Property Testers For Monotonicity:

Given list \( y_1, \ldots, y_n \)

Output sorted?

i.e. if \( y_1 \leq y_2 \leq \cdots \leq y_n \) output PASS (with prob \( \geq \frac{3}{4} \))

if \( y_1, \ldots, y_n \) E-far from sorted (need to delete \( n \)?)

Output FAIL (with prob \( \geq \frac{3}{4} \))

e.g.

Sorted: 1 2 4 5 7 11 14 19 20 21 23

close: 1 4 2 5 7 11 14 19 20 39 23

gar: 45 39 23 38 45 21 20 19 2

An easy case: \( y_i = 0, 1, 3 \) \( \forall i \)

Can do it in \( \text{poly}(1/\epsilon) \) time.
A first attempt:

Proposed algorithm: "neighbor test"

Pick random \( i \), test \( y_i < y_{in} \)

Bad input:

\[ 1, 2, 3, 4, 5, \ldots, n/4, 1, 2, 3, 4, \ldots, n/4, 1, 2, 3, 4, \ldots, n/4 \]

- \( \frac{3}{4} n \) are far from monotone
- only 3 choices of \( i \) fail

A second attempt:

Proposed algorithm: "random pair test"

Pick random \( i < j \), test \( y_i < y_j \)

Bad input: \( n/4 \) groups of 4 decreasing elements

\[ 9, 3, 2, 1, 7, 6, 5, 12, 11, 10, 9, 16, 15, 14, 13, \ldots \]

- largest monotone sequence size \( n/4 \)
- must pick \( i, j \) in same group to fail, prob \( \leq \frac{1}{n} \)
  - if see \( o(n^2) \) samples, prob \( o(1) \)
A minor simplification:

Let's assume list is distinct.

Claim: This is wlog

why? (old trick used in parallel computation)

\[ X_1 \ldots X_n \rightarrow (X_1, 1), (X_2, 2), \ldots, (X_n, n) \]

"virtually" (at runtime)
append \( i \) to each \( X_i \)

breaks ties w/o changing order

i.e. if \( X_i \leq X_{i+1} \) then \( (X_i, i) \leq (X_{i+1}, i+1) \)

A test: Given \( X_1 \ldots X_n \)

Repeat \( O(\varepsilon) \) times:

Pick \( i \in \mathbb{R} [n] \)

\( Z \leftarrow X_i \)

do binary search on \( X_1 \ldots X_n \) for \( Z \)

if see any inconsistency, \textbf{FAIL} + halt

i.e. left is bigger, right is smaller

if end up at locn \( j \neq i \), \textbf{FAIL} + halt

Pass
- If $X_1 < X_2 < \ldots < X_n$ then always passes

- To show: if need to change $z \in \mathbb{N}$ such test fails why equivalently: if test likely to pass, $X_i$'s $\epsilon$-close to monotonically
defn. $i$ "good" if bin search for $z \leq X_i$ successful

**Restatement of test:**
- Pick $O(\frac{1}{\epsilon})$ $i$'s randomly + pass if all are good
- If test likely to pass, $\geq 1 - \epsilon$ fraction of $i$'s are good
- (otherwise, in $O(\frac{1}{\epsilon})$ samples, likely to hit a bad $i$)

**Main observation:**
- "good" elements form increasing subsequence

**Proof:** if $i < j$ both good, let $K$ be least common ancestor in bin search tree.
- When hit $X_K$, search for $X_i$
- if went left + search for $X_j$
- went right.

so $X_i < X_K < X_j$
Monotonicity over Posets:

\[
def\ f\text{ is monotone over poset } P \text{ if } \forall x \leq y \text{ then } f(x) \leq f(y)\]

Examples: Can represent via dags

- Bipartite posets

- Hypercube

\[
\begin{align*}
(11111) & \quad \text{(all 1's)} \\
(01111) & \quad \text{(level 2: five 1's)} \\
(00111) & \quad \text{(level 3: six 1's)} \\
(00011) & \quad \text{(top level)} \\
(00001) & \quad \text{(level 5: six 1's)} \\
(00000) & \quad \text{bottom level: all 0's)}
\end{align*}
\]

In h.w.: Show testing monotonicity of arbitrary poset can be transformed into "equivalent" monotonicity testing problem on bipartite poset.
If a test for monotonicity can also test:

1) Given 2CNF $\varphi$ along with assignment $A = \{a_1, \ldots, a_n\}$ $a_i \in \{T,F\}$
   - Pass if $\varphi(A) = T$
   - Fail if $\forall A' s.t. A \varepsilon$-close to $A'$ $\varphi(A') = F$

2) Given $G$ with $U \subseteq V$
   - Pass if $U$ is VC
   - Fail if $\forall u \in U'$ s.t. $u$ $\varepsilon$-close to $U'$, $U'$ not VC

3) Given $G$ with $U \subseteq V$
   - Pass if $U$ is clique
   - Fail if $\forall u' \in U'$ s.t. $u'$ $\varepsilon$-close to $U$, $U'$ not clique

**Theorem**

For bipartite graphs (n nodes on each side) $\varepsilon$-mon test can be done in $O(\sqrt{n/e})$ queries.

**Hw.**

**Theorem**

$\varepsilon$-mon test requires $n^{o(1)}$ queries if nonadaptive $\exists$ open problem: Can we improve this to $O(\sqrt{n})$? for adaptive queries?

$\Rightarrow \Omega(\log n)$ queries adaptive
What about grids?

\[ f: [n] \times [n] \rightarrow [m] \]

Can test monotonicity in \( O(\log^2 n) \) time. \( \text{\textcolor{red}{\text{\it actually}}\ O(\frac{1}{\varepsilon} \log n \log m)} \)

\[ f: [n]^d \rightarrow [m] \]

Can test monotonicity in \( O(\frac{d}{\varepsilon} \log n \log m) \)

\[ f: 2^d \rightarrow \mathbb{R}^+ \]

Can test monotonicity in \( O(\frac{d^{\frac{1}{2}}}{\text{poly}(\varepsilon)} \text{ poly}(\log d)) \)

Need \( \mathcal{O}(d^{\frac{1}{4}}) \) queries (even for adaptive algorithms!)