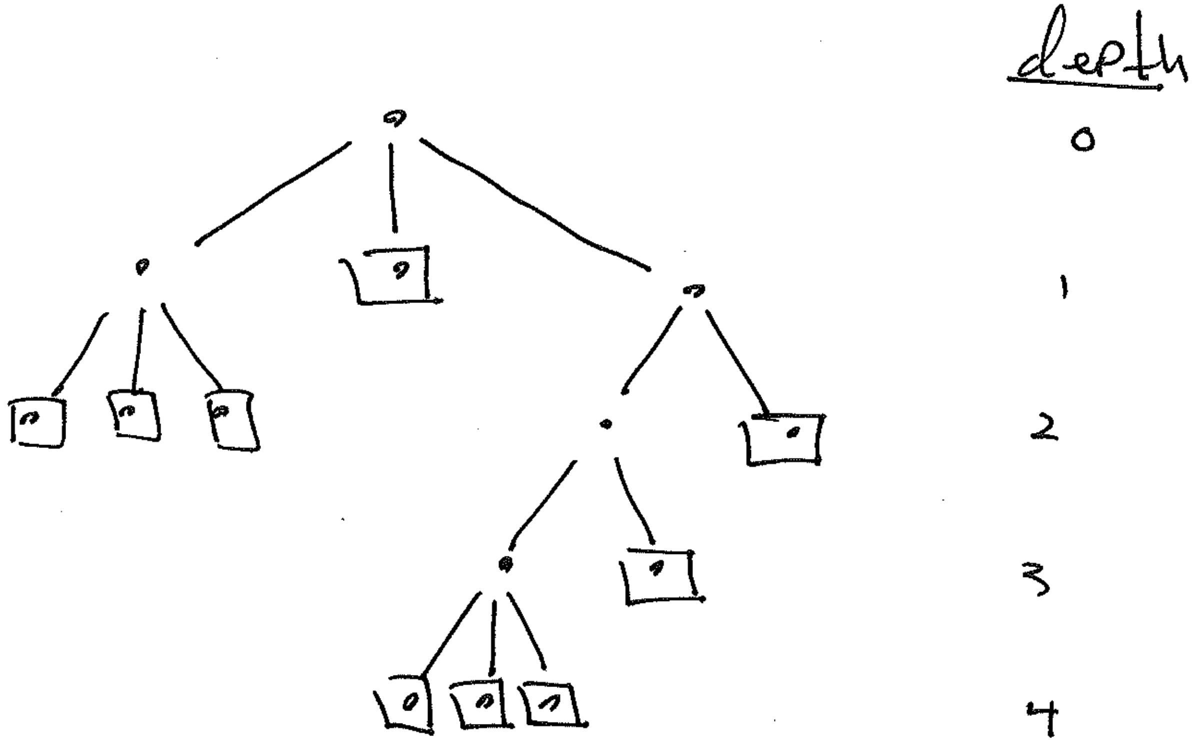
CSE 102 5-28-24

- Midterm 2 : Th 5-30-24
- hw7 : Tue (tonight) 10:00 PM
- hw8 : soon
- Programming Project : Tue 10:00 PM
- SETs are open

Defn

The average height of a rooted tree is the average depth of its leaves.

Ex



$$\frac{1\cdot 1 + 4\cdot 2 + 1\cdot 3 + 3\cdot 4}{9} = \boxed{2.66\ldots}$$

Theorem

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

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

See Brassard & Bratley for the $k=2$ case.

Apply to sorting algorithms

Theorem

Any comparison based algorithm must do, in average case, at least $\lg(n!)$ comparisons on arrays of len. n. Asymptotically this is $\Theta(n \log n)$.

Summary

(1) determine k, the max # of outcomes to each probe of data

(2) determine $f(n)$, the number of verdicts (outputs) on input of size n.

(3). conclude that any algorithm that solves P and uses only k-ary probes does at least

- $\lceil \log_k(f(n)) \rceil$ probes in worst case
- $\log_k(f(n))$ " " avg. "

Adversary Arguments

Ex. Guessing Game.

Two players

A : adversary, daemon

B : algorithm

- A pretends to choose $x \in \{1, .., n\}$.
- B asks $Q_1, Q_2, Q_3, \dots$ of binary questions

A answers in such a way as to

- never contradict prev. answers
- prolong game as much as possible.

let

$S_i = \{\text{candidates for } x \text{ after } Q_i \text{ answered}\}$

so

$S_0 = \{1, 2, \ldots, n\}$

let

$A_i(x) = $ (correct) answer to Q_i
if mystery # is x.

Ex if $n = 100$, $Q_1 =$ "is $x \leq 50$",

then $A_1(40) =$ 'yes', $A_1(60) =$ 'no'.

also define

$Y_i = \{x \in S_{i-1} \mid A_i(x) = \text{'yes'}\}$

$N_i = \{x \in S_{i-1} \mid A_i(x) = \text{'no'}\}$.

Ex $n=100$, $Q_1 =$ "is $x \le 50$", then

$$Y_1 = \{1, \dots, 50\}$$
$$N_1 = \{51, \dots, 100\}$$

We specify A's strategy:

Always answer Q_i so as to imply that S_i is the larger of Y_i, N_i.

ans

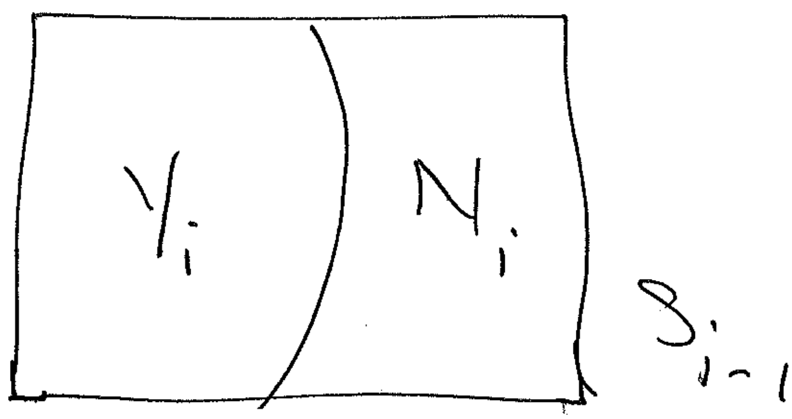
$$S_i = \begin{cases} Y_i & \text{if } |Y_i| \ge |N_i| \quad \boxed{\text{yes}} \\ N_i & \text{if } |Y_i| < |N_i| \quad \boxed{\text{no}} \end{cases}$$

Thus

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

<u>Note</u>: $S_{i-1} = Y_i \cup N_i$, and

$Y_i \cap N_i = \phi$



<u>Notation</u> $b_i = |S_i|$.

$$b_0 = n$$

$$b_1 \geq \left\lceil \frac{b_0}{2} \right\rceil = \left\lceil \frac{n}{2} \right\rceil$$

$$b_2 \geq \left\lceil \frac{b_1}{2} \right\rceil = \left\lceil \frac{\lceil n/2 \rceil}{2} \right\rceil = \left\lceil \frac{n}{2^2} \right\rceil$$

$$b_3 \geq \left\lceil \frac{n}{2^3} \right\rceil$$

$$\vdots$$

$$b_i \geq \left\lceil \frac{n}{2^i} \right\rceil$$

Process ends when $b_i = 1$, i.e.

$$\left\lceil \frac{n}{2^i} \right\rceil = 1$$

$$0 < \frac{n}{2^i} \leq 1$$

$$0 < n \leq 2^i$$

$$\downarrow$$

$$\lg n \leq i$$ ← We need the smallest such i

$$\therefore \boxed{i = \lceil \lg n \rceil}$$

<u>Recall</u>
- $N = \lfloor x \rfloor$ iff $N \leq x < N+1$
- $N = \lceil x \rceil$ iff $N-1 < x \leq N$

Thus, if B claims to know x after asking only $\lceil \lg n \rceil - 1$

questions, then $|S_i| \geq 2$, so A can claim another number as x.

It follows that any correct algorithm must ask at least $\lceil \lg n \rceil$ questions. ■

<u>Exercise</u>

Modify the argument to apply to algorithms asking k-ary questions.

lower bound: $\lceil \log_k(n) \rceil$

Summary

(1) Suppose an algorithm solving P is run against an adversary that simulates an instance of P of size n.

(2) When algorithm probes data, Adversary answers in such a way as to

- always be consistent
- prolong the game

We must specify adversary's strategy as an algorithm.

(3) Prove that there is a number $h(n)$ with property: if the algorithm halts after only $h(n)-1$ probes, then there exists at least one instance of P for which the algorithm's answer is wrong.

(4) Conclude that $h(n)$ is a lower bound for the worst case # of probes on input of size n.

Ex. Given $A[1 \cdots n]$ an array of numbers, find its maximum and where it is located:

$$(A[k], k)$$

such that $A[i] \le A[k]$ for $1 \le i \le n$.