Sublinear Algorithms
Lecture 1

Sofya Raskhodnikova
Boston University

Organizational

Course webpage:
https://cs-people.bu.edu/sofya/sublinear-course/

Use Piazza to ask questions
Office hours:
Wednesdays, 1:30PM-3:00PM

Evaluation

- Homework (about 4 assignments)
- Taking lecture notes (once or twice per person)
- Course project and presentation
- Peer grading (PhD students only)
- Class participation

Tentative Topics

Introduction, examples and general techniques.

Sublinear-time algorithms for

- graphs
- strings
- geometric properties of images
- basic properties of functions
- algebraic properties and codes
- metric spaces
- distributions

Tools: probability, Fourier analysis, combinatorics, codes, …

Sublinear-space algorithms: streaming

Tentative Plan

Introduction, examples and general techniques.

Lecture 1. Background. Testing properties of images and lists.

Lecture 2. Properties of functions and graphs. Sublinear approximation.

Lecture 3-5. Background in probability. Techniques for proving hardness. Other models for sublinear computation.

Motivation for Sublinear-Time Algorithms

Massive datasets

- world-wide web
- online social networks
- genome data
- sales logs
- census data
- high-resolution images
- scientific measurements

"Why Gramma, what big data you have!"

Long access time

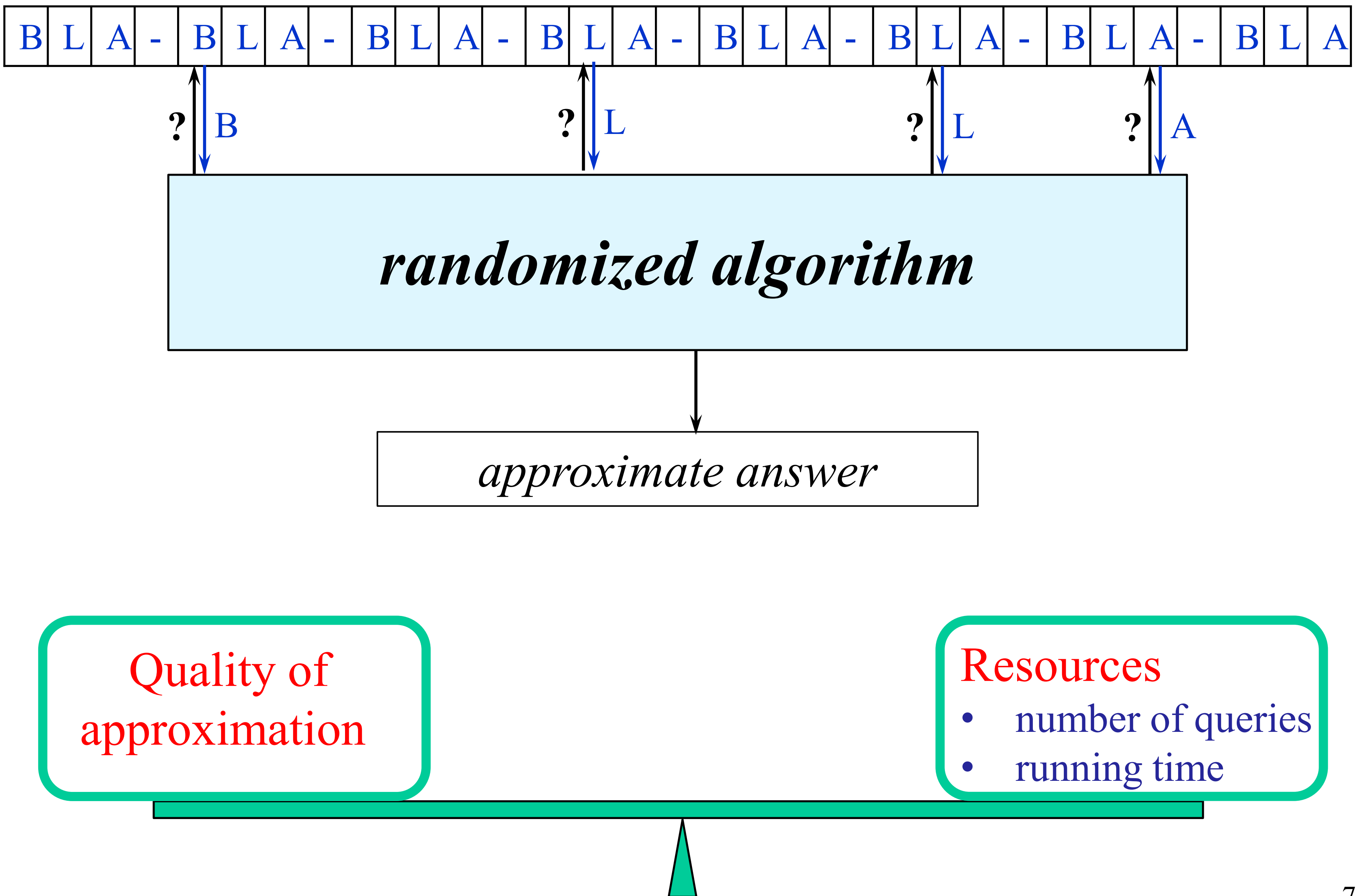
- communication bottleneck (slow connection)
- implicit data (an experiment per data point)

Do We Have To Read All the Data?

- What can an algorithm compute if it
  - reads only a **tiny** portion of the data?
  - runs in **sublinear** time?

Image source: http://apandre.wordpress.com/2011/01/16/bigdata/

A Sublinear-Time Algorithm



Goal: Fundamental Understanding of Sublinear Computation

- What computational tasks?
- How to measure quality of approximation?
- What type of access to the input?
- Can we make our computations robust (e.g., to noise or erased data)?

Types of Approximation

Classical approximation

- need to compute a value
  - output should be close to the desired value
  - example: average

Property testing

- need to answer YES or NO
  - Intuition: only require correct answers on two sets of instances that are very different from each other

Classical Approximation

A Simple Example

Approximate Diameter of a Point Set [Indyk]

Input: m points, described by a distance matrix D

- D_{ij} is the distance between points i and j
- D satisfies triangle inequality and symmetry

(Note: input size is $n = m^2$)

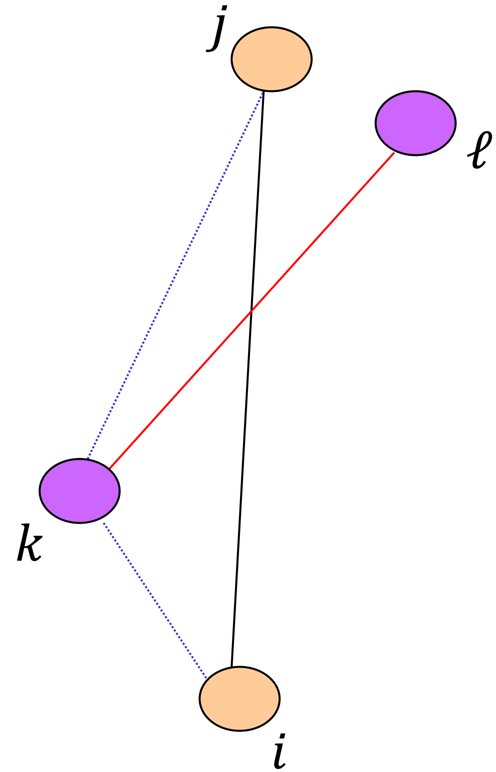
- Let i, j be indices that maximize D_{ij}.
- Maximum D_{ij} is the *diameter*.

Output: (k, ℓ) such that $D_{k\ell} \geq D_{ij}/2$

Algorithm and Analysis

Algorithm (m, D)

1. Pick k arbitrarily
2. Pick ℓ to maximize $D_{k\ell}$
3. Output (k, ℓ)



- Approximation guarantee

$$
\begin{aligned}
D_{ij} &\leq D_{ik} + D_{kj} && \text{(triangle inequality)} \\
&\leq D_{k\ell} + D_{k\ell} && \text{(choice of } \ell \text{ + symmetry of } D\text{)} \\
&\leq 2D_{k\ell}
\end{aligned}
$$

- Running time: $O(m) \;=\; O(m = \sqrt{n})$

A rare example of a deterministic sublinear-time algorithm

Property Testing

Property Testing: YES/NO Questions

Does the input satisfy some property? (YES/NO)

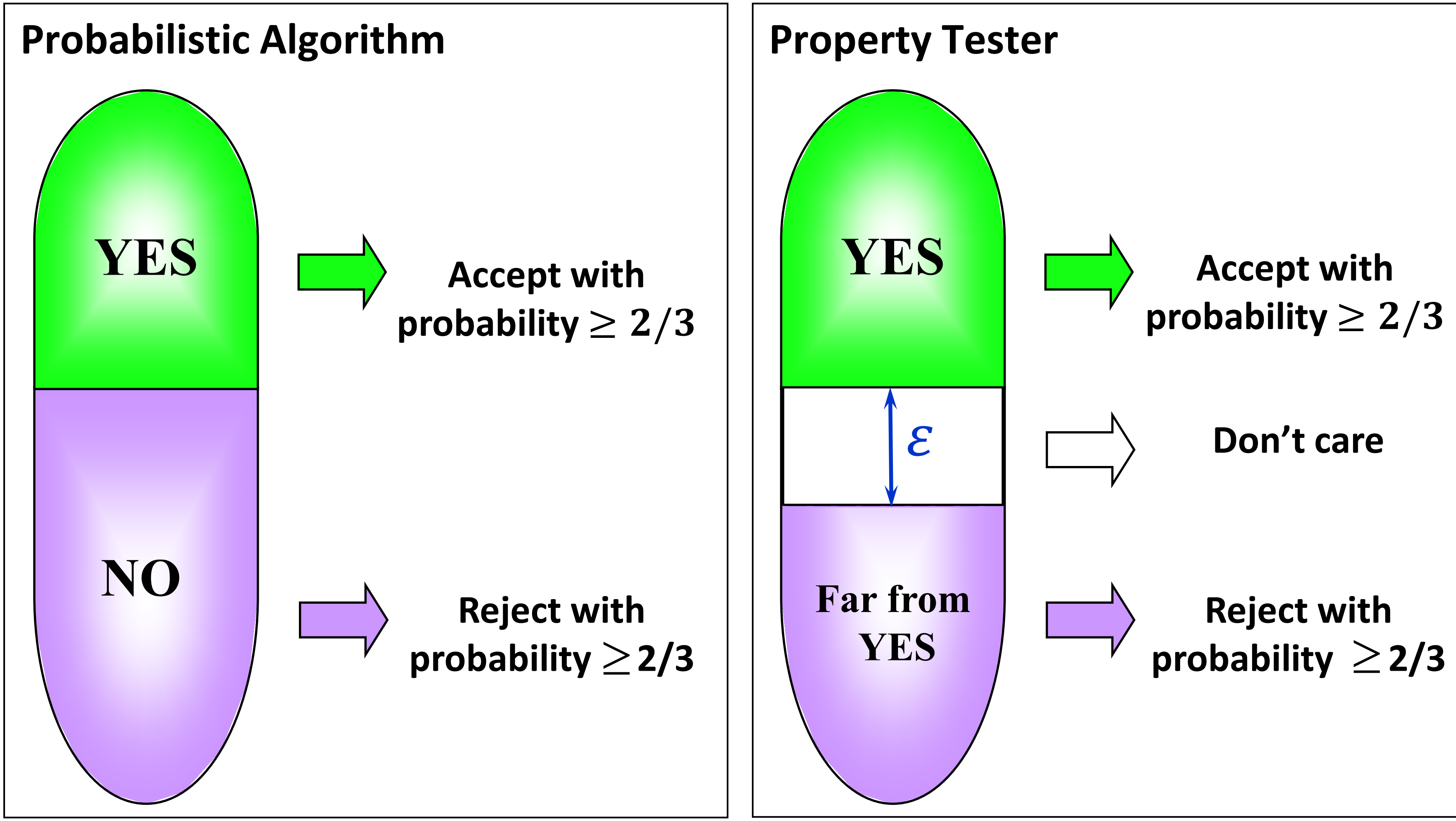
"in the ballpark" vs. "out of the ballpark"

Does the input satisfy the property or is it far from satisfying it?

- **for some applications, it is the right question (probabilistically checkable proofs (PCPs), precursor to learning)**
- **good enough when the data is constantly changing**
- **fast sanity check to rule out inappropriate inputs**

(rejection-based image processing)

Property Tester Definition



ε- far = differs in many places (≥ ε fraction of places)

Randomized Sublinear Algorithms

Toy Examples

Property Testing: a Toy Example

0	0	0	1	…	0	1	0	0

Input: a string $w \in \{0,1\}^n$

Question: Is $w = 00 \dots 0$?

Requires reading entire input.

Approximate version: Is $w = 00 \dots 0$ or does it have $\geq \varepsilon n$ 1's ("errors")?

Test (n, w)

1. Sample $s = 2/\varepsilon$ positions uniformly and independently at random
2. If 1 is found, **reject**; otherwise, **accept**

Analysis: If $w = 00 \dots 0$, it is always accepted.

Used: $1 - x \leq e^{-x}$

If w is ε-far, Pr[error] = Pr[no 1's in the sample]$\leq (1-\varepsilon)^s \leq e^{-\varepsilon s} = e^{-2} < \frac{1}{3}$

Witness Lemma

If a test catches a witness with probability $\geq p$,
then $s = \frac{2}{p}$ iterations of the test catch a witness with probability $\geq 2/3$.

Randomized Approximation: a Toy Example

Input: a string $w \in \{0,1\}^n$

0	0	0	1	…	0	1	0	0

Goal: Estimate the fraction of 1’s in w (like in polls)

It suffices to sample $s = 1 / \varepsilon^2$ positions and output the average
to get the fraction of 1’s $\pm\varepsilon$ (i.e., additive error ε) with probability ¸ 2/3

Hoeffding Bound

Let $\mathrm{Y}_1, \dots, \mathrm{Y}_s$ be independently distributed random variables in [0,1].

Let $\mathrm{Y} = \frac{1}{s} \cdot \sum_{i=1}^{s} \mathrm{Y_i}$ (called *sample mean*). Then $\Pr[|\mathrm{Y} - \mathrm{E}[\mathrm{Y}]| \geq \varepsilon] \leq 2\mathrm{e}^{-2s\varepsilon^2}$.

$\mathrm{Y_i}$ = value of sample i. Then $\mathrm{E}[\mathrm{Y}] = \frac{1}{s} \cdot \sum_{i=1}^{s} \mathrm{E}[\mathrm{Y_i}] =$ (fraction of 1’s in w)

$$\Pr[|(\text{sample mean}) - (\text{fraction of 1's in } w)| \geq \varepsilon]$$
$$\leq 2e^{-2s\varepsilon^2} = 2e^{-2} < 1/3$$

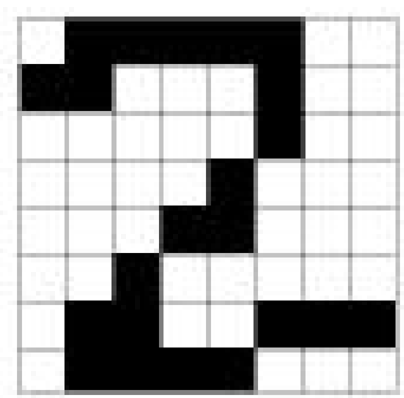
Apply Hoeffding Bound

substitute $s = 1 / \varepsilon^2$

Property Testing

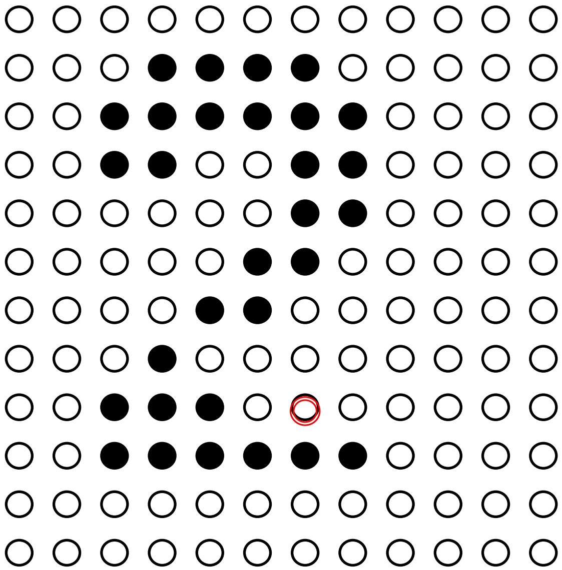
Simple Examples

Testing Properties of Images



Pixel Model

Input: $n \times n$ matrix of pixels
(0/1 values for black-and-white pictures)



Query: point (i_1, i_2)

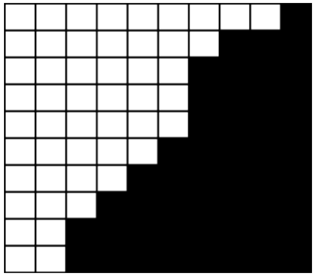
Answer: color of (i_1, i_2)

Testing if an Image is a Half-plane [R03]

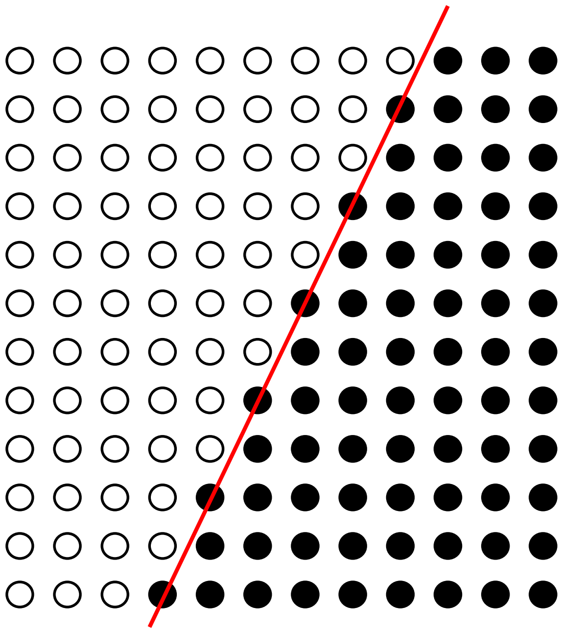
A half-plane or

ε-far from a half-plane?

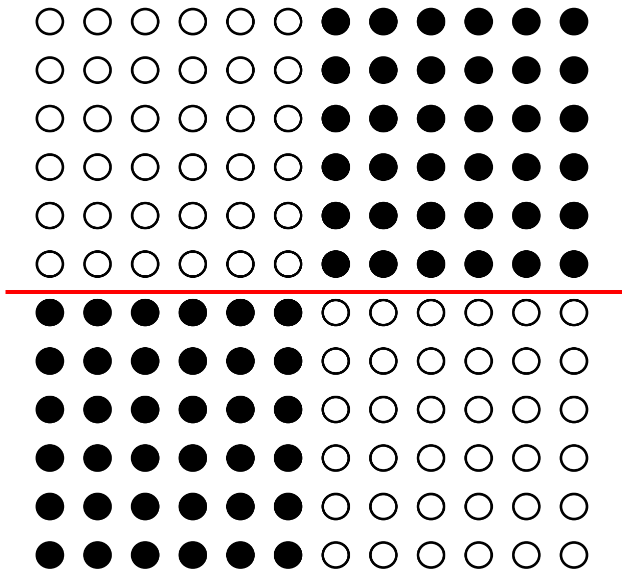
O(1/ε) time



Half-plane Instances

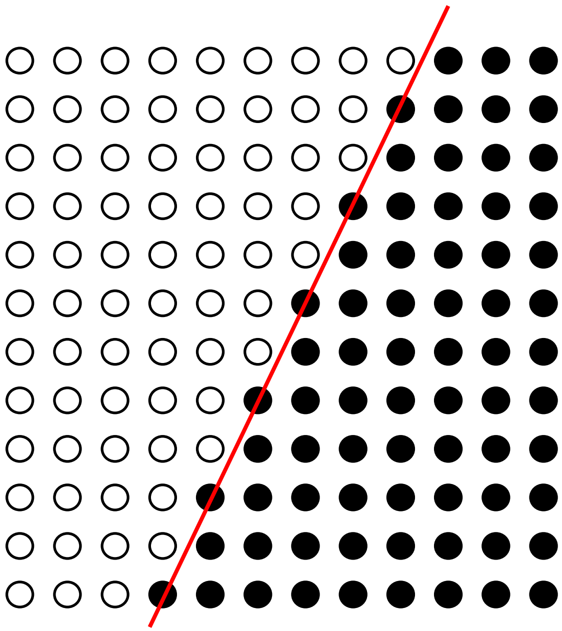


A half-plane

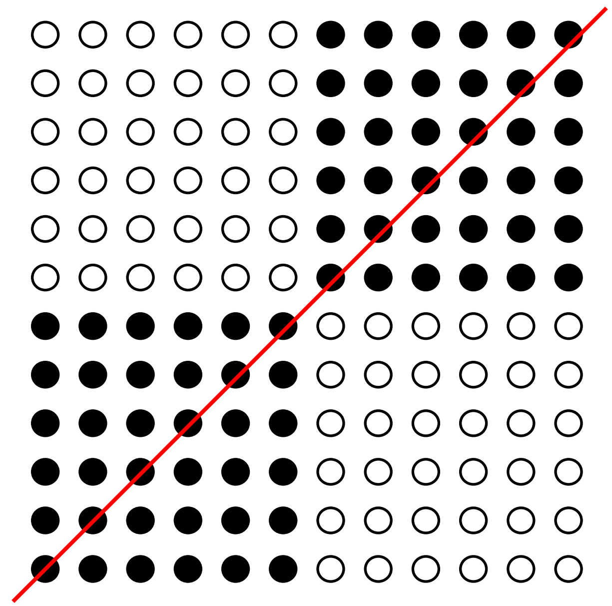


$\frac{1}{4}$-far from a half-plane

Half-plane Instances

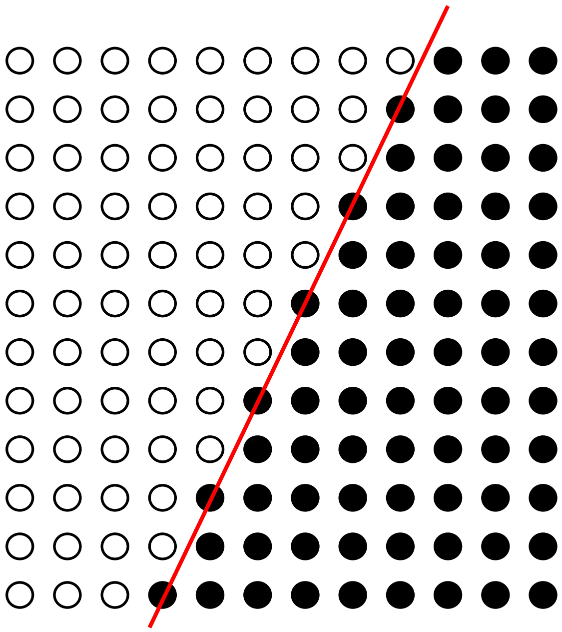


A half-plane

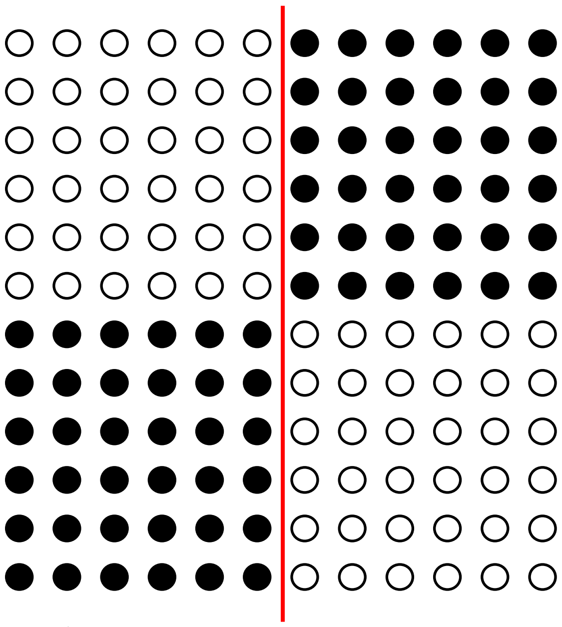


$\frac{1}{4}$-far from a half-plane

Half-plane Instances

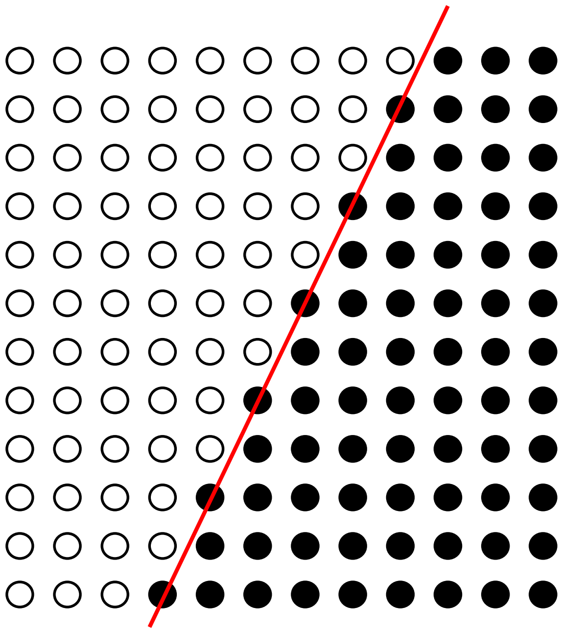


A half-plane

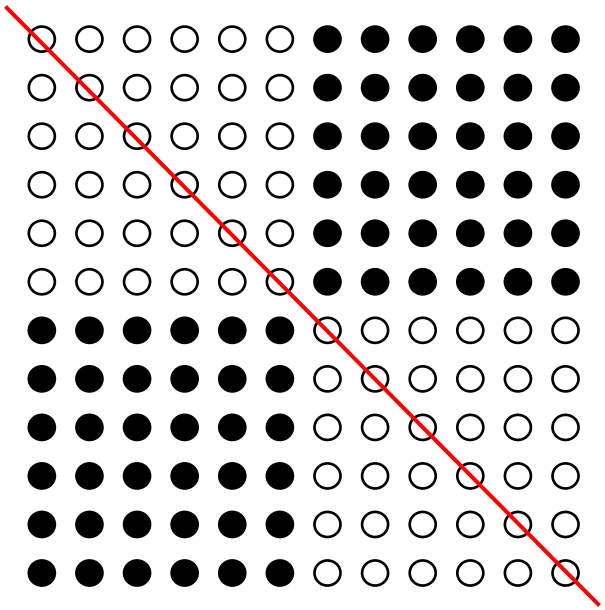


$\frac{1}{4}$-far from a half-plane

Half-plane Instances

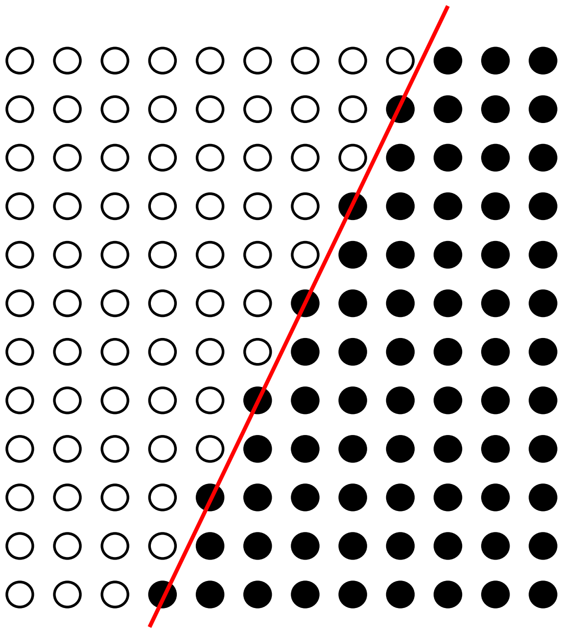


A half-plane

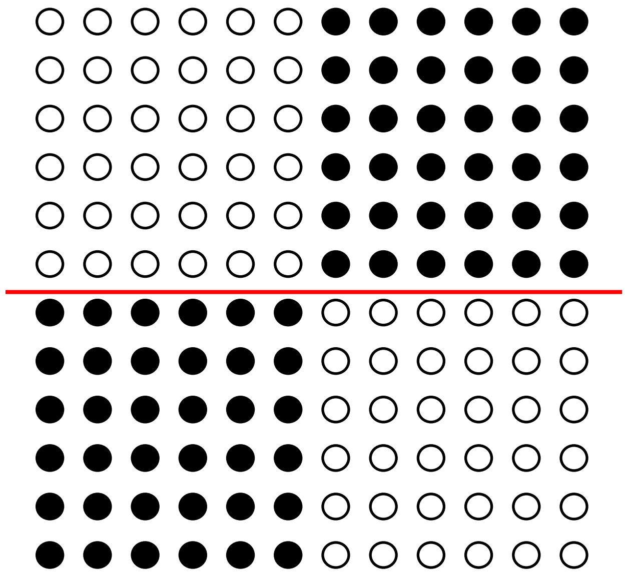


$\frac{1}{4}$-far from a half-plane

Half-plane Instances

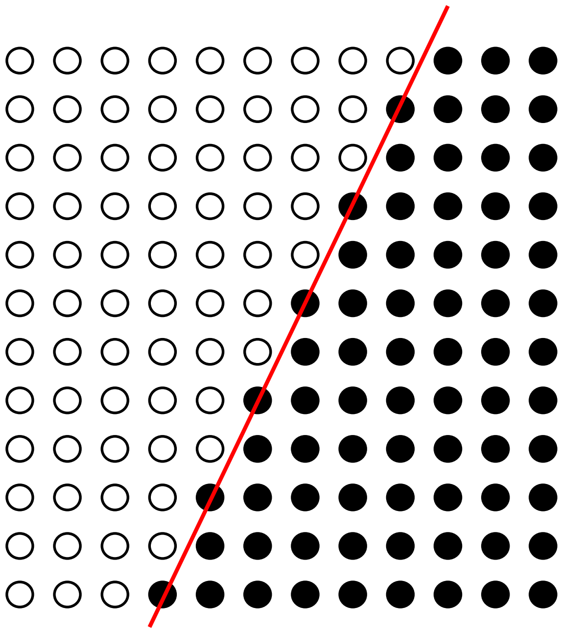


A half-plane

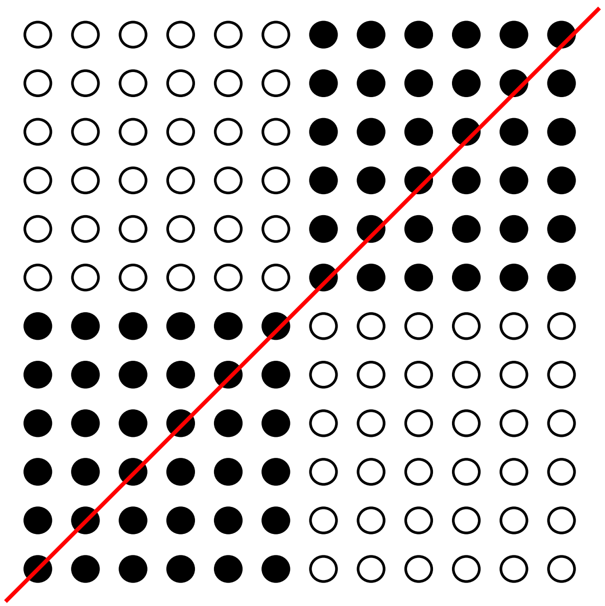


$\frac{1}{4}$-far from a half-plane

Half-plane Instances

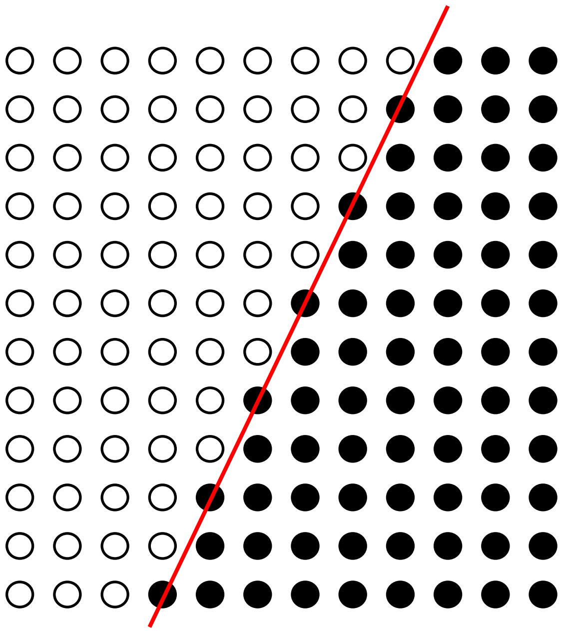


A half-plane

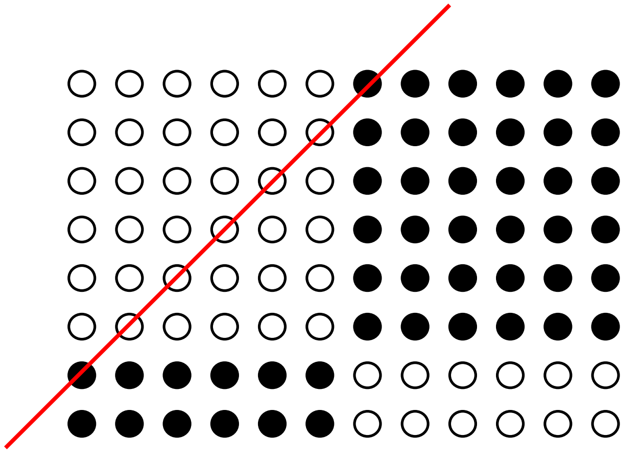


$\frac{1}{4}$-far from a half-plane

Half-plane Instances



A half-plane



$\frac{1}{4}$-far from a half-plane

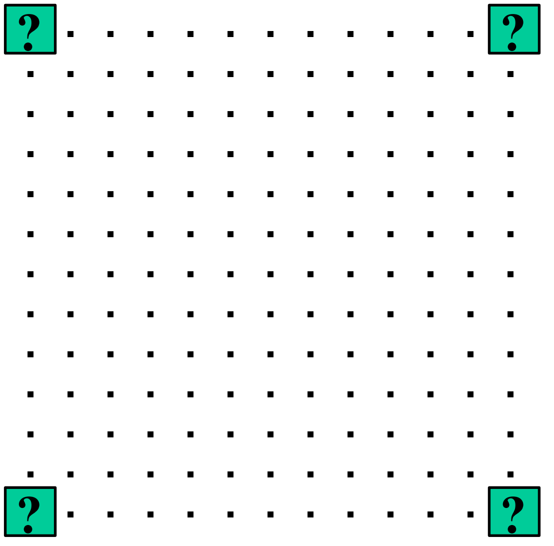
Strategy

"Testing by implicit learning" paradigm

- Learn the outline of the image by querying a few pixels.
- Test if the image conforms to the outline by random sampling, and reject if something is wrong.

Half-plane Test

Claim. The number of sides with different corners is 0, 2, or 4.

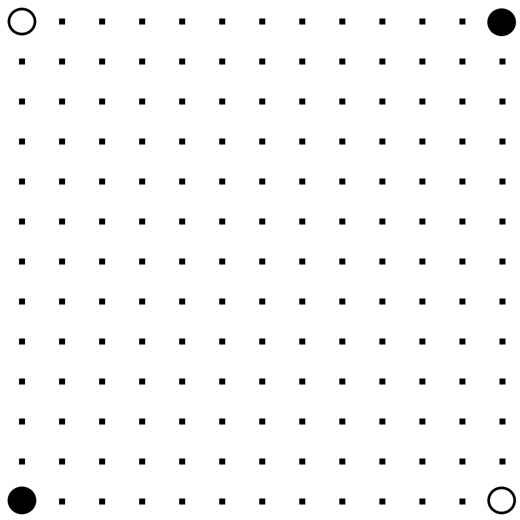


Algorithm

1. Query the corners.

Half-plane Test: 4 Bi-colored Sides

Claim. The number of sides with different corners is 0, 2, or 4.



Analysis

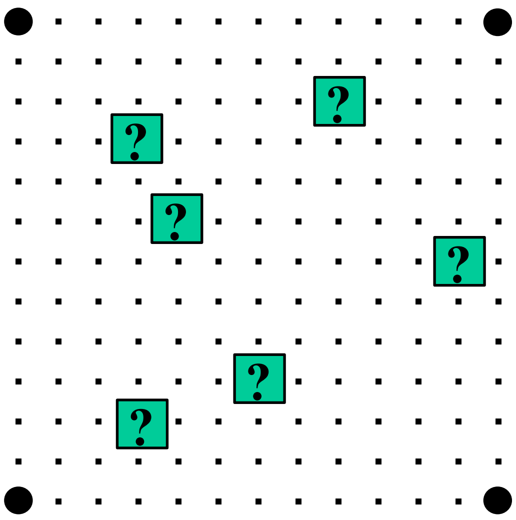
- If it is 4, the image cannot be a half-plane.

Algorithm

1. Query the corners.
2. If the number of sides with different corners is 4, **reject**.

Half-plane Test: 0 Bi-colored Sides

Claim. The number of sides with different corners is 0, 2, or 4.



Analysis

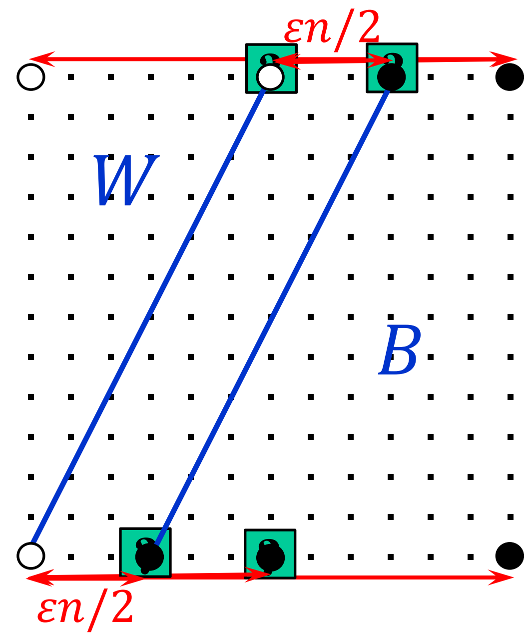
- If all corners have the same color, the image is a half-plane if and only if it is unicolored.

Algorithm

1. Query the corners.
2. If all corners have the same color c, test if all pixels have color c (as in Toy Example 1).

Half-plane Test: 2 Bi-colored Sides

Claim. The number of sides with different corners is 0, 2, or 4.



Analysis

- The area outside of $W \cup B$ has $\leq \varepsilon n^2/2$ pixels.
- If the image is a half-plane, W contains only white pixels and B contains only black pixels.
- If the image is ε-far from half-planes, it has $\geq \varepsilon n^2/2$ wrong pixels in $W \cup B$.
- By Witness Lemma, $4/\varepsilon$ samples suffice to catch a wrong pixel.

Algorithm

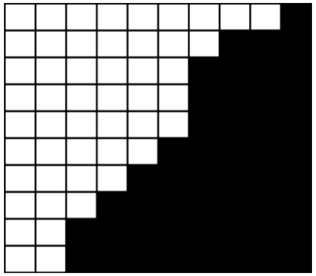
1. Query the corners.
2. If # of sides with different corners is 2, on both sides find 2 different pixels within distance $\varepsilon n/2$ by binary search.
3. Query $4/\varepsilon$ pixels from $W \cup B$
4. **Accept** iff all W pixels are white and all B pixels are black.

Testing if an Image is a Half-plane [R03]

A half-plane or

ε-far from a half-plane?

O(1/ε) time ✓



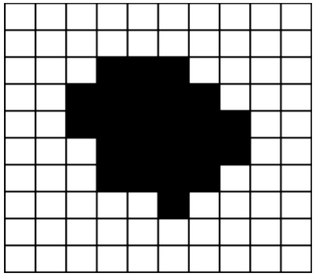
Other Results on Testing Properties of Images

- Pixel Model

Convexity [Berman Murzabulatov R 18]

Convex or ε-far from convex?

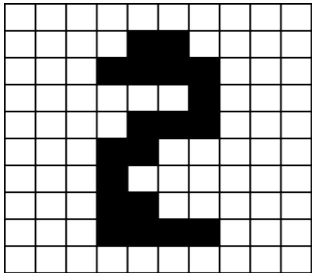
$O(1/\varepsilon)$ time



Connectedness [Berman Murzabulatov R Ristache 24]

Connected or ε-far from connected?

$O(1/\varepsilon^{3/2}\sqrt{\log 1/\varepsilon})$ time

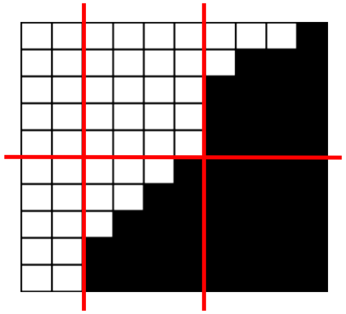


Partitioning [Kleiner Keren Newman 10]

Can be partitioned according to a template

or is ε-far?

time independent of image size



- Properties of sparse images [Ron Tsur 10]

Testing if a List is Sorted

Input: a list of n numbers $x_1, x_2, \ldots, x_n$

- Question: Is the list sorted?

  Requires reading entire list: $\Omega(n)$ time

- Approximate version: Is the list sorted or ε-far from sorted?

  (An ε fraction of x_i's must be changed to make it sorted.)

  [Ergün Kannan Kumar Rubinfeld Viswanathan 98, Fischer 01]: O((log n)/ ε) time

  Ω(log n) queries

- Best known bounds:

Θ(log $(\varepsilon n)/\varepsilon$) time

[Belovs, Chakrabarty Dixit Jha Seshadhri 15]

Testing Sortedness: Attempts

1. **Test**: Pick a uniformly random $i \in \{1, \ldots, n-1\}$ and reject if $x_i > x_{i+1}$.

Fails on:

1	1	1	1	0	0	0	0

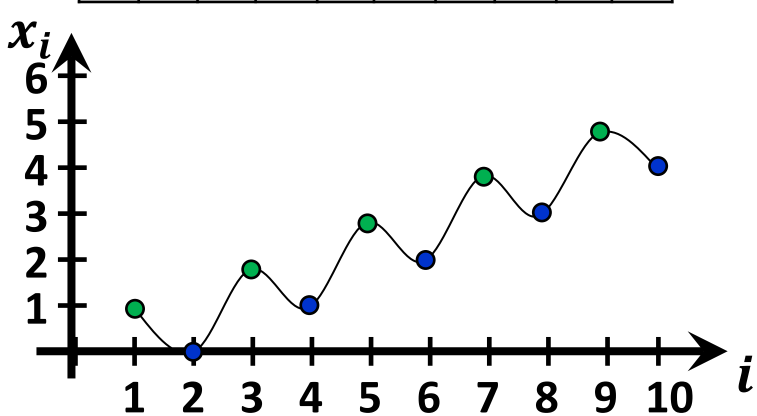
Ã 1/2-far from sorted

2. **Test**: Pick uniformly random $i < j$ in $\{1, \ldots, n\}$ and reject if $x_i > x_j$.

Fails on:

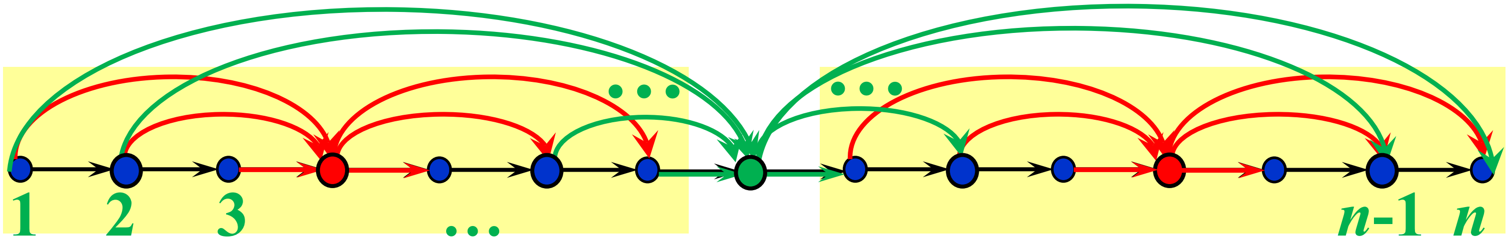
1	0	2	1	3	2	4	3	5	4

Ã 1/2-far from sorted



Is a List Sorted or ε-far from Sorted?

Idea: Associate positions in the list with vertices of the directed line.



Construct a graph (2-spanner)

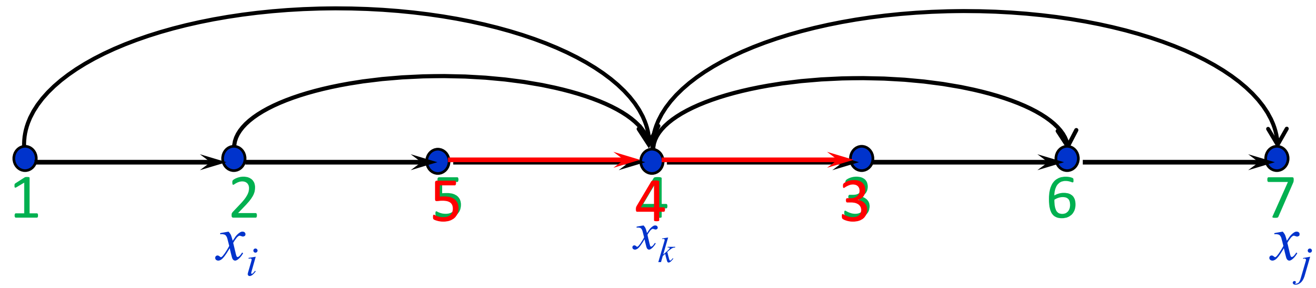
$\leq n \log n$ edges

- by adding a few "shortcut" edges (i, j) for $i < j$
- where each pair of vertices is connected by a path of length at most 2 ✓

Is a List Sorted or ε-far from Sorted?

Test [Dodis Goldreich Lehman Raskhodnikova Ron Samorodnitsky 99]

Pick a random edge (i, j) from the 2-spanner and **reject** if $x_i > x_j$.



Analysis:

- Call an edge (i, j) **violated** if $x_i > x_j$, and **satisfied** otherwise.
- If i is an endpoint of a **violated** edge, call x_i **bad**. Otherwise, call it **good**.

Claim 1. All **good** numbers x_i are sorted.

Proof: Consider any two good numbers, x_i and x_j.

They are connected by a path of (at most) two **satisfied** edges $(i, k), (k, j)$

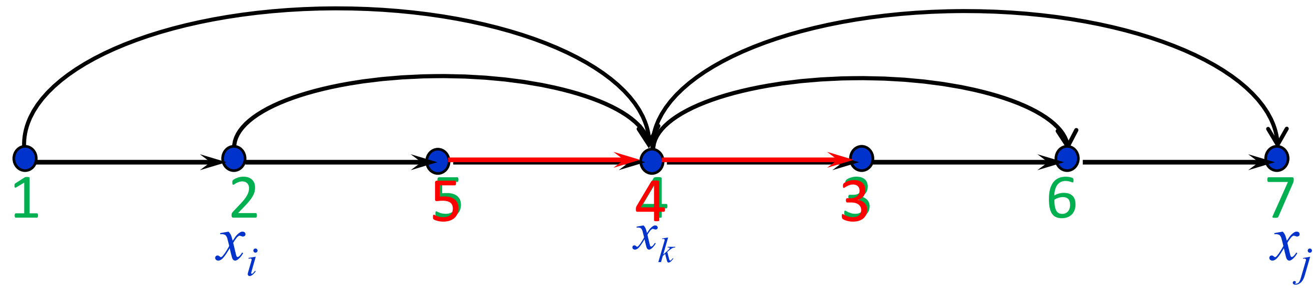
$\Rightarrow x_i \leq x_k$ and $x_k \leq x_j$

$$\Rightarrow x_i \leq x_j$$

Is a List Sorted or ε-far from Sorted?

Test [Dodis Goldreich Lehman Raskhodnikova Ron Samorodnitsky 99]

Pick a random edge (i, j) from the 2-spanner and **reject** if $x_i > x_j$.



Analysis:

- Call an edge (i, j) **violated** if $x_i > x_j$, and **satisfied** otherwise.
- If i is an endpoint of a **violated** edge, call x_i **bad**. Otherwise, call it **good**.

Claim 1. All **good** numbers x_i are sorted.

Claim 2. An ε-far list **violates** $\geq \varepsilon/(2\log n)$ fraction of edges in 2-spanner.

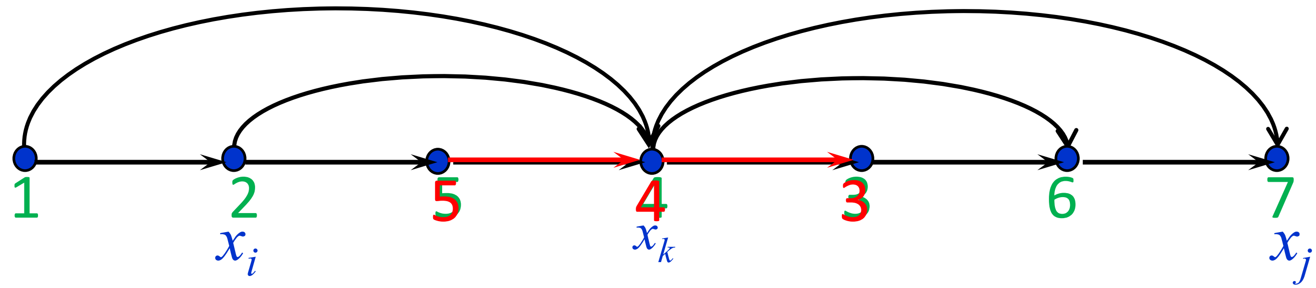
Proof: If a list is ε-far from sorted, it has $\geq \varepsilon n$ **bad** numbers. (Claim 1)

- Each **violated** edge contributes 2 **bad** numbers.
- 2-spanner has $\geq \frac{\varepsilon n}{2}$ **violated** edges out of $n \log n$.

Is a List Sorted or ε-far from Sorted?

Test [Dodis Goldreich Lehman Raskhodnikova Ron Samorodnitsky 99]

Pick a random edge (i, j) from the 2-spanner and **reject** if $x_i > x_j$.



Analysis:

- Call an edge (i, j) **violated** if $x_i > x_j$, and **satisfied** otherwise.

Claim 2. An ε-far list **violates** $\geq \varepsilon/(2 \log n)$ fraction of edges in 2-spanner.

By Witness Lemma, it suffices to sample $(4 \log n)/\varepsilon$ edges from 2-spanner.

Algorithm

Sample $\frac{4 \log n}{\varepsilon}$ edges (i, j) from the 2-spanner and **reject** if $x_i > x_j$.

Guarantee: All sorted lists are accepted. ✓

All lists that are ²-far from sorted are rejected with probability ¸ 2/3. ✓

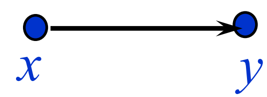
Time: O((log n)/ ε) ✓

Generalization

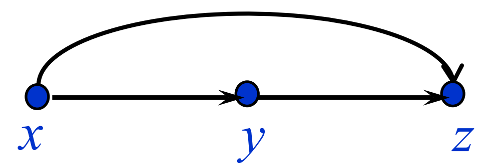
Observation:

The same test/analysis apply to any edge-transitive property of a list of numbers that allows extension.

- A property is edge-transitive if
  1) it can be expressed in terms conditions on ordered pairs of numbers



  2) it is transitive: whenever (x, y) and (y, z) satisfy (1), so does (x, z)

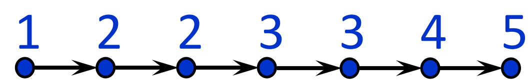


- A property allows extension if
  3) any function that satisfies (1) on a subset of the numbers can be extended to a function with the property

Testing if a Function is Lipschitz [Jha R]

A function $f : D \to R$ is Lipschitz if it has Lipschitz constant 1:
that is, if for all x,y in D,
$distance_R(f(x),f(y)) \le distance_D(x,y)$.

Consider $f : \{1,\ldots,n\} \to \mathbb{R}$:



nodes = points in the domain; edges = points at distance 1

node labels = values of the function

The Lipschitz property is *edge-transitive*:

1. a pair (x,y) is *good* if $|f(y)-f(x)| \le |y-x|$
2. (x,y) and (y,z) are *good*) (x,z) is *good*

It also allows extension for the range $\mathbb{R}$.

Testing if a function $f : \{1,\ldots,n\} \to \mathbb{R}$ is Lipschitz takes $O((\log n)/^2)$ time.

Properties of a List of n Numbers

- Sorted or ε-far from sorted?
- Lipschitz (does not change too drastically)
  or ε-far from satisfying the Lipschitz property?

$$O\left(\frac{\log n}{\varepsilon}\right) \text{ time}$$

✓

Tight bound: $\Theta\left(\frac{\log(\varepsilon n)}{\varepsilon}\right)$ [Chakrabarty Dixit Jha Seshadhri 15, Belovs 18]