Mathematical Properties of Trees

Textbook Reading:

Chapter 4, Section 4.2, pp. 143-149

Mathematical Properties of Trees

- Mathematical properties of trees are important in the analysis of algorithms.
- In this lecture we obtain
  - a result relating the number of nodes and number of edges of a general tree
  - lower bound for the depth of a binary tree in terms of the number of nodes
  - relation between the number of leaf nodes and the total number of nodes in a 2-tree
  - lower bound for the depth of a binary tree in terms of the number of leaf nodes

Number of edges vs. number of nodes in a tree

Theorem. The number m of edges of any tree T is one less than the number n of nodes.

Proof by Induction. We perform induction on the number of n of nodes.

Basis Step. A tree with one node has no edges, i.e., we have $m = 0 = n - 1$.

Induction Step

Assume true for $n = k$, i.e., any tree having k nodes has $k - 1$ edges.

Now consider a tree T having $n = k + 1$ nodes. For convenience, let $n(T)$ and $m(T)$ denote the number of nodes and edges of T.

PSN. To apply the induction hypothesis, we need to perform an operation that reduces T to a tree T' having k nodes. How to do this?

Applying Induction Hypothesis

Since T' has k nodes, we can apply the induction hypothesis (inductive assumption)

$$m(T') = n(T') - 1 \quad \text{(by Induction Hypothesis)}$$

Thus, we have

$$\begin{aligned} m(T) &= m(T') + 1 \\ &= (n(T') - 1) + 1 \qquad \text{(substituting)} \\ &= n(T') = n(T) - 1. \end{aligned}$$

This completes the induction step and the proof.

Lower Bound on Depth of a Binary Tree

Proposition 4.2.2 Every binary tree with n nodes has depth d at least

$$\lfloor \log_2 n \rfloor$$

Further, equality is achieved for the complete binary tree.

Proof

Let n_i be the number of nodes at level i, , $i = 0, 1, \ldots, d$.

Since tree is binary, each node has at most 2 children, so that

$$n_i \le 2n_{i-1}\ , i = 1, 2, \ldots, d \text{ and } n_0 = 1.$$

An easy induction proves that

$$n_i \le 2^i\ , i = 0, 1, \ldots, d. \qquad (1)$$

Clearly,

$$n = n_0 + n_1 \ldots + n_d. \qquad (2)$$

Substituting (1) into (2) and using result $x^0 + x^1 + \cdots + x^d = (x^{d+1} - 1)/(x - 1)$:

$$n \le 2^0 + 2^1 \ldots + 2^d = 2^{d+1} - 1.$$

We have

$$2^{d+1} \ge n + 1$$

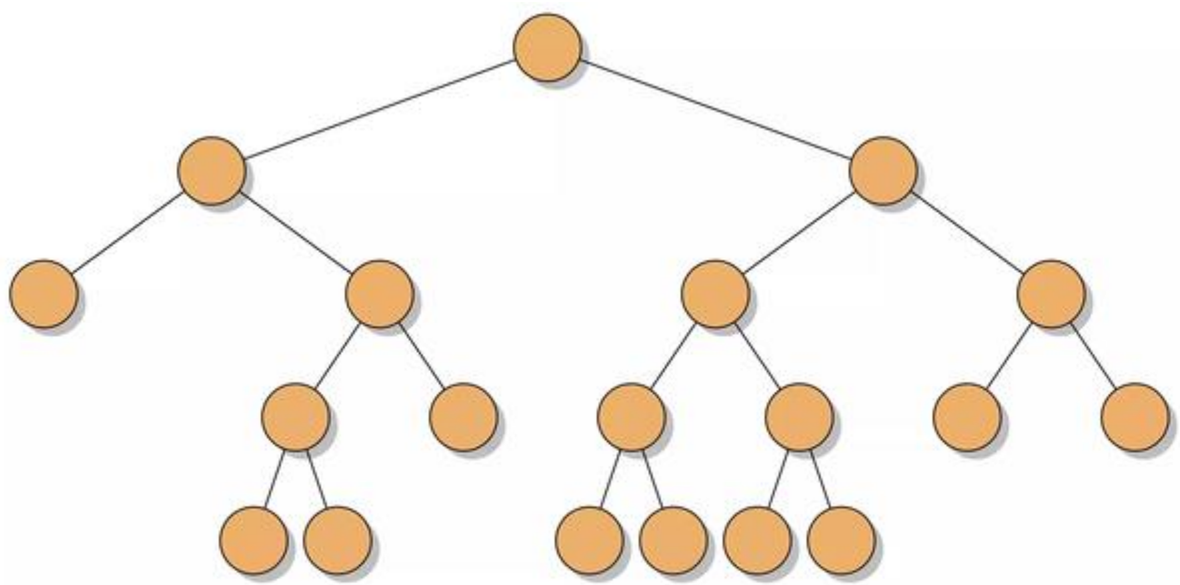
$$\Rightarrow d + 1 \ge \log_2(n+1)$$

$$\Rightarrow d \ge \log_2(n+1) - 1.$$

Using fact that d is an integer it is easy to show that $d \ge \lfloor \log_2 n \rfloor$.

2-tree

A **2-tree** is a tree where every node that is not a leaf has exactly two children.



An **internal** node is a node that is not a leaf node.

Let $I(T)$ and $L(T)$ denote the number of internal and leaf nodes of a 2-tree T, respectively. For convenience, let $I = I(T)$ and $L = L(T)$.

Proposition 4.2.3 Let T be a 2-tree. Then, $L = I + 1$.

Clearly, the total number n of nodes satisfies $n = I + L$, so we have:

Corollary 1. $n = 2I + 1$.

Corollary 2. $n = 2L – 1$.

Parametrizing the induction. We must decide, which parameter, we will perform induction on, i.e., the number I of internal nodes, the number L of leaf nodes or the total number n of nodes. We will choose L.

Basis Step

The proposition is true for $L = 1$. A single node

2-tree has one leaf node, the root, and 0 internal nodes, so we have

$$L = 1 = I + 1.$$

Induction Step

Assume proposition is true for $L = k$, i.e., all 2-trees T with k leaf nodes have $k - 1$ internal nodes.

Now consider **any** 2-tree T having $k + 1$ leaf nodes.

PSN. To apply the induction hypothesis, we need to perform an operation that reduces T to a tree T' with k leaf nodes.

How to do this?

PSN. To verify that this construction is valid, we must prove that every 2-tree *T* contains a node, both of whose children are leaf nodes. Proof this result.

Since T' has one fewer leaf nodes than T, i.e., T' has k leaf nodes, we can apply the induction hypothesis, i.e.,

$$L(T') = I(T') + 1.$$

Thus,

$$L(T) = L(T') + 1 = (I(T') + 1) + 1 = I(T) + 1$$

This completes the induction step and the proof of the Proposition.

Lower bound on depth of a binary tree in terms of number of leaf nodes

Proposition 4.2.6 Every binary tree with *L* leaf nodes has depth *d* at least

$$\lceil \log_2 L \rceil$$

Further, equality is achieved for the complete binary tree.

Proof

First assume the binary tree T is a 2-tree. By Proposition 4.2.2

$$d \geq \lfloor \log_2 n \rfloor \qquad (1)$$

And by Corollary 2 of Proposition 4.2.3

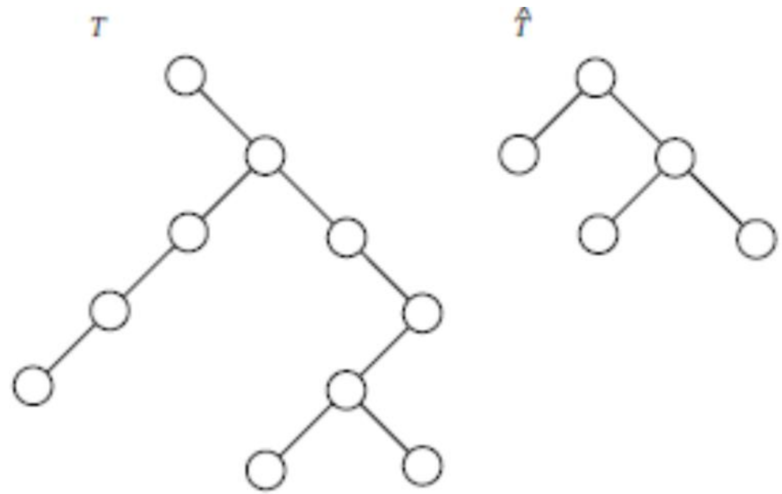
$$n = 2L - 1 \qquad (2)$$

Substituting (1) into (2) we obtain

$$d \geq \lfloor \log_2 n \rfloor = \lfloor \log_2 (2L - 1) \rfloor = \lceil \log_2 L \rceil$$

Transformation yield result for general binary tree

Now consider a general binary tree T. Construct $\hat{T}$ from T as follows



By argument on previous slide $d(\hat{T}) \geq \lfloor \log_2 L(\hat{T}) \rfloor$

Thus,

$$d(T) \geq d(\hat{T}) \geq \lfloor \log_2 L(\hat{T}) \rfloor = \lfloor \log_2 L(T) \rfloor$$

Why did the pine tree get in trouble?

Because it was being knotty.