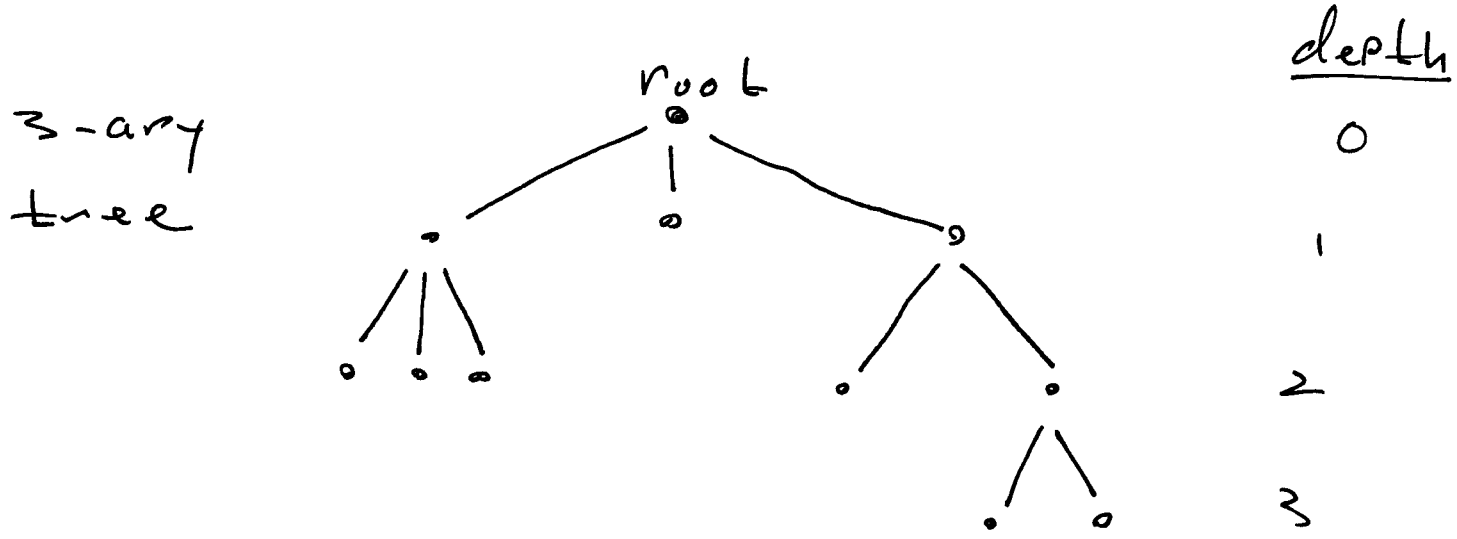
ESE 201 3-4-21

Rooted tree:



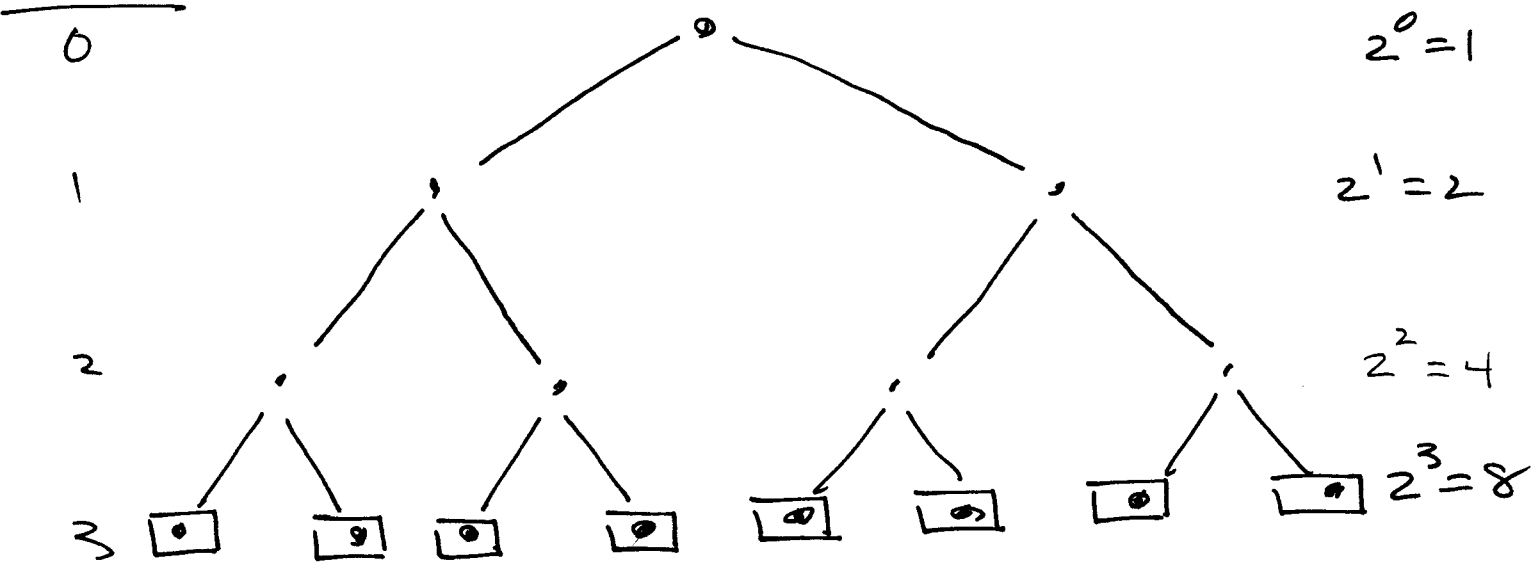
depth = dist. from root

(height of tree) = depth of deepest leaf.

Defn: a k-ary tree is a rooted tree in which each internal (non-leafs node has at most k children.

Case $K = 2$

complete binary tree :



(# nodes at depth d) $= 2^d$

if height $= h$,

(# leaves) $= 2^h$

let n = # leaves : $n = 2^h$

also $h = \lg(n)$.

Theorem

Let T be a binary tree with n leaves and height h. Then

$$h \geq \lceil \lg n \rceil$$

Proof.

Let $L(T)$, $H(T)$ be # leaves & height of a b.t. T. Use (weak) induction on $h = H(T)$.

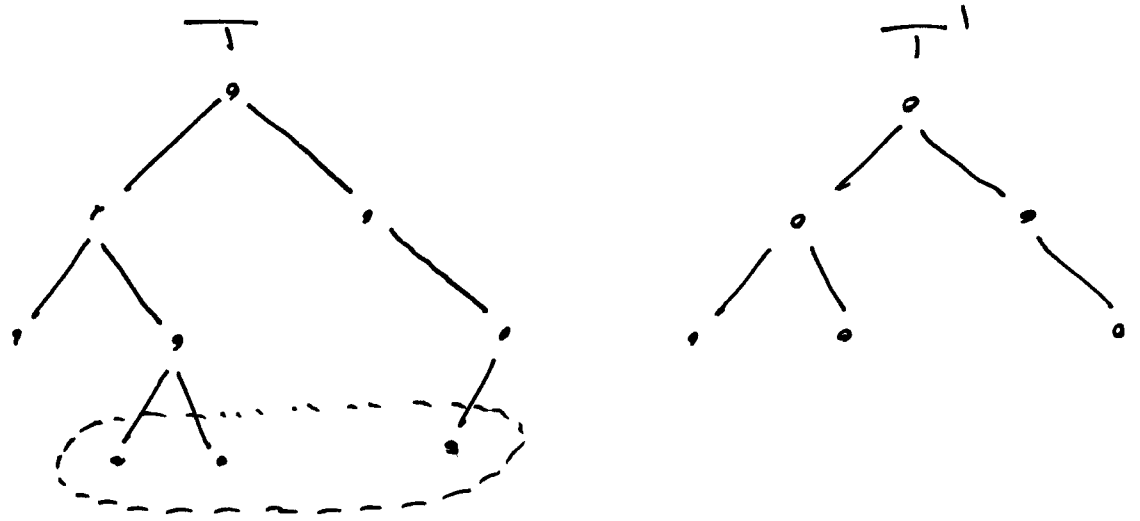
I. if $h=0$, then T has one node (root), which is a leaf. $n = L(T) = 1$. so $h \geq \lceil \lg n \rceil$ becomes $0 \geq 0$, which true.

II b. Let $h > 0$, and assume for any binary tree T' with $H(T') = h-1$ that

$$H(T') \geq \lceil \lg(L(T')) \rceil .$$

we must show $H(T) \geq \lceil \lg(L(T)) \rceil$, i.e. $h \geq \lceil \lg(n) \rceil$.

Let T' be obtained from T by deleting all leaves at depth h



note: $H(T') = h-1$.

Also note: $L(T) \leq 2L(T')$ since each node in T has at most 2 children. Thus $L(T') \geq \frac{1}{2}L(T)$.

we have

$$h-1 = H(T')$$

$$\geq \lceil \lg(L(T')) \rceil \quad \{ \text{by the ind. hyp.}$$

$$\geq \lceil \lg\left(\frac{L(T)}{2}\right) \rceil$$

$$= \lceil \lg(L(T)) - 1 \rceil$$

$$= \lceil \lg(L(T)) \rceil - 1$$

$$= \lceil \lg(n) \rceil - 1$$

$$\therefore \quad h-1 \geq \lceil \lg(n) \rceil - 1$$

$$\therefore \quad h \geq \lceil \lg(n) \rceil ,$$

as required. ∎

Exercise

Let T be a K-ary tree with n leaves and height h. Prove that $h \geq \lceil \log_K(n) \rceil$.

We reason as follows:

given a Problem P, consider all algorithms that solve P by doing a seq. basic operations, each having at most K outcomes. (call these ops. Probes or K-ary Probes.)

Any algorithm of this type can be represented by a K-ary tree. each internal node is a K-ary Probe, with at most K children. each leaf is a Verdict (return value).

Let n be size of an instance of P. Let $f(n)$ be # of possible outputs (verdicts).

Note: $f(n) \leq L(T)$

where T represents the k-ary tree representing an algorithm.

Each downward path from root to a desc. leaf, represents a particular logical path from input to output (verdict).

By preceding exercise

$$h \geq \lceil \log_k (L(T)) \rceil \geq \lceil \log_k f(n) \rceil$$

Thus

$$(\text{worst case } \# \text{ Probes}) \geq \lceil \log_k(f(n)) \rceil$$

Ex. guess $m \in \{1, \ldots, 6\}$

$n = |\{1, \ldots, 6\}| = 6$

$f(n) = 6$. Thus the worst case # of comparisons satisfies

$$(\#\text{comp}) \geq \lceil \lg(6) \rceil = 3$$

Ex guess $m \in \{1, \ldots, n\}$

$f(n) = n$, $k = 2$

$$\#\text{Questions} \geq \lceil \lg n \rceil .$$

Say $n = 1\,000\,000$

$$\#\text{questions} \geq \lceil \lg 10^6 \rceil = 20$$

Theorem

Any sorting algorithm that performs comparisons as its basic operation must do in worst case at least $\lceil \lg(n!) \rceil$ comparisons on input arrays of length n.

Proof.

A 'verdict' for input array $A[1 \ldots n]$ is a permutation of elements $\{A_1, \ldots, A_n\}$ $\therefore f(n) = n!$

Each comparison $A_i \leq A_j$ ($A_i < A_j$, $\cdot$, $\cdot$) has exactly 2 outcomes (true, false).

$\therefore k = 2$. Thus

$$|\#\text{comp}| \geq \lceil \lg(n!) \rceil = \Omega(n \log n)$$

asymptotically: $\#\text{comp} \geq \Omega(n \log n)$.

Defn

The average height of a k-ary tree is the average leaf depth.

leaf depths: $\{d_1, d_2, \ldots, d_n\}$

$$(\text{avg height}) = \frac{\sum_{i=1}^{n} d_i}{n}$$

Thm

The avg. height a of a k-ary tree with n leaves satisfies

$$a \geq \log_k(n) .$$

(see p. 416 of Brassard & Bratley for $k=2$ case.)

Thm

for a Problem P, any algorithm doing only k-ary Probes must do at least $\log_k(f(n))$ such Probes in avg. case. Here n is Prob. size, $f(n)$ is # of Verdicts.

Summary: Decision tree argument

(1) Determine k, max # of outcomes to each Probe of the data.

(2) Determine $f(n)$, the number of Verdicts (outputs) as a fcn. of input size.

(3) Conclude any algorithm for P that does only Probes in (1) does at least $\lceil \log_k(f(n)) \rceil$ Probes (worst case) & at least $\log_k(f(n))$ " (avg. case).

Adversary arguments

Ex. Same guessing game

find $x \in \{1, 2, \ldots, n\}$. Two

Players A, R.

A: chooses $x \in \{1, \ldots, n\}$

R: asks $Q_1, Q_2, \ldots,$ a seq.

of questions.

A cheats!

A's goal is to prolong the game

as far as possible. Each answer

must be consistent with preceding

answers.