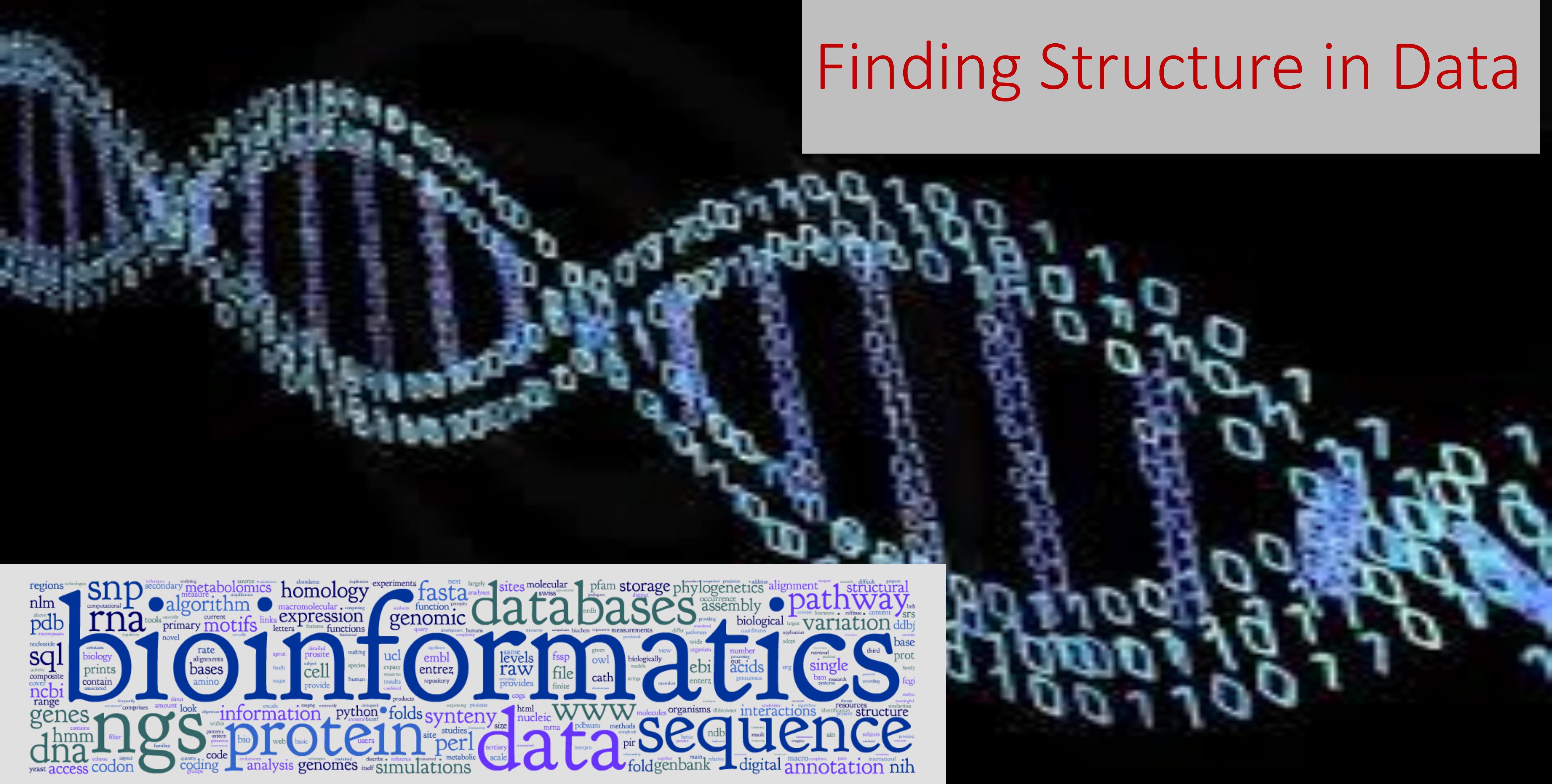
Samson Zhou

Pattern Matching over Noisy Data Streams

PURDUE UNIVERSITY

Finding Structure in Data



Pattern Matching

- Finding all instances of a pattern within a string

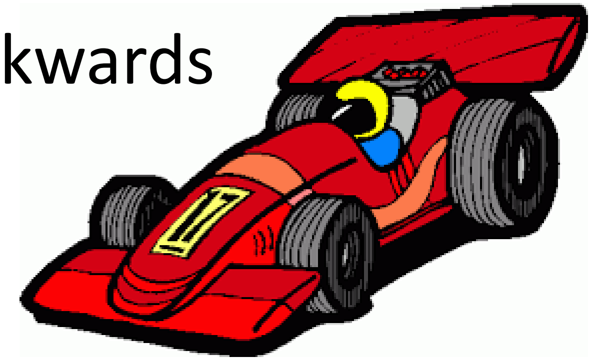
  ABCD

  ABCAABCDAACAABCDBCABCDADDDEAEABCDA

- Knuth-Morris-Pratt'70

Palindrome

- A string that reads the same forwards and backwards
- Manacher'75
- $S = S^R$
- RACECAR
- RACECAR



- AIBOHPHOBIA
- AIBOHPHOBIA

Alignment

- For strings S and T, indices i, j, and a metric $dist$: S and T have an alignment of length $i - j + 1$ if $S[i,j] = T[i,j]$
- S = ALGORITHM
- T = LOGARITHM

Periodicity

- A portion of a string that repeats

  ABCDABCDABCDABCD

  ABCDABCDABCDABCD

Streaming Model

- String of length n arrives one symbol at a time
- Use $o(n)$ space, ideally $O(polylog\ n)$

abaacabaccbabbbcbabbccababbccb

abaacabaccbabbbcbabbccababbccb

abaacabaccbabbbcbabbccababbccb

Finding Structure in Noisy Data

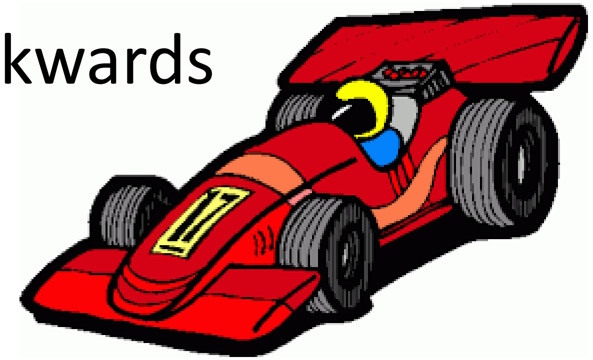
Palindrome

❖ A string that reads the same forwards and backwards

❖ $S = S^R$

❖ RACECAR

❖ RACECAR



❖ AIBOHPHOBIA

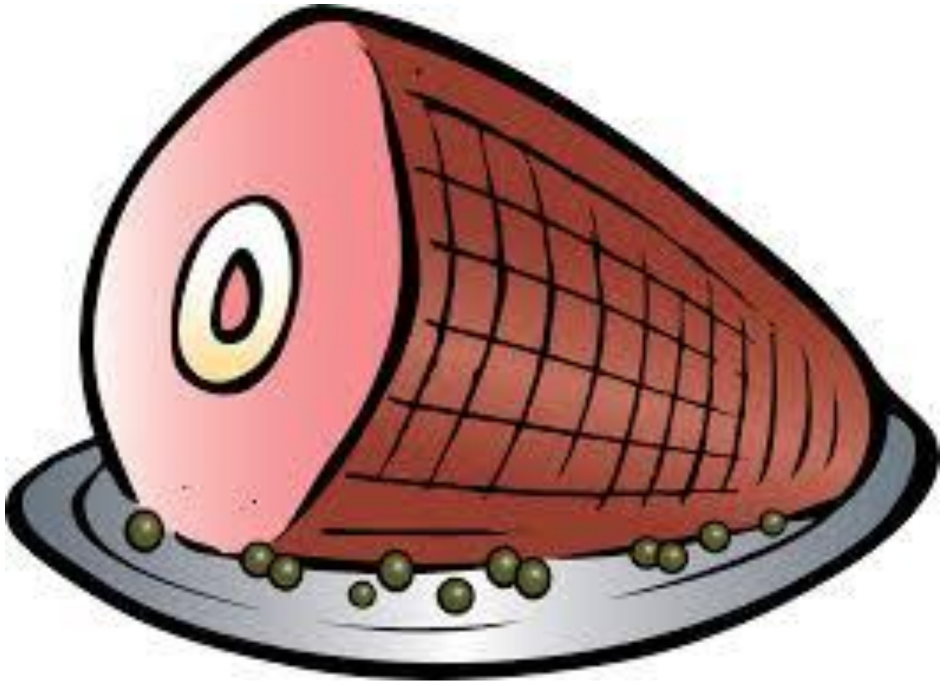
❖ AIBOHPHOBIA

d-Near-Palindrome

- A string that "almost" reads the same forwards and backwards
- Given a metric $dist$, a d-near-palindrome has $dist(S, S^R) \leq d$.
- RACECAR
- FACECAR

Hamming Distance

- Given strings X, Y, the Hamming distance between X and Y is defined as the positions i at which $X_i \neq Y_i$.
- S = FACECAR
- S^R = RACECAF
- $\text{HAM}(S, S^R) = 2$



Longest d-Near-Palindrome Problem

- Given a string S of length n, which arrives in a data stream, identify the longest d-near-palindrome in space $o(n)$.

- Given a string S of length n, which arrives in a data stream, find a "long" d-near-palindrome in space $o(n)$.

Related Work

- $O(\log n)$ space to provide a $(1+\varepsilon)$ multiplicative approximation to the length of the longest palindrome (Berenbrink,Ergün,Mallmann-Trenn,Sadeqi Azer ‘14)
- $O(\sqrt{n})$ space to provide a $\sqrt{n}$ additive approximation to the length of the longest palindrome (BEMS14)
- $O(\sqrt{n})$ space to find the longest palindrome in two passes (BEMS14)
- $\Omega\left(\frac{\log n}{\varepsilon \log(1+\varepsilon)}\right)$ space for $(1+\varepsilon)$ multiplicative approximation (GMSU16)
- $\Omega\left(\frac{n}{E}\right)$ space for E additive approximation (GMSU16)

Our results

- ❖ $O\left(\frac{d\log^7 n}{\varepsilon\log(1+\varepsilon)}\right)$ space to provide a $(1+\varepsilon)$ multiplicative approximation to the length of the longest d-near-palindrome
- ❖ $O(d\sqrt{n}\log^6 n)$ space to provide a $\sqrt{n}$ additive approximation to the length of the longest d-near-palindrome
- ❖ $O(d^2\sqrt{n}\log^6 n)$ space to find the longest d-near-palindrome in two passes
- ❖ $\Omega(d\log n)$ space LB for $(1+\varepsilon)$ multiplicative approximation
- ❖ $\Omega\left(\frac{dn}{E}\right)$ space LB for E additive approximation

Comparison

	Longest Palindrome	Longest d-Near-Palindrome (Here)
$(1+\varepsilon)$ multiplicative	$O(\log^2 n)$ (BEMS14)	$O\left(\frac{d \log^7 n}{\varepsilon \log(1+\varepsilon)}\right)$
$\sqrt{n}$ additive	$O(\sqrt{n} \log n)$ (BEMS14)	$O(d\sqrt{n} \log^6 n)$
two pass exact	$O(\sqrt{n} \log n)$ (BEMS14)	$O(d^2\sqrt{n} \log^6 n)$
$(1+\varepsilon)$ multiplicative LB	$\Omega\left(\frac{\log n}{\log(1+\varepsilon)}\right)$ (GMSU16)	$\Omega(d \log n)$
E additive LB	$\Omega\left(\frac{n}{E}\right)$ (GMSU16)	$\Omega\left(\frac{dn}{E}\right)$

Warm-up

❖ Suppose we see string S, followed by string T. How can we determine if $S = T$, with high probability?

Karp-Rabin Fingerprints

- Given base B and a prime P, define $\phi(S) = \sum_{i=1}^{n} B^i S[i] \ (mod\ P)$
- If $S = T$, then $\phi(S) = \phi(T)$
- If $S \neq T$, then $\phi(S) \neq \phi(T)$ w.h.p. (Schwartz-Zippel)

Properties of Karp-Rabin Fingerprints

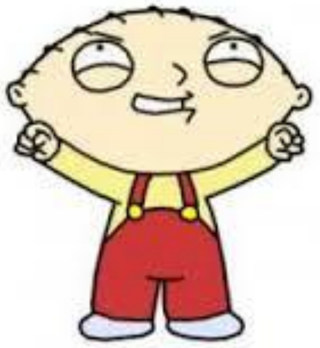
- $\phi(S[1:y]) = \phi(S[1:x]) + B^x \phi(S[x:y])$
- Define $\phi^R(S) = \sum_{i=1}^{n} B^{-i} S[i] \ (mod\ P)$
- $\phi(S^R[1:x]) = B^{x+1} \phi^R(S[1:x])$
- $\phi^R(S[1:y]) = \phi^R(S[1:x]) + B^{-x} \phi^R(S[x:y])$

Identifying Palindromes

- 111101011100001010010101001111101011100001010010101001
- 111101011100001010010101001111101011100001010010101001

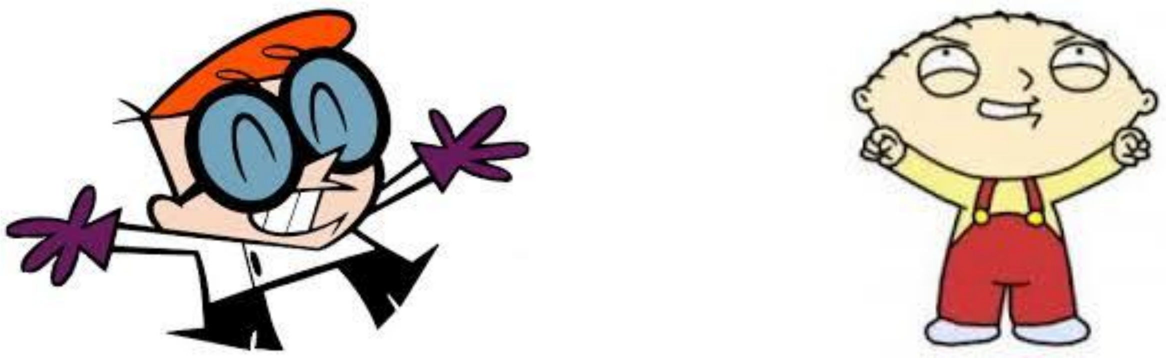
Identifying Near-Palindromes?

- 111101011100001010010101001111101011100001010010101001
- 111101011100001010010101001111101011100001010010101001

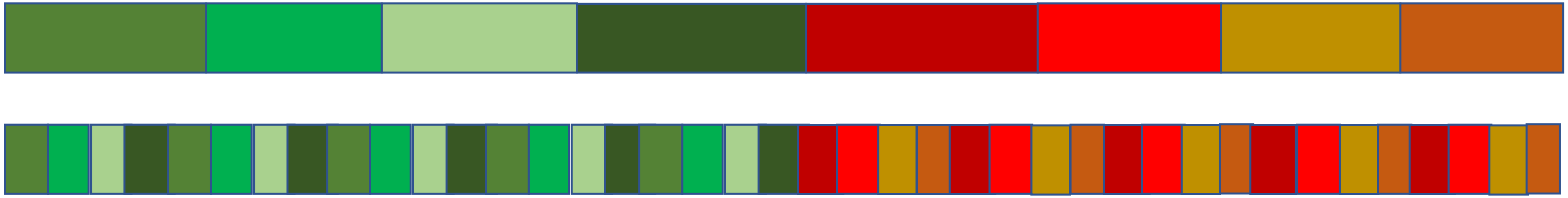


Identifying Near-Palindromes?

- 111101011100001010010101001111101011100001010010101001
- 111101011100001010010101001111101011100001010010101001

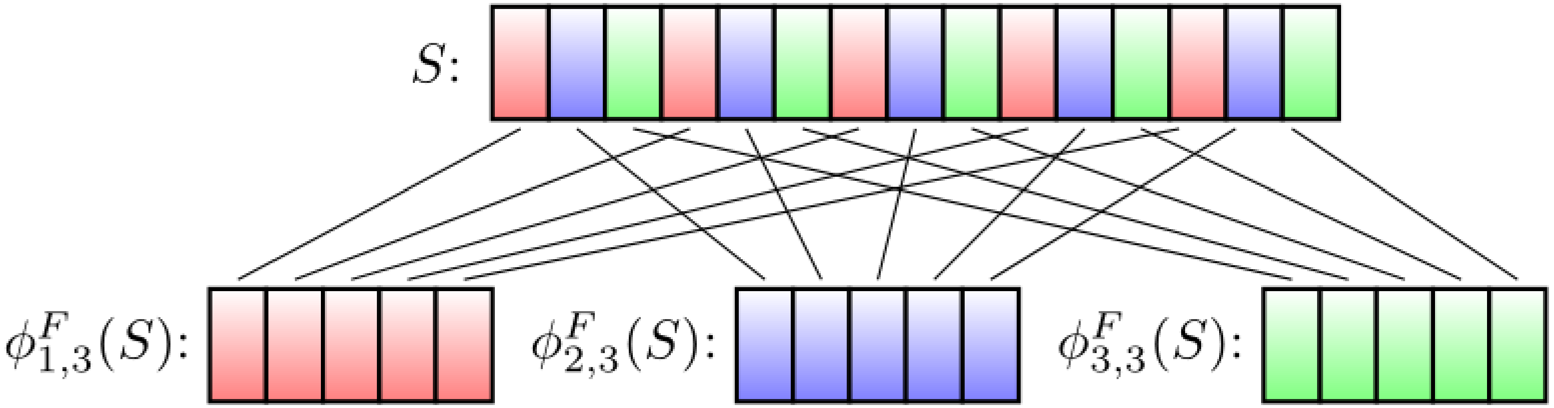


Identifying Near-Palindromes? (CFP+16)



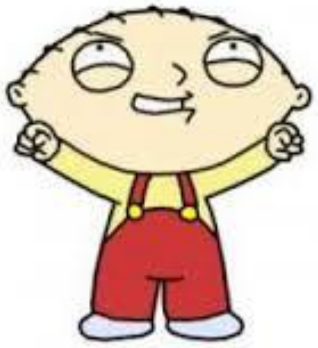
Karp-Rabin Fingerprints for subpatterns

- $S_{a,b} = S[a]S[a+b]S[a+2b]S[a+3b]\ldots$
- $\phi_{a,b}(S) = \phi(S_{a,b}) = B * S[a] + B^2 * S[a+b] + B^3 * S[a+2b]\ldots$



Identifying Near-Palindromes?

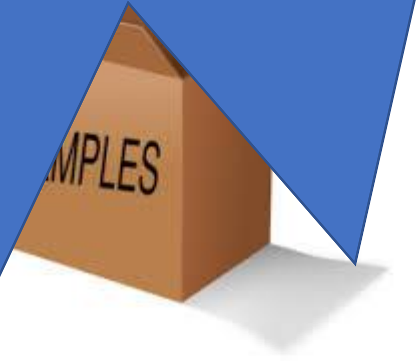
- Let $\Delta = \#\{a \mid \phi_{a,b}(S) \neq B^k \phi^R_{a,b}(S) \ (mod\ P)\}$
- Then $\Delta \leq \mathrm{HAM}(S, S^R)$



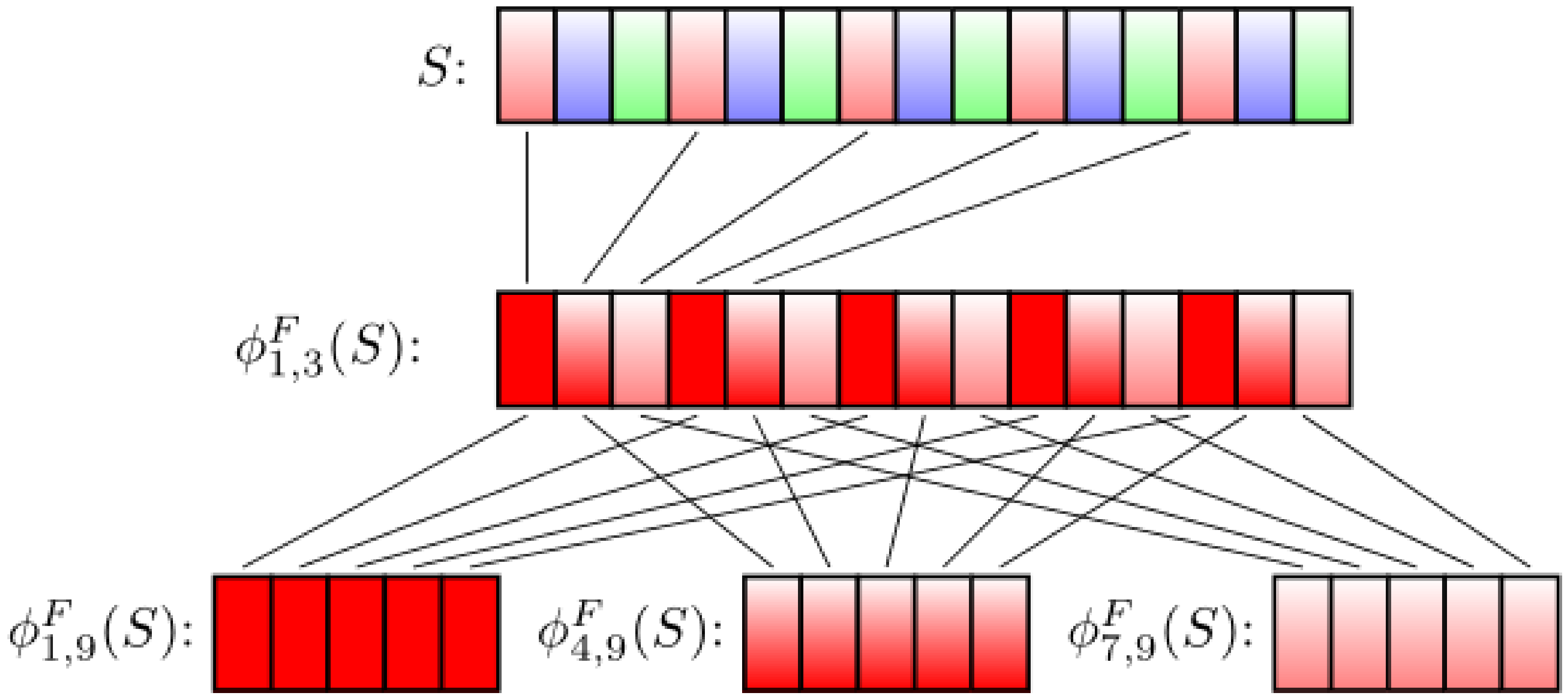
Identifying Near-Palindrome ?

- Sample log … imes p_1 … fro … log^2 … 4 $d \log^2 n$].
- Let $\Delta = \max \#$ … $P)\}$
- $\Delta \leq$ …
- If $\mathrm{HAM}(S, S$ … 16)

What about
$\mathrm{HAM}(S, S^R) \leq 2d$?



Karp-Rabin Fingerprints for sub-subpatterns



Second level Karp-Rabin Fingerprints

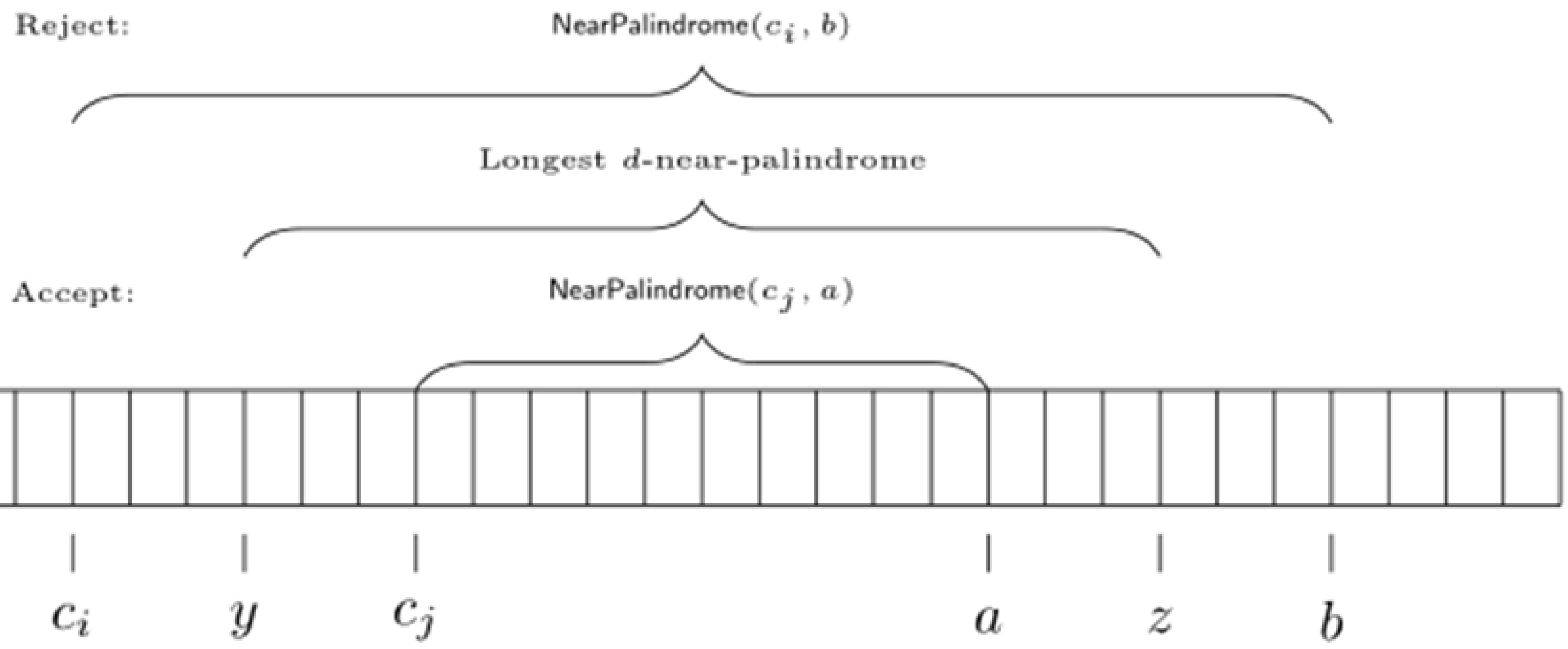
- Call a mismatch *isolated* under p_i if it is the only mismatch under some subpattern S_{a,p_i}. Let I be the number of isolated mismatches.
- If $\mathrm{HAM}(S, S^R) \leq 2d$, then $I = \mathrm{HAM}(S, S^R)$ w.h.p. (CFP+16)

In review

- There exists a data structure of size $O(d \log^6 n)$ bits which recognizes whether $\mathrm{HAM}(S, S^R) \leq d$ w.h.p.

Additive Error Algorithm

❖ Initialize a data structure every $\frac{\sqrt{n}}{2}$ positions!

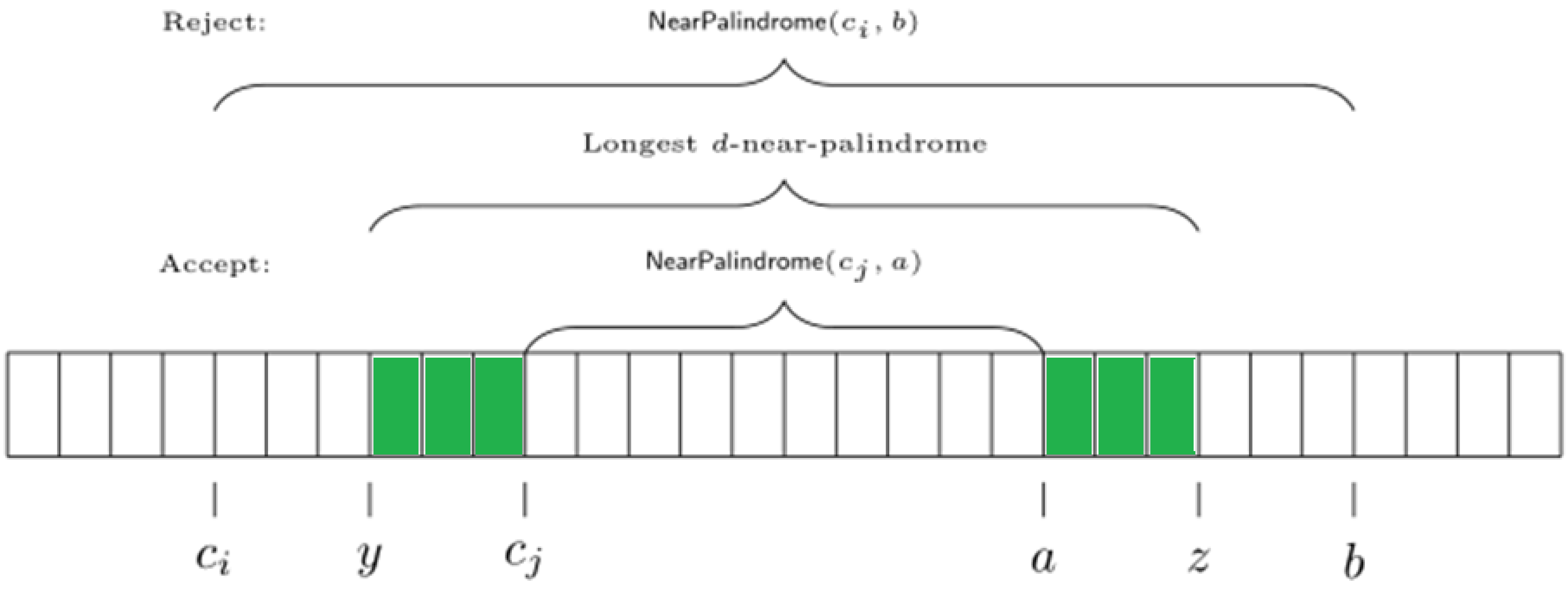


Additive Error Algorithm

- $\frac{\sqrt{n}}{2}$ sketches, each of size $O(d \log^6 n)$ bits
- Total space: $O(d\sqrt{n} \log^6 n)$ bits

2-Pass Exact Algorithm

- Can we modify 1-pass additive algorithm to 2-pass exact?
- Missing characters before checkpoint!



2-Pass Exact Algorithm

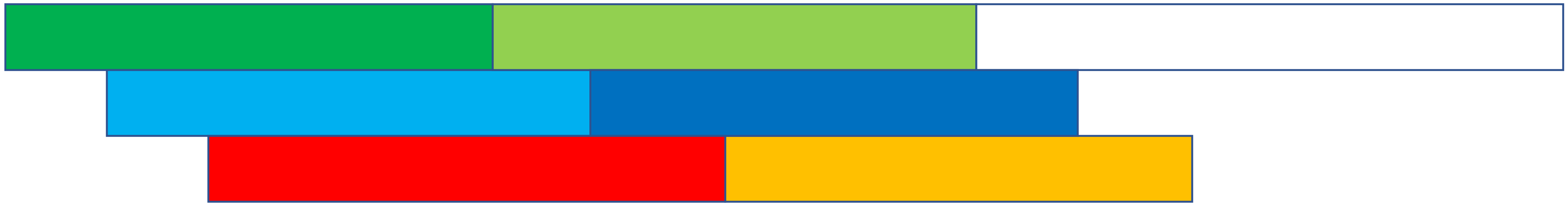
- Idea: keep all characters before each checkpoint in the second pass
- What if there are $O(n)$ candidates?

- Structural result of palindromes (BEMS14)

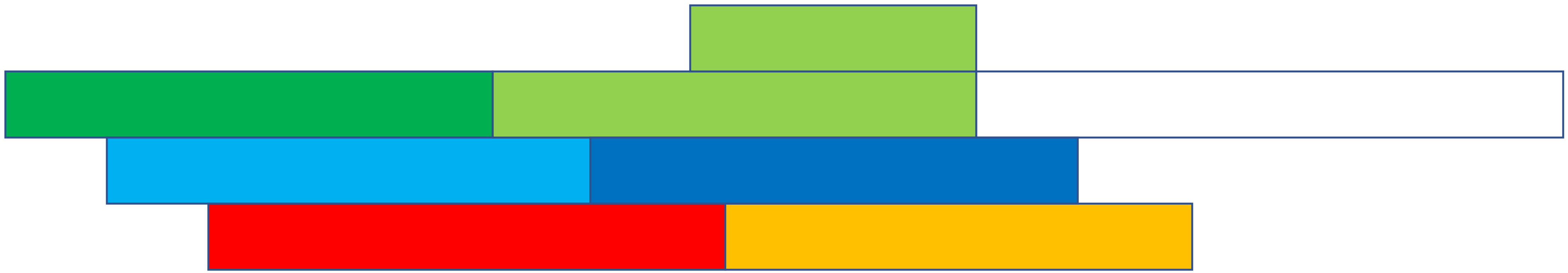
Structural Result of Palindromes (BEMS14)

Structural Result of Palindromes (BEMS14)

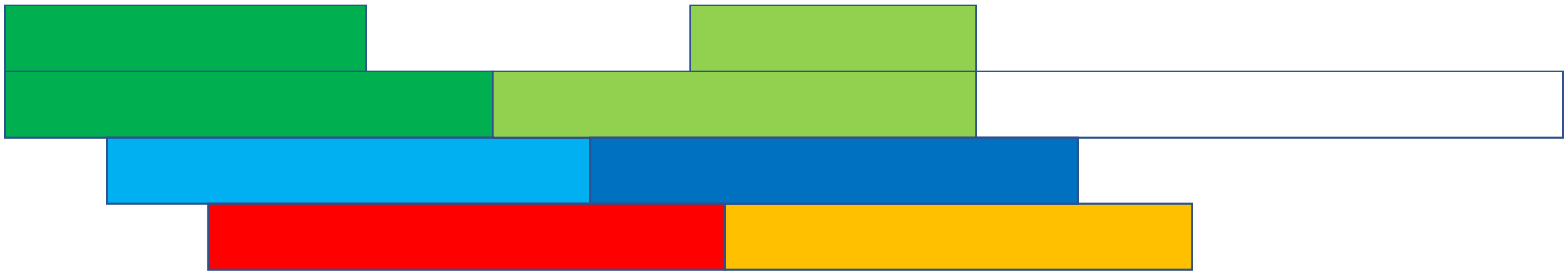
Structural Result of Palindromes (BEMS14)



Structural Result of Palindromes (BEMS14)



Structural Result of Palindromes (BEMS14)



Structural Result of Palindromes (BEMS14)

Structural Result of Palindromes (BEMS14)

Structural Result of Near-Palindromes

- Not quite periodic (at most $2d - 1$ different words)
- Need to save at most $2d - 1$ fingerprints of words

2-Pass Exact Algorithm

- Not quite periodic (at most $2d-1$ different words)
- Need to save at most $2d-1$ fingerprints of words

2-Pass Exact Algorithm

- First pass: $O(d^2\sqrt{n}\log^7 n)$ bits
- At most $2d-1$ fingerprints, each of size $O(d^2\log^6 n)$ words
- Need to save at $\sqrt{n}$ characters before $2d-1$ checkpoints: $O(d\sqrt{n})$
- Total space: $O(d^2\sqrt{n}\log^7 n)$ bits

Multiplicative Lower Bounds

- Yao's Principle: to show that any randomized algorithm fails, show that every deterministic algorithm fails over random inputs
- Let ν be the prefix of $101100111000111110000 \ldots = 1^1 0^1 1^2 0^2 \ldots$ of length $\frac{n}{4}$ (GMSU16).
- Take $x \in X = \left\{\text{strings of length } \frac{n}{4} \text{ with weight } d\right\}$
- Take $y \in Y = \{y \mid \mathrm{HAM}(x, y) = d \text{ or } \mathrm{HAM}(x, y) = d + 1\}$
- Define $s(x, y) = \nu^R x y^R \nu$.

Multiplicative Lower Bounds

YES:
If $\mathrm{HAM}(x, y) \leq d$,
then the longest d-near-palindrome of $s(x, y)$ has length n.

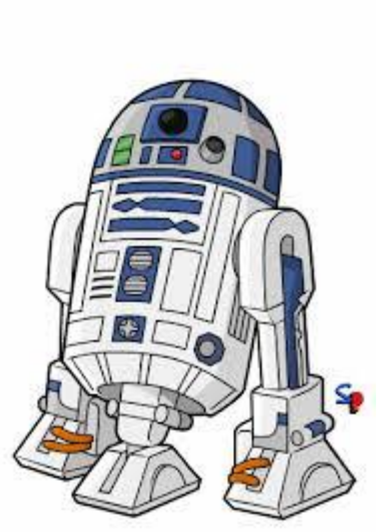
NO:
If $\mathrm{HAM}(x, y) > d$,
then the longest d-near-palindrome of $s(x, y)$ has length at most $200d^2 + \frac{n}{2}$.

Multiplicative Lower Bounds

- A $(1+\varepsilon)$ multiplicative algorithm differentiates whether $\mathrm{HAM}(x,y) \leq d$ or $\mathrm{HAM}(x,y) > d$.
- Just need to show cannot differentiate whether $\mathrm{HAM}(x,y) \leq d$ or $\mathrm{HAM}(x,y) > d$ in $o(d \log n)$ space!

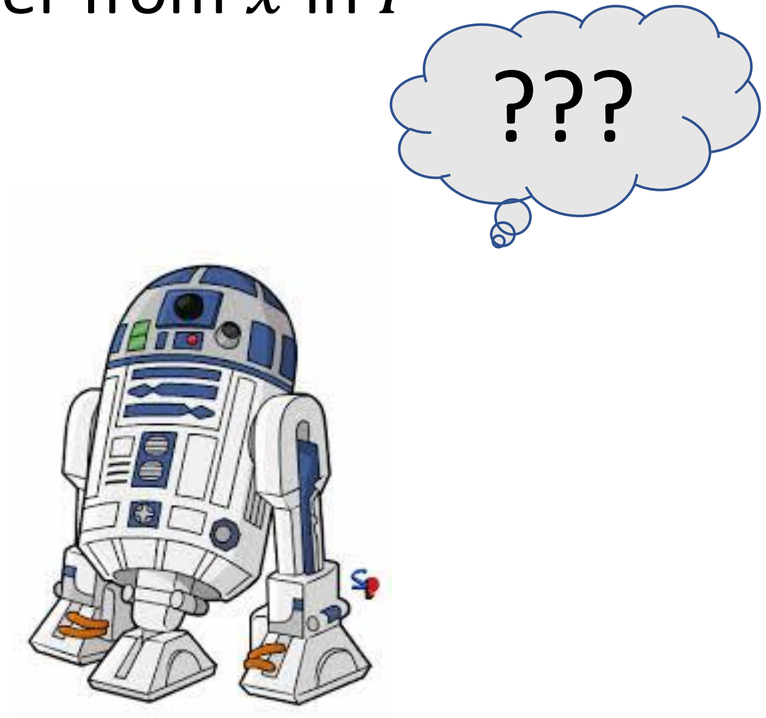
Multiplicative Lower Bounds

- Save x in $\frac{d \log n}{3}$ bits.
- Since $x \in X = \left\{\text{strings of length } \frac{n}{4} \text{ with weight } d\right\}$, there are $\frac{|X|}{4}$ pairs (x, x') which are mapped to the same configuration.



Multiplicative Lower Bounds

- Let I be the set of indices for which $x_i = 1$ or $x_i' = 1$
- Suppose $\mathrm{HAM}(x, y) = d$ but y does not differ from x in I
- x: 101100000010001000000100100000
- x': 100000010010101000000100100000
- y: 111101100010001011100100100010
- Then $\mathrm{HAM}(x', y) > d$!
- Errs on either $s(x, y)$ or $s(x', y)$.



Multiplicative Lower Bounds

- There are $\frac{|X|}{4}$ values of x mapped to the wrong configuration, each with $\binom{\frac{n}{4}-2d}{d}$ values of y, where algorithm is incorrect.
- Probability of failure:

$$\frac{\frac{|X|}{4}\binom{\frac{n}{4}-2d}{d}}{|X||Y|} \geq \frac{1}{n}$$

In review

- Provided a distribution over which any deterministic algorithm with $o(d \log n)$ bits fails to distinguish $\mathrm{HAM}(x, y) \leq d$ or $\mathrm{HAM}(x, y) > d$ at least $\frac{1}{n}$ of the time
- A $(1 + \varepsilon)$ multiplicative algorithm differentiates whether $\mathrm{HAM}(x, y) \leq d$ or $\mathrm{HAM}(x, y) > d$
- Showed every deterministic algorithm fails over random inputs

Additive Lower Bounds

- Define $s(x,y) = 1^E x_1 1^{\frac{E}{d}} x_2 1^{\frac{E}{d}} x_3 \ldots x_{\frac{n'}{2}} y_{\frac{n'}{2}} \ldots y_3 1^{\frac{E}{d}} y_2 1^{\frac{E}{d}} y_1 1^E$
- Take $x \in X = \left\{\text{all strings of length } \frac{n'}{2}\right\}$
- Take $y \in Y = \{\text{HAM}(x,y) = d \text{ or } \text{HAM}(x,y) = d+1\}$

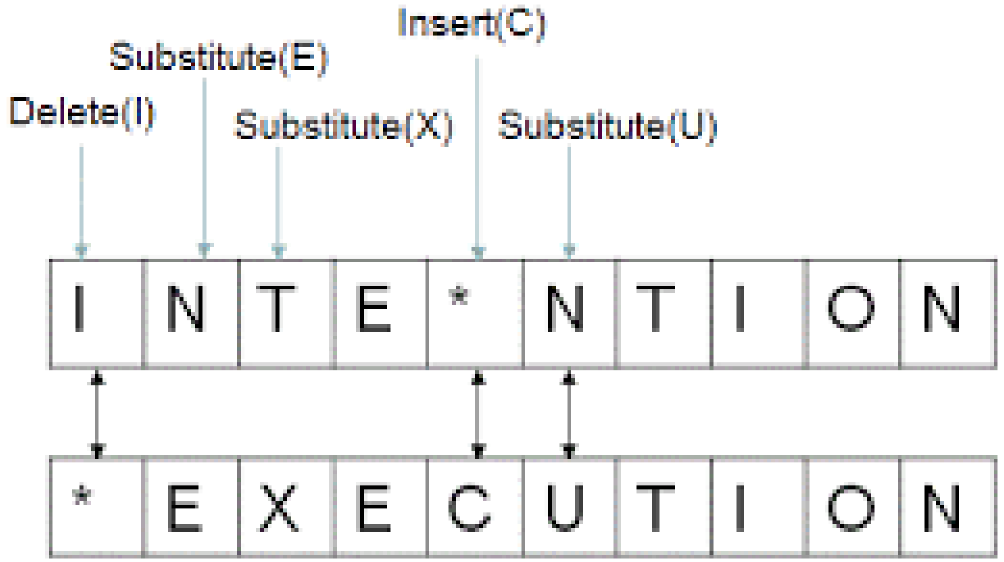
Questions?

d-Near-Alignment

- For strings S and T, indices i, j, and a metric $dist$: S and T have a d-near-alignment of length $i - j + 1$ if $dist(S[i,j], T[i,j]) \leq d$.
- S = RACECAR
- T = FACECAR

Edit (Levenshtein) Distance

- Given strings X, Y, the edit distance between X and Y is defined as the minimum number of deletions, insertions, and substitutions performed on X to obtain Y.
- S = 1010101010101010
- T = 0101010101010101
- $\text{HAM}(S, T) = 16$
- $\text{ed}(S, T) = 2$



Edit (Levenshtein) Distance

- Classical offline solution: dynamic programming $O(n^2)$ time (WF74)
- Cannot be computed in $O(n^{2-\delta})$ time assuming SETH (BI15)
- Any linear sketch which distinguishes the cases $\mathrm{ed}(x, y) = 2$ and $\mathrm{ed}(x, y) = 1$ requires $\Omega(n)$ space (AGMP13)

Longest d-Near-Alignment Problem

❖ Given strings S and T of length n, which arrive in a data stream, identify the longest d-near-alignment in space $o(n)$.

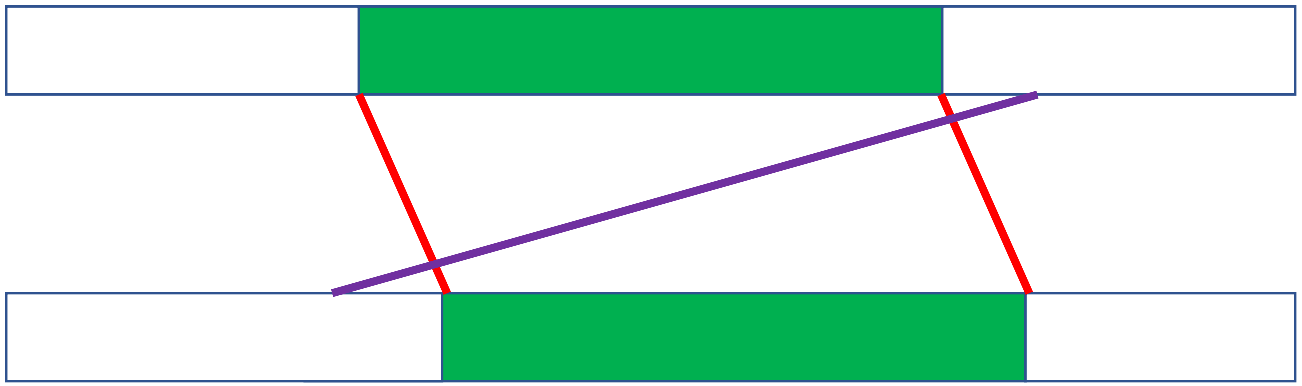
❖ Given strings S and T of length n, which arrive *simultaneously* in a data stream, identify the longest d-near-alignment in space $o(n)$.

Results (All Edit Distance)

- $O\left(\frac{d \log n}{\varepsilon \log(1+\varepsilon)}\right)$ space to provide a $(1+\varepsilon)$ multiplicative approximation to the length of the d-near-alignment (simultaneous)
- $O\left(\frac{dn \log n}{E}\right)$ space to provide an E additive approximation to the length of the d-near-alignment (simultaneous)
- $O(d^2 + d \log n)$ space to find the longest d-near-alignment (simultaneous)
- $\Omega(d \log n)$ space LB for $(1+\varepsilon)$ multiplicative approximation in streaming model

Longest d-Near-Alignment

- Observation #1: If $d + 1$ consecutive characters in S are matched to $d + 1$ consecutive characters in T, no character before the *region* can be matched to a character after the region by any other alignment



Longest d-Near-Alignment

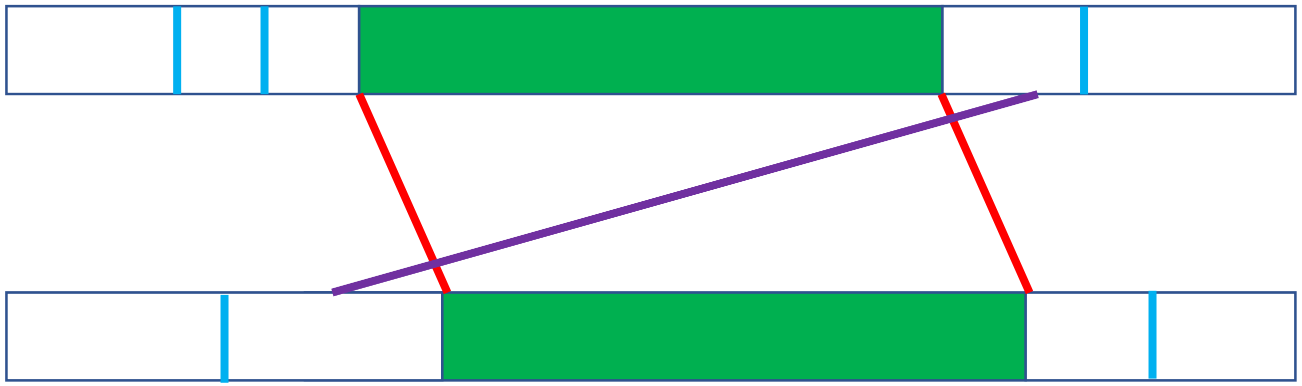
- Observation #2: If $(d+1)^2$ consecutive characters in S and T does not contain a region (of length $d+1$), then it requires d edit operations to be aligned

Longest d-Near-Alignment

❖ Sliding window of size $(d+1)^2$ identifies either the most recent region or the most recent d edit operations

Longest d-Near-Alignment

- Algorithm keeps the most recent d edit operations, location of the latest region, and the sliding window of size $(d+1)^2$
- Edit operations before the region are fixed



Longest d-Near-Alignment

- Window of size $(d+1)^2$
- Locations of d edit operations, each requiring space $O(\log n)$
- Total space: $O(d^2 + d \log n)$

Questions?

Periodicity

- A portion of a string that repeats

  ABCDABCDABCDABCD

  ABCDABCDABCDABCD

Periodicity

- Alternate definition: prefix is the same as suffix
- If S has length n, and $S[1{:}n-p] = S[p+1{:}n]$, then we say S has period p.

ABCDABCDABCDABCD

ABCDABCDABCD

ABCDABCDABCD

ABCDABCDABCDABCD

Hamming Distance

❖ Given strings X, Y, the Hamming distance between X and Y is defined as the positions i at which $X_i \neq Y_i$.

S = HAMMING

T = FALLING

$$HAM(S, T) = 3$$

k-Periodicity

- A string that is "almost" periodic, robust to k changes.
- Periodicity: $S[1{:}n-p] = S[p+1{:}n]$
- k-Periodicity: $\mathrm{HAM}(S[1{:}n-p], S[p+1, n]) \le k$.

ABCDABCDABCEABCE

ABCDABCDABCEABCE

ABCDABCDABCE

ABCDABCEABCE

ABCDABCDABCEABCE

1-period: 4

- Long term periodic changes, but also encompasses "natural" definition.

k-Periodicity Problem

- Given a string S of length n, which arrives in a data stream, identify the smallest k-period in space $o(n)$.

- Given a string S of length n, which arrives in a data stream, identify the smallest k-period in space $o(n)$, with two passes.

Related Work

- $O(\log^2 n)$ space to find the shortest period in one-pass, if $p \leq \frac{n}{2}$. (ErgunJowhariSaglam10)
- $\Omega(n)$ space to find the period in one-pass, if $p > \frac{n}{2}$. (EJS10)
- $O(\log^2 n)$ space to find the shortest period in two-passes, even if $p > \frac{n}{2}$. (EJS10)

- k-Mismatch Problem: $O(k^2 \log^8 n)$ space to find all instances of a pattern P within a text T with up to k errors. (CliffordFontainePoratSachStarikovskaya16)

k-Periodicity (Our results)

- $O(k^4 \log^9 n)$ space to find the shortest k-period in one-pass, if $p \leq \frac{n}{2}$.
- $O(k^4 \log^9 n)$ space to find the shortest k-period in two-passes, even if $p > \frac{n}{2}$.
- $\Omega(n)$ space to find the k-period, if $p > \frac{n}{2}$, in one-pass.
- $\Omega(k \log n)$ space to find the k-period, even if $p \leq \frac{n}{2}$, in one-pass.

Ideas from Streaming Periodicity

- A period p satisfies $S[1:n-p] = S[p+1,n]$.
- If $p \leq \frac{n}{2}$, then $S\left[1:\frac{n}{2}\right] = S\left[p+1, p+\frac{n}{2}\right]$.

  ABCDABCDABCDABCD

  ABCDABCDABCDABCD

  ABCDABCDABCDABCD

  ABCDABCDABCDABCD
- If $p > \frac{n}{2}$, then for some m, $S[1:2^m] = S[p+1, p+2^m]$.

Karp-Rabin Fingerprints

- Given base B and a prime P, define $\phi(S) = \sum_{i=1}^{n} B^i S[i] \ (mod\ P)$
- If $S = T$, then $\phi(S) = \phi(T)$
- If $S \neq T$, then $\phi(S) \neq \phi(T)$ w.h.p. (Schwartz-Zippel)

Ideas from Streaming Periodicity

- First pass: Find all positions p such that first $\frac{n}{2}$ characters match.

$$S\left[1 : \frac{n}{2}\right] = S\left[p + 1, p + \frac{n}{2}\right].$$

ABCDABCDABCDABCD

ABCDABCDABCDABCD

- Second pass: For each p, check whether p is a k-period.

$$S[1 : n - p] = S[p + 1, n].$$

ABCDABCDABCDABCD

ABCDABCDABCDABCD

Overall Idea

- A period p satisfies $\mathrm{HAM}(S[1:n-p], S[p+1,n]) \leq k$.
- If $p \leq \frac{n}{2}$, then $\mathrm{HAM}\left(S\left[1:\frac{n}{2}\right], S\left[p+1, p+\frac{n}{2}\right]\right) \leq k$.
- First pass: Find all positions p that match the first $\frac{n}{2}$ characters.

$$\mathrm{HAM}\left(S\left[1:\frac{n}{2}\right], S\left[p+1, p+\frac{n}{2}\right]\right) \leq k.$$

- Second pass: For each p, check whether p is a k-period.

$$\mathrm{HAM}(S[1:n-p], S[p+1,n]) \leq k.$$

- Reduction to Pattern Matching / k-Mismatch

First Pass to Second Pass?

- First pass: Find all positions p, "candidate" k-periods.

$$\mathrm{HAM}\left(S\left[1:\frac{n}{2}\right], S\left[p+1, p+\frac{n}{2}\right]\right) \le k.$$

- Second pass: For each p, check whether p is a k-period.

$$\mathrm{HAM}(S[1:n-p], S[p+1, n]) \le k.$$

- ABCDABCDABCDABCDABCD
- Candidate positions $p = \{4, 8, 12, 16, \dots\}$.
- Candidates form an arithmetic progression!

First Pass to Second Pass?

- If p and q are periods, then $d = \gcd(p, q)$ is a period.
- Does not work for k-periodicity!
- AAAABA, $k = 1$
- $p = 2$: AAAABA, AAAABA

  AAAA
  AABA
  1 mismatch

- $p = 3$: AAAABA, AAAABA

  AAA
  ABA
  1 mismatch

- $p = 1$: AAAABA, AAAABA

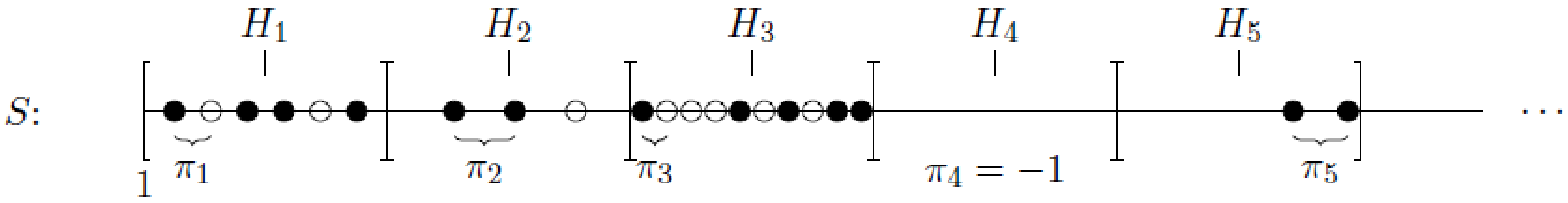
  AAAAB
  AAABA
  2 mismatches!

First Pass to Second Pass?

- Periodicity: Candidate positions $p = \{4,8,12,16, \ldots\}$
  - What's actually happening in the second pass?
  - Using $S[1:4]$, $S[5:8]$, $S[9:12]$,… to build $S[5:n]$, $S[9:n]$, $S[13:n]$,…
  - Can do this because $S[1:4]$, $S[5:8]$, $S[9:12]$ are all the same!
- k-periodicity: Candidate positions $p = \{8,16,20,28,32 \ldots\}$?
- Attempt: Candidate positions $p = \{4,8,12,16,20,24,28,32 \ldots\}$?
  - Can still do above construction if "most" of $S[1:4]$, $S[5:8]$, $S[9:12]$ are the same
  - Not sure if true…

First Pass to Second Pass?

- Candidates $p = \{8,16,20,27,30,39,45,55\}$?
- Candidates $p = \{8,12,16,20\},\{27,30,33,36,39\},\{45,50,55\}$



Structural Results

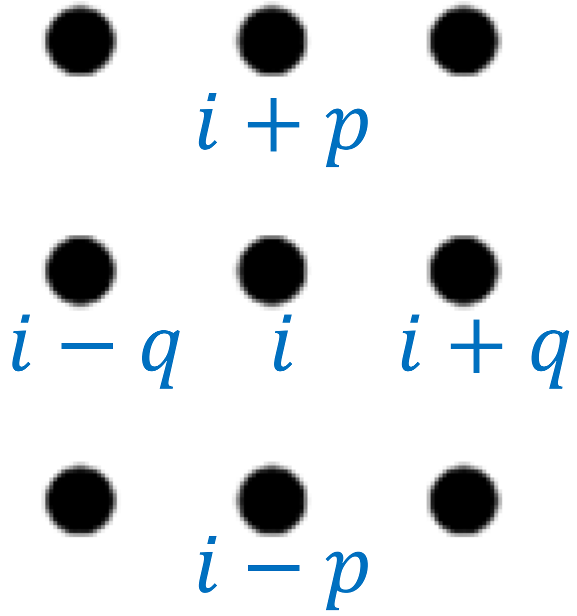
- If p and q are periods, then $d = \gcd(p, q)$ is a period.
- If p and q are "small", then $d = \gcd(p, q)$ is a $O(k^2)$-period.
  - At most $O(k^2)$ of the substrings $S[1:d], S[d+1:2d], S[2d+1:3d]$, can be different

Structural Results

❖ If p and q are "small", then $d = \gcd(p, q)$ is a $O(k^2)$-period.

If there are at most k indices i such that $S[i] \neq S[i+p]$, and at most k indices j such that $S[j] \neq S[j+q]$, then there are at most $O(k^2)$ indices l such that $S[l] \neq S[l+d]$.

❖ Consider the indices as a grid.

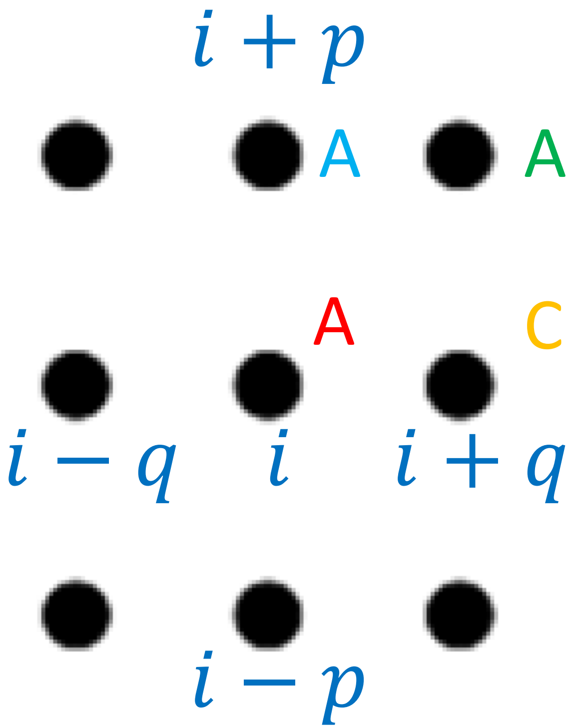


Structural Results

...AABAAABCCAA...

$p = 3, q = 7$

- Bound the number of indices l such that $S[l] \neq S[l + d]$.



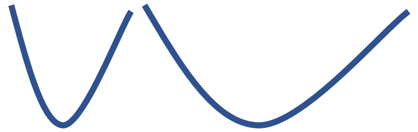
Structural Results

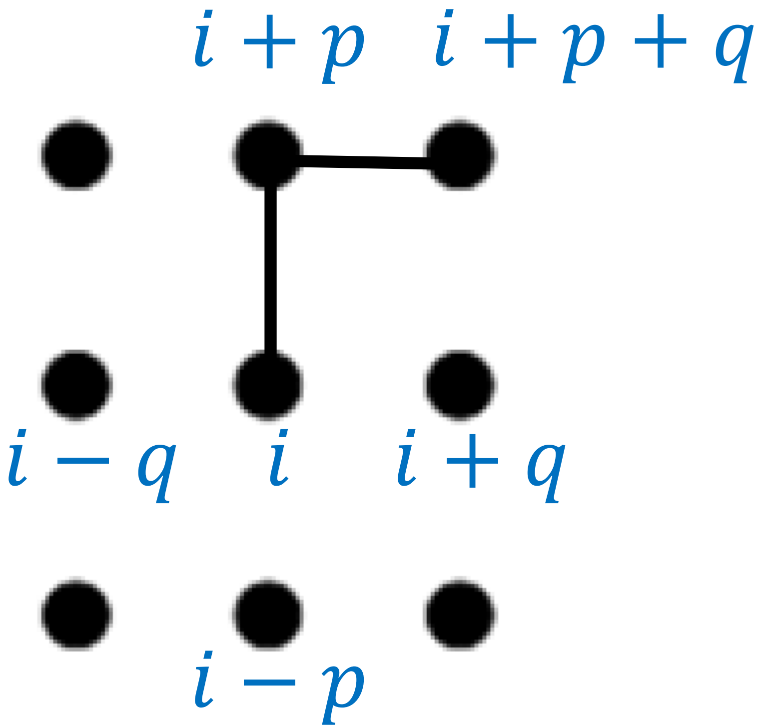
- Connect adjacent points with edges.
- “Good edge” if $S[i] = S[i+p]$.
- “Bad edge” if $S[i] \neq S[i+p]$.
- If there exists a path from i to j which “hops” along good edges, then $S[i] = S[j]$.

…AABAAABCCAA…

$p = 3, q = 7$

…AABAAABCCAA…

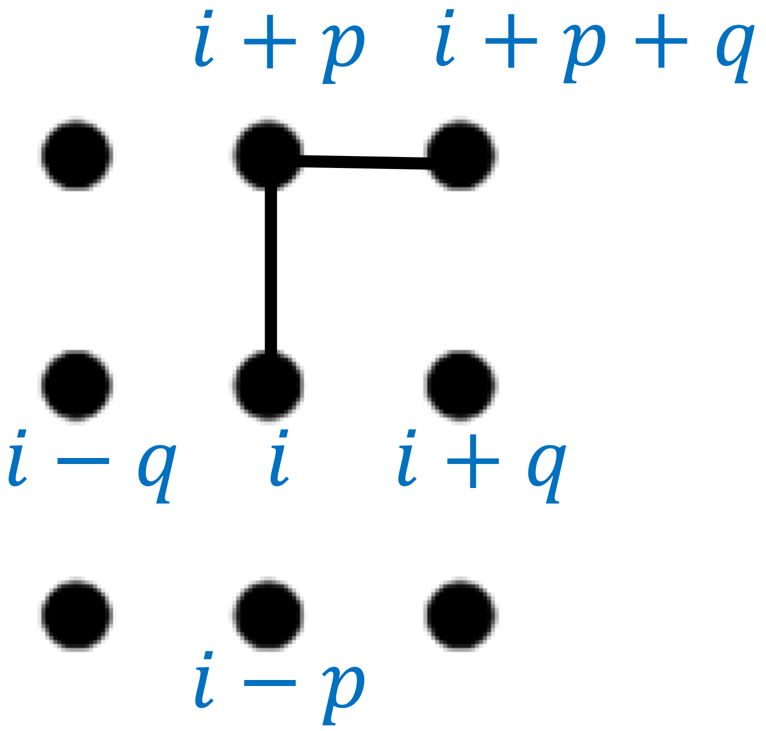




Structural Results

If there are at most k indices i such that $S[i] \neq S[i+p]$, and at most k indices j such that $S[j] \neq S[j+q]$, then there are at most $O(k^2)$ indices l such that $S[l] \neq S[l+d]$.

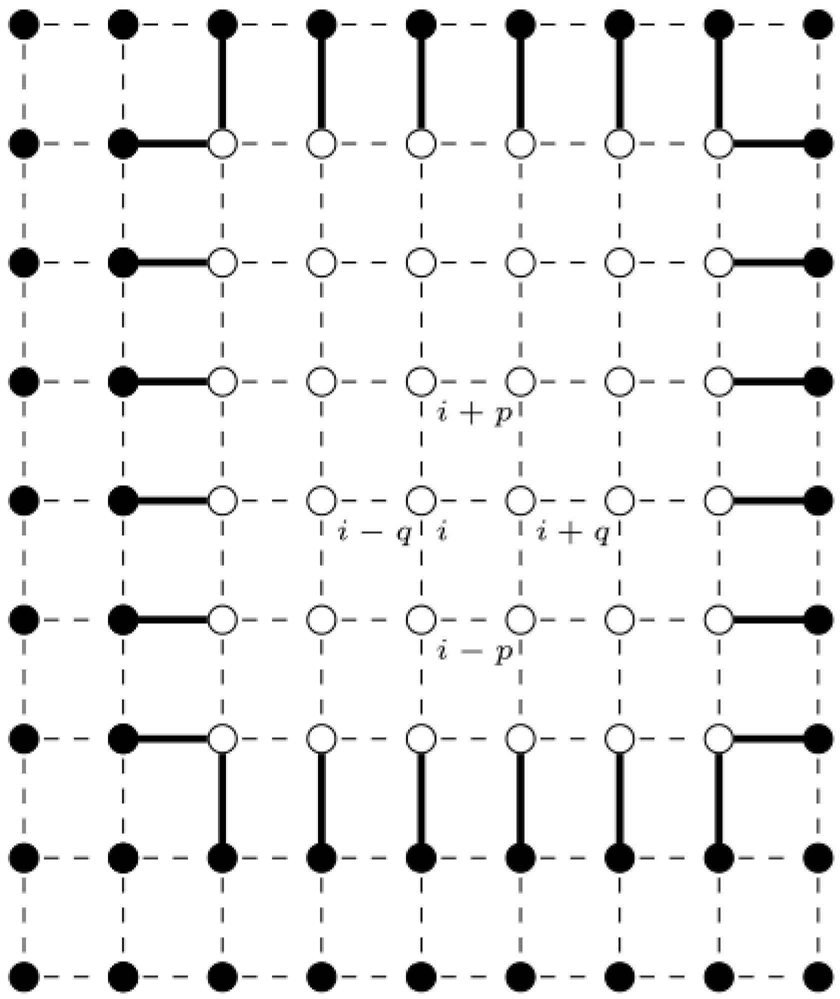
- Bound the number of indices l such that $S[l] \neq S[l+d]$.
- If $S[l] \neq S[l+d]$, then l must be enclosed by bad edges.
- There are at most $2k$ bad edges.
- How many enclosed points can there be?



Structural Results

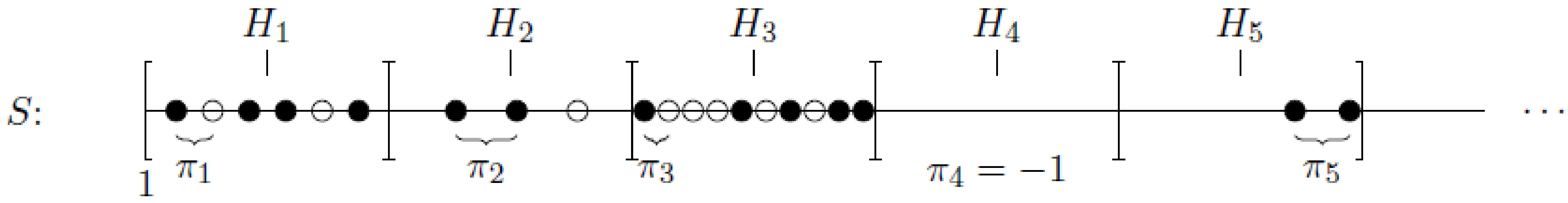
❖ If there are at most $2k$ bad edges, there are $O(k^2)$ enclosed points.

❖ There are $O(k^2)$ indices l such that $S[l] \neq S[l+d]$.



In review

- If p and q are “small”, then $d = \gcd(p, q)$ is a $O(k^2)$-period.
- Positions $p = \{8,16,20,27,30,39,45,55\}$?
- Positions $p = \{8,12,16,20\}, \{27,30,33,36,39\}, \{45,50,55\}$



In review

- First pass: Find all positions p such that

$$\mathrm{HAM}\left(S\left[1{:}\frac{n}{2}\right], S\left[p+1, p+\frac{n}{2}\right]\right) \le k.$$

- Second pass: For each p, check if

$$\mathrm{HAM}(S[1{:}n-p], S[p+1, n]) \le k.$$

Open Problems

- What can we say about these problems with other distance metrics (particularly, edit distance)?
- Can we improve the space usage? Specifically, the k^4 dependence comes from the structural property and the k-Mismatch Problem algorithm.
- Can we find the longest d-near-alignment in space $o(d^2)$?
- Longest palindromic subsequence

Questions?


thank you
danke
謝謝
ngiyabonga
teşekkür ederim
рахмат
Баярлалаа
спасибо
faafetai lava
mersi
kia ora
barka
welalin
tack
spas
dank je
misaotra
matondo
paldies
grazzi
mahalo
tapadh leat
gracias
vinaka
спасибі
blagodaram
kiitos
dankie
nanni
nandri
dhanyavad
bedankt
bayarlalaa
gràçie
hvala
maururu
köszönöm
enkosi
dziękuję
akun
dankon
ačiū
хвала
asante
manana
obrigada
murakoze
tenki
djiere dieuf
tau
mochchakkeram
дякую
mamnun
chokrane
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
go raibh maith agat
sobodi
děkuji
obrigado
mèsi
sagolun
sukriya
kop khun krap
ありがとう
grazie
arigatô
takk
dakujem
trugarez
didi madloba
najis tuke
kam sah hamnida
rahmat
terima kasih
tanemirt
rahmet
diolch
dhanyavadagalu
shukriya
mercé
мерси
তোমাকে ধন্যবাদ
감사합니다
xièxie
ευχαριστώ
merci