Sublinear Algorithms

LECTURE 2

Last time

- Introduction
- Basic models for sublinear-time computation
- Simple examples of sublinear algorithms

Today

- Properties of lists and functions.
- Testing if a list is sorted/Lipschitz
  and if a function is monotone.

Sofya Raskhodnikova;Boston University

Reminders

HW1 is due Thursday at 10am
It is posted on the course webpage:
https://cs-people.bu.edu/sofya/sublinear-course/

Use Piazza for questions and discussions

Office hours (on zoom):
Wednesdays, 1:00PM-2:30PM

Testing if a List is Sorted

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

- 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 have to be changed to make it sorted.)

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

  $\Omega(\log n)$ queries

- Best known bounds:

$$\Theta(\log(\varepsilon n)/\varepsilon) \text{ time}$$

[Belovs, Chakrabarty Dixit Jha Seshadhri 15]

Testing Sortedness: Attempts

1. **Test**: Pick a random i and reject if $x_i > x_{i+1}$

Fails on:

1	1	1	1	0	0	0	0

← 1/2-far from sorted

2. **Test**: Pick random $i < j$ and reject if $x_i > x_j$

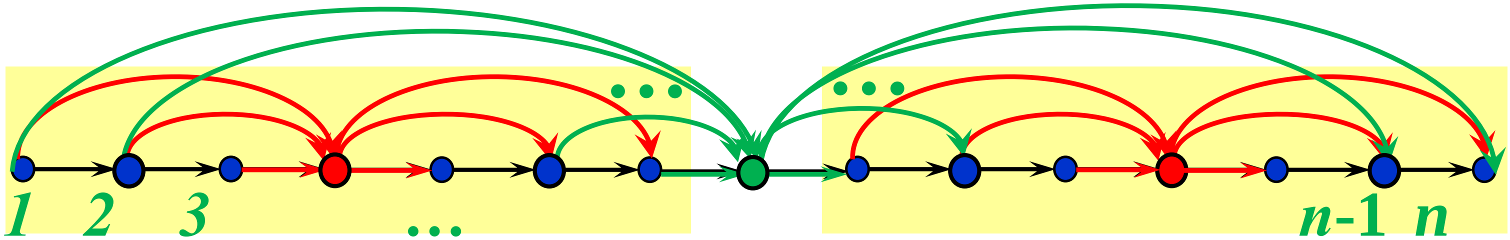
Fails on:

1	0	2	1	3	2	4	3	5	4

← 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)

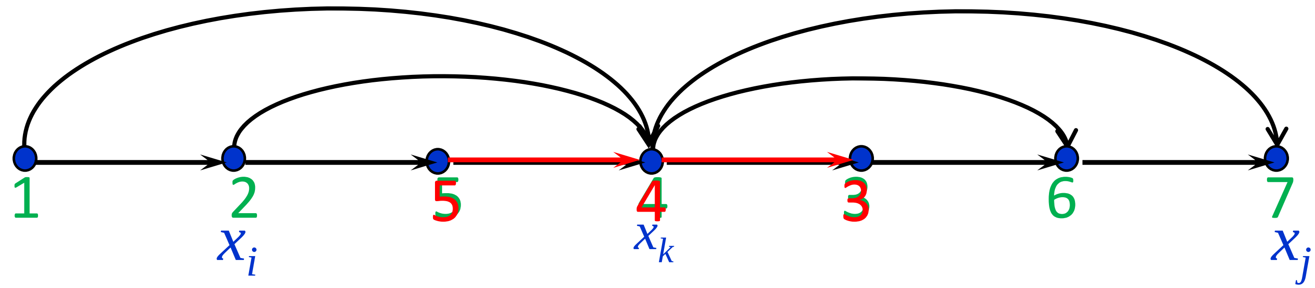
≤ *n* log *n* edges

- 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 (x_i,x_j) from the 2-spanner and **reject** if $x_i > x_j$.



Analysis:

- Call an edge (x_i,x_j) **violated** if $x_i > x_j$, and **good** otherwise.
- If x_i is an endpoint of a **violated** edge, call it **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 **good** edges (x_i,x_k), (x_k,x_j).

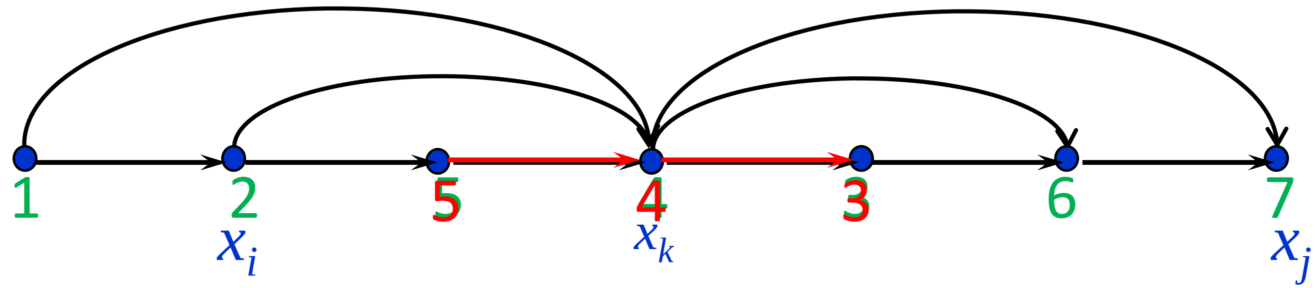
$\Rightarrow$ $x_i \leq x_k$ and $x_k \leq x_j$

$\Rightarrow$ $x_i \leq x_j$

Is a list sorted or ϵ-far from sorted?

Test [Dodis Goldreich Lehman Raskhodnikova Ron Samorodnitsky 99]

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



Analysis:

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

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

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

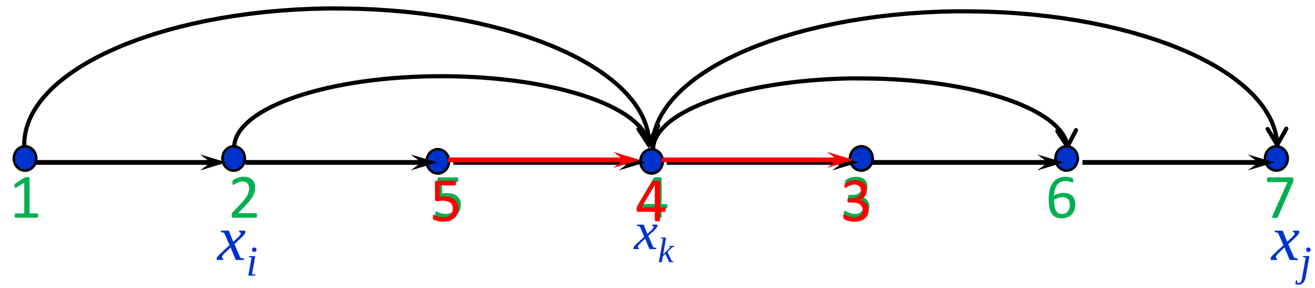
Proof: If a list is ϵ-far from sorted, it has $\geq \epsilon n$ **bad** numbers. (Claim 1)

- Each **violated** edge contributes 2 **bad** numbers.
- 2-spanner has $\geq \epsilon n/2$ **violated** edges out of $\leq n \log n$.

Is a list sorted or ϵ-far from sorted?

Test [Dodis Goldreich Lehman Raskhodnikova Ron Samorodnitsky 99]

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



Analysis:

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

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

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

Algorithm

Sample $(4 \log n)/\epsilon$ edges (x_i, x_j) from the 2-spanner and **reject** if $x_i > x_j$.

Guarantee: All sorted lists are accepted. ✓

All lists that are ϵ-far from sorted are rejected with probability $\geq 2/3$. ✓

Time: $O((\log n)/\epsilon)$ ✓

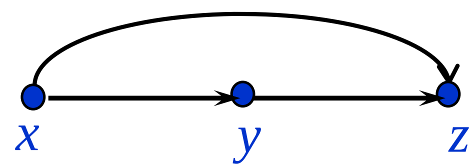
Generalization

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)

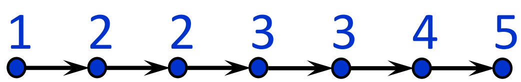


- A property allows extension if
  3) any function that satisfies (1) on a subset of the numbers can be extended to a function with the property

Testing if a Function is Lipschitz [Jha R]

A function $f : D \rightarrow R$ 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 $f : \{1,\ldots,n\} \rightarrow \mathbb{R}$:



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* $\Rightarrow$ (x,z) is *good*

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

ting if a function $f : \{1,\ldots,n\} \rightarrow \mathbb{R}$ is Lipschitz takes $O((\log n)/\epsilon)$ time.

Does the spanner-based test apply if the range is $\mathbb{R}^2$ with Euclidean distances? $\mathbb{Z}^2$ with Euclidean distances?

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(log n/ε) time ✓

This bound is tight (unless ε is really tiny w.r.t. n)

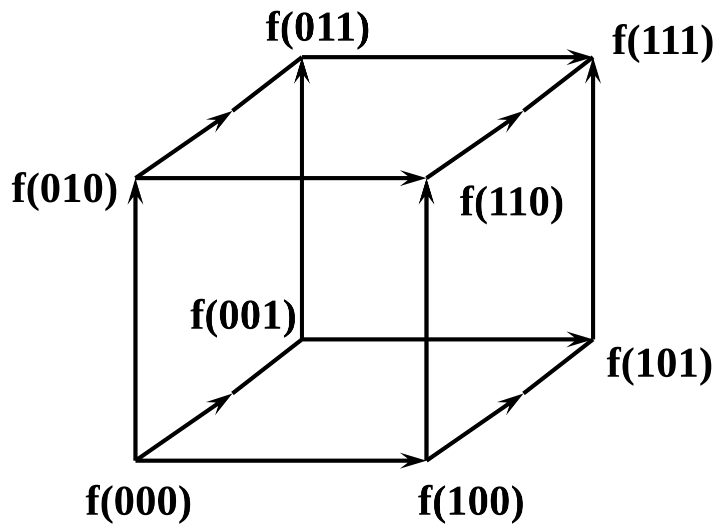
[Chakrabarty Dixit Jha Seshadhri 15]

Basic Properties of Functions

Boolean Functions $f : \{0,1\}^n \to \{0,1\}$

Graph representation:

n-dimensional hypercube



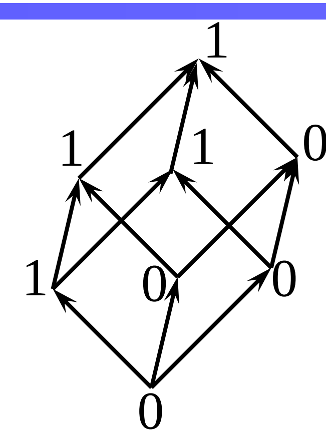
- vertices: bit strings of length n
- edges: (x, y) is an edge if y can be obtained from x by increasing one bit from 0 to 1

x	001001
y	011001

- each vertex x is labeled with $f(x)$

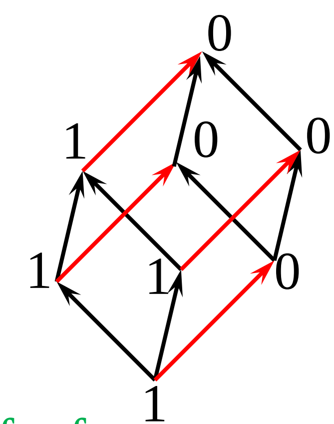
Monotonicity of Functions

[Goldreich Goldwasser Lehman Ron Samorodnitsky,
Dodis Goldreich Lehman Raskhodnikova Ron Samorodnitsky
Fischer Lehman Newman Raskhodnikova Rubinfeld Samorodnitsky]



- A function $f : \{0,1\}^n \rightarrow \{0,1\}$ is monotone if increasing a bit of x does not decrease $f(x)$.

- Is f monotone or ε-far from monotone (f has to change on many points to become monontone)?
  - Edge $x \rightarrow y$ is violated by f if $f(x) > f(y)$.



Time:

- $O(n/\varepsilon)$, logarithmic in the size of the input, 2^n
- $\Omega(\sqrt{n}/\varepsilon)$ for restricted class of tests
- Advanced techniques: $\Theta(\sqrt{n}/\varepsilon^2)$ for nonadaptive tests, $\Omega(\sqrt[3]{n})$

  [Khot Minzer Safra 15, Chen De Servidio Tang 15, Chen Waingarten Xie 17]

Monotonicity Test [GGLRS, DGLRRS]

Idea: Show that functions that are far from monotone violate many edges.

EdgeTest (f, ε)

1. Pick $2n/\varepsilon$ edges (x, y) uniformly at random from the hypercube.
2. **Reject** if some (x, y) is violated (i.e. $f(x) > f(y)$). Otherwise, **accept**.

Analysis

- If f is monotone, EdgeTest always accepts.
- If f is ε-far from monotone, by Witness Lemma, it suffices to show that $\geq \varepsilon/n$ fraction of edges (i.e., $\frac{\varepsilon}{n} \cdot 2^{n-1} n = \varepsilon 2^{n-1}$ edges) are violated by f.
  - Let $V(f)$ denote the number of edges violated by f.

Contrapositive: If $V(f) < \varepsilon\, 2^{n-1}$,

f can be made monotone by changing $< \varepsilon\, 2^n$ values.

Repair Lemma

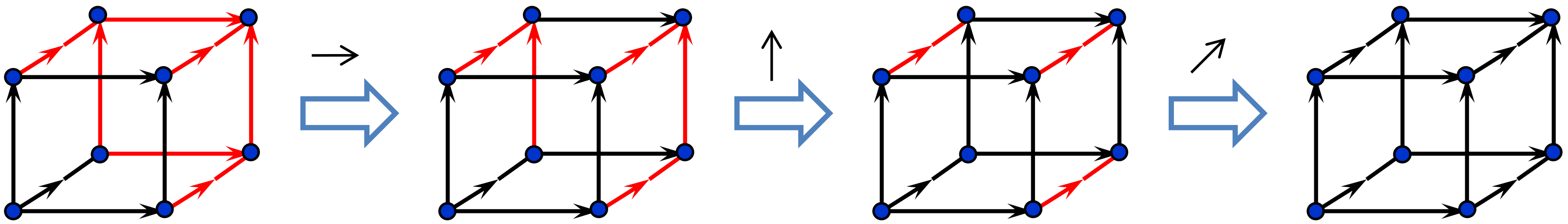
f can be made monotone by changing $\leq 2 \cdot V(f)$ values.

Repair Lemma: Proof Idea

Repair Lemma

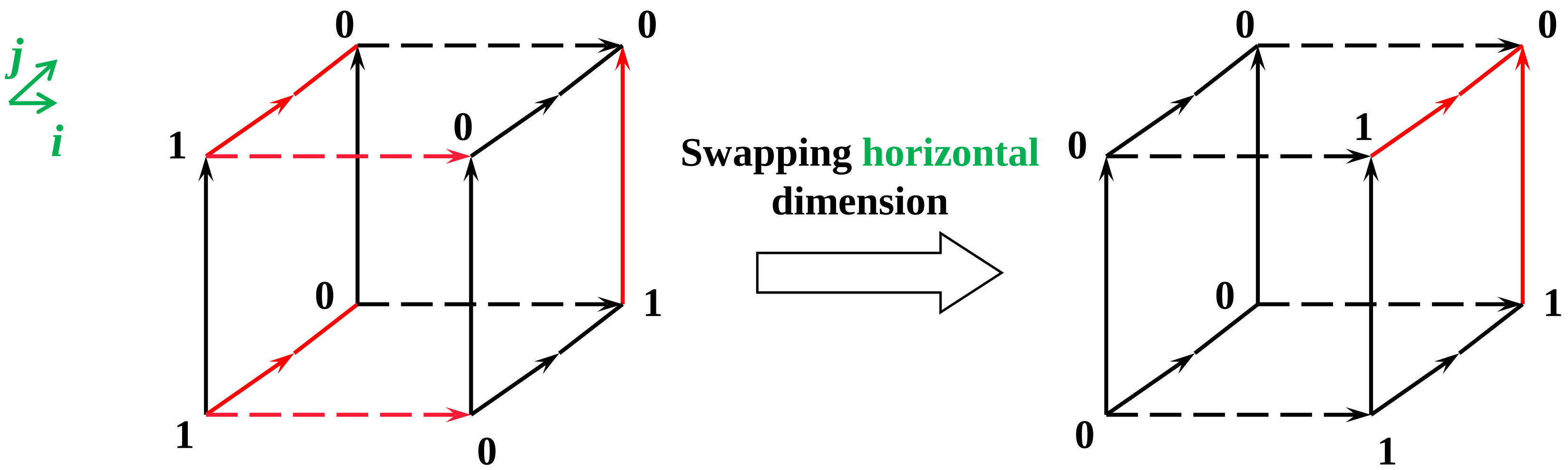
f can be made monotone by changing $\leq 2 \cdot V(f)$ values.

Proof idea: Transform f into a monotone function by repairing edges in one dimension at a time.



Repairing Violated Edges in One Dimension

Swap violated edges $1 \rightarrow 0$ **in one dimension to** $0 \rightarrow 1$**.**



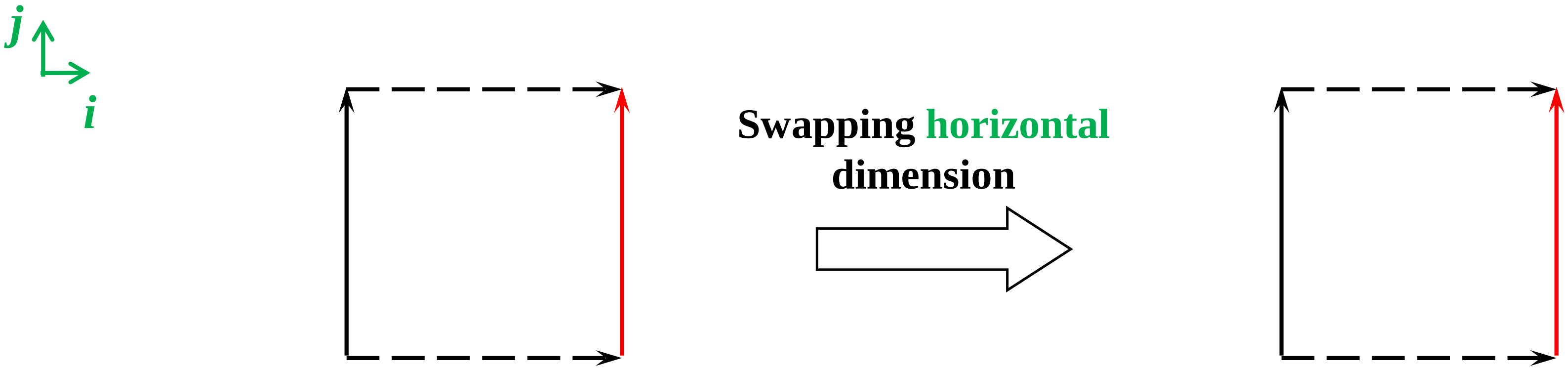
Let V_j = # of violated edges in dimension j

Claim. Swapping in dimension i does not increase V_j for all dimensions $j \neq i$

Enough to prove the claim for squares

Proof of The Claim for Squares

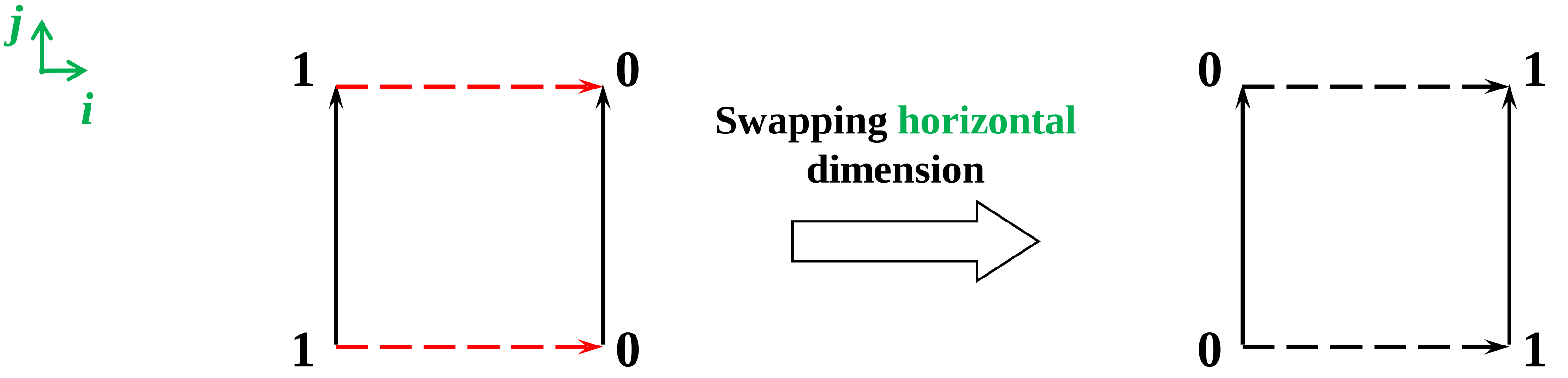
Claim. Swapping in dimension i does not increase V_j for all dimensions $j \neq i$



- If no horizontal edges are violated, no action is taken.

Proof of The Claim for Squares

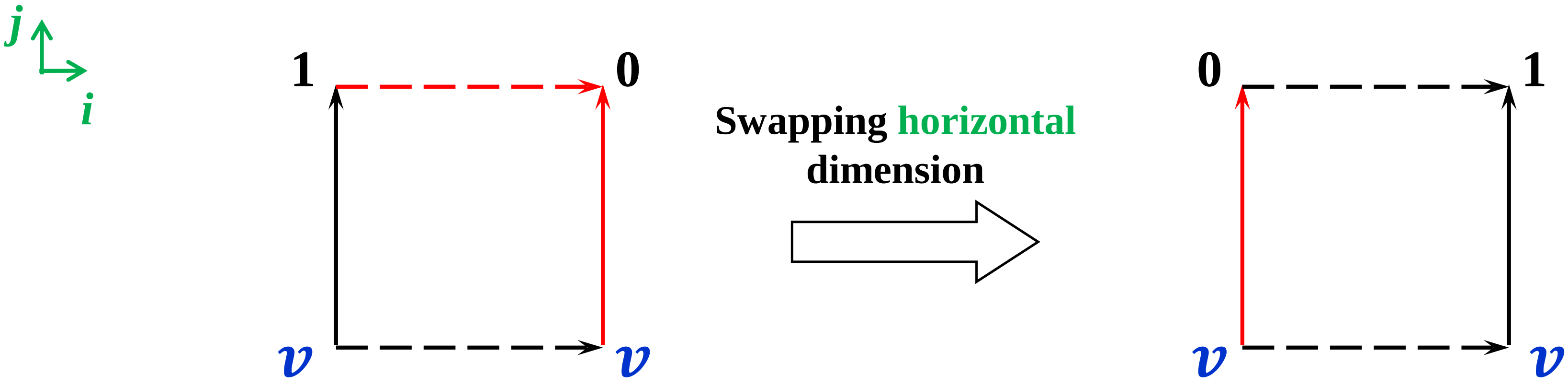
Claim. Swapping in dimension i does not increase V_j for all dimensions $j \neq i$



- If both horizontal edges are violated, both are swapped, so the number of vertical violated edges does not change.

Proof of The Claim for Squares

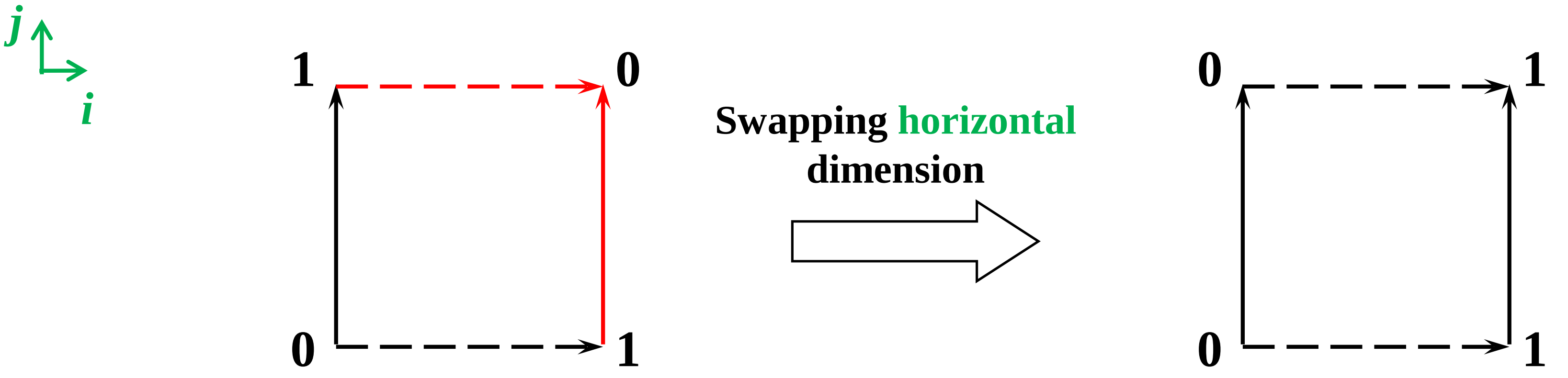
Claim. Swapping in dimension i does not increase V_j for all dimensions $j \neq i$



- Suppose one (say, top) horizontal edge is violated.
- If both bottom vertices have the same label, the vertical edges get swapped.

Proof of The Claim for Squares

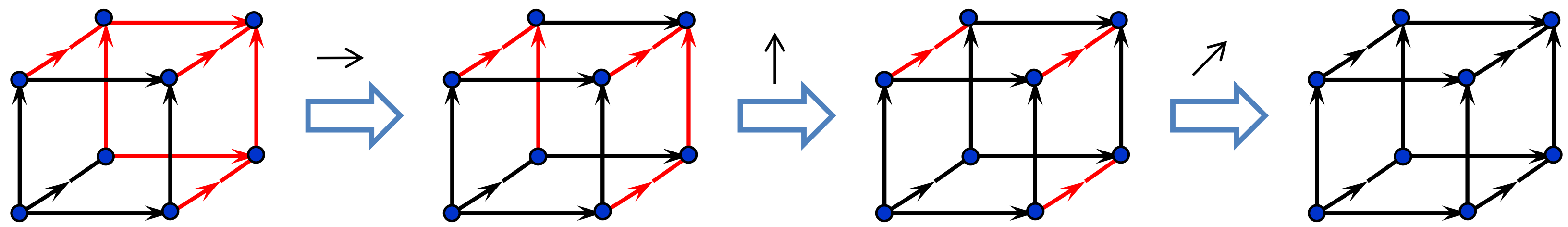
Claim. Swapping in dimension i does not increase V_j for all dimensions $j \neq i$



- Suppose one (say, top) horizontal edge is violated.
- If both bottom vertices have the same label, the vertical edges get swapped.
- Otherwise, the bottom vertices are labeled $0 \rightarrow 1$, and the vertical violation is repaired.

Proof of The Claim for Squares

Claim. Swapping in dimension i does not increase V_j for all dimensions $j \neq i$ ✔



After we perform swaps in all dimensions:

- f becomes monotone
- # of values changed:

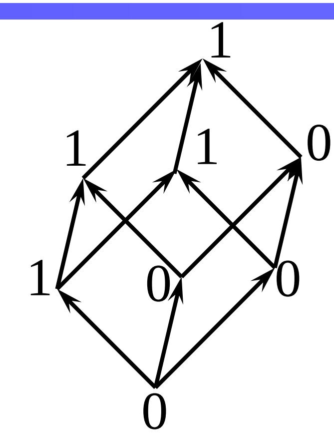
$$2 \cdot V_1 + 2 \cdot (\text{# violated edges in dim 2 after swapping dim 1})$$
$$+ 2 \cdot (\text{# violated edges in dim 3 after swapping dim 1 and 2})$$
$$+ \ldots \leq 2 \cdot V_1 + 2 \cdot V_2 + \cdots 2 \cdot V_n = 2 \cdot V(f)$$

Repair Lemma ✔

f can be made monotone by changing $\leq 2 \cdot V(f)$ values.

Improve the bound by a factor of 2.

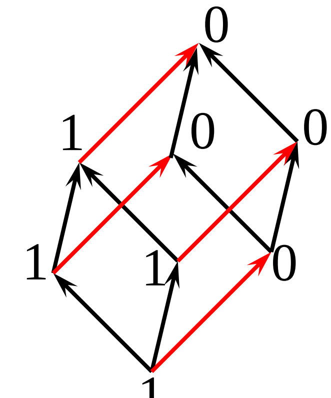
Testing if a Functions $f : \{0,1\}^n \rightarrow \{0,1\}$ *is monotone*



monotone

Monotone or

ε-far from monotone?



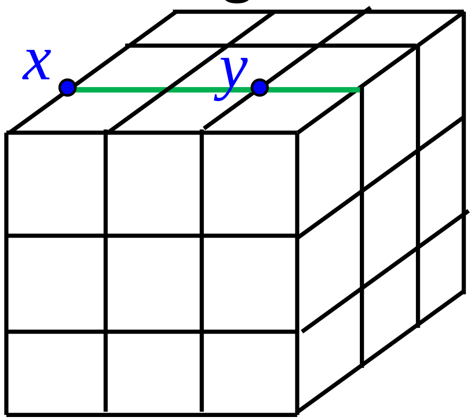
$\frac{1}{2}$-far from monotone

O(n/ε) time ✓

(logarithmic in the size of the input)

Testing Properties of High-Dimensional Functions

In polylogarithmic time, we can test a large class of properties of functions $f: \{1, \ldots, n\}^d \rightarrow \mathbb{R}$, including:



- Lipschitz property [Jha **R**]
- Bounded-derivative properties [Chakrabarty Dixit Jha Seshadhri]
- Unateness [Baleshzar Chakrabarty Pallavoor **R** Seshadhri]