

Binary tree T .

PROOF.

Let T be a binary tree with $h = H(T)$ and $n = L(T)$. We show by induction on h that $h \geq \lceil \lg n \rceil$.

I. If $h = 0$ then T contains just one node, and $n = 0$. Thus in this case $h \geq \lceil \lg n \rceil$.

II. Let $h > 0$ and assume the result holds for any binary tree of height $h-1$. Let T' be the binary tree obtained by deleting all leaves at depth h from T (along with all incident edges.)

Observe that $H(T') = h-1$, and by the induction hypothesis,

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

Since each node in T can have at most 2 children, we also have $L(T) \leq 2 L(T')$, whence

$$L(T') \geq \frac{L(T)}{2}$$

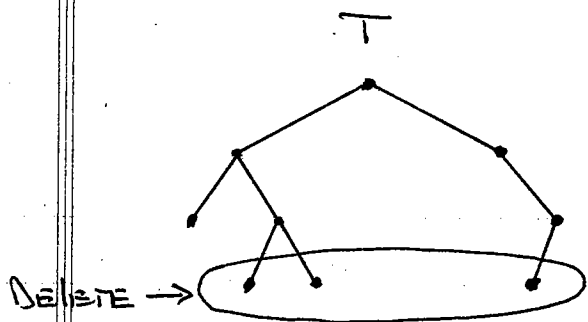
POTTING THESE INEQUALITIES TOGETHER GIVES

$$\begin{aligned}
h-1 &= H(T') \\
&\geq \lceil \lg L(T') \rceil \\
&\geq \lceil \lg \frac{L(T)}{2} \rceil \\
&= \lceil \lg \left(\frac{n}{2}\right) \rceil \\
&= \lceil \lg n - 1 \rceil \\
&= \lceil \lg n \rceil - 1,
\end{aligned}$$

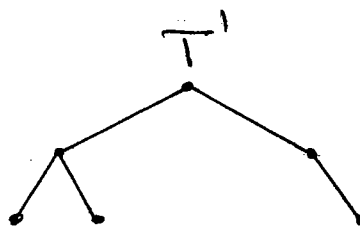
AND THEREFORE $h \geq \lceil \lg n \rceil$ AS REQUIRED.

///

ILLUSTRATION:



$$\begin{aligned}
h = H(T) &= 3 \\
n = L(T) &= 4
\end{aligned}$$



$$\begin{aligned}
H(T') &= 2 \\
L(T') &= 3
\end{aligned}$$

NOTE ONE CAN DEFINE IN A SIMILAR MANNER COMPLETE AND ALMOST COMPLETE K-ARY TREES, AND PROVE FORMULAS $h = \log_k n$ AND $h = \lceil \log_k n \rceil$ RESPECTIVELY.

EXERCISE: CARRY OUT THE ABOVE DEFINITIONS AND PROOFS.

THEOREM:

THE HEIGHT h OF ANY K-ARY TREE WITH n LEAVES SATISFIES

$$h \geq \lceil \log_k n \rceil$$

EXERCISE: PROVE THIS BY INDUCTION ON THE HEIGHT h ,

A DECISION TREE IS A WAY TO REPRESENT THE WORKINGS OF AN ALGORITHM ON ALL POSSIBLE INPUTS OF A GIVEN SIZE, USING A K-ARY TREE.

EACH INTERNAL NODE REPRESENTS AN OPERATION OR TEST OF SOME KIND ON THE INPUT DATA. EACH LEAF REPRESENTS AN OUTPUT OR VERDICT.

EACH DOWNWARD PATH FROM THE ROOT TO A LEAF REPRESENTS A PARTICULAR SEQUENCE OF TESTS LEADING TO A CONCLUSION, i.e. A PARTICULAR LOGICAL PATHWAY TAKEN BY THE ALGORITHM.

K = MAXIMUM NUMBER OF POSSIBLE OUTCOMES TO EACH TEST

n = NUMBER OF LEAVES
= NUMBER OF POSSIBLE VERDICTS

h = HEIGHT
= MAXIMUM NUMBER OF TESTS NECESSARY TO REACH A VERDICT.

EX

WE RETURN TO THE EXAMPLE OF FINDING $m \in \{1, \dots, 6\}$ BY ASKING ONLY YES/NO QUESTIONS. IS THERE AN ALGORITHM WHICH DETERMINES (ANY) m USING AT MOST 2 QUESTIONS?

VERDICTS = $n = 6$

OUTCOMES ON EACH TEST = $K = 2$

MAX # OF QUESTIONS = h

By THE PREVIOUS THEOREM $h \geq \lceil \lg 6 \rceil = 3$.

OBSERVE THAT THE OPERATION OF ANY ALGORITHM WHICH SOLVES THIS PROBLEM CAN BE REPRESENTED BY A DECISION TREE WITH $k=2$ AND $n=6$ LEAVES.

SINCE $h \geq 3$, WE KNOW 3 IS A LOWER BOUND ON THE WORST CASE NUMBER OF QUESTIONS NECESSARY IN ANY ALGORITHM WHICH SOLVES THIS PROBLEM.

\therefore 2 QUESTIONS DO NOT SUFFICE.

EX

GIVEN $1 \leq m \leq 10^6$, FIND A LOWER BOUND ON THE NUMBER OF YES/NO QUESTIONS NECESSARY TO DETERMINE m .

ANS. $k=2$, $n=10^6$, $h \geq \lceil \lg(10^6) \rceil = 20$.

\therefore THIS PROBLEM CANNOT BE SOLVED (IN GENERAL) WITH 19 OR FEWER QUESTIONS.

NOTE! THIS DOES NOT SHOW THAT 20 QUESTIONS WILL SUFFICE, ONLY THAT 19 WILL NOT.

EXERCISE

SHOW THAT BINARY SEARCH (WITH TARGET = m) SOLVES THIS PROBLEM IN NO MORE THAN 20 QUESTIONS (I.E. COMPARISONS.)

EX

SAME PROBLEM, BUT NOW WE ARE ALLOWED TO ASK QUESTIONS WITH 3 ALTERNATIVES.

ANS. $k=3$, $n=10^6$, $h \geq \lceil \log_3(10^6) \rceil = 13$.

THUS ANY ALGORITHM WHICH SOLVES THIS PROBLEM MUST ASK AT LEAST 13 QUESTIONS IN WORST CASE.

AGAIN, THIS DOES NOT SHOW THAT 13 QUESTIONS WILL SUFFICE, ONLY THAT 12 WILL NOT.

THEOREM

ANY COMPARISON BASED SORTING ALGORITHM MUST DO, IN WORST CASE, AT LEAST $\lceil \lg(n!) \rceil$ COMPARISONS ON INPUT ARRAYS OF LENGTH n .

PROOF:

THE DECISION TREE CORRESPONDING TO ANY SUCH ALGORITHM HAS $k=2$ OUTCOMES FOR EACH COMPARISON AND $n!$ POSSIBLE VERDICTS (SINCE EACH OUTPUT IS A PERMUTATION OF THE INPUT ARRAY.) THE HEIGHT OF SUCH A TREE MUST SATISFY

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

THEREFORE, ANY ALGORITHM WHICH SOLVES THIS PROBLEM MUST DO AT LEAST $\lceil \lg n! \rceil$ COMPARISONS, IN WORST CASE.

///

Corollary

ANY SUCH ALGORITHM RUNS IN (WORST CASE) TIME $\Omega(n \lg n)$.

PROOF:

By STIRLING'S FORMULA $\lceil \lg n! \rceil = \Omega(n \lg n)$.

///

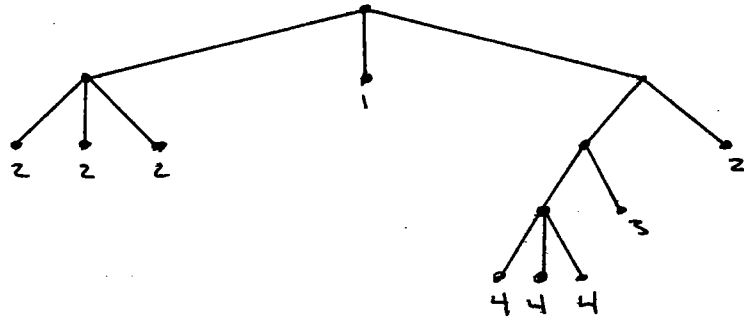
DEFN

THE AVERAGE HEIGHT OF A k -ARY TREE IS THE AVERAGE DEPTH OF EACH OF ITS LEAVES.

i.e. if T has n leaves at depths d_1, \dots, d_n , then the average height of T is

$$a = \frac{\sum_{i=1}^n d_i}{n}$$

Ex. $k=3, n=9$



$h=4$
 $a=2.67$

$$a = \frac{2+2+2+1+2+3+4+4+4}{9} = \frac{24}{9} = \frac{8}{3} = 2.67$$

Just as the height of a decision tree gives the worst case number of tests performed by an algorithm, the average height gives the average number of tests performed, assuming that each verdict (i.e. leaf) is equally likely.