Streaming for Aibohphobes: Longest Near-Palindrome under Hamming Distance

Elena Grigorescu, Purdue University

Erfan Sadeqi Azer, Indiana University

Samson Zhou, Purdue University

Structure of Talk

- Background
- 1-Pass Additive Algorithm
- 2-Pass *Exact* Algorithm
- Lower Bounds

Finding Structure in Noisy Data

FSTTCSIITKAN

ALPPATTERNS

FSTTCSPATTERNIITKANPURINDIAO

STREAMINGALGORITHMPATTERNU

PERIODPERIODPERIODPERIODPER

FSTTCSTHEORYCSASBRICBCAUON

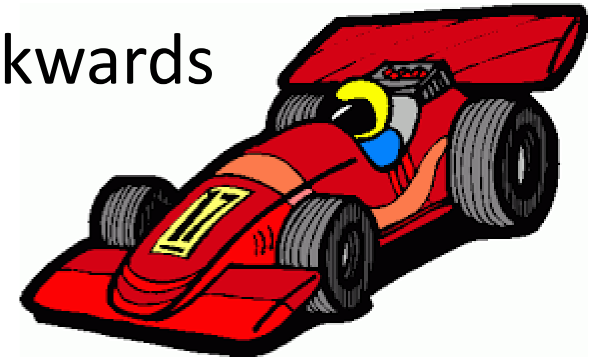
LONGPALINDROMEEMORDNILAPGN

OLFSTTCSIITKANPURINDIAGENXAS

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
- $\mathrm{HAM}(S, S^R) = 2$

Streaming Model

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

abaacabaccbabbbcbabbccababbccb

abaacabaccbabbbcbabbccababbccb

abaacabaccbabbbcbabbccababbccb

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 $o(n)$.

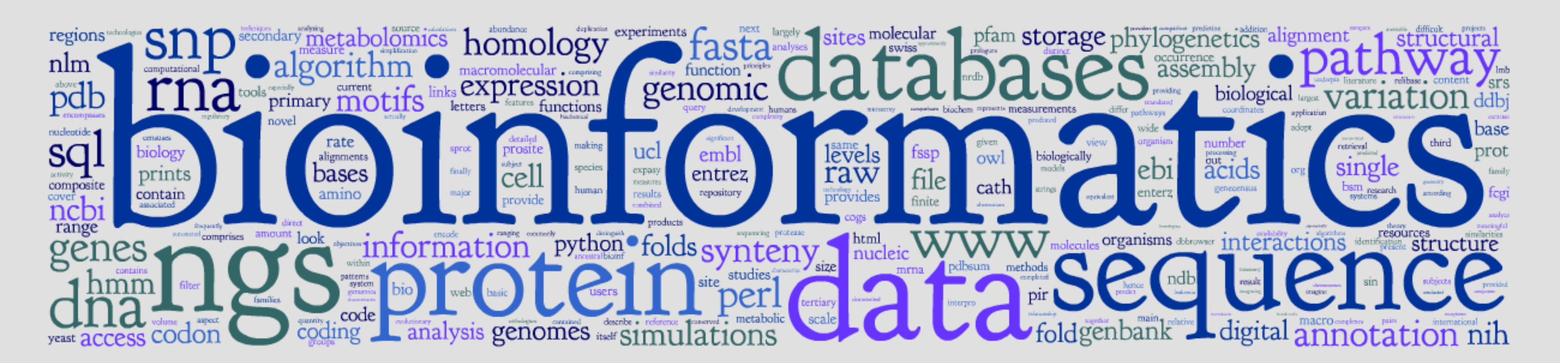
- ❖ Given a string S of length n, which arrives in a data stream, find a "long" d-near-palindrome in space $o(n)$.

Applications

TGCTTAAGCGCTTGCAAGCGCTTAAGCA

CAAGCGCTTAAGCA

ACGAATTCGCGAAC



Related Work (Palindromes in Data Streams)

- $O(\log n)$ 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 (Gawrychowski,Merkurev,Shur,Uznanski’16)
- $\Omega\left(\frac{n}{E}\right)$ 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)	$\Omega\left(\frac{dn}{E}\right)$

Structure of Talk

- Background
- 1-Pass Additive Algorithm
- 2-Pass *Exact* Algorithm
- Lower Bounds

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

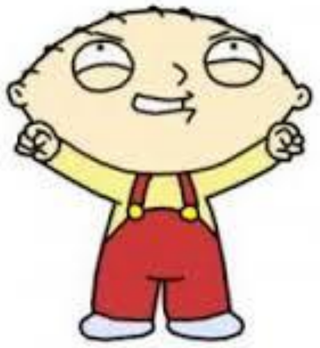
- $\phi(S[1:y]) = \phi(S[1:x]) + B^x \phi(S[x:y])$ (concatenation)
- Define $\phi^R(S) = \sum_{i=1}^{n} B^{-i} S[i] \ (mod\ P)$ (reversal)
- $\phi(S^R[1:x]) = B^{x+1} \phi^R(S[1:x])$
- $\phi^R(S[1:y]) = \phi^R(S[1:x]) + B^{-x} \phi^R(S[x:y])$
- Can be computed on the fly

Identifying Palindromes

- 111101011100001010010101001111101011100001010010101001
- 111101011100001010010101001111101011100001010010101001

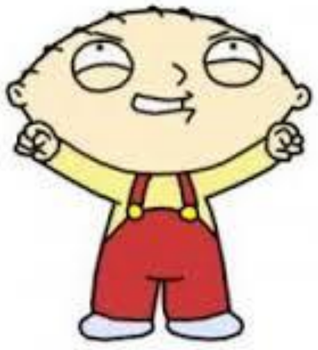
Identifying Near-Palindromes?

- 111101011100001010010101001111101011100001010010101001
- 111101011100001010010101001111101011100001010010101001



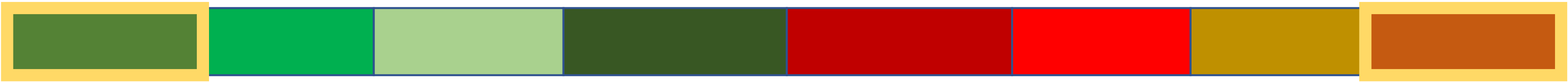
Identifying Near-Palindromes?

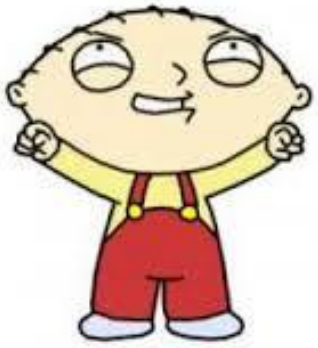
- 111101011100001010010101001111101011100001010010101001
- 111101011100001010010101001111101011100001010010101001



Identifying Near-Palindromes?

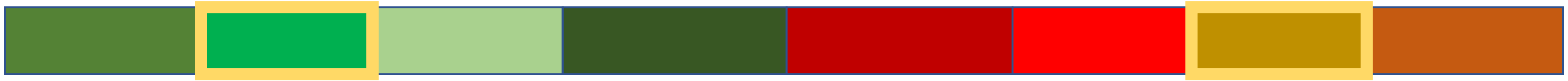
- 111101011100001010010101001111101011100001010010101001
- 111101011100001010010101001111101011100001010010101001

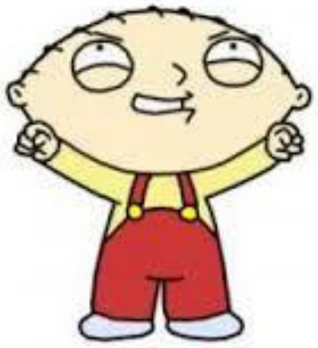




Identifying Near-Palindromes?

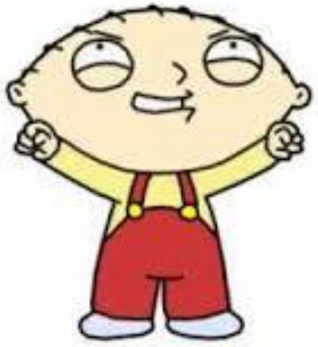
- 111101011100001010010101001111101011100001010010101001
- 111101011100001010010101001111101011100001010010101001



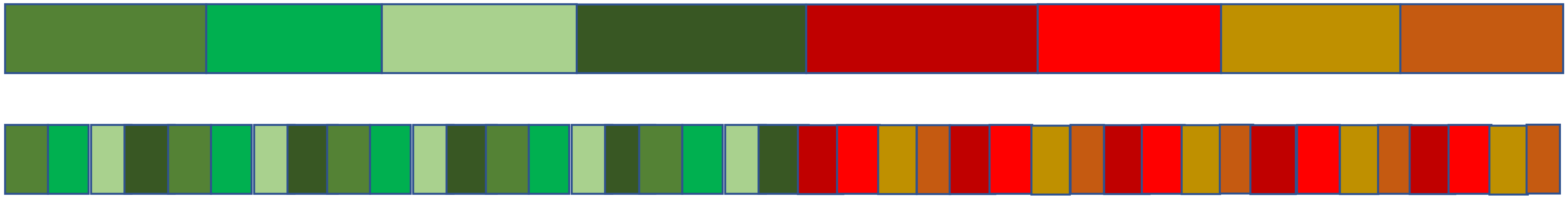


Identifying Near-Palindromes?

- 111101011100001010010101001111101011100001010010101001
- 111101011100001010010101001111101011100001010010101001

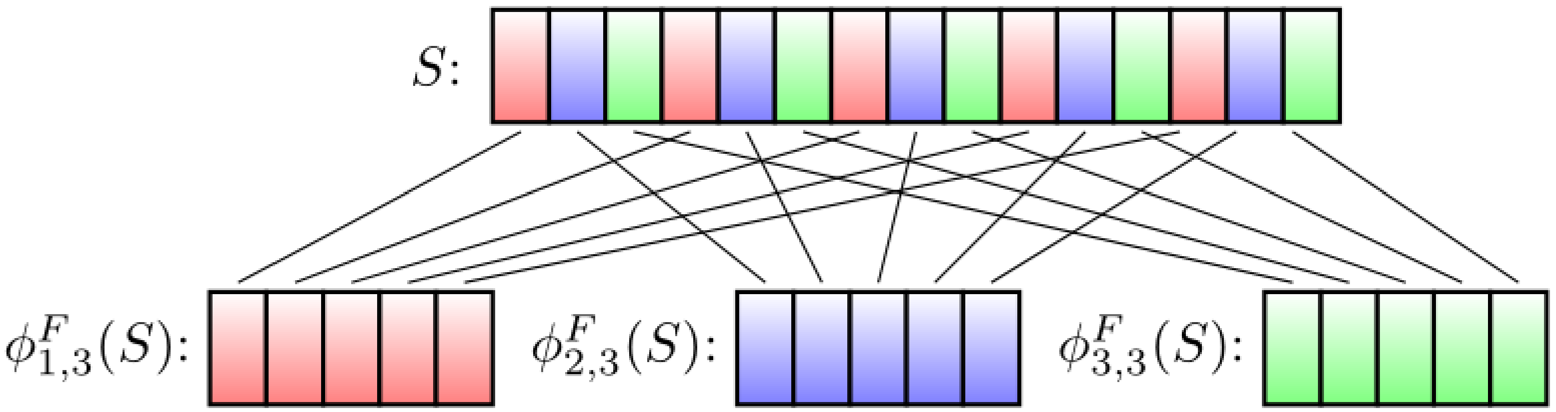


Identifying Near-Palindromes? (CFP+16)



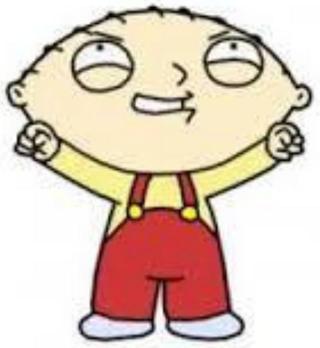
Karp-Rabin Fingerprints for Subpatterns

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



Identifying Near-Palindromes?

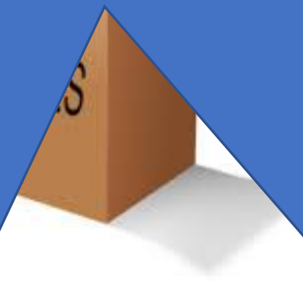
- Let $\Delta = \#\{a \mid \phi_{a,b}(S) \neq B^k \phi^R_{a,b}(S)\}$
- Then $\Delta \leq \mathrm{HAM}(S, S^R)$



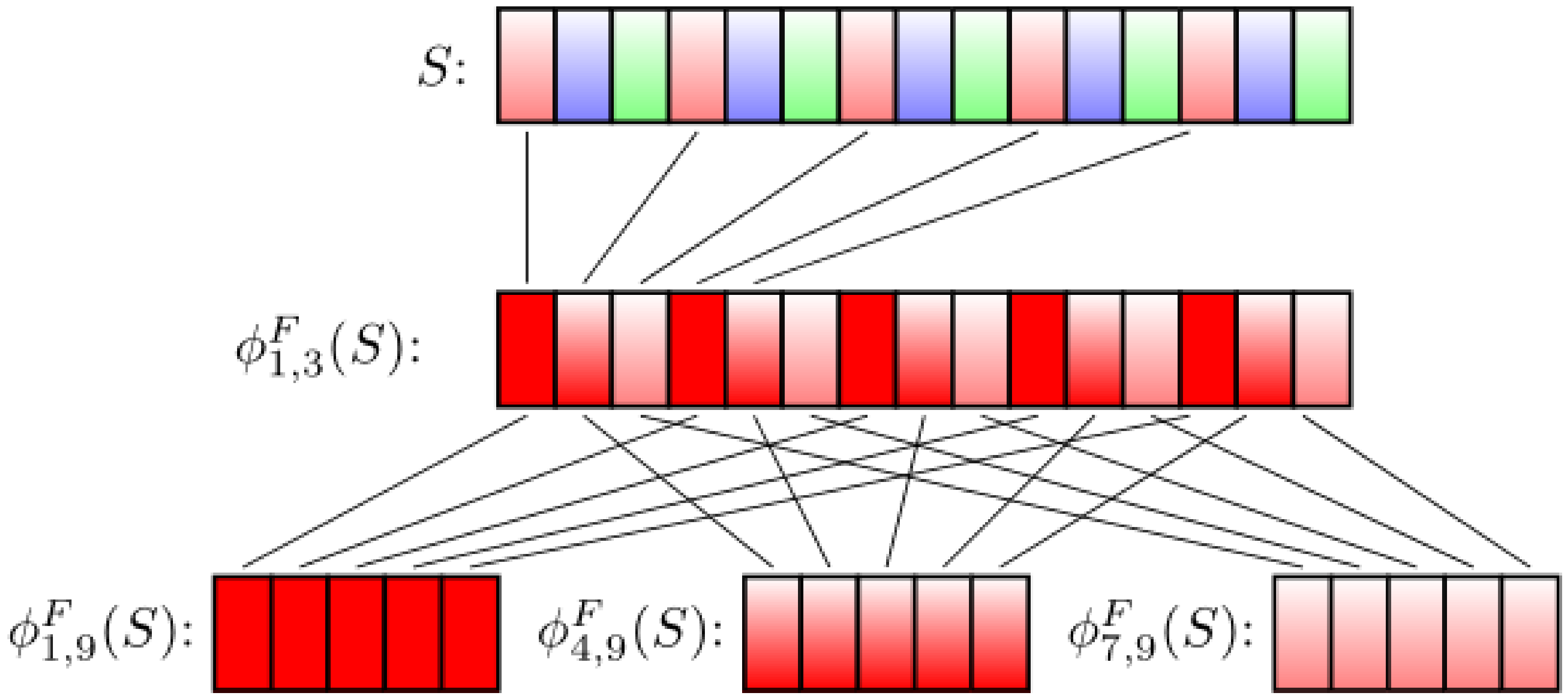
Identifying Near-Palindromes?

- Sample ... primes p_1, p_2 ... from $[1$ ... $g^2\, n, 544\, d \log^2 n]$.
- Let $\Delta = \max$ ... $($ ...
- $\Delta \leq \mathrm{HAM}(S, S^R)$ ...
- If $\mathrm{HAM}(S,$ ... 16)

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



Karp-Rabin Fingerprints for Sub-Subpatterns



Second-Level Karp-Rabin Fingerprints

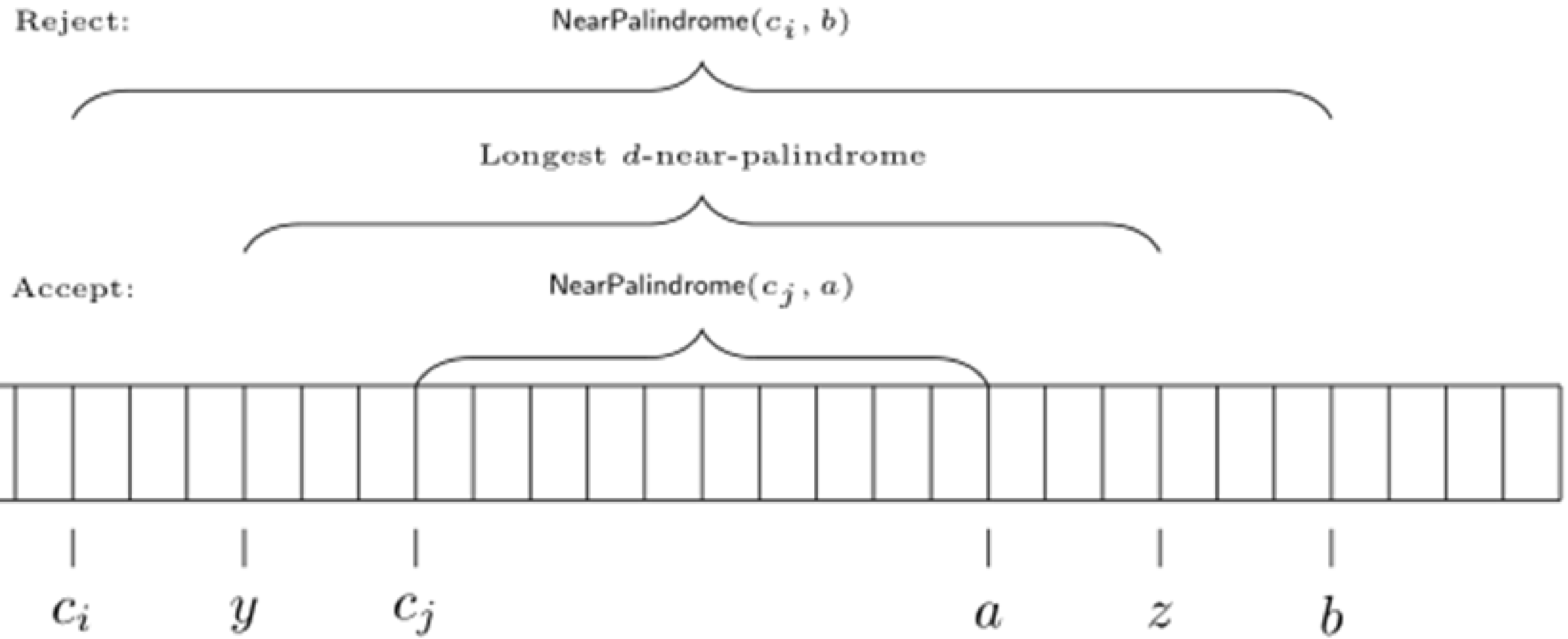
- Call a mismatch *isolated* under p_i if it is the only mismatch under some subpattern S_{a,p_i}. Let I be the number of isolated mismatches.
- If $\mathrm{HAM}(S, S^R) \leq 2d$, then $I = \mathrm{HAM}(S, S^R)$ w.h.p. (CFP+16)

In Review

- There exists a data structure of size $O(d \log^6 n)$ bits that recognizes whether $\mathrm{HAM}(S, S^R) \leq d$ w.h.p.
- Recently, this has been improved to $O(d \log n)$. (Clifford, Kociumaka, Porat '17)
- Through black-box reduction, improves our results by $O(\log^5 n)$.

Additive Error Algorithm

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



Additive Error Algorithm

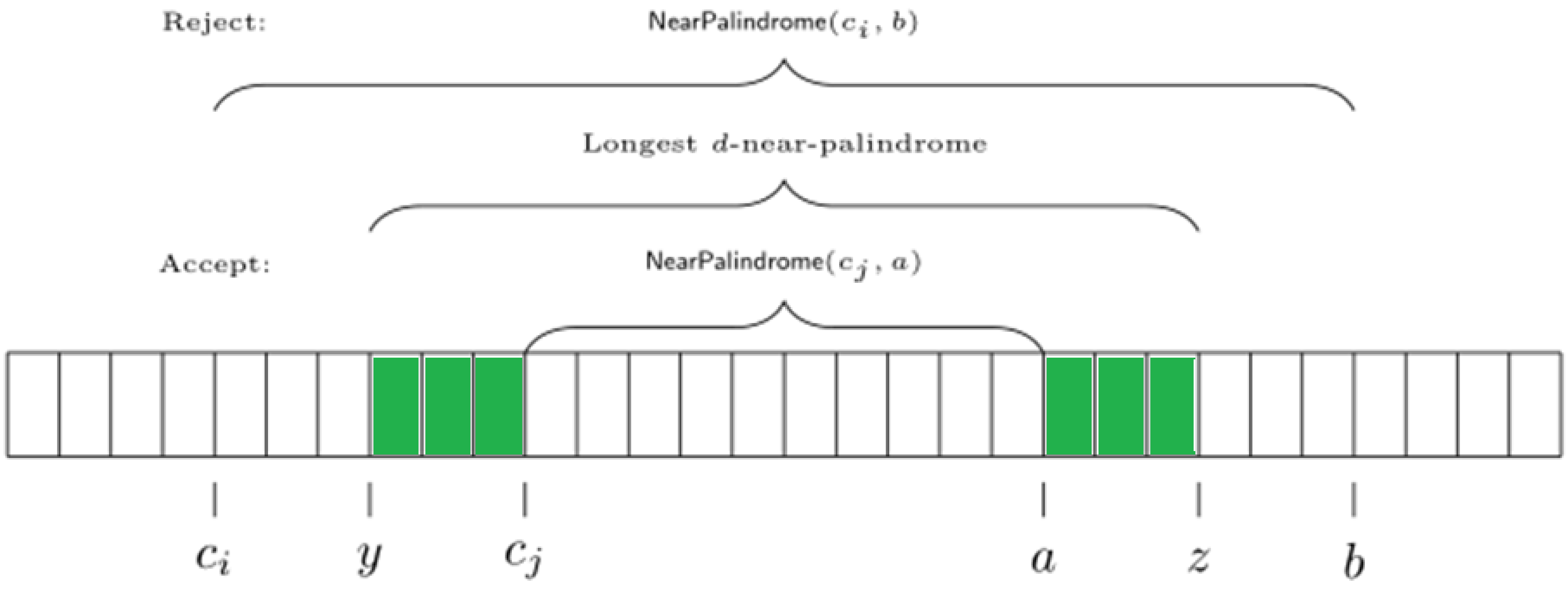
- $2\sqrt{n}$ sketches, each of size $O(d \log^6 n)$ bits
- Total space: $O(d\sqrt{n} \log^6 n)$ bits

Structure of Talk

- Background
- 1-Pass Additive Algorithm
- 2-Pass *Exact* Algorithm
- Lower Bounds

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 $\Omega(n)$ candidates?

- Structural result of palindromes (BEMS14)

Structural Result of Near-Palindromes

❖ Goal #1: Recover fingerprints of all overlapping “long” near-palindromes

❖ Goal #2: Use sublinear space in compression

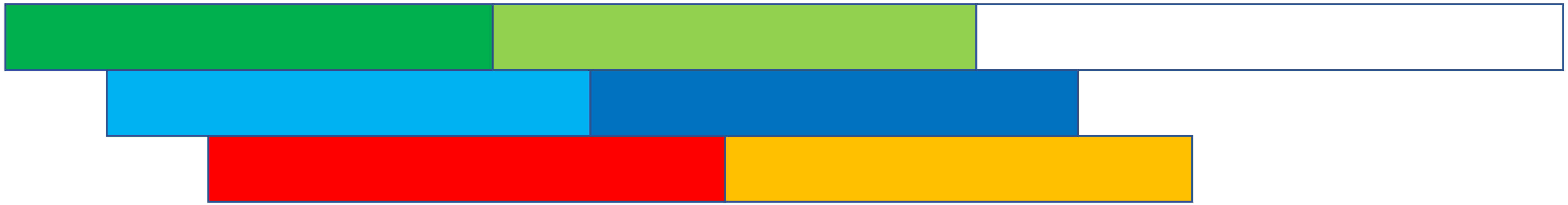
Structural Result of Near-Palindromes

❖ Goal #1: Recover fingerprints of all overlapping “long” near-palindromes

❖ Goal #2: Use sublinear space in compression

Structural Result of Near-Palindromes

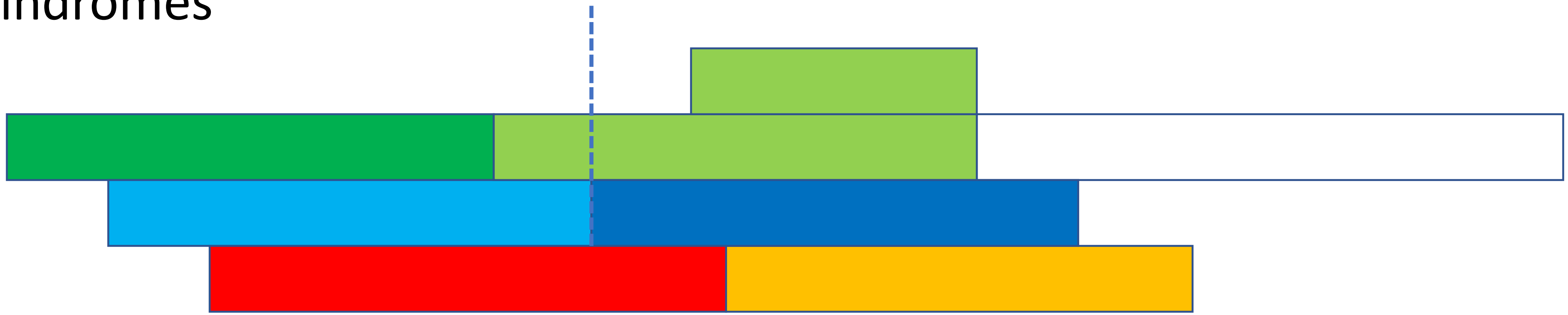
- Goal #1: Recover fingerprints of all overlapping “long” near-palindromes



- Goal #2: Use sublinear space in compression

Structural Result of Near-Palindromes

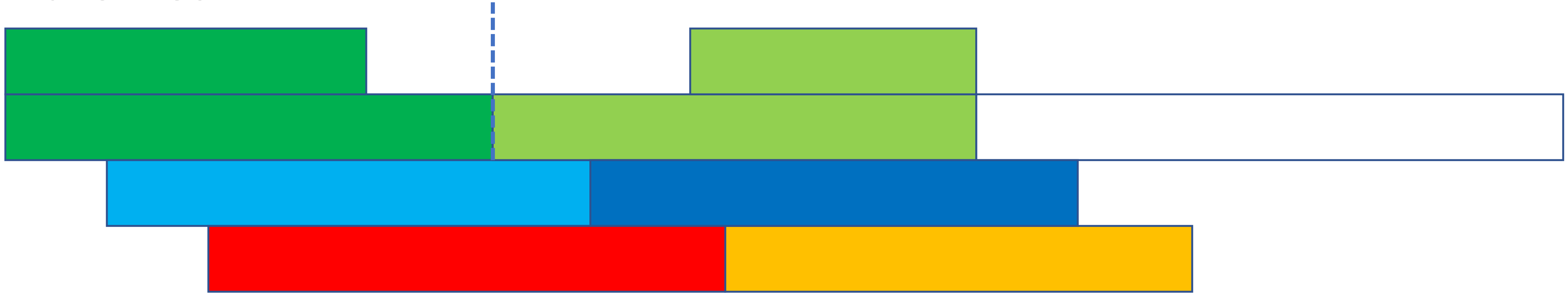
- ❖ Goal #1: Recover fingerprints of all overlapping “long” near-palindromes



- ❖ Goal #2: Use sublinear space in compression

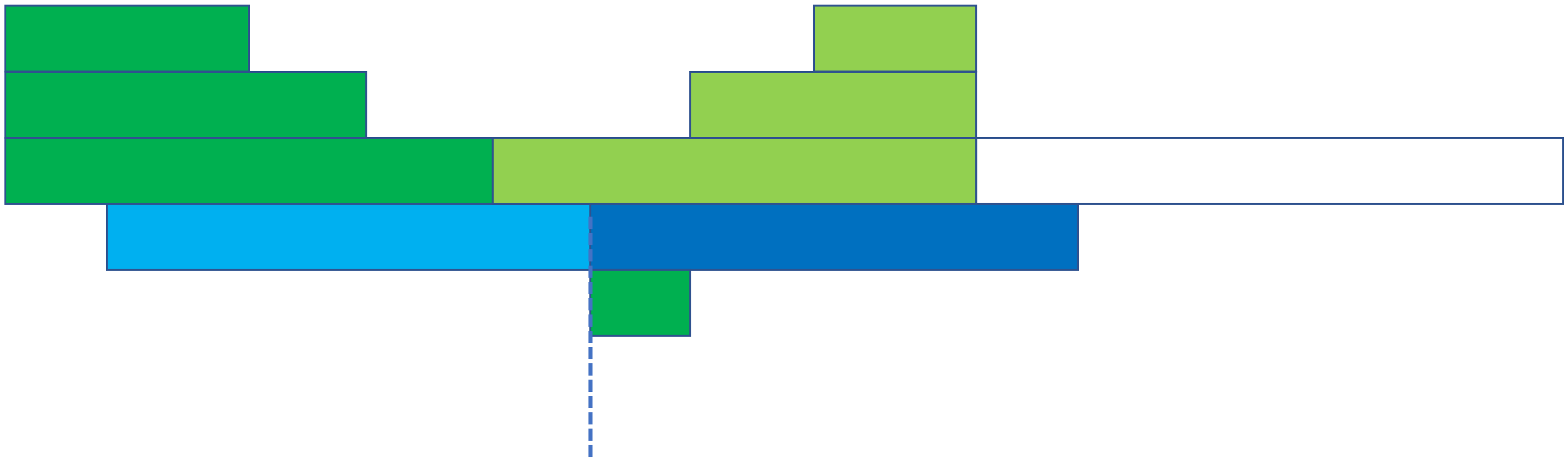
Structural Result of Near-Palindromes

- Goal #1: Recover fingerprints of all overlapping "long" near-palindromes



- Goal #2: Use sublinear space in compression

Structural Result of Near-Palindromes



Structural Result of Near-Palindromes

- ❖ Goal #1: Recover fingerprints of all overlapping “long” near-palindromes

- ❖ Goal #2: Use sublinear space in compression

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

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

Structure of Talk

- Background
- 1-Pass Additive Algorithm
- 2-Pass *Exact* Algorithm
- Lower Bounds

Multiplicative Lower Bounds

- Yao's Principle: find "hard" distribution for deterministic algorithms
- Let v be the prefix of $101100111000011110000\ldots = 1^1 0^1 1^2 0^2 \ldots$ 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 $s(x, y) = v^R x y^R v$.

Multiplicative Lower Bounds

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

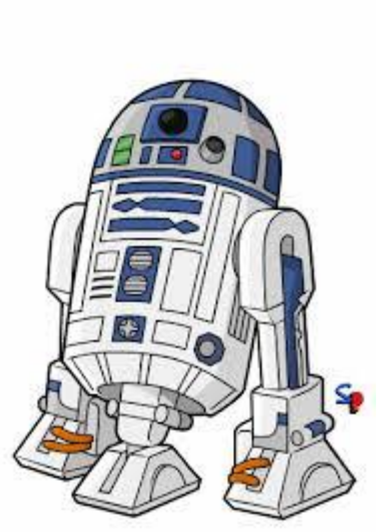
NO:
If $\mathrm{HAM}(x,y) > d$,
then the longest
d-near-palindrome of
$s(x,y)$ has length at
most $200d^2 + \frac{n}{2}$.

Multiplicative Lower Bounds

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

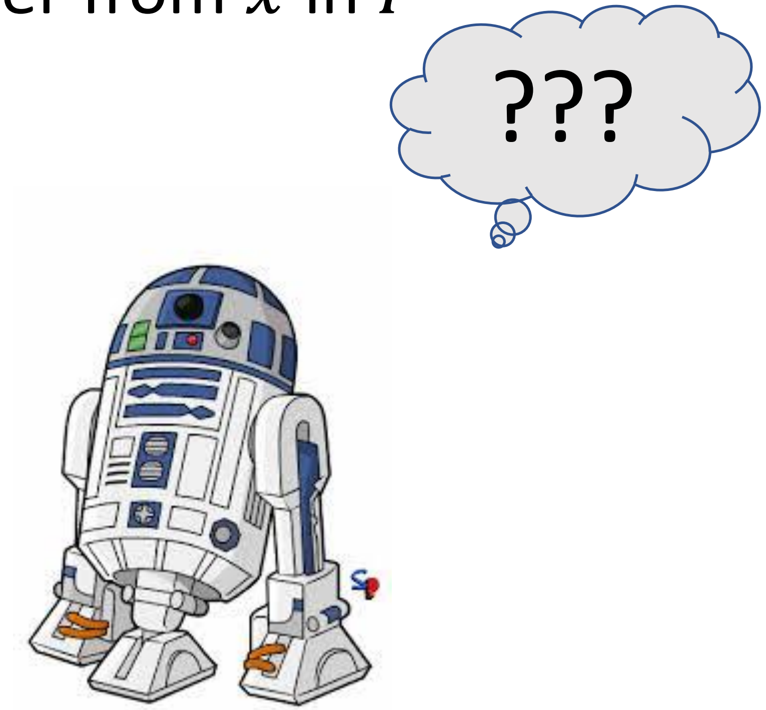
Multiplicative Lower Bounds

- 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: 1011000000100010000000100100000
- x': 1000000100101010000000100100000
- y: 1111011000100010111001001000010
- 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 $y \in Y = \{\mathrm{HAM}(x,y) = d \text{ or } \mathrm{HAM}(x,y) = d+1\}$

Open Problems

- Can we find the longest d-near-palindrome in the *edit* distance?
- Longest palindromic subsequence


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

Questions?