Lecture 11:

Lower bounds via Yao's method

How to prove lower bounds?

Big difficulty: Property testing algorithms are randomized

how do you argue about their behavior?

Useful tool for lower bounding randomized algorithms:

Yao's Principle

If there is probability distribution D on union of "positive" ("yes"/"pass") + "negative" ("no"/"Fail") inputs, s.t. any deterministic algorithm of query complexity $\leq t$ outputs incorrect answer with prob $\geq 1/3$ for inputs chosen according to D, then t is a lower bound on the randomized query complexity.

moral: average case deterministic l.b. $\Rightarrow$ randomized worst case l.b.

principle works for all types of randomized algorithms

Why?

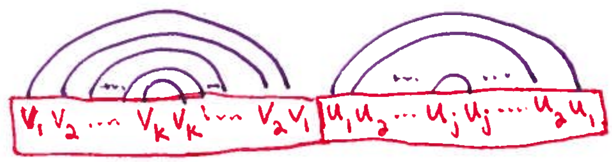
proof omitted

Game theoretic view:

Alice selects deterministic algorithm A } payoff = cost of A(x)
Bob selects input x

Von Neuman's minimax ⇒ Bob has randomized strategy which is as good when A randomized

An example:



$L_n = \{ w \mid w$ is n-bit string, $w = vv^R ww^R \}$

w is concatenation of palindromes

Note: testing is w is ε-close to a palindrome i.e. $w = vv^R$ can be done with $O(\frac{1}{\varepsilon})$ queries

def w is "ε-close to L_n" if $\exists\ w' \in L_n$ s.t. w & w' differ on $\le \varepsilon \cdot n$ characters (this is different from edit distance)

Thm if A satisfies
$\forall\ x \in L_n$, $Pr[A(x) = \text{Pass}] \ge 2/3$
$\forall\ x$ ε-far from L_n, $Pr[A(x) = \text{fail}] \ge 2/3$

then A makes $\Omega(\sqrt{n})$ queries

Proof

Plan: give distribution on inputs that is hard for all det. algs with $o(\sqrt{n})$ queries.
then Yao $\Rightarrow$ randomized l.b. of $\Omega(\sqrt{n})$

- w.l.o.g. assume $6 \mid n$

- distribution on negative inputs: ← should output "Fail" on these

  $N =$ random string of distance $\geq \varepsilon n$ from L_n

- distribution on positive inputs: ← should output "PASS" on these

  $P = \begin{cases} 1. \text{ pick } K \in_R [\frac{n}{6}+1, \frac{n}{3}] \\ 2. \text{ pick random } v, u \text{ s.t. } \\ \quad |v| = K \\ \quad |u| = \frac{n-2K}{2} \\ 3. \text{ output } v v^R u u^R \end{cases}$

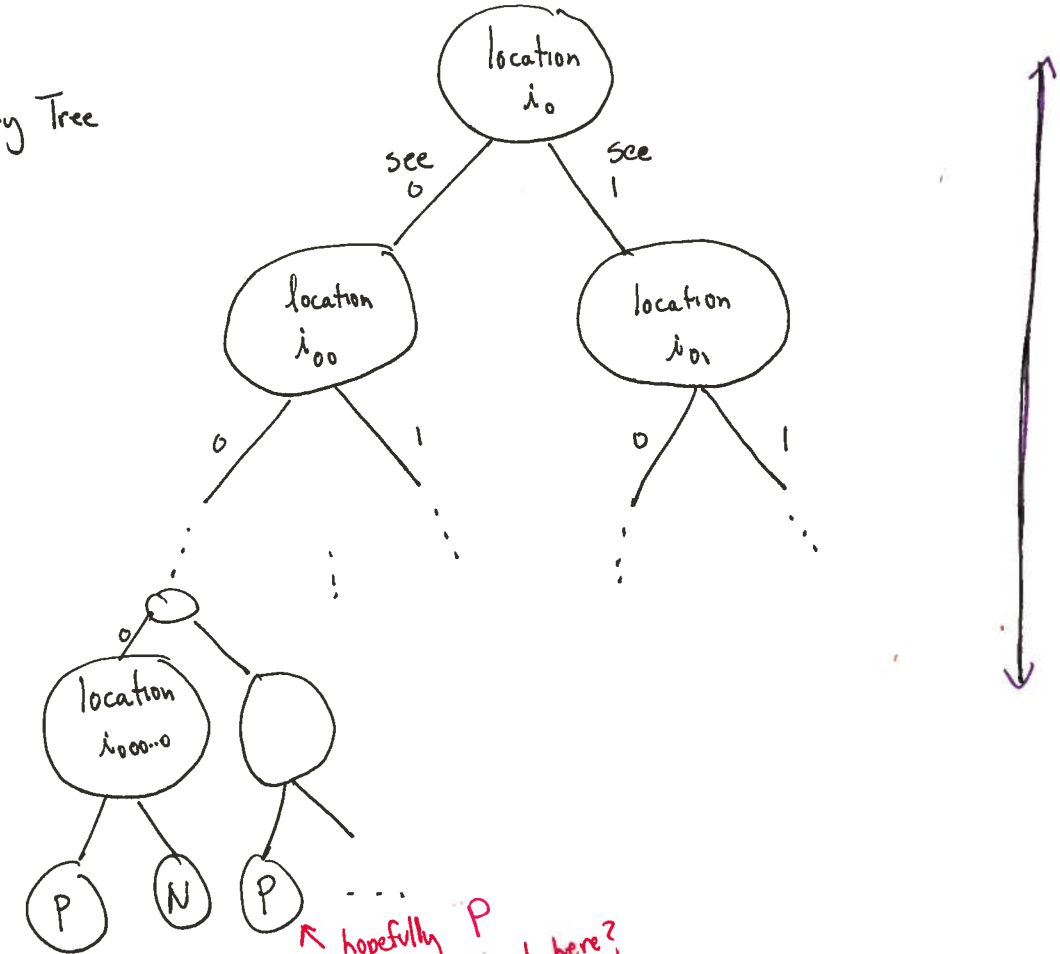
  ← note: some strings can be generated via ≥ 1 K.

- distribution D:
  - flip coin
  - if H output according to N

    else " " " P

- Assume deterministic algorithm A uses $\leq t = o(\sqrt{n})$ queries

Query Tree



hopefully P inputs reach here?

depth t

$\leq 2^t$ root-leaf paths

wlog all leaves have depth t

output leaves labelled with A's answer following path + seeing bits labelling edges

NOTE: we can calculate probability of reaching leaf since we know input distribution

Error of leaf: $E^-(\ell) = \{ \text{inputs } w \in \{0,1\}^n \mid w \ \varepsilon\text{-far} + w \text{ reaches leaf } \ell \}$ ← w should fail

$E^+(\ell) = \{ \text{inputs } w \in \{0,1\}^n \mid w \in L + w \text{ reaches leaf } \ell \}$ ← w should pass

Total error of A on D

$$= \sum_{\substack{\ell \\ \text{passing}}} \Pr_{w \in D}[w \in E^-(\ell)] + \sum_{\substack{\ell \\ \text{failing}}} \Pr_{w \in D}[w \in E^+(\ell)]$$

should fail but reach passing leaf

should pass but reach failing leaf

Why is there a problem?

lots of inputs from N + P end up at $\underline{\underline{\text{all}}}$ leaves.

<u>Claim 1</u> if $t = o(n)$, $\forall \ell$ at depth t

$$\Pr_D[w \in E^-(\ell)] \geq \left(\tfrac{1}{2} - o(1)\right) 2^{-t}$$

but, each leaf only has 1 label so almost ½ will get wrong label

<u>Claim 2</u> if $t = o(\sqrt{n})$, $\forall \ell$ at depth t

$$\Pr_D[w \in E^+(\ell)] \geq \left(\tfrac{1}{2} - o(1)\right) 2^{-t}$$

So error of A on D

$$= \sum_{\substack{\ell \\ \text{passing}}} \left(\tfrac{1}{2} - o(1)\right) 2^{-t} + \sum_{\substack{\ell \\ \text{failing}}} \left(\tfrac{1}{2} - o(1)\right) 2^{-t} \geq \tfrac{1}{2} - o(1) \gg \tfrac{1}{3} \quad \blacksquare$$

still need to prove the claims...

Pf of Claim 1:

idea: N is close to U

+ U would end up uniformly distributed at each leaf

$$\Rightarrow \Pr_{w \in U}[w \in E^{-}(\ell)] = \frac{2^{n-t}}{2^n} = 2^{-t}$$

How much can distribution change by using N instead of U?

$$|L_n| \le \underbrace{2^{\frac{n}{2}}}_{\text{choice of } u,v} \cdot \underbrace{\frac{n}{2}}_{\text{choice of } i}$$

\# words at dist $\le \varepsilon$ from L_n:

$$\le 2^{\frac{n}{2}} \cdot \frac{n}{2} \cdot \sum_{i=0}^{\varepsilon n} \binom{n}{i} \le 2^{\frac{n}{2} + 2\varepsilon \log(\frac{1}{\varepsilon}) n}$$

so

$$E^{-}(\ell) \ge \underbrace{2^{n-t}}_{\#\text{ strings in } U \text{ that reach } \ell} - \underbrace{2^{\frac{n}{2} + 2\varepsilon \log(\frac{1}{\varepsilon}) n}}_{\#\text{ words at dist} \le \varepsilon} = (1 - o(1))\, 2^{n-t}$$

assume $\varepsilon << 1/8$

t is $o(n)$

so 1st term swamps 2nd term!

So $\Pr_D[w \in E^{-}(\ell)] \ge \frac{1}{2} \Pr_N[w \in E^{-}(\ell)]$

$$\ge \frac{1}{2} \frac{|E^{-}(\ell)|}{2^n} \ge \left(\frac{1}{2} - o(1)\right) 2^{-t}$$

$\blacksquare$

Proof of Claim 2

Will show: For every fixed set of $o(\sqrt{n})$ queries, lots of strings in L_n follow that path.

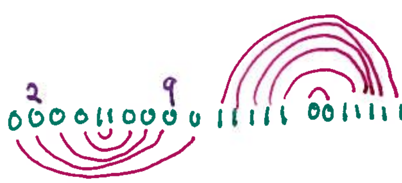
Count # strings agreeing with t queries of leaf?

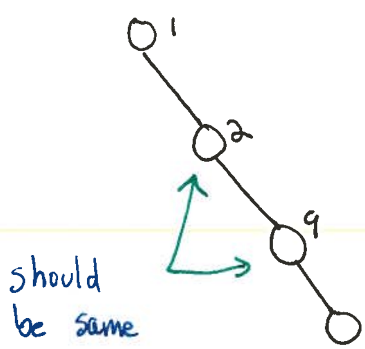
$$= 2^{n-t}$$

Count # strings in L_n agreeing with t queries of leaf?

$$\geq 2^{n-t} - ?$$

Main difficulty:





Fix $k = 10$

should see same value at locns:

1, 10
2, 9
3, 8
4, 7
5, 6
11, n
12, n-1
⋮

☹ maybe no string in L_n follows path?

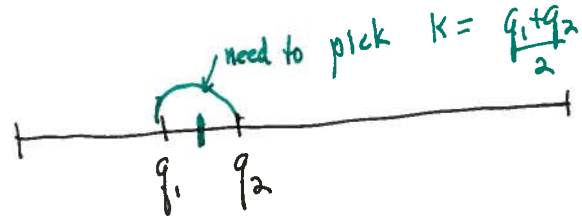
☺ that's why k is picked randomly in $[\frac{n}{6} .. \frac{n}{3}]$!
not <u>all</u> queries can be bad

Given leaf ℓ, let $Q_\ell \leftarrow$ indices queried along the way

For each of $\binom{t}{2}$ pairs of queries $q_1, q_2 \in Q_\ell$

at most 2 choices of k for which

q_1, q_2 is symmetric to k or $\frac{n}{2}+k$



Only 1 choice in this case!

"good k"

$\Rightarrow$ # choices of k s.t. no pair in Q_ℓ symmetric around k or $\frac{n}{2}+k$ is

$$\geq \frac{n}{6} - 2\binom{t}{2} = (1-o(1))\left(\frac{n}{6}\right)$$

For these good k, # strings that follow path $= 2^{n/2 - t}$

So
$$\Pr_D[w \in E^+(\ell)] = \sum_w \sum_k \underbrace{\Pr_D[w \mid k]}_{2^{-n/2}} \underbrace{\Pr[\text{choose } k]}_{\frac{6}{n}} \cdot 1_{w \in E^+(\ell)}$$

$$\geq \frac{1}{\left(\frac{n}{6}\right)\left(2^{\frac{n}{2}}\right)} \left[(1-o(1)) \cdot \frac{n}{6}\right] \cdot 2^{\frac{n}{2}-t} = (1-o(1)) \cdot 2^{-t}$$

$$\Rightarrow \Pr_D[w \in E^+(\ell)] = \left(\tfrac{1}{2} - o(1)\right) 2^{-t} \quad \blacksquare$$