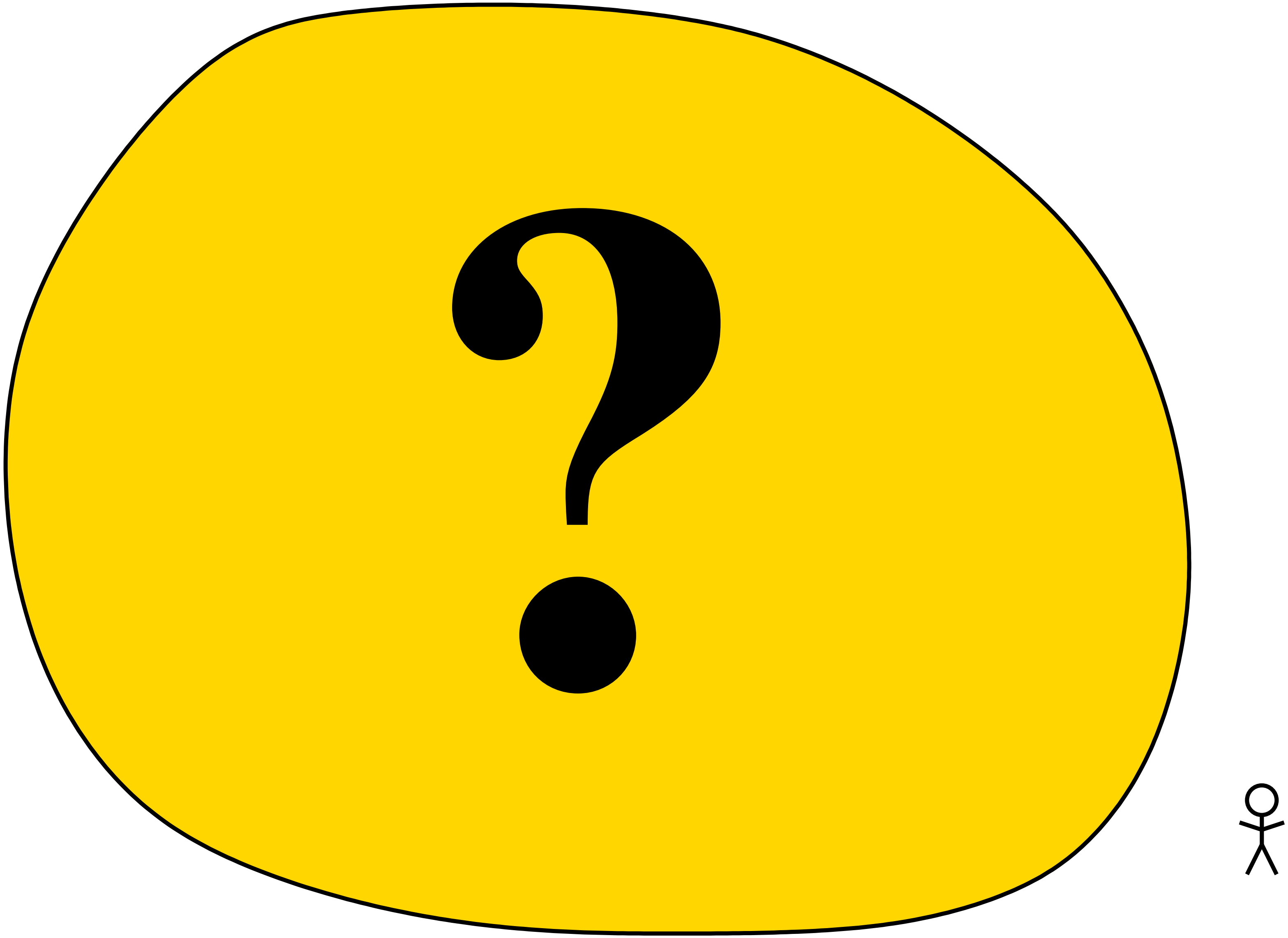
External Sampling

Krzysztof Onak
MIT

Joint work with **Alexandr Andoni**,
Piotr Indyk, and **Ronitt Rubinfeld**

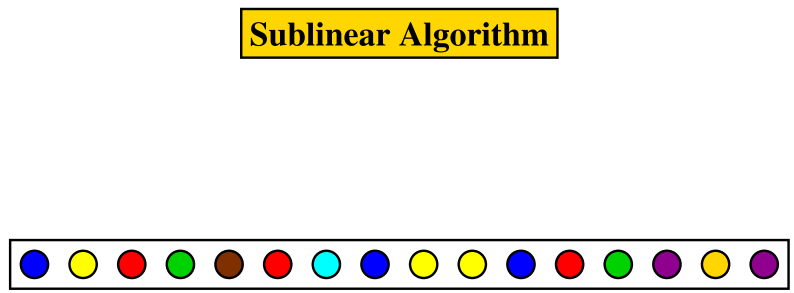
Massive Data



Massive Data

Various models have been developed:

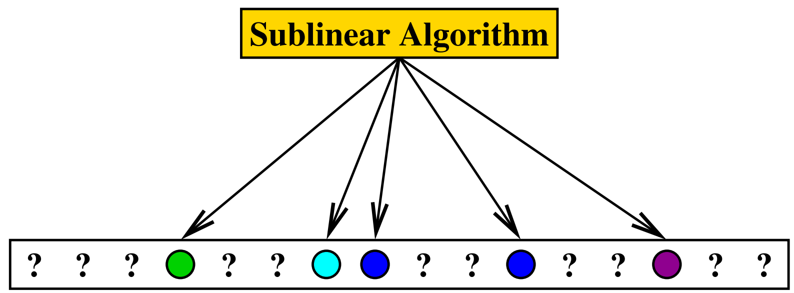
- Sublinear time algorithms
  (for instance, random sampling)



Massive Data

Various models have been developed:

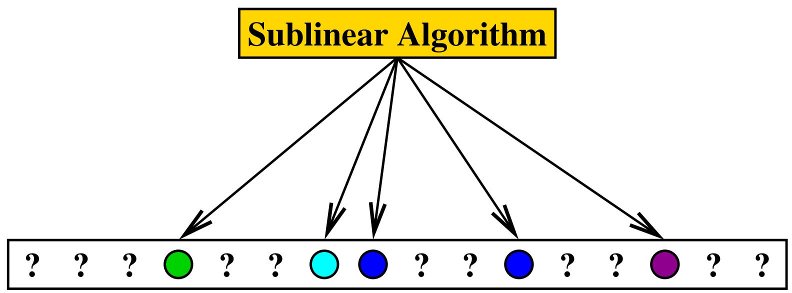
- Sublinear time algorithms
  (for instance, random sampling)



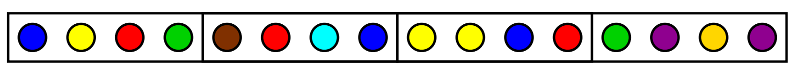
Massive Data

Various models have been developed:

- Sublinear time algorithms
  (for instance, random sampling)



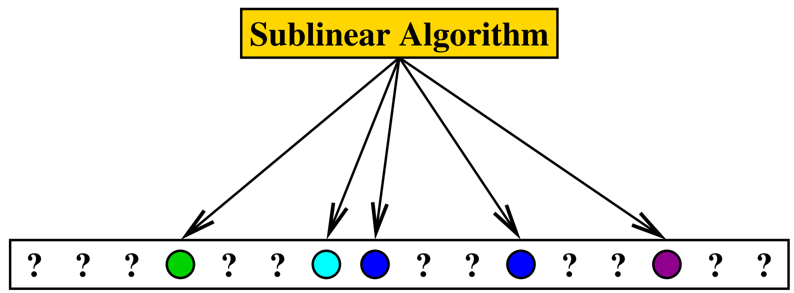
- External memory algorithms for data on disk



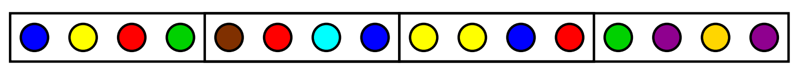
Massive Data

Various models have been developed:

- Sublinear time algorithms
  (for instance, random sampling)



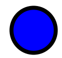
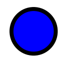
- External memory algorithms for data on disk

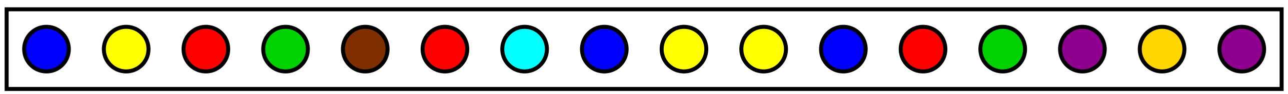


Can combine the two?

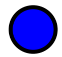
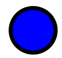
- Has to read entire block to get single sample
- Can decrease the number of block reads?

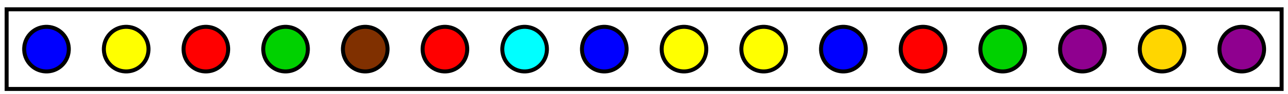
Estimating frequency

- Problem:
  - if frequency of  $\geq 2f$, report YES
  - if frequency of  $\leq f$, report NO



Estimating frequency

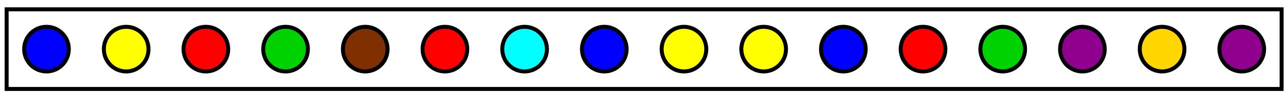
- Problem:
  - if frequency of  $\geq 2f$, report YES
  - if frequency of  $\leq f$, report NO



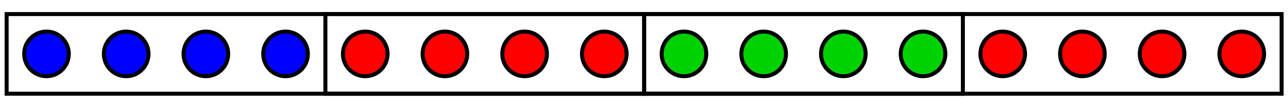
- Complexity: $\Theta(1/f)$ random samples

Estimating frequency

- Problem:
  - if frequency of $\geq 2f$, report YES
  - if frequency of $\leq f$, report NO

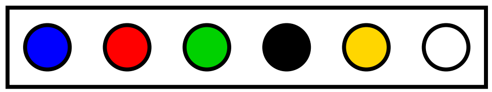
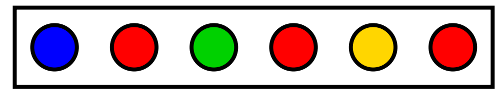


- Complexity: $\Theta(1/f)$ random samples
- Sampling blocks doesn't help!



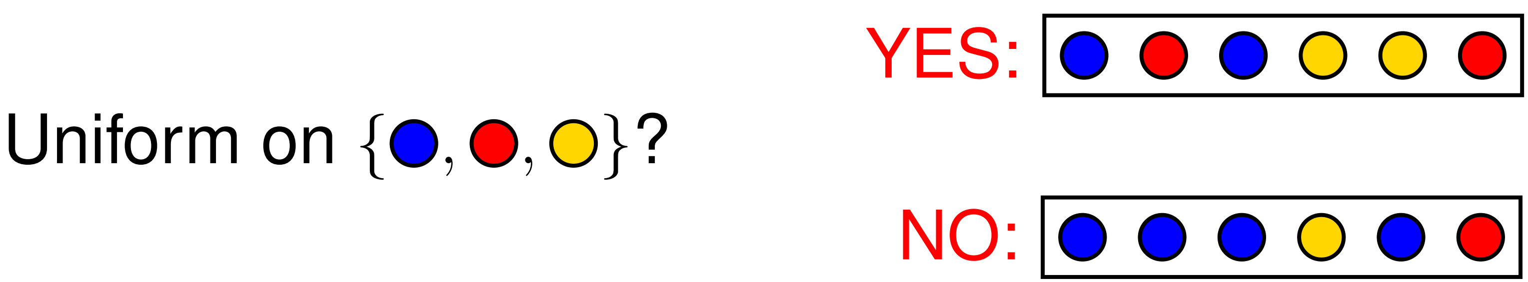
Our Results

- Problems:
  - Distinctness
    - YES: all elements different
    - NO: must remove $\geq \epsilon n$ elements for distinctness

YES:  NO: 

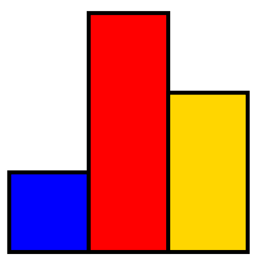
Our Results

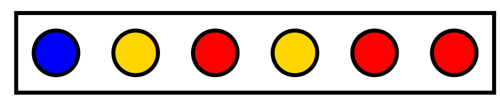
- Problems:
  - Distinctness
    - YES: all elements different
    - NO: must remove $\geq \epsilon n$ elements for distinctness
  - Uniformity [Goldreich-Ron, Batu-Fortnow-Rubinfeld-Smith-White]
    - YES: uniformly distributed over known set of size $m \leq n$
    - NO: must modify $\geq \epsilon n$ elements for uniformity

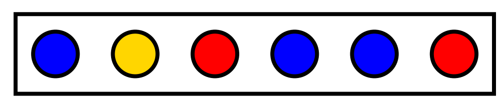


Our Results

- Problems:
  - Distinctness
    - YES: all elements different
    - NO: must remove $\geq \epsilon n$ elements for distinctness
  - Uniformity [Goldreich-Ron, Batu-Fortnow-Rubinfeld-Smith-White]
    - YES: uniformly distributed over known set of size $m \leq n$
    - NO: must modify $\geq \epsilon n$ elements for uniformity
  - Identity [Batu-Fortnow-Fischer-Kumar-Rubinfeld-White]
    - YES: distributed according to a known distribution
    - NO: must modify $\geq \epsilon n$ elements for the property

Distributed like  ?

YES: 

NO: 

Our Results

- Problems:
  - Distinctness
    - YES: all elements different
    - NO: must remove $\geq \epsilon n$ elements for distinctness
  - Uniformity [Goldreich-Ron, Batu-Fortnow-Rubinfeld-Smith-White]
    - YES: uniformly distributed over known set of size $m \leq n$
    - NO: must modify $\geq \epsilon n$ elements for uniformity
  - Identity [Batu-Fortnow-Fischer-Kumar-Rubinfeld-White]
    - YES: distributed according to a known distribution
    - NO: must modify $\geq \epsilon n$ elements for the property
- All require $\tilde{\Theta}(\sqrt{n})$ samples for fixed ϵ
- We improve by factor $\tilde{\Theta}(\sqrt{B})$ to $\tilde{\Theta}(\sqrt{n/B})$ block reads
- Can show improvement (nearly) optimal

Distinctness

Standard Algorithm

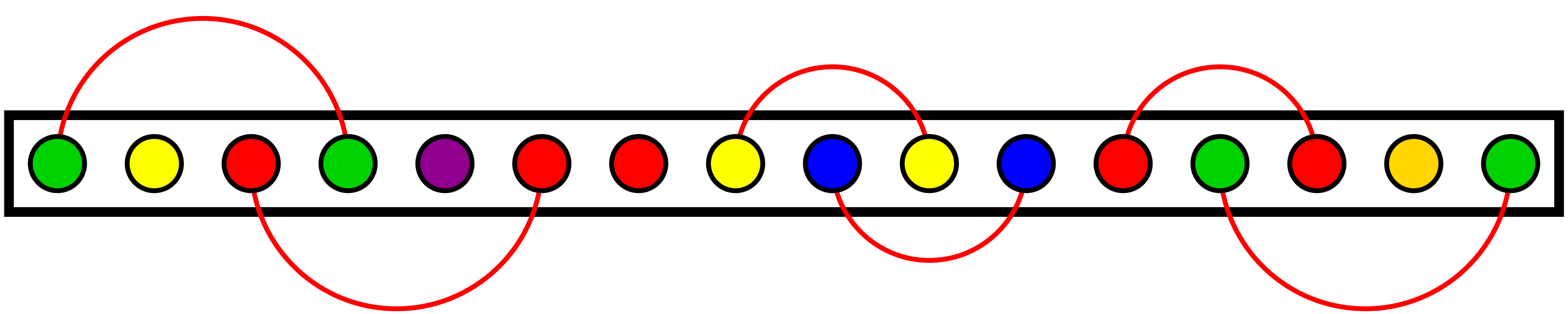
- Algorithm:
  - Sample $O(\sqrt{n/\epsilon})$ elements from different positions
  - If two sampled elements are equal, REJECT
  - Otherwise, ACCEPT

Standard Algorithm

- Algorithm:
  - Sample $O(\sqrt{n/\epsilon})$ elements from different positions
  - If two sampled elements are equal, REJECT
  - Otherwise, ACCEPT
- YES instance: always accepts

Standard Algorithm

- Algorithm:
  - Sample $O(\sqrt{n/\epsilon})$ elements from different positions
  - If two sampled elements are equal, REJECT
  - Otherwise, ACCEPT
- YES instance: always accepts
- NO instance:
  - $\Omega(\epsilon n)$ disjoint pairs of identical elements
  - Birthday paradox: algorithm samples one of them with constant probability

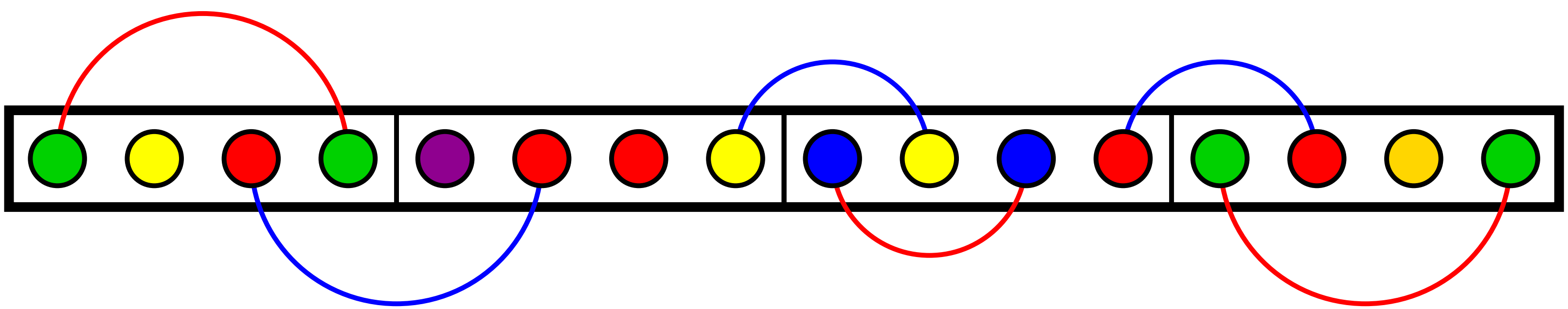


Algorithm for Blocks

- This time: Sample $O(\sqrt{n/(\epsilon B)})$ blocks

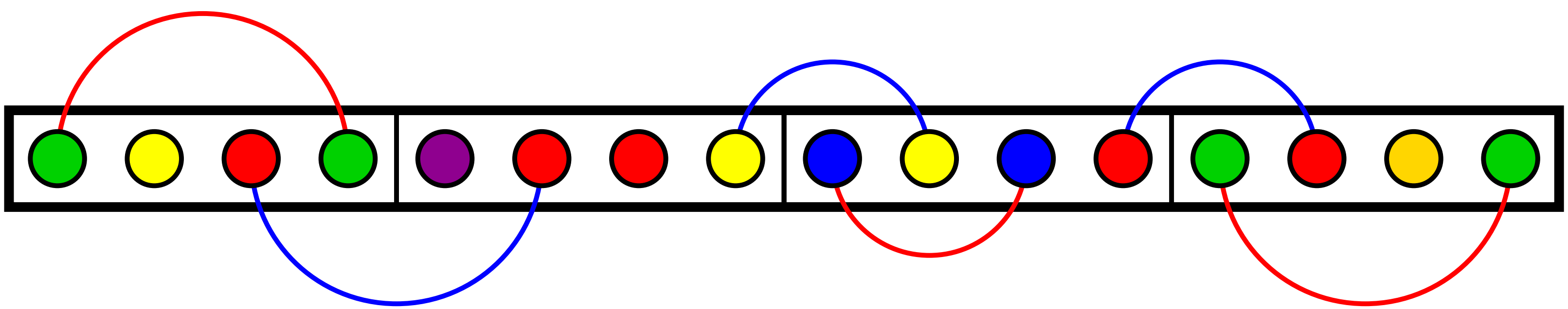
Algorithm for Blocks

- This time: Sample $O(\sqrt{n/(\epsilon B)})$ blocks
- Two kinds of pairs:
  - intra-block
  - inter-block



Algorithm for Blocks

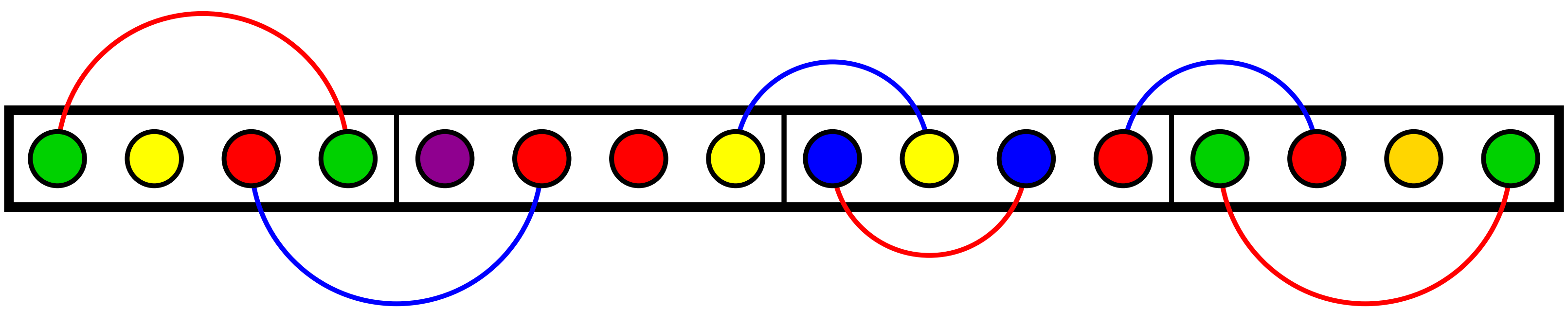
- This time: Sample $O(\sqrt{n/(\epsilon B)})$ blocks
- Two kinds of pairs:
  - intra-block
  - inter-block



- NO instance: $\Omega(\epsilon n)$ pairs of one of the kinds

Algorithm for Blocks

- This time: Sample $O(\sqrt{n/(\epsilon B)})$ blocks
- Two kinds of pairs:
  - intra-block
  - inter-block



- NO instance: $\Omega(\epsilon n)$ pairs of one of the kinds
- Matching lower bound

Testing Distributions

Uniformity

- Problem: Is uniform over $\{1, \ldots, m\}$?

Uniformity

- Problem: Is uniform over $\{1, \ldots, m\}$?
- Standard Algorithm:
  - Collect $t = O\left(\frac{1}{\epsilon}\sqrt{n}\right)$ samples
  - If number of identical pairs $> \left(1 + \frac{\epsilon}{2}\right) \cdot \binom{t}{2} \cdot \frac{1}{m}$,
    REJECT
  - Otherwise, ACCEPT

Uniformity

- Problem: Is uniform over $\{1, \ldots, m\}$?
- Standard Algorithm:
  - Collect $t = O\left(\frac{1}{\epsilon}\sqrt{n}\right)$ samples
  - If number of identical pairs $> \left(1 + \frac{\epsilon}{2}\right) \cdot \binom{t}{2} \cdot \frac{1}{m}$,
    REJECT
  - Otherwise, ACCEPT
- Algorithm for blocks:
  - almost the same
  - first checks if no item too frequent
  - more careful variance analysis

Uniformity

- Problem: Is uniform over $\{1, \ldots, m\}$?
- Standard Algorithm:
  - Collect $t = O\left(\frac{1}{\epsilon}\sqrt{n}\right)$ samples
  - If number of identical pairs $> \left(1 + \frac{\epsilon}{2}\right) \cdot \binom{t}{2} \cdot \frac{1}{m}$, REJECT
  - Otherwise, ACCEPT
- Algorithm for blocks:
  - almost the same
  - first checks if no item too frequent
  - more careful variance analysis
- Can extend to testing identity

Other Problems

Applications of Our Techniques

- Graph Isomorphism [Fischer-Matsliah]
  - Two graphs: known G and unknown H
    - YES: G and H isomorphic
    - NO: $\geq \epsilon n^2$ edges of H must be modified for isomorphism
  - Allowed queries: Is (u, v) edge of H?
  - Block model: adjacency matrix row by row on disk
  - Identity testing dominates the complexity

Applications of Our Techniques

- Graph Isomorphism [Fischer-Matsliah]
  - Two graphs: known G and unknown H
    - YES: G and H isomorphic
    - NO: $\geq \epsilon n^2$ edges of H must be modified for isomorphism
  - Allowed queries: Is (u, v) edge of H?
  - Block model: adjacency matrix row by row on disk
  - Identity testing dominates the complexity
- Metric Properties of Points [O.]
  - Does set of points embed into a tree metric? an ultrametric? ℓ_2^d? ℓ_1^2? ℓ_∞^2?
  - Searching for k-tuple
  - Standard algorithms: $\approx O(n^{1-1/k})$ samples for fixed ϵ
  - Block model: $O((n/B)^{1-1/k})$ samples

Further Problems

- Monotonicity
  - Input: sequence of n numbers
    - YES: monotone
    - NO: must delete ϵn elements for monotonicity
  - Can improve from $O(\frac{1}{\epsilon}\log n)$ to $O(\frac{1}{\epsilon}\log(n/B))$

Further Problems

- Monotonicity
  - Input: sequence of n numbers
    - YES: monotone
    - NO: must delete ϵn elements for monotonicity
  - Can improve from $O(\frac{1}{\epsilon}\log n)$ to $O(\frac{1}{\epsilon}\log(n/B))$
- Weak Estimation of Edit Distance [Batu-Ergün-Kilian-Magen-Raskhodnikova-Rubinfeld-Sami]

Further Problems

- Monotonicity
  - Input: sequence of n numbers
    - YES: monotone
    - NO: must delete ϵn elements for monotonicity
  - Can improve from $O(\frac{1}{\epsilon}\log n)$ to $O(\frac{1}{\epsilon}\log(n/B))$
- Weak Estimation of Edit Distance [Batu-Ergün-Kilian-Magen-Raskhodnikova-Rubinfeld-Sami]
- OPEN: Equality of Distributions [Batu-Fortnow-Rubinfeld-Smith-White]
  - Input: two sequences on disk
  - Problem: same distribution of items or at distance $\geq \epsilon$?
  - Standard model: $\tilde{O}(n^{2/3} \cdot \mathrm{poly}(1/\epsilon))$

Further Problems

- Monotonicity
  - Input: sequence of n numbers
    - YES: monotone
    - NO: must delete ϵn elements for monotonicity
  - Can improve from $O(\frac{1}{\epsilon}\log n)$ to $O(\frac{1}{\epsilon}\log(n/B))$
- Weak Estimation of Edit Distance [Batu-Ergün-Kilian-Magen-Raskhodnikova-Rubinfeld-Sami]
- OPEN: Equality of Distributions [Batu-Fortnow-Rubinfeld-Smith-White]
  - Input: two sequences on disk
  - Problem: same distribution of items or at distance $\geq \epsilon$?
  - Standard model: $\tilde{O}(n^{2/3} \cdot \mathrm{poly}(1/\epsilon))$
- Homework:

Check your favorite sublinear algorithm!!!