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 PROCEEDS 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.

AS USUAL THIS STATEMENT DOES NOT ASSERT THAT THE PROBLEM CAN BE SOLVED WITH ONLY $\lceil \lg n \rceil$ COMPARISONS, ONLY THAT $\lceil \lg n \rceil$ ARE NECESSARY.

IN FACT, THE BEST KNOWN ALGORITHM DOES $n-1$ COMPARISONS, MUCH WORSE THAN $\lceil \lg n \rceil$.

FindMax (A)

```
1.) n ← length(A)
2.) max ← A[1], imax ← 1
3.) for   ← 2 TO n
4.)       if A[i] > max
5.)             max ← A[i]
6.)             imax ← i
7.) return (max, imax)
```

EX. $n = 4$

DECISION TREE : $\#\text{COMP} \geq \lceil \lg 4 \rceil = 2$

BEST KNOWN : $\#\text{COMP} = 3$

EXERCISE

DRAW A DECISION TREE FOR THE OPERATION OF FindMax IN THIS CASE. OBSERVE THAT ITS HEIGHT IS 3 NOT 2.

EX $n = 1001$

DECISION TREE : # COMP $\geq \lceil \lg 1001 \rceil = 10$

BEST KNOWN : # COMP $= 1000$

WE MUST EITHER FIND A BETTER ALGORITHM (NOT POSSIBLE) OR OBTAIN A TIGHTER LOWER BOUND

ADVERSARY ARGUMENT :

CONSIDER ANY COMPARISON BASED ALGORITHM FOR THIS PROBLEM AND LET IT RUN ON AN ARRAY $A[1..n]$, AS YET UNSPECIFIED.

DAEMON'S STRATEGY:
ANSWER EACH QUESTION CONCERNING A COMPARISON AS IF $A[i] = i$ $(1 \leq i \leq n)$. i.e. AS IF $A = (1, 2, \ldots, n)$. IN OTHER WORDS, WHEN THE ALGORITHM ASKS "IS $A[i] < A[j]$", THE DAEMON ANSWERS

$$\begin{cases} \text{TRUE} & \text{IF} \quad i < j \quad (\text{"}i \text{ HAS LOST A COMPARISON"}) \\ \text{FALSE} & \text{IF} \quad j < i \quad (\text{"}j \text{ HAS LOST A COMPARISON"}) \end{cases}$$

WHEN THIS HAPPENS WE SAY THAT THE SMALLER OF i AND j HAS "LOST A COMPARISON".

NOW ASSUME THAT THE ALGORITHM DOES FEWER THAN $M = n-1$ COMPARISONS BEFORE IT HALTS AND OUTPUTS THE INDEX k (OR THE PAIR $(A[k], k)$). I.E. THE ALGORITHM CLAIMS THAT $A[k]$ IS MAXIMUM IN ARRAY A.

LET j BE AN INTEGER SATISFYING

- $1 \le j \le n$
- $j \neq k$
- j HAS NOT LOST ANY COMPARISONS.

SUCH AN INTEGER MUST EXIST SINCE BY ASSUMPTION AT MOST $n-2$ COMPARISONS HAVE BEEN PERFORMED, AND EACH NEW COMPARISON CREATES AT MOST ONE NEW LOSER, HENCE THERE ARE AT MOST $n-2$ LOSERS.

AT THIS POINT THE DAEMON CAN SAY THE ALGORITHM IS WRONG BY CLAIMING THAT ARRAY A IS GIVEN BY

$$A[i] = \begin{cases} i & i \neq j \\ n+1 & i = j \end{cases}$$

INDEED $A[k] = k$ IS NOT MAXIMUM IN

THIS ARRAY, YET THE DAEMONS ANSWERS ARE ALL CONSISTENT WITH IT.

THEREFORE NO CORRECT ALGORITHM CAN SOLVE THIS PROBLEM WITH FEWER THAN $M = n-1$ COMPARISONS, AND OUR BEST KNOWN ALGORITHM CANNOT BE IMPROVED UPON.

///.

GRAPH CONNECTIVITY

LET $G = (V, E)$ BE AN (UNDIRECTED) GRAPH ON $|V| = n \geq 2$ VERTICES

PROBLEM: DETERMINE WHETHER OR NOT G IS CONNECTED.

WE CONSIDER ONLY ALGORITHMS WHICH ARE ALLOWED TO ASK QUESTIONS OF THE FORM "IS VERTEX u ADJACENT TO VERTEX v?"

THE DECISION TREE LOWER BOUND IS TRIVIAL:

OUTCOMES PER QUESTION = 2 (YES/NO)
VERDICTS = 2 (CONNECTED / DISCONNECTED.)
$\therefore h \geq \lceil \lg 2 \rceil = 1$
$\therefore$ AT LEAST 1 QUESTION IS NECESSARY.

DEPTH FIRST SEARCH (DFS) SOLVES THIS PROBLEM IN TIME $\Omega(n^2)$. (SEE SECTION 22.3 FOR A DESCRIPTION.)

ADVERSARY LOWER BOUND:

CONSIDER ANY "ADJACENCY" BASED ALGORITHM FOR THIS PROBLEM AND START IT ON AN (UNSPECIFIED) GRAPH $G=(V,E)$ WITH $n=|V|$.

DAEMON'S STRATEGY:
PARTITION V INTO TWO SUBSETS X AND Y OF SIZES $\lfloor n/2 \rfloor$ AND $\lceil n/2 \rceil$ RESPECTIVELY. I.E.

$$X \cup Y = V,\ X \cap Y = \phi,\ |X| = \left\lfloor \tfrac{n}{2} \right\rfloor,\ |Y| = \left\lceil \tfrac{n}{2} \right\rceil.$$

WHENEVER THE ALGORITHM ASKS "is u ADJACENT TO v?" THE DAEMON ANSWERS YES IFF u AND v BELONG TO THE SAME SUBSET. I.E.

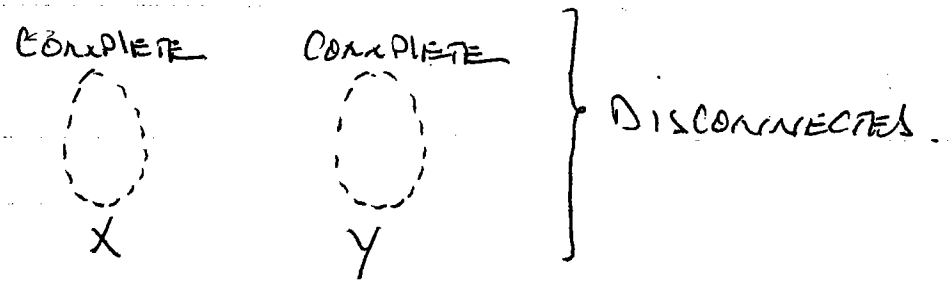
$$\begin{cases} \text{YES} & \text{IF } u,v \in X \text{ OR } u,v \in Y \\ \text{NO} & \text{IF } u \in X, v \in Y \text{ OR } u \in Y, v \in X. \end{cases}$$

IN OTHER WORDS, THE DAEMON ANSWERS AS IF G CONSISTS OF THE DISJOINT UNION OF TWO COMPLETE GRAPHS ON X AND Y RESPECTIVELY.

(A GRAPH IS CALLED <u>COMPLETE</u> IF EACH PAIR OF DISTINCT VERTICES ARE JOINED BY EXACTLY ONE EDGE.)

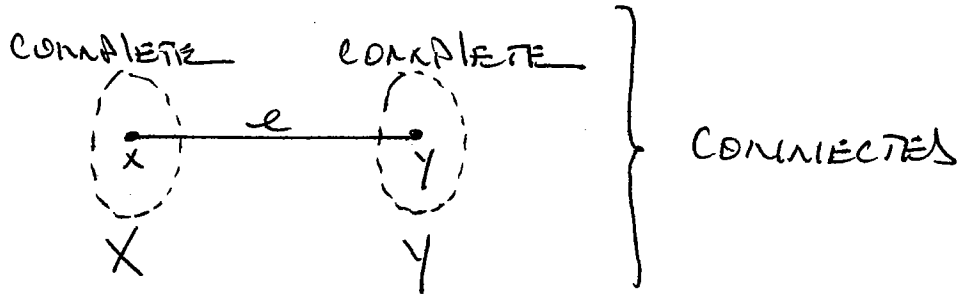
NOW SUPPOSE THE ALGORITHM HALTS AND RETURNS AN ANSWER (CONNECTED OR DISCONNECTED) AFTER ASKING FEWER THAN $M = \lfloor N/2 \rfloor \cdot \lceil N/2 \rceil$ QUESTIONS. THERE MUST THEN EXIST A PAIR $x \in X$ AND $y \in Y$ ABOUT WHICH THE ALGORITHM HAS NOT INQUIRED.

IF THE ALGORITHM SAYS G IS CONNECTED, THE DAEMON CAN CLAIM THAT G CONSISTS OF TWO DISJOINT COMPLETE GRAPHS ON X AND Y.



THIS GRAPH IS DISCONNECTED AND IS CLEARLY CONSISTENT WITH THE DAEMONS SEQUENCE OF ANSWERS.

ON THE OTHER HAND, IF THE ALGORITHM SAYS G is DISCONNECTED, THE DAEMON CAN CLAIM THAT G ACTUALLY CONSISTS OF TWO COMPLETE GRAPHS ON X AND Y, WITH A SINGLE EDGE $e = xy$ ADDED.



THIS GRAPH is CONNECTED AND is CONSISTENT WITH ALL THE DAEMON'S ANSWERS, SINCE THE EDGE $e = xy$ WAS NOT PROBED BY THE ALGORITHM.

IN EITHER CASE THE DAEMON CAN CLAIM THE ALGORITHM is WRONG. THUS ANY ALGORITHM WHICH DOES NOT ASK AT LEAST $\lfloor n/2 \rfloor \lceil n/2 \rceil = \Omega(n^2)$ QUESTIONS (IN WORST CASE) CANNOT BE CORRECT.

///

THUS DFS CANNOT BE IMPROVED UPON, EXCEPT PERHAPS FOR IMPROVEMENTS IN HIDDEN CONSTANTS.

NOTE: A COMPLETE GRAPH ON n VERTICES HAS $\binom{n}{2} = \frac{n(n-1)}{2}$ EDGES, SINCE EACH EDGE CORRESPONDS TO A UNIQUE 2-ELEMENT SUBSET OF $V(G)$.

EXERCISE:
GIVE A MORE SOPHISTICATED ADVERSARY ARGUMENT SHOWING THAT ANY "ADJACENCY BASED" ALGORITHM TO DETERMINE CONNECTEDNESS MUST ASK AT LEAST $\binom{n}{2}$ QUESTIONS (IN WORST CASE.) NOTE: $\binom{n}{2} = \lfloor n/2 \rfloor \cdot \lceil n/2 \rceil$

IN OTHER WORDS, A CORRECT ALGORITHM MUST INQUIRE ABOUT EACH OF THE $\binom{n}{2}$ POTENTIAL EDGES.

EXERCISE
USE AN ADVERSARY ARGUMENT TO SHOW THAT $\binom{n}{2}$ "ADJACENCY" QUESTIONS ARE NECESSARY (IN WORST CASE) TO DETERMINE IF A GRAPH G IS ACYCLIC.

EXERCISE
GIVE AN ADVERSARY ARGUMENT SHOWING THAT A COMPARISON SORT MUST DO AT LEAST $\lceil \lg n! \rceil$ COMPARISONS IN WORST CASE, ON INPUT OF SIZE n.

(NOTE THIS IS ESSENTIALLY NO DIFFERENT FROM THE ADVERSARY LOWER BOUND FOR 20 QUESTIONS SINCE SORTING n ELEMENTS is REALLY A SEARCH OF $n!$ PERMUTATIONS. THE DAEMON MUST ANSWER IN A WAY WHICH KEEPS THE POOL OF CANDIDATE PERMUTATIONS AS LARGE AS POSSIBLE.)

PROBLEM

LET $b = b_1 b_2 b_3 b_4 b_5$ BE A BIT STRING OF LENGTH 5. DETERMINE WHETHER OR NOT b CONTAINS THE SUBSTRING 111 (i.e. 3 CONSECUTIVE 1's).

CONSIDER ALGORITHMS WHOSE ONLY ALLOWABLE OPERATION is TO PEEK AT A BIT.

OBVIOUSLY 5 PEEKS ARE SUFFICIENT. A DECISION TREE ARGUMENT PROVES THE (USELESS) FACT THAT AT LEAST 1 PEAK is NECESSARY

EXERCISE.

a.) USE AN ADVERSARY ARGUMENT TO SHOW THAT 4 PEEKS ARE NECESSARY IN GENERAL.

b.) DESIGN AN ALGORITHM WHICH SOLVES THE PROBLEM IN ONLY 4 PEEKS.