Property Testers For Monotonicity:

Given list $y_1, \dots, y_n$

Output sorted?

i.e. if $y_1 \le y_2 \le \dots \le y_n$ output PASS (with prob $\ge 3/4$)

if $y_1 \cdots y_n$ ε-far from sorted $\left(\begin{array}{l}\text{need to delete } \varepsilon n \text{?} \\ \text{need to change } \varepsilon n \text{?}\end{array}\right)$

output FAIL (with prob $\ge 3/4$)

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
far	45	39	23	1	38	4	5	21	20	19	2

An easy case: $y_i \in \{0,1\}$ $\forall i$

Can do it in $\text{poly}(1/\varepsilon)$ time.

A first attempt:

Proposed algorithm: "neighbor test"

Pick random i, test $y_i \le y_{i+1}$

Bad input:

$1, 2, 3, 4, 5, \dots, n/4, 1, 2, 3, 4, \dots, n/4, 1, 2, 3, 4, \dots n/4, 1, 2, 3, 4, \dots n/4$

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

A second attempt:

Proposed algorithm: "random pair test"

Pick random $i < j$, test $y_i \le y_j$

Bad input: $n/4$ groups of 4 decreasing elements

$\underbrace{4, 3, 2, 1}, \underbrace{8, 7, 6, 5}, \underbrace{12, 11, 10, 9}, \underbrace{16, 15, 14, 13}, \dots$

- largest monotone sequence size $n/4$
- must pick i, j in same group to fail, prob $\le \frac{1}{n}$
  if see $o(\sqrt{n})$ samples, prob $o(1)$

A minor simplification:

Lets assume list is distinct

<u>Claim</u> This is wlog

why? (old trick used in parallel computation)

$x_1 \cdots x_n \rightarrow (x_1, 1), (x_2, 2), \ldots, (x_n, n)$

"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 $(x_i, i) < (x_{i+1}, i+1)$

A test: Given $x_1 \cdots x_n$

Repeat $O(\frac{1}{\varepsilon})$ times:

runtime $O(\frac{\log n}{\varepsilon})$

Pick $i \in_R [n]$

$z \leftarrow x_i$

do binary search on $x_1 \cdots x_n$ for z

if see any inconsistency FAIL + halt

i.e. left is bigger
right is smaller

if end up at locn $j \neq i$ FAIL + halt

Pass

- If $x_1 < x_2 < \ldots < x_n$ then always passes
- To show: if need to change $\geq \varepsilon n$ x_i's then test fails whp

equivalently: if test likely to pass, x_i's ε-close to monotone

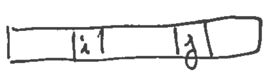
<u>defn</u>. i "good" if bin search for $z \in x_i$ successful

<u>restatement of test:</u>
Pick $O(\frac{1}{\varepsilon})$ i's randomly + pass if all are good

if test likely to pass, $\geq 1-\varepsilon$ fraction of i's are good
(otherwise, in $O(\frac{1}{\varepsilon})$ samples, likely to hit a bad i)

<u>main observation</u>:
"good" elements form increasing subsequence

<u>Proof</u> if $i<j$ both good, let k be least common ancestor in bin search tree.



When hit x_k, search for x_i went left + search for x_j went right.

so $x_i < x_k < x_j$ ∎

Monotonicity over Posets :

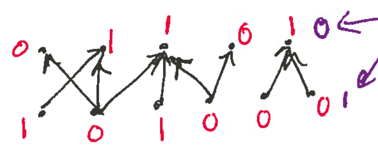
def. f is monotone over poset P if

$\forall\ x \prec y$

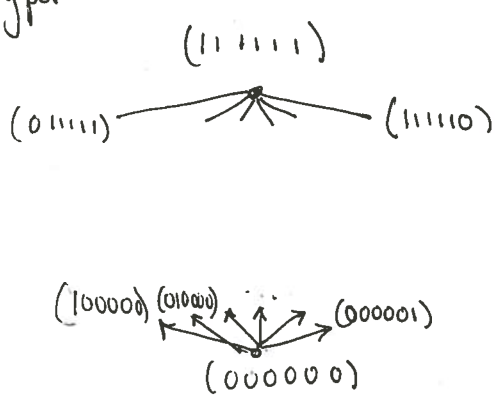
then $f(x) \le f(y)$

examples: Can represent via dags

- bipartite posets



- hypercube



top level : all 1's
level 5 : five 1's
⋮
level i : i 1's
⋮

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

If can test monotonicity, can also test:

1) Given 2CNF φ along with assignment $A = \{a_1, \dots a_n\}$ $a_i \in \{T, F\}$

- Pass if $\varphi(A) = T$
- Fail if $\forall A'$ s.t. A ε-close to A' $\varphi(A) = F$

whp

2) Given G with $U \subseteq V$

- Pass if U is VC
- Fail if $\forall U'$ s.t. U ε-close to U', U' not VC

\# nodes in $U' \Delta U$

3) Given G with $U \subseteq V$

- Pass if U is clique
- Fail if $\forall U'$ s.t. U' ε-close to U, U' not clique

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

Pf. h.w.

Thm ε-mon test requires $n^{\Omega(1)}$ queries ~~if~~ nonadaptive

$\xRightarrow{h.w.}$ $\Omega(\log n)$ queries adaptive

open problem: Can we improve this to $\Omega(\sqrt{n})$? for adaptive queries ??

What about grids?

$f : [n] \times [n] \to [m]$

Can test monotonicity in $O(\log^2 n)$ time

actually $O(\frac{1}{\varepsilon} \log n \log m)$

$f : [n]^d \to [m]$

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

$f : 2^d \to \{0,1\}$

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

need $\Omega(d^{1/4})$ queries (even for adaptive algorithms!)