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 in correct
answer with prob $\geq \frac{1}{3}$ for inputs chosen according to $D$,
then $t$ is a lower bound on the randomized
query complexity.

moral: average case deterministic lb. $\Rightarrow$
randomized worst case lb.
Why?

Proof omitted

Game theoretic view:

Alice selects deterministic algorithm $A$; payoff = $\sum \text{cost of } A(x)

Bob selects input $x$

Von Neumann's minimax $\Rightarrow$ Bob has randomized strategy which is as good when $A$ randomized

An example:

$\begin{align*}
L_n = \{ & w \mid w \text{ is } n\text{-bit string} \} \\
& w = v^R w w^R.
\end{align*}$

$w$ is concatenation of palindromes

Note: testing is $w$ is $\varepsilon$-close to a palindrome, i.e. $w = v^R v$

can be done with $O(\frac{1}{\varepsilon})$ queries

$\defn w$ is "$\varepsilon$-close to $L_n" \iff \exists w \in L_n

s.t. w + w^R$ differ on $\leq \varepsilon n$ characters

(this is different from edit distance)

Thm: if $A$ satisfies

$\forall x \in L_n, \Pr[A(x) = \text{Pass}] \geq 2/3$

$\forall x, \varepsilon$-far from $L_n, \Pr[A(x) = \text{Fail}] \geq 2/3$

then $A$ makes $\Omega(\varepsilon n)$ queries
Proof:

Plan: give distribution on inputs that is hard for alldet. algs with \( o(\sqrt{n}) \) queries, then \( \text{ Yao } \Rightarrow \text{ randomized 1.6 of } \Omega(\sqrt{n}) \)

- w.l.o.g. assume \( b/n \)
- distribution on negative inputs:
  - \( N \) = random string of distance \( \geq 2n \) from \( L_n \)
- distribution on positive inputs:
  \[
  P = \begin{cases}
    1. \text{ pick } k \in_R \left[ \frac{n}{b+1}, \frac{n}{3} \right] \\
    2. \text{ pick random } v, u \text{ s.t. } \\
    \quad |v| = k \\
    \quad |w| = n - 2k \\
    3. \text{ output } vv^k uu^k \\
  \end{cases}
  \]

- distribution \( D \):
  - flip coin
  - if \( H \) output according to \( N \)
  - else \( " \) output according to \( P \)
Assume deterministic algorithm $A$ uses $\leq t = o(\sqrt{n})$ queries.

**Query Tree**

- **Location $l_0$**
  - **Location $l_{00}$**
    - **Location $l_{000}$**
      - **Outputs**: $P$, $N$, $P$ (indicates yes, no, yes)
      - **Paths** labeled with inputs reaching the root.
      - **Depth $t$**
      - **$2^t$ root-leaf paths**
      - **WLOG, all leaves have depth $t$**

**NOTE:** we can calculate probability of reaching leaf $w$ since we know input distribution.

**Error of leaf**:

- $E^-(x) = \sum_{w \in \{0,1\}^n} |w|$ if $x$ is far and $w$ reaches leaf $L$.
- $E^+(x) = \sum_{w \in \{0,1\}^n} |w|$ if $x \in L$ and $w$ reaches leaf $L$.

$E^-(x)$ should fail.
$E^+(x)$ should pass.
Total error of \( A \) on \( D \)

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

Why is there a problem?

lots of inputs from \( N + P \) end up at all leaves.

Claim 1: if \( t = o(n) \), \( \forall l \) at depth \( t \)

\[
\Pr_{D} [w \in E^{-}(l)] \leq (\frac{1}{2} - o(1)) \cdot 2^{-t}
\]

Claim 2: if \( t = o(\sqrt{n}) \), \( \forall l \) at depth \( t \)

\[
\Pr_{D} [w \in E^{+}(l)] \geq (\frac{1}{2} - o(1)) \cdot 2^{-t}
\]

So error of \( A \) on \( D \)

\[
= \sum_{\text{passing}} \frac{1}{2} - o(1) \cdot 2^{-t} + \sum_{\text{failing}} \frac{1}{2} - o(1) \cdot 2^{-t} \geq \frac{1}{2} - o(1) \geq \frac{1}{3}
\]

still need to prove the claims...
**Proof of Claim 1:**

**Idea:** $N$ is close to $U$.

$U$ would end up uniformly distributed at each leaf.

$$\Rightarrow P_{u \in U} \left[ w \in E^{-t}(U) \right] = \frac{2^{n-t}}{2^n} = 2^{-t}$$

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

$$|L_n| \leq 2^\frac{n}{2} \cdot \frac{n}{2}$$

**Choice of $u,v$**

**# words at dist $\leq \varepsilon$ from $L_n$:**

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

So,

$$E^{-t}(U) \geq 2^{n-t} - 2^\frac{n}{2} + 2\varepsilon \log(2)n = (1 - o(1))2^{n-t}$$

**# strings in $U$ that reach $l$**

**# words at dist $\leq \varepsilon$**

Assume $\varepsilon < \frac{1}{8}$

$t = o(n)$

So $1^{st}$ term swamps $2^{nd}$ term!

So,

$$Pr_{D} \left[ w \in E^{-t}(U) \right] \geq \frac{1}{2} \cdot Pr_{N} \left[ w \in E^{-t}(U) \right]$$

$$\geq \frac{1}{2} \cdot \frac{|E^{-t}(U)|}{2^n} \geq (\frac{1}{2} - o(1))2^{-t}$$
Proof of Claim 2

Will show: For every fixed set of $o(n^2)$ 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:

$1, 10$
$2, 9$
$3, 8$
$4, 7$
$5, 6$
$n, n$
$12, n-1$
$\vdots$

maybe no string in $L_n$ follows path?

that's why $k$ is picked randomly in $\left[ \frac{n}{6}, \frac{n}{3} \right]$!

not all queries can be bad
Given leaf $l$, let $Q_l$ be indices queried along the way.

For each of $(\frac{t}{2})$ pairs of queries $q_1, q_2 \in Q_l$ at most 2 choices of $k$ for which $q_1, q_2$ is symmetric to $k$ or $\frac{n}{2} + k$ need to pick $k = \frac{q_1 + q_2}{2}$ only 1 choice in this case!

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

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

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

So $Pr_p[\omega \in E^+(l)] = \sum_w \sum_k Pr_p[w | k] \cdot Pr[\text{choose } k] \cdot Pr_\omega[\omega \in E^+(l)]$

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

$\Rightarrow Pr_\omega[\omega \in E^+(l)] = \left(\frac{1}{b} - o(1)\right) 2^t$