CMPS 201 5-17-10

Thm.

The height h of any k-ary tree with n leaves satisfies

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

Thm

Given a Problem P consider all algorithms that solve P by performing a seq. of basic ops. (i.e. tests, probes) each of which results in one of (at most) k possible outcomes.

let n denote the size of an instance of P, and f(n) denote the # of possible verdicts (i.e. # outputs) for such an instance. Then the # of basic operations performed by any algorithm that solves P is bounded below by

$$\lceil \log_k (f(n)) \rceil$$

<u>Proof:</u>
any such algorithm can be represented by a k-ary tree where each internal

Probe 3

node represents a test or probe of the input data, and each leaf represents a verdict or algorithm output. So

$$\# \text{leaves} \geq \# \text{verdicts}$$

Each descending path from root to leaf represents a distinct logical path that the algorithm can take. The length of such a path is the # of probes, so height of tree is the worst case run time.

so worst case # Probes satisfies

$$\#\text{Probes} = \text{height} \geq \lceil \log_k (\#\text{leaves}) \rceil$$

$$\geq \lceil \log_k (\#\text{verdicts}) \rceil$$

$$= \lceil \log_k (f(n)) \rceil .$$

///.

Ex.
Given $1 \leq m \leq 10^6$, find a lower bound on # of yes/no questions necessary to determine m.

<u>Soln</u>: K = # outcomes per test = 2

verdicts = 10^6

$\therefore$

Questions $\geq \lceil \log_2 10^6 \rceil = 20$,

so at least 20 questions are necessary (in worst case.)

This argument does not show that there exists an algorithm that solves the problem in 20 questions (worst case), only that there does not exist an algorithm that solves problem in 19 Questions (W.C.)

EXERCISE:

show that 'binary search' does solve the Problem in 20 questions.

Ex. Same Problem: gues m in the range $1 \le m \le 10^6$ but ask questions with 3 Possible outcomes.

$$\#\text{questions} \ge \lceil \log_3(10^6) \rceil = 13.$$

□

<u>Thm</u>.

any comparison based sorting algorithm must do, in worst case, at least $\lceil \log_2(n!) \rceil$ comparisons on input arrays of length n.

<u>Proof</u>:

comparisons like $A[i] < A[j]$ have 2 outcomes (true/false). # of verdicts $= n!$ so by prev. thm

$$\# \text{ comparisons} \geq \lceil \log_2(n!) \rceil.$$

///.

Coroliary:

Worst case runtime of a Comparison sort is $\Omega(n \log n)$.

Pf:

$$\# \text{comp} \geq \lceil \log_2(n!) \rceil = \Theta(\log(n!))$$
$$= \Theta(n \log n).$$

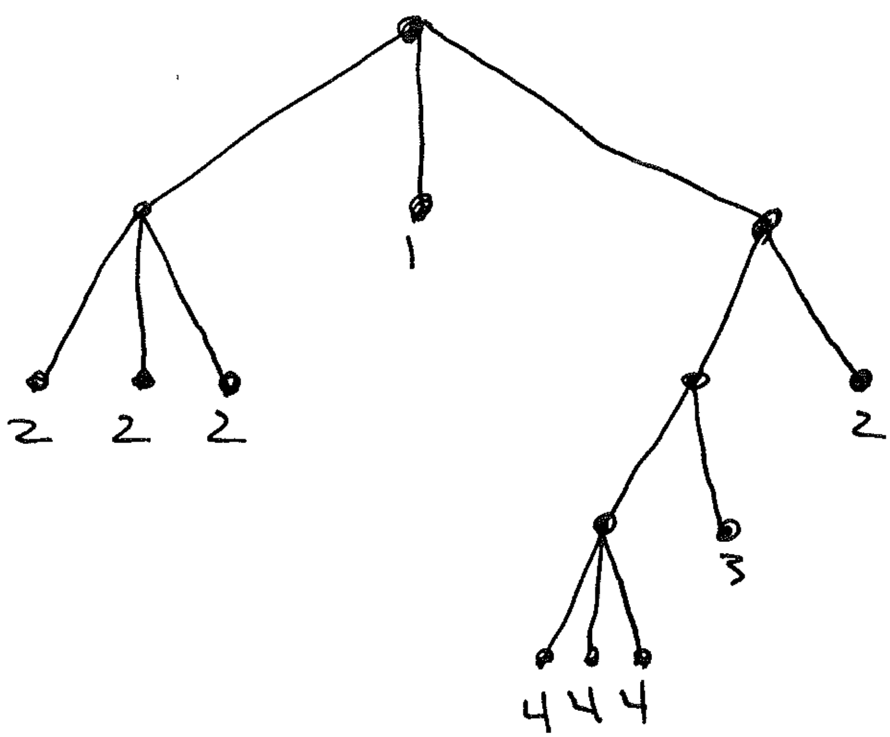
///.

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 average height of T is

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

Ex. $K=3$, $n=9$



$$a = \frac{2+2+2+1+2+3+4+4+4}{9} = \frac{8}{3} = 2.66\ldots$$

Thm

The average height a of a k-ary tree having n leaves satisfies

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

Brassard & Bratley proves case $k = 2$.

Thm

any comparison sort must do, on average, at least $\lg(n!)$ comparisons on input arrays of length n.

$\therefore$ avg. case runtime is

$$\Omega(n \log n)$$

Summary of Decision tree lower bounds

(1) Determine max # outcomes per test of data.

(2) Determine # verdicts $f(n)$ (i.e. # possible algorithm outputs) as a fcn of input size n

e.g. searching $f(n) = n$

sorting $f(n) = n!$

(3) Conclude that any algorithm that solves problem using tests in (1), must do at least

worst case: $\lceil \log_K(f(n)) \rceil$

avg case: $\log_K(f(n))$

· Basic ops. (from (1)) on input of size n.

Adversary Arguments

Consider 20 Questions again i.e. guess a # m in range

$$1 \le m \le 10^6 .$$

Players: A & B

A: thinks of # m
Cheats!!

B: tries to guess it, asks only yes/no questions.

A: Pretends to pick $x \in \{1, \ldots, 10^6\}$

B: asks a seq. of Questions $Q_1, Q_2, Q_3, \ldots$ (yes/no).

<u>note</u>: each Question Depends on Preceding Seq. of answers.

A: Always gives an answer which is consistent with his previous answers, but which is designed to prolong the game. <u>note</u>: there must always exist at least one $x \in \{1, \ldots, 10^6\}$ which would elicit the answer sequence (from an honest player.)

How long can A keep this up?

The answer provides a lower bound on # of questions that are necessary.

A's strategy must be specified explicitly.

We call A the <u>Adversary</u> or <u>Daemon</u>.

Let S_i Denote Remaining set of canditates for x after i^{th} Question has been asked and answered.

$$S_0 = \{1, 2, \ldots, 10^6\}$$

let Q_i = i^{Th} Question, and let $A_i(x)$ denote the honest answer to Q_i if mystery # 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. if $Q_1 =$ "is $x \leq 500\,000$?"

$$Y_1 = \{1, \ldots, 500\,000\}$$

$$N_1 = \{500\,001, \ldots, 1\,000\,000\}.$$

<u>note</u>: $Y_i \cap N_i = \emptyset$. and also

$$Y_i \cup N_i = S_{i-1}$$

$$\therefore \; |Y_i| + |N_i| = |S_{i-1}|.$$

Also note: at least one of Y_i or N_i must contain at least

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

numbers. (Pigeonhole principle)

(pf: if $a < \lceil \frac{c}{2} \rceil$ and $b < \lceil \frac{c}{2} \rceil$ then

$a + b < c$, so $a + b \neq c$.)

so either

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

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

Adversary's (i.e. A's) strategy:

Always answer Q_i in such a WAY which implies that x is in the larger of 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 $b_i \geq \left\lceil \frac{b_{i-1}}{2} \right\rceil$.

Then

$$b_0 = 10^6$$

$$b_1 \geq \left\lceil \frac{10^6}{2} \right\rceil$$

$$b_2 \geq \left\lceil \frac{\lceil 10^6/2 \rceil}{2} \right\rceil = \left\lceil \frac{10^6}{2^2} \right\rceil$$

$$\vdots$$

$$b_{19} \geq \left\lceil \frac{10^6}{2^{19}} \right\rceil = 2$$

$$b_{20} \geq \left\lceil \frac{10^6}{2^{20}} \right\rceil = 1.$$