THEOREM

THE AVERAGE HEIGHT a OF A k-ARY TREE HAVING n LEAVES SATISFIES

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

COROLLARY

THE AVERAGE HEIGHT a OF A BINARY TREE HAVING n LEAVES SATISFIES

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

SEE P. 416 OF BRASSARD & BRATLEY FOR PROOF OF COROLLARY ($k=2$).

THEOREM.

ANY COMPARISON SORT MUST, IN AVERAGE CASE, AT LEAST $\lg(n!)$ COMPARISONS ON INPUT ARRAYS OF LENGTH n.

COROLLARY

ANY COMPARISON BASED SORTING ALGORITHM RUNS IN (AVERAGE CASE) TIME $\Omega(n \lg n)$.

DECISION TREE ARGUMENTS FOR LOWER BOUNDS ARE VERY EASY TO USE:

(1) DETERMINE THE MAXIMUM NUMBER OF OUTCOMES TO EACH TEST OF THE INPUT DATA, CALL THAT NUMBER K.

(2) DETERMINE THE NUMBER OF VERDICTS $f(n)$ (i.e. POSSIBLE ALGORITHM OUTPUTS) AS A FUNCTION OF THE INPUT SIZE n.
e.g. SEARCHING: $f(n) = n$
SORTING: $f(n) = n!$

(3) CONCLUDE THAT ANY ALGORITHM WHICH SOLVES THE PROBLEM USING ONLY TESTS OF THE KIND ANALYSED IN (1) MUST DO AT LEAST

$$\text{WORST CASE: } \lceil \log_K f(n) \rceil$$
$$\text{AVERAGE CASE: } \log_K f(n)$$

TESTS ON INPUT OF SIZE n.

NOTE: THESE ARGUMENTS DO NOT ASSERT THE EXISTENCE OF ALGORITHMS WHICH PERFORM THE NUMBER OF TESTS CITED ABOVE. INSTEAD THEY ASSERT THE NON-EXISTENCE OF ALGORITHMS WHICH DO FEWER TESTS.

ADVERSARY ARGUMENTS

CONSIDER AGAIN THE GAME OF 20 QUESTIONS, EXCEPT NOW YOUR OPPONENT CHEATS.

CALL THE PLAYERS A AND B.

A: PRETENDS TO PICK $x \in \{1, \ldots, 10^6\}$

B: ASKS A SEQUENCE $Q_1, Q_2, \ldots$ OF YES/NO QUESTIONS (EACH QUESTION DEPENDS ON PRECEEDING ANSWERS.)

A: ALWAYS GIVES AN ANSWER WHICH IS CONSISTENT WITH ALL PREVIOUS ANSWERS, BUT WHICH IS DESIGNED TO PROLONG THE GAME AS FAR AS POSSIBLE.

FOR A'S ANSWERS TO BE CONSISTENT, THERE MUST ALWAYS EXIST AN $x \in \{1, \ldots, 10^6\}$ SUCH THAT IF x HAD BEEN PICKED (BY AN HONEST PLAYER), THE TRUE ANSWERS WOULD HAVE BEEN THOSE GIVEN BY A.

HOW LONG CAN A KEEP THIS UP? THE ANSWER TO THIS QUESTION PROVIDES A LOWER BOUND ON THE WORST CASE NUMBER OF QUESTIONS WHICH MUST BE ASKED BY ANY ALGORITHM PLAYING THE PART OF B.

WE MUST SPECIFY B'S STRATEGY FOR ANSWERING QUESTIONS.

LET S_i DENOTE THE REMAINING SET OF CANDIDATES FOR THE MYSTERY NUMBER x AFTER THE i^{TH} QUESTION HAS BEEN ASKED (AND ANSWERED.)

e.g. $S_0 = \{1, \ldots, 10^6\}$

LET $Q_i = i^{TH}$ QUESTION, AND LET $A_i(x)$ DENOTE THE (TRUE) ANSWER TO Q_i IF THE MYSTERY NUMBER IS x.

e.g. $Q_1 =$ " is $x \leq 500{,}000$ "
$A_1(400{,}000) =$ 'YES'
$A_1(600{,}000) =$ 'NO'

LET $Y_i = \{x \in S_{i-1} \mid A_i(x) = \text{'YES'}\}$
$N_i = \{x \in S_{i-1} \mid A_i(x) = \text{'NO'}\}$

e.g. $Q_1 =$ " is $x \leq 500{,}000$ "
$Y_1 = \{1, \ldots, 500000\}$
$N_1 = \{500001, \ldots, 1000000\}$

NOTE $Y_i \cap N_i = \phi$ AND $Y_i \cup N_i = S_{i-1}$. THUS

$$|Y_i| + |N_i| = |S_{i-1}|$$

BY THE PIGEONHOLE PRINCIPLE, AT LEAST ONE OF Y_i OR N_i MUST CONTAIN AT LEAST

$$\left\lceil \frac{|S_{i-1}|}{2} \right\rceil$$

NUMBERS.

A'S STRATEGY IS TO ALWAYS ANSWER Q_i IN A WAY WHICH IMPLIES x IS IN THE LARGER OF THE TWO SETS Y_i OR N_i.

THUS

$$S_i = \begin{cases} Y_i & |Y_i| \geq |N_i| \\ N_i & |Y_i| < |N_i| \end{cases}.$$

THEREFORE

$$|S_i| \geq \left\lceil \frac{|S_{i-1}|}{2} \right\rceil.$$

LET $b_i = |S_i|$ SO THAT $b_i \geq \lceil b_{i-1}/2 \rceil$ FOR $i > 0$. THEN

$$\begin{aligned} b_0 &= 10^6 \\ b_1 &\geq \lceil 10^6/2 \rceil \\ b_2 &\geq \lceil 10^6/2^2 \rceil \\ &\vdots \\ b_{19} &\geq \lceil 10^6/2^{19} \rceil = 2 \\ b_{20} &\geq \lceil 10^6/2^{20} \rceil = 1 \end{aligned}$$

THUS IF B CLAIMS TO KNOW x AFTER NO MORE THAN 19 QUESTIONS, THEN A CAN CLAIM AT LEAST ONE NUMBER OTHER THAN x (WHICH IS CONSISTENT WITH ALL HIS PREVIOUS ANSWERS) AS HIS TRUE PICK.

THEREFORE ANY ALGORITHM WHICH PLAYS THE PART OF B MUST ASK AT LEAST 20 QUESTIONS, IN WORST CASE.

AN ADVERSARY ARGUMENT FOR A LOWER BOUND PROCESS AS FOLLOWS.

- START THE ALGORITHM ON AN INPUT WHICH IS UNSPECIFIED, EXCEPT AS TO SIZE.
- WHENEVER THE ALGORITHM PROBES THE INPUT DATA A MALEVOLENT ADVERSARY (ALSO CALLED A DAEMON) ANSWERS IN A WAY WHICH MAKES THE ALGORITHM WORK HARD. HE IS CONSISTENT IN THAT THERE MUST ALWAYS EXIST AN INPUT (OF THE GIVEN SIZE) WHICH WOULD ELICIT THE DAEMON'S SEQUENCE OF ANSWERS.

- THE DAEMON'S STRATEGY FOR PROLONGING THE RUN TIME MUST BE SPECIFIED EXACTLY.

- PROVE THAT THERE IS A NUMBER M (DEPENDING ON INPUT SIZE) SUCH THAT IF THE ALGORITHM USED TO PRODUCE AN OUTPUT AFTER FEWER THAN M PROBES, THEN THERE EXISTS AT LEAST ONE INPUT WHICH IS CONSISTENT WITH ALL THE DAEMON'S ANSWERS, BUT WHOSE CORRECT SOLUTION IS DIFFERENT FROM THE ALGORITHM OUTPUT.

- CONCLUDE THAT M IS A LOWER BOUND ON THE WORST CASE NUMBER OF PROBES WHICH MUST BE PERFORMED BY ANY ALGORITHM WHICH SOLVES THE PROBLEM.

JUST AS WITH ANY LOWER BOUNDS ARGUMENT, THE ABOVE PROGRAM DOES NOT PROVE THE EXISTENCE OF AN ALGORITHM WHICH CAN SOLVE THE PROBLEM USING AT MOST M PROBES. INSTEAD IT PROVES THE NON-EXISTENCE OF AN ALGORITHM WHICH SOLVES THE PROBLEM USING AT MOST M-1 PROBES.

ADVERSARY ARGUMENTS ARE MORE COMPLICATED THAN DECISION TREE ARGUMENTS. WHY DO WE USE THEM?

OFTEN THE LOWER BOUND OBTAINED BY A DECISION TREE IS FAR FROM TIGHT.

PROBLEM

GIVEN AN ARRAY $A[1 \ldots n]$ OF NUMBERS, FIND ITS MAXIMUM ENTRY AND THE INDEX WHERE IT IS LOCATED.

WE CONSIDER ONLY ALGORITHMS WHICH ARE RESTRICTED TO PERFORMING COMPARISONS OF ARRAY ELEMENTS. (i.e. NO ARITHMETIC OPERATIONS ON ELEMENTS.) THIS IS ANALOGOUS TO THE CLASS OF COMPARISON SORTS.

DECISION TREE LOWER BOUND:

$$\#\text{OUTCOMES PER TEST} = k = 2$$
$$\#\text{VERDICTS} = n$$
$$\therefore\ h \geq \lceil \lg n \rceil$$

THUS ANY ALGORITHM DOES AT LEAST $\lceil \lg n \rceil$ COMPARISONS (IN WORST CASE) ON INPUT OF SIZE n.