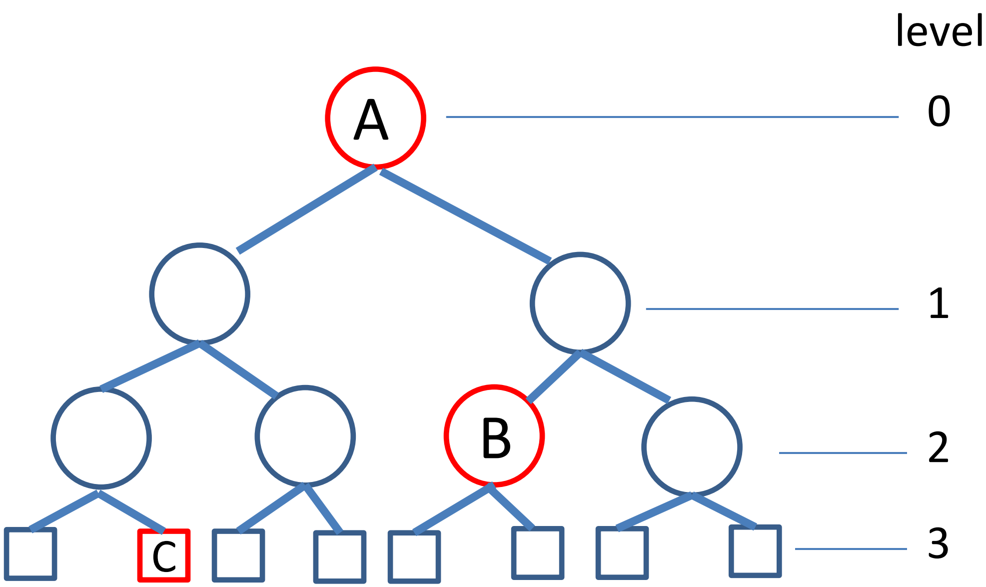


Node	level	depth	height
A			
B			
C			

Practice:
- Give the level, depth and height for each of the red nodes.
- How many nodes are on each level? ____________________________________

Terminology - Answers

- The ***level*** of the root is defined to be 0.
- The ***level*** of each node is defined to be 1+ the level of its parent.
- The ***depth*** of a node is the number of edges from the root to the node. (It is equal to the level of that node.)
- The ***height*** of a node is the number of edges *from the node to the deepest leaf*. (Treat that node as the root of a small tree)



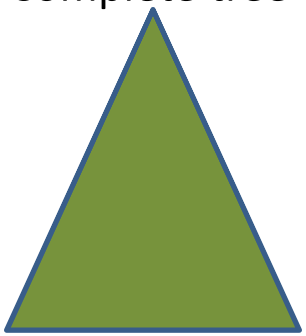
Node	level	depth	height
A	0	0	3
B	2	2	1
C	3	3	0

Practice:
- Give the level, depth and height for each of the red nodes.
- How many nodes are on each level? __________1, 2, 4, 8____________________

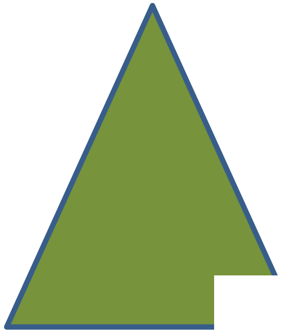
Types of binary trees

- **Complete (**or **perfect)**– each internal node has exactly 2 children and all the leaves are on the same level.
  - E.g. ancestry tree (anyone will have exactly 2 parents).
- **Full [binary]** – every node has exactly 0 or 2 children.
  - Tree of recursive calls when there are 2 recursive calls
    - E.g. tree generated by the Fibonacci recursive calls.
    - *=> his properties are used in analyzing time complexity of such rec fcts.*
  - *Binary* tree.
- **Nearly complete tree** – every level, except for possibly the last one is completely filled and on the last level, all the nodes are as far on the left as possible.
  - E.g. the heap tree.
  - Height: $\lfloor \lg N \rfloor$ and it can be stored as an array.

Complete tree



Nearly Complete tree



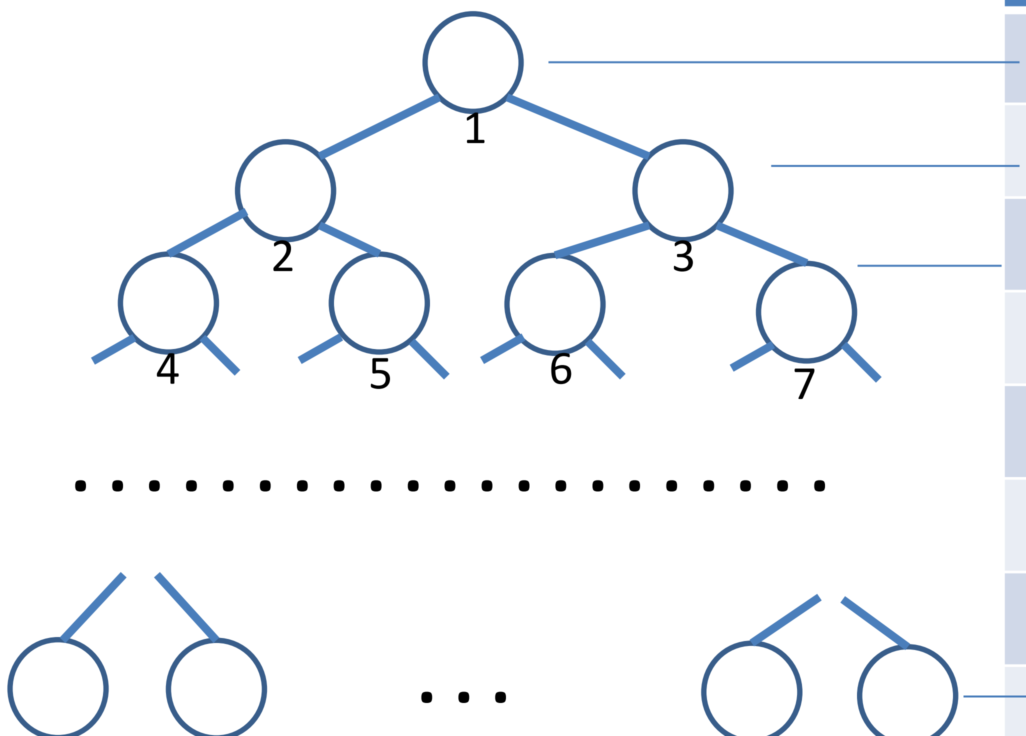
Complete Binary Trees

A **complete binary tree** with N nodes and height h, has:

- $\lceil N/2 \rceil$ leaves (half the nodes are on the last level)
- $\lfloor N/2 \rfloor$ internal nodes (half the nodes are internal)
- Height : $h = \lfloor \lg N \rfloor$
- Levels : $\lfloor \lg N \rfloor + 1$ (= lg(N+1))

$$\sum_{k=0}^{h} 2^k = 2^{h+1} - 1$$

In the other direction: N = 2^{h+1} -1

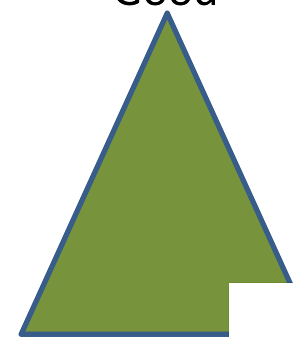


Level	Nodes per level	Sum of nodes from root up to this level
0	2^0 (=1)	$2^1 - 1$ (=1)
1	2^1 (=2)	$2^2 - 1$ (=3)
2	2^2 (=4)	$2^3 - 1$ (=7)
…	…	
i	2^i	$2^{i+1} - 1$
…	…	
h-1	2^{h-1}	2^h-1
h	2^h	$2^{h+1} - 1$

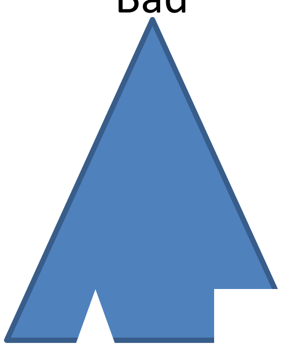
Nearly Complete Tree

- All levels are full, except possibly for the last level.
- At the last level:
  - Nodes are on the left.
  - Empty positions are on the right.
- There is “no hole”

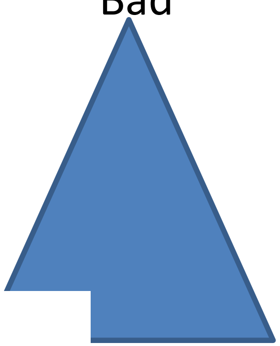
Good



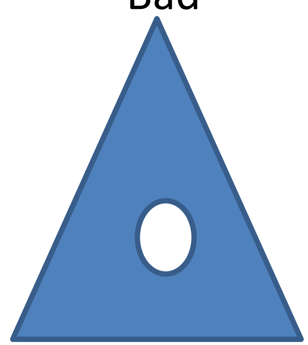
Bad

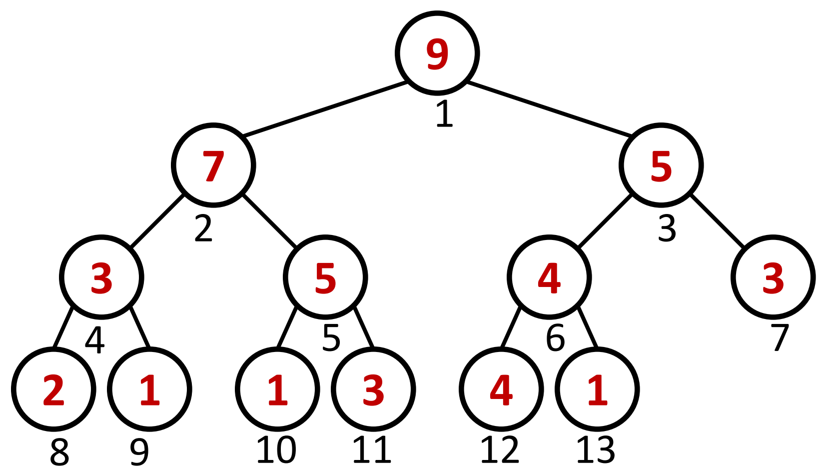


Bad



Bad





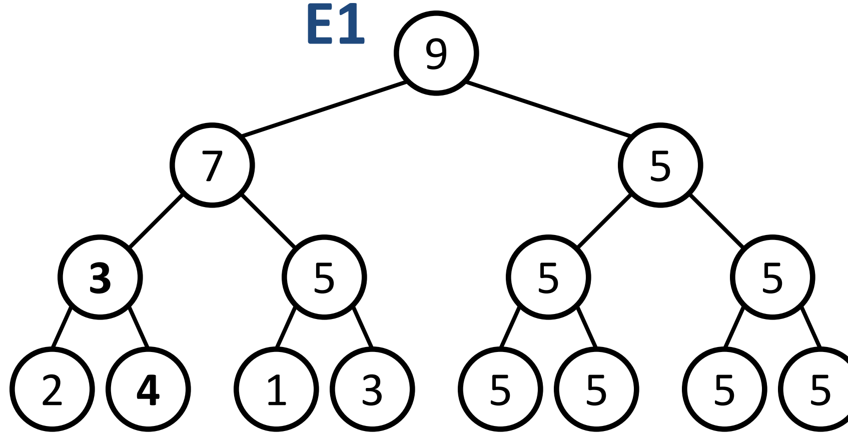
Worksheet

- Self study: Give examples of trees that are:
  - Complete
  - Full but not nearly complete
  - Nearly complete but not full
  - Neither full not nearly complete

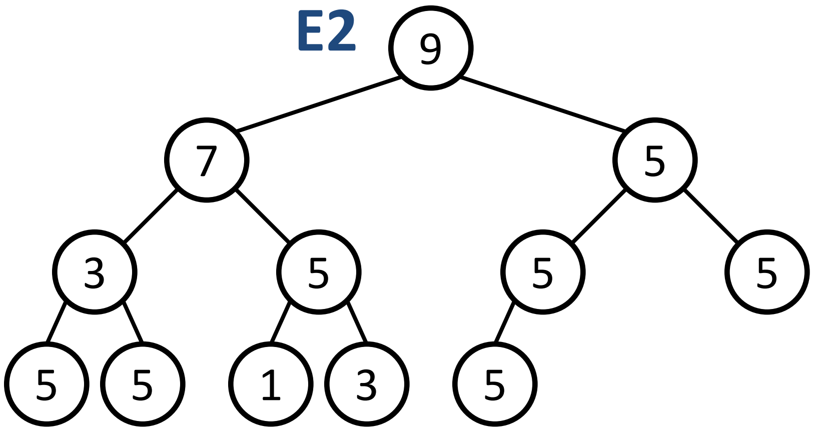
Worksheet

For each tree, say if it is full, nearly complete or complete.

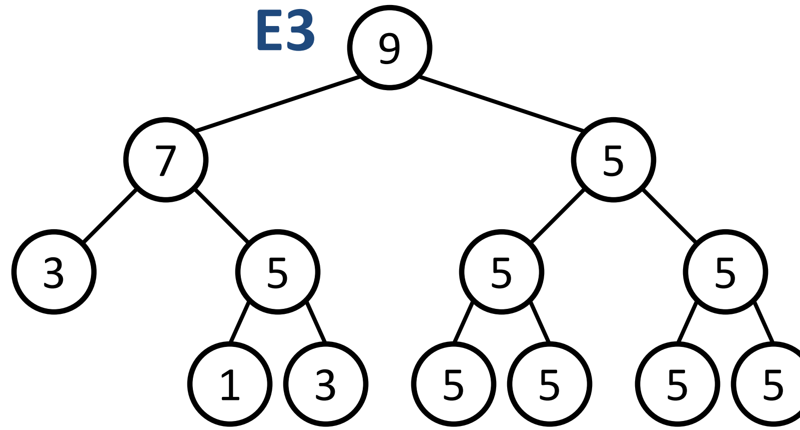
E1



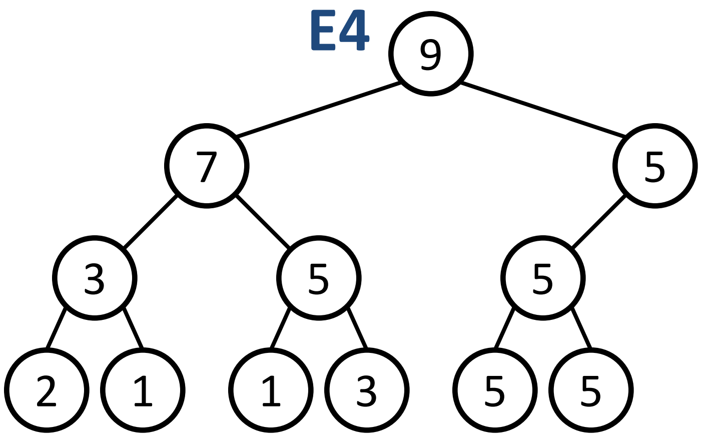
E2



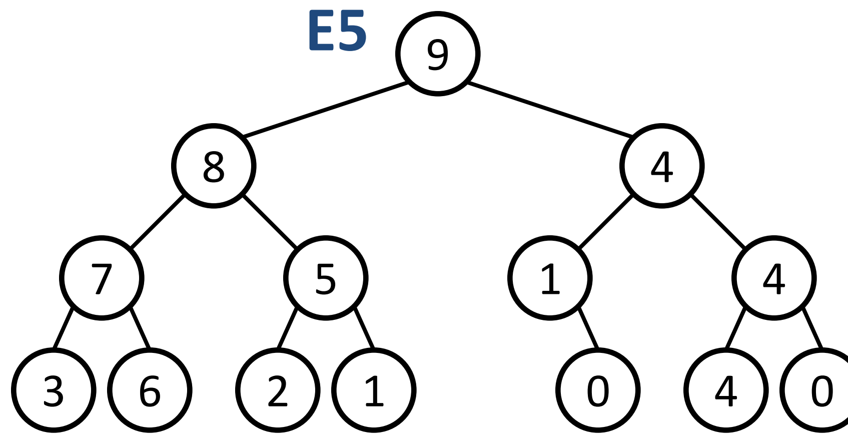
E3



E4



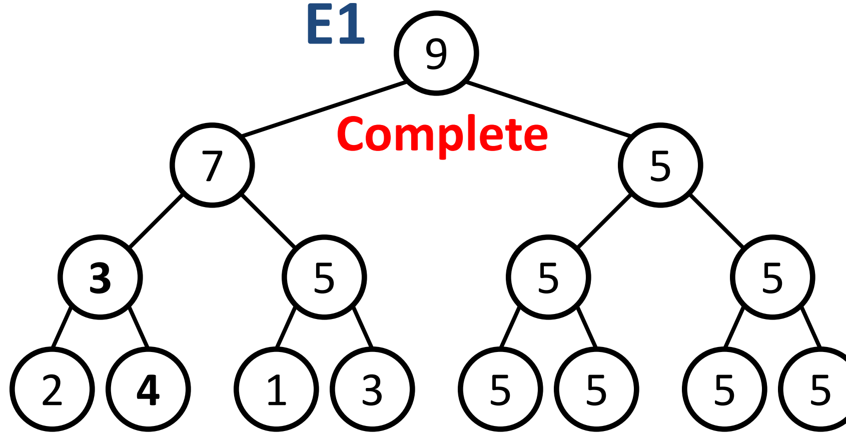
E5



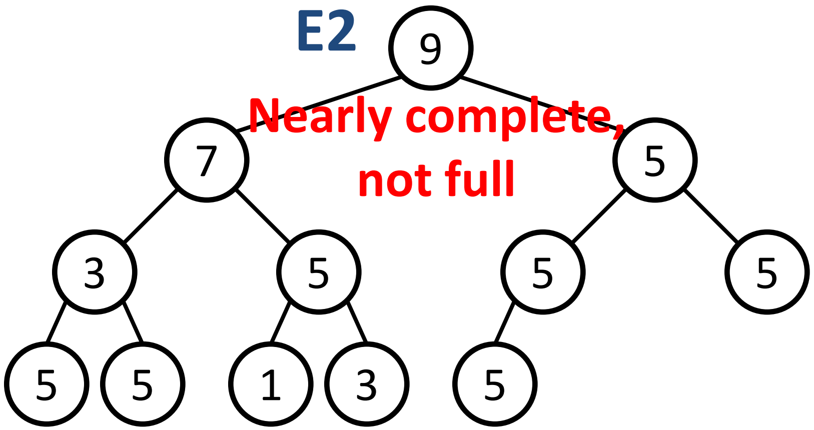
Answers

For each tree, say if it is perfect, nearly complete or complete.

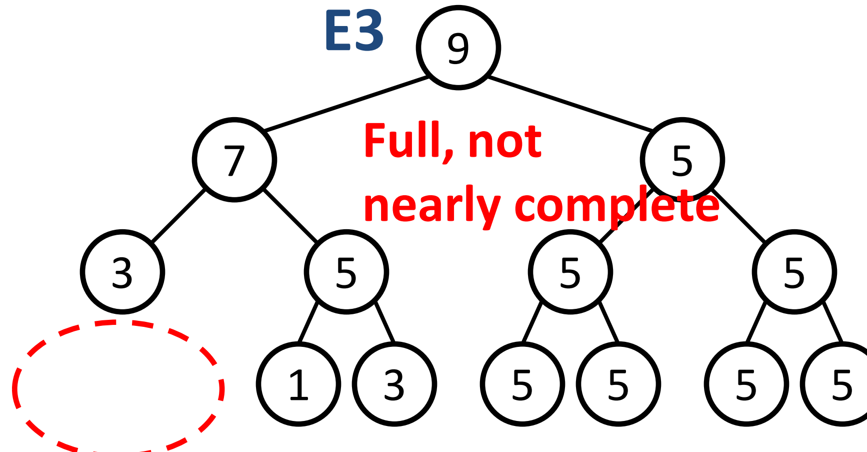
E1



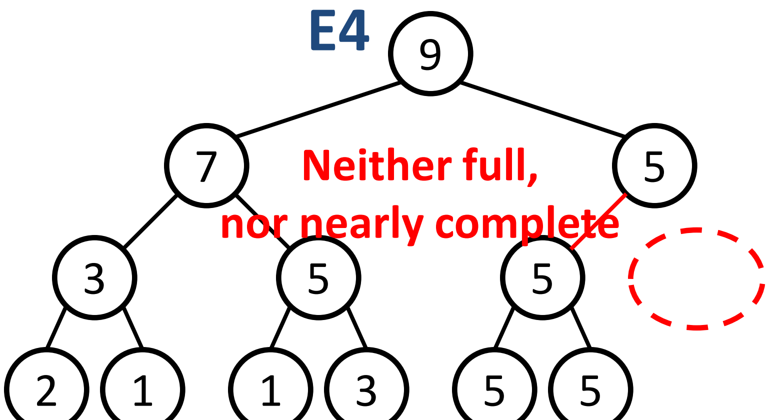
E2



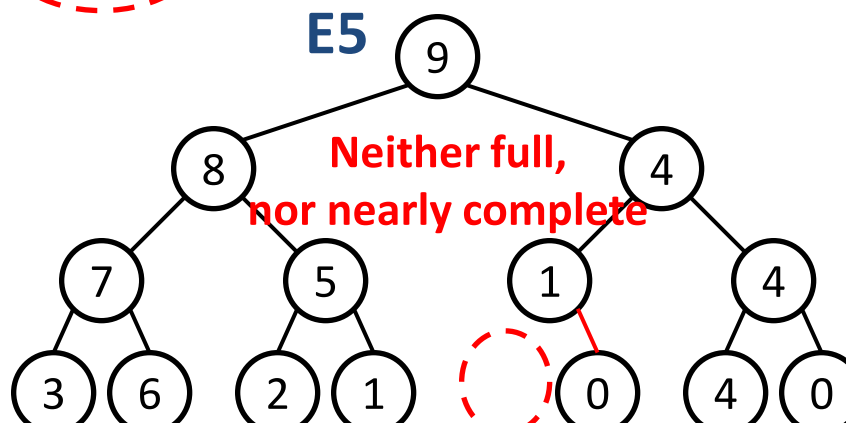
E3



E4

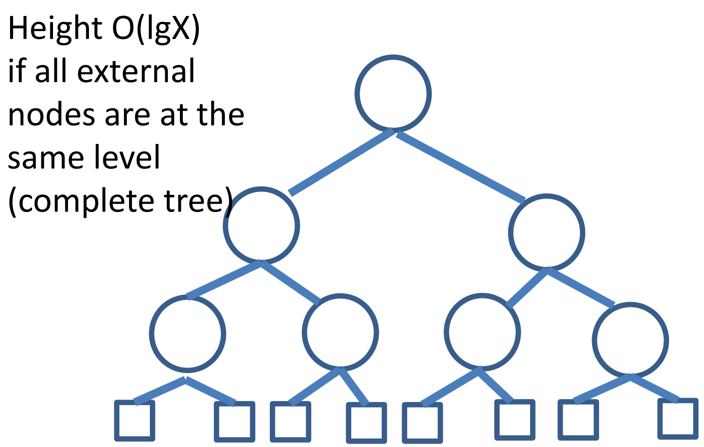


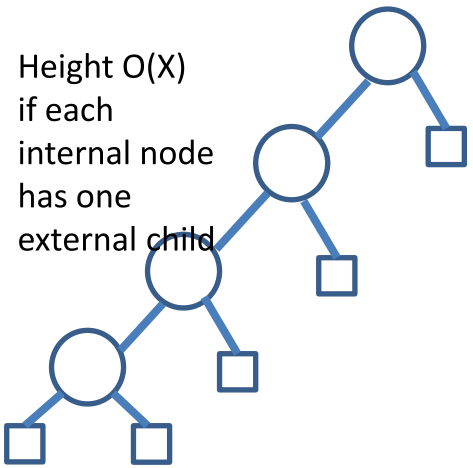
E5



Properties of Full Trees

- **Full** binary tree : every node has exactly 0 or 2 children. No nodes with only 1 child.
- A **full** binary tree with X internal nodes has:
  - X+1 external nodes.
  - 2X edges (links).
  - N = 2X+1 (total number of nodes)
  - height at least lg X and at most X:





Proof

Prove that a full tree with P internal nodes has P+1 external leaves.

- Full tree property:
- What proof style will you use?

Proof

Prove that a full tree with P internal nodes has P+1 external leaves.

- Full tree property:
- What proof style will you use?