Lecture 10:

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 a probability distribution $D$ on union of "positive" ("yes"/"pass") + "negative" ("no"/"fail") inputs, s.t. any deterministic algorithm of query complexity $\leq t$ outputs a 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 \( A \)
Bob selects input \( x \)

\[ \text{payoff} = \sum \text{cost of } A(x) \]

Von Neumann's minmax \( \Rightarrow \) Bob has randomized strategy
which is as good when \( A \) randomized

An example:

\[ L_n = \{ w | w \text{ is } n\text{-bit string} \} \]
\[ w = v v^R \text{ where } R \text{ is } v \text{ is palindrome} \]

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

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

\[ \text{def } w \text{ is } \varepsilon\text{-close to } L_n \text{ if } \exists w \in L_n \]

\[ \text{st. } w + w^i \text{ differ on } \varepsilon n \text{ characters} \]

(this is different from edit distance)

\[ \text{Thm } \text{if } A \text{ satisfies} \]

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

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

Then \( A \) makes \( \Omega(\frac{1}{\varepsilon^2}) \) queries
Proof:

Plan: give distribution on inputs that is hard for all det. alg. 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 $\geq en$ from $L_n$

  - distribution on positive inputs:

    $P = \begin{cases} 
    1. \text{ pick } k \in_R \left[ \frac{n}{b+1}, \frac{n}{3} \right] \\
    2. \text{ pick random } v, u \text{ s.t. } \\
       |v| = k \\
       |u| = \frac{n-2k}{2} \\
    3. \text{ output } vv^Ruu^R 
    \end{cases}$

    should output "Fail"

  - distribution $D$:

    - flip coin
      - if $H$ output according to $N$
      - else $\ldots$ $\mathbf{P}$
Assume deterministic algorithm $A$ uses $\leq t = o(n^{0.5})$ queries.

**Query Tree**

- 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:

$E^- (A) = \# \text{ inputs } w \in \{0, 1\}^n | w \text{ errors } + w \text{ reaches leaf } l^-$

$E^+ (A) = \# \text{ inputs } w \in \{0, 1\}^n | w \in L + w \text{ reaches leaf } l^+$

We should fail $w$ 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 $t$

$$\Pr_D [w \in E^-(l)] = (\frac{1}{2} - o(1)) 2^{-t}$$

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

$$\Pr_D [w \in E^+(l)] = (\frac{1}{2} - o(1)) 2^{-t}$$

So error of $A$ on $D$

$$= \sum_{\text{passing}} (\frac{1}{2} - o(1)) 2^{-t} + \sum_{\text{failing}} (\frac{1}{2} - o(1)) 2^{-t} \geq \frac{1}{2} - o(1)$$

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

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

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

\[ |L_n| = 2^\frac{n}{2} \cdot \frac{n}{2} \]

↑ choice of \(U\)
↑ choice of \(U\)

# words at dist \( \leq \varepsilon \) from \(L_n\):

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

so

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

# strings in \(U\) that reach \(L\)

↑ # words at dist \( \leq \varepsilon \)
assume \( \varepsilon \ll \frac{1}{8} \)
\( t \) is \( o(n) \)
so 1st term swamps 2nd term!

So

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

\[ \geq \frac{1}{2} \frac{|E^-(U)|}{2^n} \geq \left( \frac{1}{2} - o(1) \right) 2^{-t} \]
Proof of Claim 2

Will show: For every fixed set of $o(c\ln n)$ 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:

should be same

maybe no string in $L_n$ follows path?

That's why $k$ is picked randomly in $[\frac{n}{6}, \frac{n}{3}]$.

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

For each of \( \binom{\ell}{2} \) pairs of queries \( q_i, q_j \in Q_l \)

at most 2 choices of \( k \) for which

\( q_i, q_j \) is symmetric to \( k \) or \( \frac{n}{2} + k \)

need to pick \( k = \frac{q_i + q_j}{2} \)

only 1 choice in this case!

\[ \Rightarrow \ \text{"good } k\text{"} \]

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

\[ \geq \ \frac{n}{6} - 2\chi(\frac{t}{a}) = (1 - o(1)) \left( \frac{n}{6} \right) \]

\[ \Rightarrow \ \sum_{w} \sum_{k} P_r[w | k] P_r[\text{choose } k] \cdot \mathbf{1}_{w \in E^+(l)} \]

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

\[ \Rightarrow \ P_r[w \in E^+(l)] = (\frac{1}{2} - o(1)) \cdot 2^{-t} \]