Lecture 18:

Lower bound techniques
How to prove lower bounds?

easy? sublinear time algorithms see very little of input.
difficult? sublinear time algorithms are usually randomized
How to prove lower bounds?

Easy? Sublinear time algorithms see very little of input.

Difficult? Sublinear time algorithms are usually randomized.

Useful lower bound tool:

Yao's Principle: Given distribution \( D \) on union of "positive" (Yes, PASS) instances and "negative" (No, FAIL) inputs, such that any deterministic algorithm of query complexity \( \leq t \) is incorrect with prob \( \geq 1/3 \) on inputs chosen from \( D \), then \( t \) is a lower bound on randomized query complexity.

(Proof omitted) (see Wikipedia)
Game theoretic view:

Alice selects deterministic alg A \rightarrow \text{payoff} = \text{cost of } A(x)

Bob selects input x

\( \forall \text{a} \text{ random} \) deterministic algorithm \( \Rightarrow \) A picks random deterministic algorithm (includes random bits)

Von Neumann’s minimax \( \Rightarrow \) when A randomized,

Bob can do just as well as when A deterministic

\( \Rightarrow \) if want to show l.b. need only show a “bad” distribution on inputs that is “hard” for any deterministic algorithm
Example application of Yao's method:

$2^{2^2^2} = \{ w \mid w \text{ is a pair of palindromes} \}$

Concatenation of 2 palindromes

e.g. 00111111 00 011100 00 1100

$V, v_2, \ldots, V_k | V_k | v_k, \ldots, v_1 | u_1, u_2, \ldots, u_{2^3}$

Note that testing $PAL = \{ w \mid w = V V^R \}$ is "easy"

pick random $i$, if $w_i \neq w_{n-i}$ FAIL

Can test $2^{2^2^2}$ in $O(\sqrt{n})$ time

Can you do better?
Thm: any property tester for $2\text{PAL}$ needs $\sqrt[3]{n}$ queries.

e.g. if $A$ satisfies $\forall x \in 2\text{PAL}, \Pr[A(x) = \text{PASS}] \geq 2/3$

+ $\forall x \in \text{far from } 2\text{PAL}, \Pr[A(x) = \text{FAIL}] \geq 2/3$

then $A$ makes $\Omega(\sqrt[3]{n})$ queries.

Pf. Plan: give distribution on inputs that is hard for all deterministic algorithms using $O(\sqrt[3]{n})$ queries.

$\forall A_0 \Rightarrow$ randomized l.b. of $\Omega(\sqrt[3]{n})$
Distribution on "Fail" inputs:

\[ F = \text{random string of distance } \geq 3n \text{ from } 2PAL \]

Distribution on "Pass" inputs: (wlog assume \( b/n \))

\[ P = \frac{1}{2} \begin{cases} 1. & \text{pick } K \in R \left[ \frac{n}{b + 1}, \frac{n}{3} \right] \\ 2. & \text{pick random } v, u \text{ s.t. } |v| = k \\ 3. & \text{output } v v \text{ if } u \text{ k } u k \text{ if } u l \text{ if } u l = \frac{n - k}{2} \\ \end{cases} \]

Note: some strings can be generated by multiple K's

E.G.: 111111

Bad Distribution:

\[ D = \frac{1}{2} \begin{cases} H: \text{output according to } F \\ T: \text{flip coin} \end{cases} \]
Assume deterministic algorithm $A$

- Each input follows exactly one branch reached leaf, which is hopefully labelled by correct answer.

- Suppose $w \in \{0,1\}^n$; $\Pr[w \text{ reaches leaf } l] = 2^{-t}$

- Assume no "repeats" along path.
  - Depth of decision tree is $t$
  - Wlog assume all leaves have depth $t$ (complete binary tree).
  - $2^t$ root-leaf paths.

- We can calculate prob of reaching each leaf given (input dist $D$).
For each leaf $l$:

$$E^-(l) = \{ w \in \mathbb{Q}_1 \mathbb{S}^n \text{ s.t. } \text{dist}(w, 2\text{PAL}) \geq 3n \text{ and } w \text{ reaches leaf } l \}$$

$w$ should fail

$$E^+(l) = \{ w \in \mathbb{Q}_1 \mathbb{S}^n \land 2\text{PAL} \text{ and } w \text{ reaches leaf } l \}$$

$w$ should pass

Total error of $A$ on $0$:

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

$\uparrow$ output of leaf

$\uparrow$ correct answer is Fail

$\uparrow$ should Pass
For each leaf $l$: 

$$E^-(l) = \sum_{w \in \mathcal{L}} \text{st. dist}(w, 2\text{PAL}) \geq EN + w \text{ reaches leaf } \ell^3$$ 

$$E^+(l) = \sum_{w \in \mathcal{L}} \text{st. dist}(w, 2\text{PAL}) \cap \text{PAL} + w \text{ reaches leaf } \ell^3$$

**Claim 1** if $t = o(n)$, $\forall \ell$ at depth $t$, 

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

But each leaf has to choose a label so will be wrong on almost $\frac{1}{2}$ inputs that reach it.

**Claim 2** if $t = O(\sqrt{n})$, $\forall \ell$ at depth $t$, 

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

3 so fail inputs show up at all leaves

Total error of $A$ on $D$: 

$$= \sum_{l} \left( \frac{1}{2} - o(1) \right) 2^{-t} + \sum_{l} \left( \frac{1}{2} - o(1) \right) 2^{-t} \geq \frac{1}{2} - o(1) > \frac{1}{3}$$
Claim 1: if \( t = o(n) \), \( \forall \ell \) at depth \( t \)

\[
\Pr_D[ w \in E^*(U) ] \geq (\frac{1}{2} - o(1)) 2^{-t}
\]

**Proof:**

Plan:

- \( F \) is close to \( U \)
- \( U \) is uniformly distributed at each leaf
  (each leaf has random bit, so go left/right with equal probability)

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

But how much can distribution on leaves change using \( F \)?

\[
|2^{\operatorname{PAL}_n}| \leq \frac{n}{2} \cdot 2^{n/2} \Rightarrow \text{choice of } u, v
\]

\[
\# \text{words at dist} \leq 3^n \text{ from } 2^{\operatorname{PAL}} \leq \left( 2^{n/2} \right)^3 \leq 2^{\frac{3n}{2}} < 2^{n+2} \leq 2 \cdot \log \frac{1}{\varepsilon} n
\]

\( F = \) random string of distance \( \leq 3^n \) from \( 2^{\operatorname{PAL}} \)

\[
P = \left\{ \begin{array}{l}
1. \text{ pick } k \in \mathbb{R} \left[ \frac{n}{6} + 1, \frac{n}{3} \right] \\
2. \text{ pick random } v_i, w_i \text{ s.t. } |v_i| = k \quad |w_i| = \frac{k}{2}
\end{array} \right.
\]

\[
D = \left\{ \begin{array}{l}
1. \text{ flip coin } \text{ if } H \text{ output according to } F \\
\text{ else } " " \text{ " } \text{ " } P
\end{array} \right.
\]
so \( E^-(x) \geq 2^{n-t} - 2^{\frac{n}{2} + 2\epsilon \log 2 \cdot n} = (1 - o(1))2^{n-t} \) since \( t \ll \frac{n}{2} \)

so \( \Pr_D [ W \in E^-(L) ] \geq \frac{1}{2} \cdot \Pr_F [ W \in E^-(L) ] \)

\[
= \frac{1}{2} \frac{|E^-(L)|}{2^n} \geq (\frac{1}{2} - o(1))2^{-t}
\]
Claim 2: if \( t = O(\sqrt{n}) \), \( \forall \ell \) at depth \( t \)

\[
\Pr_0 [ w \in E^+(\ell) ] \geq (\frac{1}{2} - 6(\ell)) 2^{-t}
\]

**Proof**: Plan: for every fixed set of \( o(\sqrt{n}) \) queries, lots of strings in 2PAC follow the path.

- How many strings agree with leaf \( \ell \)? \( 2^{-t} \)
- How many \( n \)-bit strings in 2PAC agree with leaf \( \ell \)?

\[
\geq 2^n - 2^k - 2^k
\]

**Difficulty**: Fix \( K = 10 \); should see some value at \( 1, 10 \)

Lots of dependences.

\( F \): random string of distance \( \geq 3n \) from 2PAC.

**P**:
1. pick \( K \in R \left[ \frac{n}{k+1}, \frac{n}{3} \right] \)
2. pick \( u, v \) at random
3. output \( uv^ku^v \)

**D**: flip coin
- if \( H \) output according to \( F \)
- else "" "" "" "" \( P \)

Maybe no string in 2PAC follows the path?
but $k$ is picked randomly! in $\left[ \frac{n}{6} + 1, \ldots, \frac{n}{3} \right]$

hope paths that pair up dependent queries for one $k$ will do badly on most others?

Consider leaf $l$,

$Q_l = \{10, 22, 13, 63\}$

$Q_l \leftarrow$ indices queried along way

Up pair $q_1, q_2 \in Q_l$, at most 2 choices of $k$ "pair" them! $\leq 1$?

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

\[ \text{Pr}[w \in E(t) \mid t] \geq (1-0.12^t)^{2-P} \]

For every fixed set of \( o(n) \) queries, \( t \) of strings in 2PAL follow the path

so "PASS" inputs show up at all leaves

\[ F = \text{random string of distance } 2n \text{ from } 2\text{PAL} \]

D = \{ 1. pick \( k \in \{ \frac{n}{2}+1, 3 \} \) \}

2. if \( H \) output according to \( P \)

3. output \( W/k \) or \( W \)