Fine-Grained Complexity - Lecture 8:

SETH-Hardness for Longest Common Subsequence

Sequences (a.k.a. strings):

An ordered set of n symbols over some alphabet Σ.

Examples:

DNA, text, programs, machine code, curves (e.g. GPS trajectories), time-series (e.g. audio), …

20.12.21

Fine-Grained Complexity - Lecture 8:

SETH-Hardness for Longest Common Subsequence

(and other similarity measures)

Sequences (a.k.a. strings):

An ordered set of n symbols over some alphabet Σ.

Examples:

DNA, text, programs, machine code, curves (e.g. GPS trajectories), time-series (e.g. audio), …

20.12.21

Fine-Grained Complexity - Lecture 8:

SETH-Hardness for Longest Common Subsequence

(and "better" SETH)

(and other similarity measures)

Sequences (a.k.a. strings):

An ordered set of n symbols over some alphabet Σ.

Examples:

DNA, text, programs, machine code, curves (e.g. GPS trajectories), time-series (e.g. audio), …

20.12.21

Longest Common Subsequence (**LCS**)

Input: two sequences of length n

S = cddcabbabcbaa

T = adbdbbcabacdd

Output: the length of the longest common subsequence

Longest Common Subsequence (**LCS**)

Input: two sequences of length n

S = cddcabbabcbaa

T = adbdbbcabacdd

Output: the length of the longest common subsequence

Longest Common Subsequence (**LCS**)

Input: two sequences of length n

S = cddcabbabcbaa

T = adbdbbcabacdd

Output: the length of the longest common subsequence

Classic Dynamic Programming: $O(n^2)$ [Wagner-Fisher '74]

$$M[i,j] = \max \begin{cases} M[i-1, j-1] + (S[i] == T[j]), \\ M[i-1, j], \\ M[i, j-1] \end{cases}$$

Longest Common Subsequence (**LCS**)

Input: two sequences of length n

S = cddcabbabcbaa

T = adbdbbcabacdd

Output: the length of the longest common subsequence

Classic Dynamic Programming: $O(n^2)$ [Wagner-Fisher '74]

$$M[i,j] = \max \begin{cases} M[i-1,j-1] + (S[i] == T[j]), \\ M[i-1,j], \\ M[i,j-1] \end{cases}$$

[Masek - Paterson '80]

$$O\left(\frac{n^2}{\log n}\right)$$

Longest Common Subsequence (**LCS**)

Input: two sequences of length n

S = cddcabbabcbaa

T = adbdbbcabacdd

Output: the length of the longest common subsequence

Classic Dynamic Programming: *O(n²)* → $O(n^2)$ [Wagner-Fisher ’74]

$$M[i,j] = \max \begin{cases} M[i-1,j-1] + (S[i] == T[j]), \\ M[i-1,j], \\ M[i,j-1] \end{cases}$$

[Masek - Paterson ’80]

$$O\left(\frac{n^2}{\log n}\right)$$

Can we do better?

How similar are two sequences?

The following popular similarity measures can be computed in $O(n^2)$ time with dynamic programming:

- **LCS**
- **Edit-Distance**
- **Local Alignment (BLAST)**
- **Frechet**
- **Dynamic Time Warping**

And cannot be computed in $O(n^{1.99})$ time assuming SETH.

How similar are two sequences?

The following popular similarity measures can be computed in $O(n^2)$ time with dynamic programming:

- **LCS**
- **Edit-Distance**
- **Local Alignment (BLAST)** [A. - Vassilevska W. - Weimann ICALP'14]
- **Frechet**
- **Dynamic Time Warping**

And cannot be computed in $O(n^{1.99})$ time assuming SETH.

How similar are two sequences?

The following popular similarity measures can be computed in $O(n^2)$ time with dynamic programming:

- **LCS**
- **Edit-Distance**
- **Local Alignment (BLAST)** [A. - Vassilevska W. - Weimann ICALP'14]
- **Frechet** [Bringmann FOCS'14]
- **Dynamic Time Warping**

And cannot be computed in $O(n^{1.99})$ time assuming SETH.

How similar are two sequences?

The following popular similarity measures can be computed in $O(n^2)$ time with dynamic programming:

- **LCS**
- **Edit-Distance** [Backurs - Indyk STOC'15]
- **Local Alignment (BLAST)** [A. - Vassilevska W. - Weimann ICALP'14]
- **Frechet** [Bringmann FOCS'14]
- **Dynamic Time Warping**

And cannot be computed in $O(n^{1.99})$ time assuming SETH.

A celebrated result!

[Backurs-Indyk STOC'15]:

"$\forall \varepsilon > 0 : O(n^{2-\varepsilon})$ for **Edit Distance** refutes SETH"

The Boston Globe

Menu | Ideas | SPOTLIGHT The story behind Spotlight in our free e-book | Subscribe Starting at 99 cents | Members Sign In

BRAINIAC

For 40 years, computer scientists looked for a solution that doesn't exist

24

Top 10 Trending Articles

Most Viewed | Most Commented | Most Shared

- Words matter in 'ISIS' war, so use 'Daesh'
- Governor Charlie Baker is opposed to allowing Syrian refugees in Mass. for now
- Paris and the fall of Rome
- Chris Kimball to leave America's Test Kitchen after contract dispute
- Paris attacks: Baker, others wrong to reject Syrian refugees
- Paris attacks | Andrew J. Bacevich: A war the West cannot win
- A Patriots-Giants rematch in February: You can't beat that
- How will Patriots fill void while Julian Edelman is out?

How similar are two sequences?

The following popular similarity measures can be computed in $O(n^2)$ time with dynamic programming:

- **LCS** [A. - Backurs - Vassilevska W. FOCS'15] [Bringmann - Kunnemann FOCS'15]
- **Edit-Distance** [Backurs - Indyk STOC'15]
- **Local Alignment (BLAST)** [A. - Vassilevska W. - Weimann ICALP'14]
- **Frechet** [Bringmann FOCS'14]
- **Dynamic Time Warping** [A. - Backurs - Vassilevska W. FOCS'15] [Bringmann - Kunnemann FOCS'15]

And cannot be computed in $O(n^{1.99})$ time assuming SETH.

Today:

[A. - Backurs - Vassilevska W. FOCS'15] and [Bringmann - Kunnemann FOCS'15]:

"$\forall \varepsilon > 0 : O(n^{2-\varepsilon})$ for **LCS** refutes SETH"

Today:

[A. - Backurs - Vassilevska W. FOCS'15] and [Bringmann - Kunnemann FOCS'15]:

"$\forall \varepsilon > 0 : O(n^{2-\varepsilon})$ for **LCS** refutes SETH"

The Reduction:

OV → **LCS**

Today:

[A. - Backurs - Vassilevska W. FOCS'15] and [Bringmann - Kunnemann FOCS'15]:

"$\forall \varepsilon > 0 : O(n^{2-\varepsilon})$ for **LCS** refutes SETH"

The Reduction:

OV $\longrightarrow$ **LCS**

OV: Given $A, B \subseteq \{0,1\}^d$, $|A| = |B| = n$,
is there $a \in A$, $b \in B$ s.t. $\langle a, b \rangle = 0$?

Last time: $\forall \varepsilon > 0 : O(n^{2-\varepsilon} \cdot \text{poly}(d))$
for OV refutes SETH.

Today:

[A. - Backurs - Vassilevska W. FOCS'15] and [Bringmann - Kunnemann FOCS'15]:

"$\forall \varepsilon > 0 : O(n^{2-\varepsilon})$ for **LCS** refutes SETH"

The Reduction:

OV → **LCS**

An instance of OV on two sets of n vectors of dimension d

$O(n')$ time

An instance of LCS on two strings of length $n' = O(n \cdot \text{poly}(d))$

OV: Given $A, B \subseteq \{0,1\}^d$, $|A| = |B| = n$, is there $a \in A$, $b \in B$ s.t. $\langle a, b \rangle = 0$?

Last time: $\forall \varepsilon > 0 : O(n^{2-\varepsilon} \cdot \text{poly}(d))$ for OV refutes SETH.

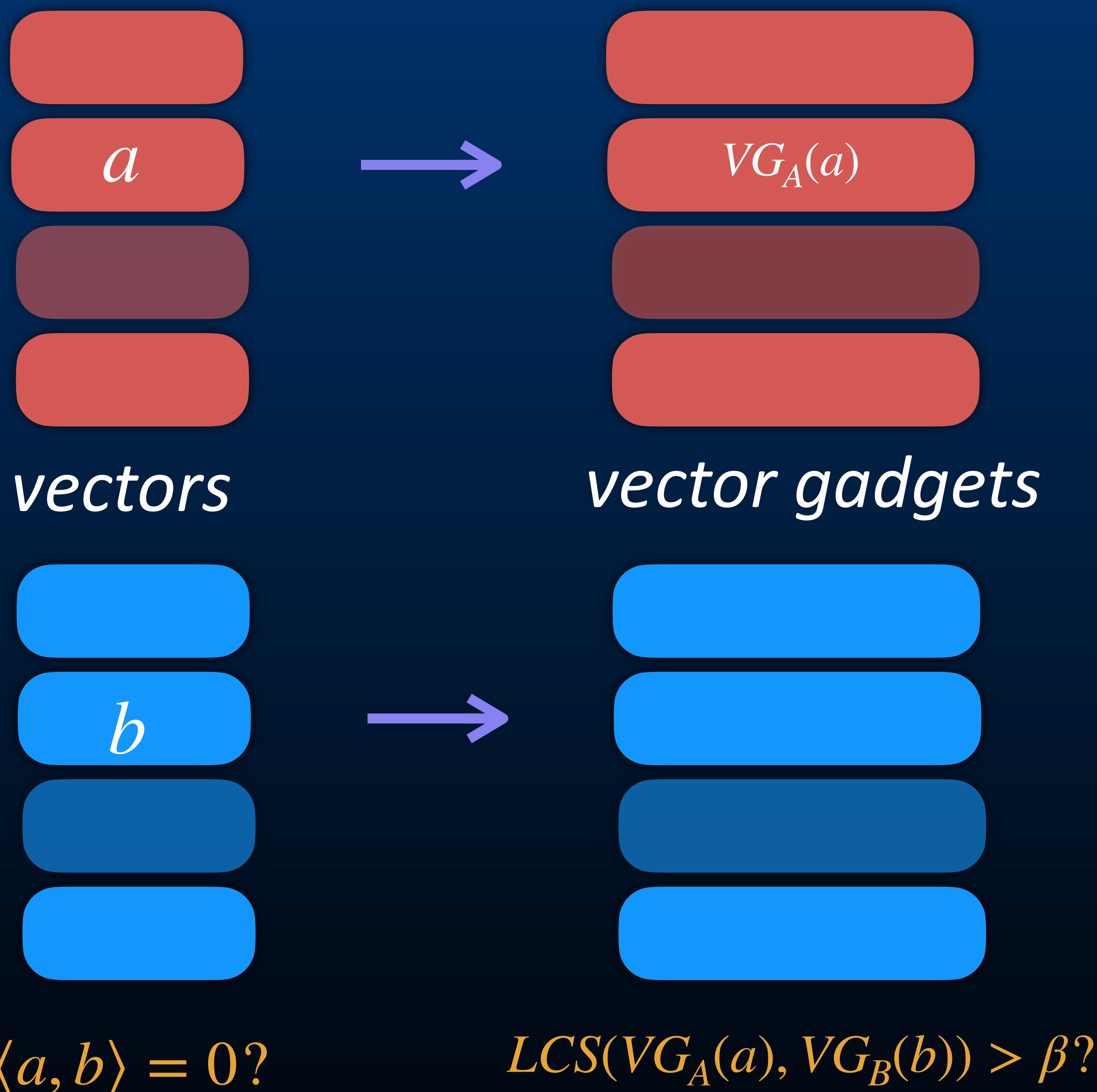
Reduction Plan:

OV: Given $A, B \subseteq \{0,1\}^d$, $|A| = |B| = n$, is there $a \in A$, $b \in B$ s.t. $\langle a, b \rangle = 0$?

$\langle a, b \rangle = 0?$

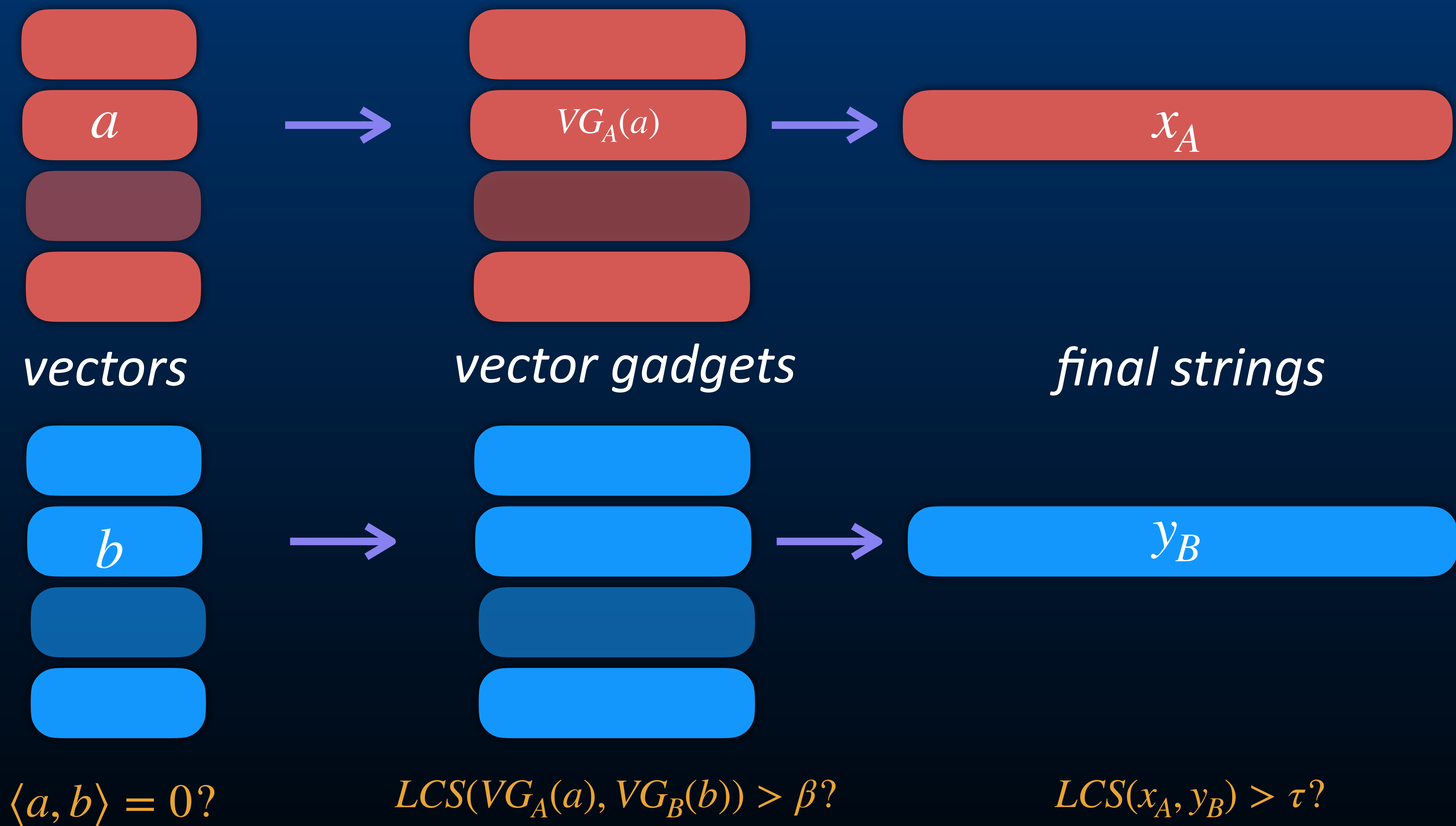
Reduction Plan:

OV: Given $A, B \subseteq \{0,1\}^d$, $|A| = |B| = n$, is there $a \in A$, $b \in B$ s.t. $\langle a, b \rangle = 0$?



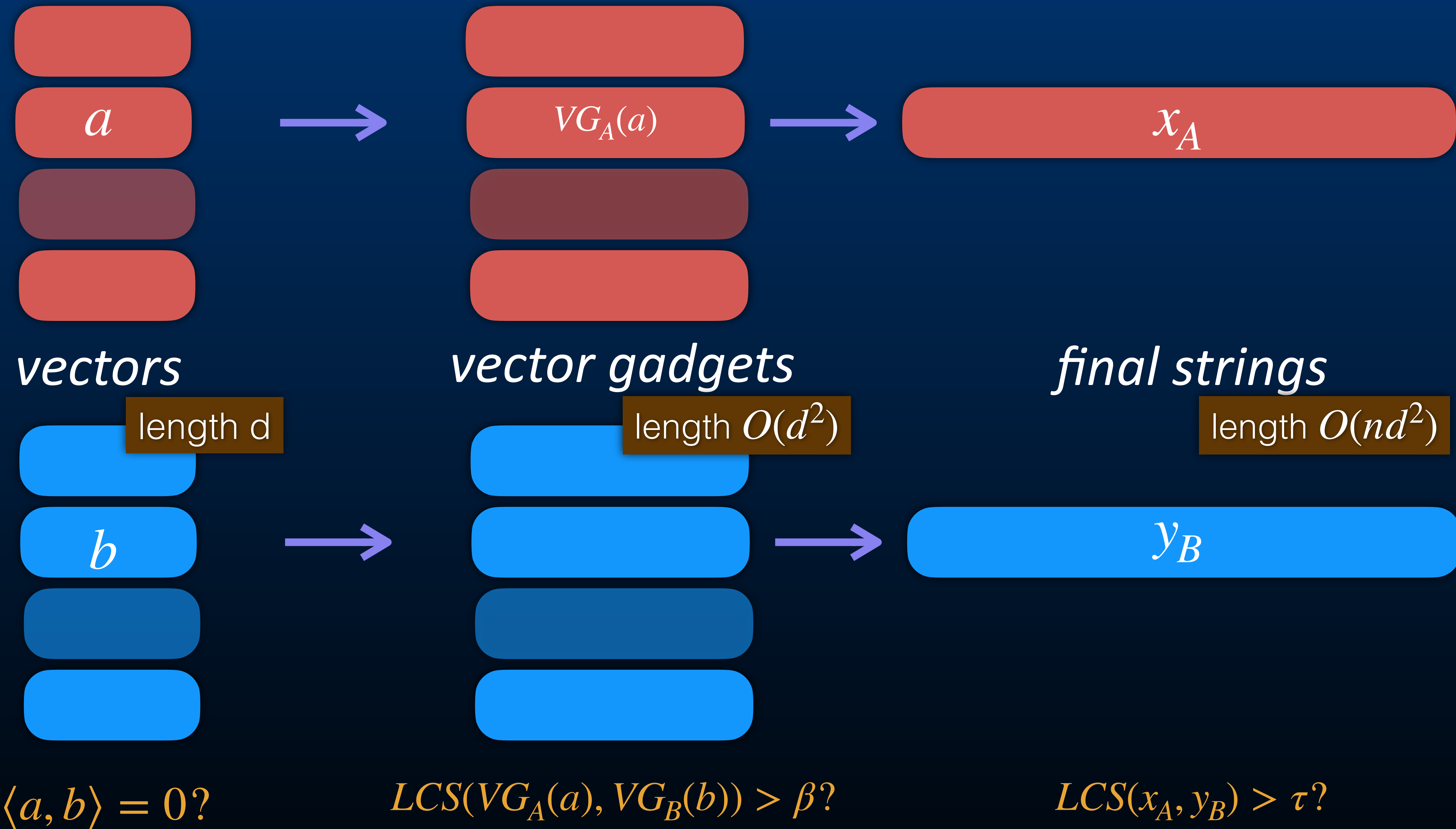
Reduction Plan:

OV: Given $A, B \subseteq \{0,1\}^d$, $|A| = |B| = n$, is there $a \in A$, $b \in B$ s.t. $\langle a, b \rangle = 0$?



Reduction Plan:

OV: Given $A, B \subseteq \{0,1\}^d$, $|A| = |B| = n$, is there $a \in A$, $b \in B$ s.t. $\langle a, b \rangle = 0$?



Step 0: **Coordinate Gadgets**

Lemma 0: There are atomic strings $CG_A(0), CG_A(1)$ and $CG_B(0), CG_B(1)$ of length 3 s.t. for any $s, t \in \{0,1\}$:

$s \cdot t = 0$	$\Longrightarrow$	$LCS(CG_A(s), CG_B(t)) = 2$	(Large LCS)
$s \cdot t \neq 0$	$\Longrightarrow$	$LCS(CG_A(s), CG_B(t)) = 0$	(Small LCS)

Step 0: **Coordinate Gadgets**

Lemma 0: There are atomic strings $CG_A(0), CG_A(1)$ and $CG_B(0), CG_B(1)$ of length 3 s.t. for any $s, t \in \{0,1\}$:

$s \cdot t = 0 \implies LCS(CG_A(s), CG_B(t)) = 2$ (Large LCS)

$s \cdot t \neq 0 \implies LCS(CG_A(s), CG_B(t)) = 0$ (Small LCS)

Proof:

$CG^A(0) :=$

$CG^A(1) :=$

$CG^B(0) :=$

$CG^B(1) :=$

Step 0: **Coordinate Gadgets**

Lemma 0: There are atomic strings $CG_A(0), CG_A(1)$ and $CG_B(0), CG_B(1)$ of length 3 s.t. for any $s, t \in \{0,1\}$:

$s \cdot t = 0 \implies LCS(CG_A(s), CG_B(t)) = 2$ (Large LCS)

$s \cdot t \neq 0 \implies LCS(CG_A(s), CG_B(t)) = 0$ (Small LCS)

Proof:

$CG^A(0)$:= 001

$CG^A(1)$:= 111

$CG^B(0)$:= 011

$CG^B(1)$:= 000

Step 1: **Vector Gadgets**

Lemma 1: There are mappings $VG_A, VG_B : \{0,1\}^d \rightarrow \{0,1,2,3\}^{6d^2-d-2}$ computable in $O(d^2)$ time s.t. for any vectors $a, b \in \{0,1\}^d$:

$\langle a,b \rangle = 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d + 2$ (Large LCS)

$\langle a,b \rangle \neq 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d$ (Small LCS)

where $\beta_d := 3d^2 - d - 2$ is a fixed value.

Step 1: **Vector Gadgets**

Lemma 1: There are mappings $VG_A, VG_B : \{0,1\}^d \to \{0,1,2,3\}^{6d^2-d-2}$ computable in $O(d^2)$ time s.t. for any vectors $a, b \in \{0,1\}^d$:

$$\langle a,b \rangle = 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d + 2 \quad \text{(Large LCS)}$$

$$\langle a,b \rangle \neq 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d \quad \text{(Small LCS)}$$

where $\beta_d := 3d^2 - d - 2$ is a fixed value.

Attempts:

Step 1: **Vector Gadgets**

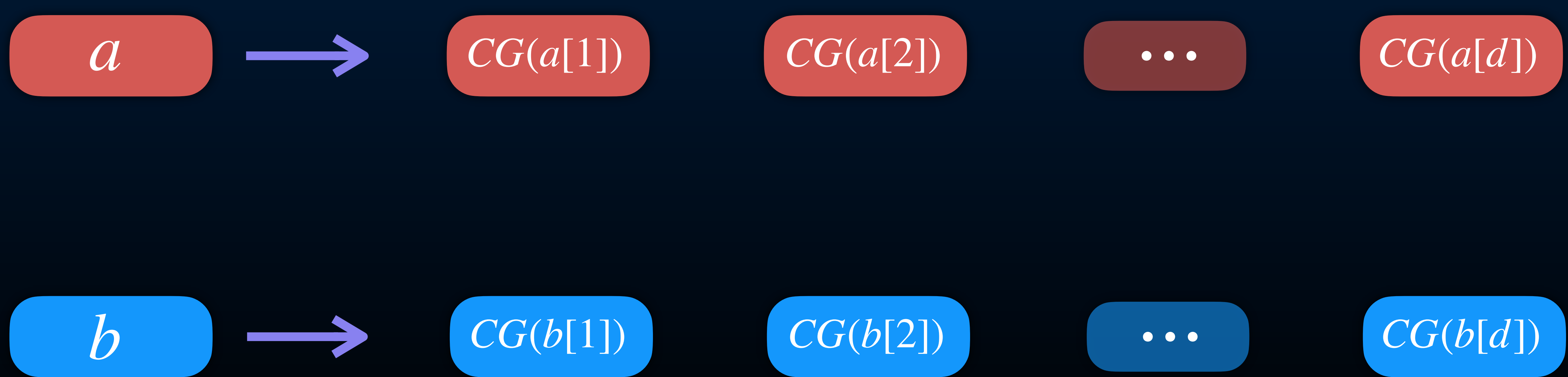
Lemma 1: There are mappings $VG_A, VG_B : \{0,1\}^d \rightarrow \{0,1,2,3\}^{6d^2-d-2}$ computable in $O(d^2)$ time s.t. for any vectors $a, b \in \{0,1\}^d$:

$$\langle a,b\rangle = 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d + 2 \qquad \text{(Large LCS)}$$

$$\langle a,b\rangle \neq 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d \qquad \text{(Small LCS)}$$

where $\beta_d := 3d^2 - d - 2$ is a fixed value.

Attempts:



Step 1: **Vector Gadgets**

Lemma 1: There are mappings $VG_A, VG_B : \{0,1\}^d \to \{0,1,2,3\}^{6d^2-d-2}$ computable in $O(d^2)$ time s.t. for any vectors $a, b \in \{0,1\}^d$:

$$\langle a, b\rangle = 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d + 2 \quad \text{(Large LCS)}$$

$$\langle a, b\rangle \neq 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d \quad \text{(Small LCS)}$$

where $\beta_d := 3d^2 - d - 2$ is a fixed value.

Attempts:

$a \longrightarrow CG(a[1])\ 2^{3d}\ CG(a[2])\ 2^{3d}\ \cdots\ 2^{3d}\ CG(a[d])$

$b \longrightarrow CG(b[1])\ 2^{3d}\ CG(b[2])\ 2^{3d}\ \cdots\ 2^{3d}\ CG(b[d])$

Step 1: **Vector Gadgets**

Lemma 1: There are mappings $VG_A, VG_B : \{0,1\}^d \to \{0,1,2,3\}^{6d^2-d-2}$ computable in $O(d^2)$ time s.t. for any vectors $a, b \in \{0,1\}^d$:

$$\langle a, b\rangle = 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d + 2 \quad \text{(Large LCS)}$$

$$\langle a, b\rangle \neq 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d \quad \text{(Small LCS)}$$

where $\beta_d := 3d^2 - d - 2$ is a fixed value.

Attempts:

a → $CG(a[1])$ 2^{3d} $CG(a[2])$ 2^{3d} $\cdots$ 2^{3d} $CG(a[d])$

Length d | Length $O(d^2)$

b → $CG(b[1])$ 2^{3d} $CG(b[2])$ 2^{3d} $\cdots$ 2^{3d} $CG(b[d])$

Step 1: **Vector Gadgets**

Lemma 1: There are mappings $VG_A, VG_B : \{0,1\}^d \rightarrow \{0,1,2,3\}^{6d^2-d-2}$ computable in $O(d^2)$ time s.t. for any vectors $a, b \in \{0,1\}^d$:

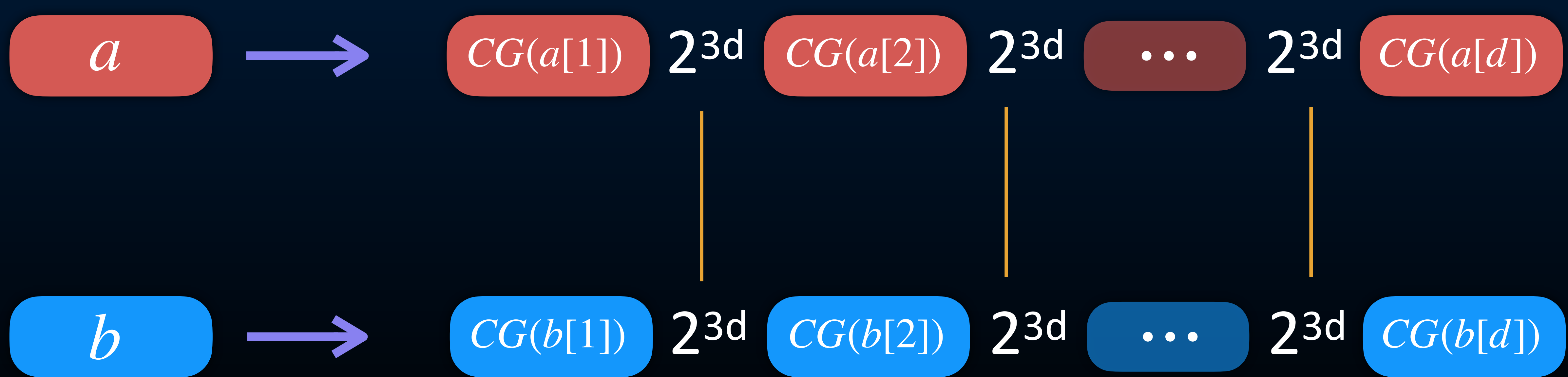
$$\langle a,b\rangle = 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d + 2 \quad \text{(Large LCS)}$$

$$\langle a,b\rangle \neq 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d \quad \text{(Small LCS)}$$

where $\beta_d := 3d^2 - d - 2$ is a fixed value.

Attempts:

Claim: LCS = $(d-1)\cdot 3d + 2\cdot(d - \langle a,b\rangle)$



Step 1: **Vector Gadgets**

Lemma 1: There are mappings $VG_A, VG_B : \{0,1\}^d \to \{0,1,2,3\}^{6d^2-d-2}$ computable in $O(d^2)$ time s.t. for any vectors $a, b \in \{0,1\}^d$:

$$\langle a,b \rangle = 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d + 2 \quad \text{(Large LCS)}$$

$$\langle a,b \rangle \neq 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d \quad \text{(Small LCS)}$$

where $\beta_d := 3d^2 - d - 2$ is a fixed value.

Attempts:

Claim: LCS = $(d-1) \cdot 3d + 2 \cdot (d - \langle a,b \rangle)$

Step 1: **Vector Gadgets**

Lemma 1: There are mappings $VG_A, VG_B : \{0,1\}^d \to \{0,1,2,3\}^{6d^2-d-2}$ computable in $O(d^2)$ time s.t. for any vectors $a, b \in \{0,1\}^d$:

$$\langle a,b \rangle = 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d + 2$$ Done! (Large LCS)

$$\langle a,b \rangle \neq 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d$$ (Small LCS)

where $\beta_d := 3d^2 - d - 2$ is a fixed value.

Attempts:

Claim: LCS = $(d-1) \cdot 3d + 2 \cdot (d - \langle a,b \rangle)$

$a \to CG(a[1])\ 2^{3d}\ CG(a[2])\ 2^{3d} \cdots 2^{3d}\ CG(a[d])$

Loses $3d > |CG|$ ... never worth it.

$b \to CG(b[1])\ 2^{3d}\ CG(b[2])\ 2^{3d} \cdots 2^{3d}\ CG(b[d])$

Step 1: **Vector Gadgets**

Lemma 1: There are mappings $VG_A, VG_B : \{0,1\}^d \to \{0,1,2,3\}^{6d^2-d-2}$ computable in $O(d^2)$ time s.t. for any vectors $a, b \in \{0,1\}^d$:

$$\langle a, b \rangle = 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d + 2$$

Done! arge LCS)

$$\langle a, b \rangle \neq 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d$$

Almost done: $\leq \beta_d$)

where $\beta_d := 3d^2 - d - 2$ is a fixed value.

Attempts:

Claim: LCS = $(d-1) \cdot 3d + 2 \cdot (d - \langle a, b \rangle)$

$a \longrightarrow CG(a[1])\ 2^{3d}\ CG(a[2])\ 2^{3d}\ \cdots\ 2^{3d}\ CG(a[d])$

Loses $3d > |CG|$ ... never worth it.

$b \longrightarrow CG(b[1])\ 2^{3d}\ CG(b[2])\ 2^{3d}\ \cdots\ 2^{3d}\ CG(b[d])$

Step 1: Normalized **Vector Gadgets**

Lemma 1: There are mappings $VG_A, VG_B : \{0,1\}^d \to \{0,1,2,3\}^{6d^2-d-2}$ computable in $O(d^2)$ time s.t. for any vectors $a, b \in \{0,1\}^d$:

$$\langle a,b \rangle = 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d + 2 \quad \text{(Large LCS)}$$

$$\langle a,b \rangle \neq 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d \quad \text{(Small LCS)}$$

where $\beta_d := 3d^2 - d - 2$ is a fixed value.

Proof:

$a \longrightarrow CG(a[1])\ 2^{3d}\ CG(a[2])\ 2^{3d} \cdots 2^{3d}\ CG(a[d])$

$b \longrightarrow CG(b[1])\ 2^{3d}\ CG(b[2])\ 2^{3d} \cdots 2^{3d}\ CG(b[d])$

Step 1: Normalized **Vector Gadgets**

Lemma 1: There are mappings $VG_A, VG_B : \{0,1\}^d \to \{0,1,2,3\}^{6d^2-d-2}$ computable in $O(d^2)$ time s.t. for any vectors $a, b \in \{0,1\}^d$:

$$\langle a,b\rangle = 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d + 2 \quad \text{(Large LCS)}$$

$$\langle a,b\rangle \neq 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d \quad \text{(Small LCS)}$$

where $\beta_d := 3d^2 - d - 2$ is a fixed value.

Proof:

$a \longrightarrow 3^{\beta_d}\ CG(a[1])\ 2^{3d}\ CG(a[2])\ 2^{3d} \cdots 2^{3d}\ CG(a[d])$

$b \longrightarrow CG(b[1])\ 2^{3d}\ CG(b[2])\ 2^{3d} \cdots 2^{3d}\ CG(b[d])\ 3^{\beta_d}$

Step 1: Normalized **Vector Gadgets**

Lemma 1: There are mappings $VG_A, VG_B : \{0,1\}^d \rightarrow \{0,1,2,3\}^{6d^2-d-2}$ computable in $O(d^2)$ time s.t. for any vectors $a, b \in \{0,1\}^d$:

$$\langle a,b\rangle = 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d + 2 \quad \text{(Large LCS)}$$

$$\langle a,b\rangle \neq 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d \quad \text{(Small LCS)}$$

where $\beta_d := 3d^2 - d - 2$ is a fixed value.

Proof:

$a \longrightarrow 3^{\beta_d}$ $CG(a[1])$ 2^{3d} $CG(a[2])$ 2^{3d} $\cdots$ 2^{3d} $CG(a[d])$

If we match any 3, we cannot match any non-3's.

$b \longrightarrow$ $CG(b[1])$ 2^{3d} $CG(b[2])$ 2^{3d} $\cdots$ 2^{3d} $CG(b[d])$ 3^{β_d}

Step 1: Normalized **Vector Gadgets**

Lemma 1: There are mappings $VG_A, VG_B : \{0,1\}^d \to \{0,1,2,3\}^{6d^2-d-2}$ computable in $O(d^2)$ time s.t. for any vectors $a, b \in \{0,1\}^d$:

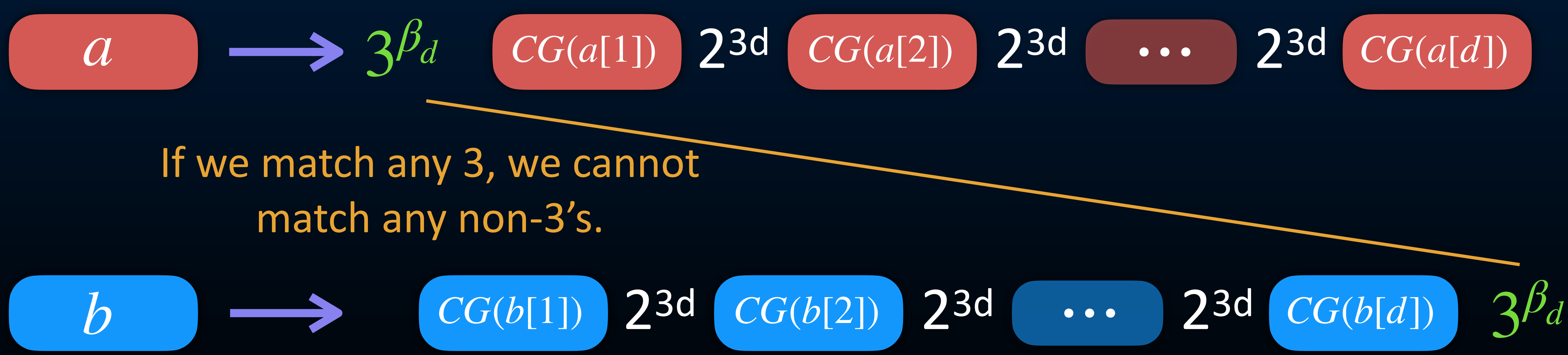
$$\langle a,b\rangle = 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d + 2 \quad \text{(Large LCS)}$$

$$\langle a,b\rangle \neq 0 \implies LCS(VG_A(a), VG_B(b)) = \beta_d \quad \text{(Small LCS)}$$

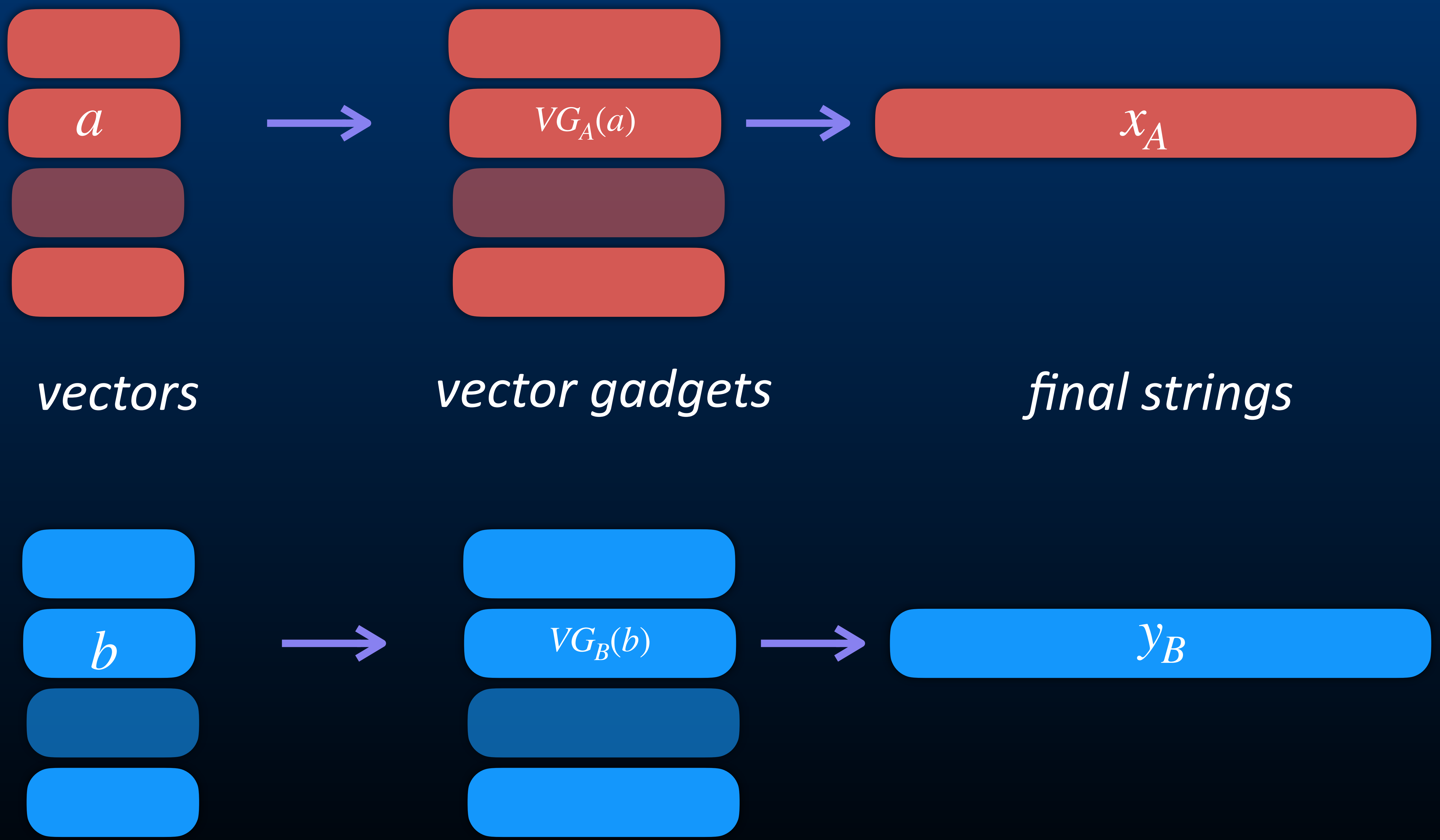
where $\beta_d := 3d^2 - d - 2$ is a fixed value.

Proof:

Claim: LCS = $\max\{(d-1)\cdot 3d + 2\cdot(d - \langle a,b\rangle), \beta_d\}$



Step 2: **Outer-OR Gadget**



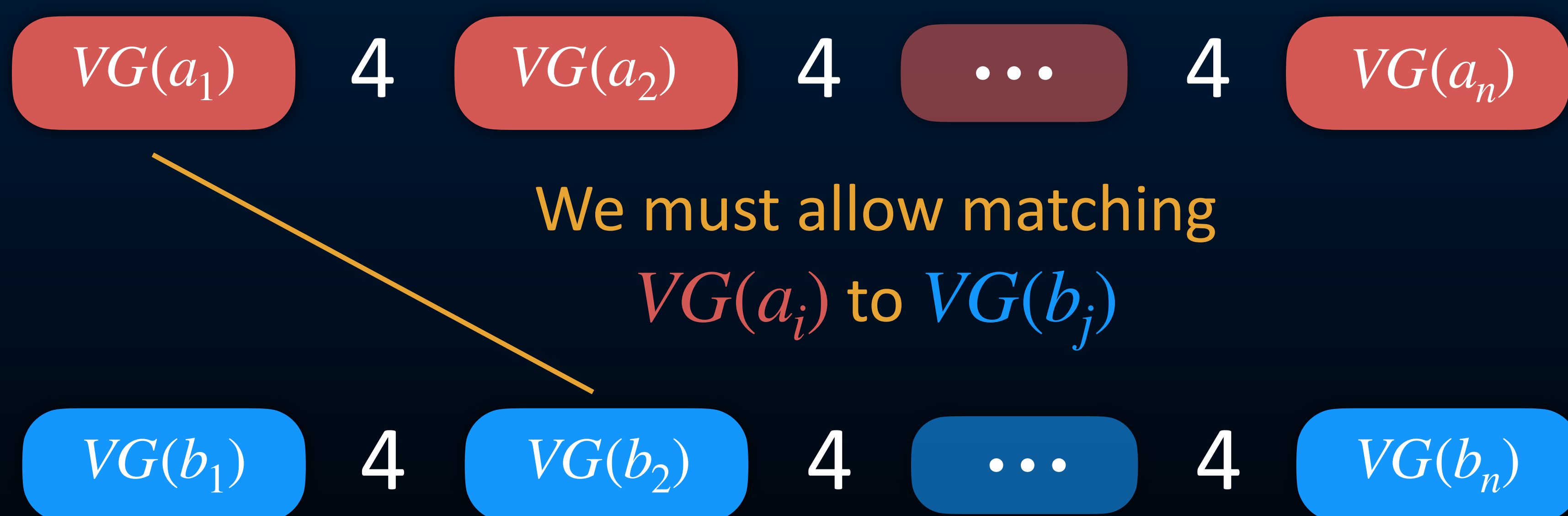
Step 2: **Outer-OR Gadget**

Thm: Given $A, B \subseteq \{0,1\}^d$ can construct x_A , y_B $\in \{0,1,2,3,4\}^{O(nd^2)}$ and an integer τ, in $O(nd^2)$ time s.t.

$\exists a \in A, b \in B : \langle a, b \rangle = 0 \implies LCS(x_A, y_B) \geq \tau$ (Large LCS)

$\forall a \in A, b \in B : \langle a, b \rangle \neq 0 \implies LCS(x_A, y_B) < \tau$ (Small LCS)

Attempts:



Step 2: **Outer-OR Gadget**

Thm: Given $A, B \subseteq \{0,1\}^d$ can construct x_A , y_B $\in \{0,1,2,3,4\}^{O(nd^2)}$ and an integer τ, in $O(nd^2)$ time s.t.

$\exists a \in A, b \in B : \langle a, b \rangle = 0 \implies LCS(x_A, y_B) \geq \tau$ (Large LCS)

$\forall a \in A, b \in B : \langle a, b \rangle \neq 0 \implies LCS(x_A, y_B) < \tau$ (Small LCS)

Attempts:

$VG(a_1)$ 4 $\cdots$ 4 $VG(a_n)$ 4 $VG(a_1)$ 4 $\cdots$ 4 $VG(a_n)$

$VG(b_1)$ 4 $\cdots$ 4 $VG(b_n)$

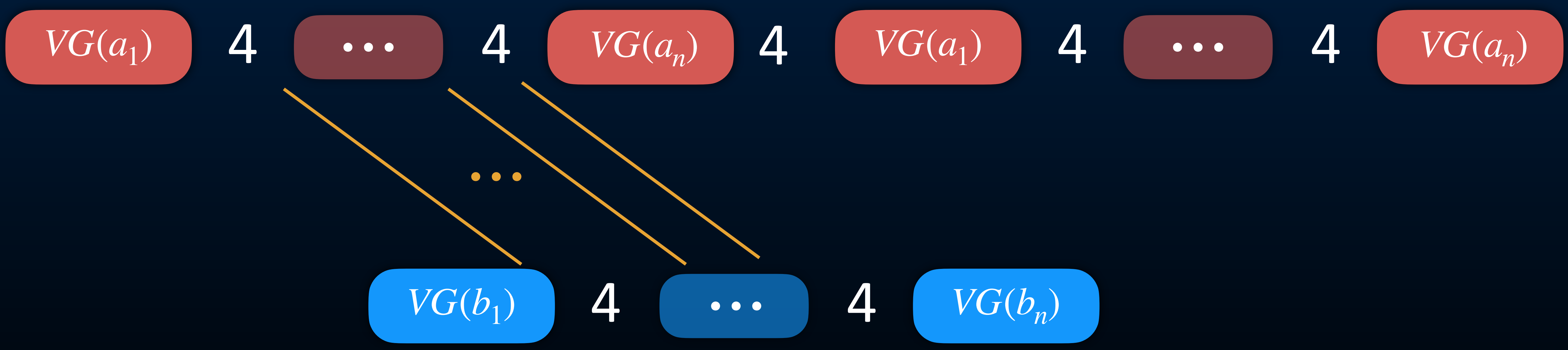
Step 2: **Outer-OR Gadget**

Thm: Given $A, B \subseteq \{0,1\}^d$ can construct x_A , y_B $\in \{0,1,2,3,4\}^{O(nd^2)}$ and an integer τ, in $O(nd^2)$ time s.t.

$\exists a \in A, b \in B : \langle a, b \rangle = 0 \implies LCS(x_A, y_B) \geq \tau$ (Large LCS)

$\forall a \in A, b \in B : \langle a, b \rangle \neq 0 \implies LCS(x_A, y_B) < \tau$ (Small LCS)

Attempts:



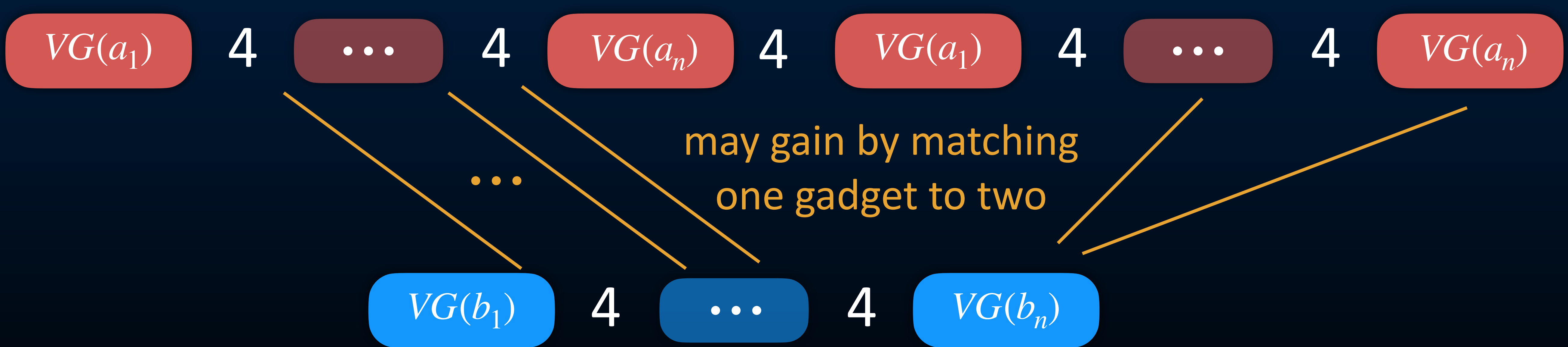
Step 2: **Outer-OR Gadget**

Thm: Given $A, B \subseteq \{0,1\}^d$ can construct x_A , y_B $\in \{0,1,2,3,4\}^{O(nd^2)}$ and an integer τ, in $O(nd^2)$ time s.t.

$\exists a \in A, b \in B : \langle a,b \rangle = 0 \implies LCS(x_A, y_B) \geq \tau$ (Large LCS)

$\forall a \in A, b \in B : \langle a,b \rangle \neq 0 \implies LCS(x_A, y_B) < \tau$ (Small LCS)

Attempts:



Step 2: **Outer-OR Gadget**

Thm: Given $A, B \subseteq \{0,1\}^d$ can construct x_A , y_B $\in \{0,1,2,3,4\}^{O(nd^2)}$ and an integer τ, in $O(nd^2)$ time s.t.

$\exists a \in A, b \in B : \langle a, b \rangle = 0 \implies LCS(x_A, y_B) \geq \tau$ (Large LCS)

$\forall a \in A, b \in B : \langle a, b \rangle \neq 0 \implies LCS(x_A, y_B) < \tau$ (Small LCS)

Proof:

$VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$ 4^γ $VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$

$(4^\gamma)^n$ $VG(b_1)$ 4^γ $\cdots$ 4^γ $VG(b_n)$ $(4^\gamma)^n$

Set: $\gamma := 6d^2 - d - 2 = |VG_B(b)|$, and $\tau := (2n-1) \cdot \gamma + n\beta_d + 2$.

Step 2: **Outer-OR Gadget**

Thm: Given $A, B \subseteq \{0,1\}^d$ can construct x_A , y_B $\in \{0,1,2,3,4\}^{O(nd^2)}$ and an integer τ, in $O(nd^2)$ time s.t.

$\exists a \in A, b \in B : \langle a, b \rangle = 0 \implies LCS(x_A, y_B) \geq \tau$ (Large LCS)

$\forall a \in A, b \in B : \langle a, b \rangle \neq 0 \implies LCS(x_A, y_B) < \tau$ (Small LCS)

Proof:

Claim 1: $\exists a, b : \langle a, b \rangle = 0 \implies LCS(x_A, y_B) \geq \tau$

$VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$ 4^γ $VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$

$(4^\gamma)^n$ $VG(b_1)$ 4^γ $\cdots$ 4^γ $VG(b_n)$ $(4^\gamma)^n$

Set: $\gamma := 6d^2 - d - 2 = |VG_B(b)|$, and $\tau := (2n-1) \cdot \gamma + n\beta_d + 2$.

Step 2: **Outer-OR Gadget**

Thm: Given $A, B \subseteq \{0,1\}^d$ can construct x_A , y_B $\in \{0,1,2,3,4\}^{O(nd^2)}$ and an integer τ, in $O(nd^2)$ time s.t.

$\exists a \in A, b \in B : \langle a,b \rangle = 0 \implies LCS(x_A, y_B) \geq \tau$ (Large LCS)

$\forall a \in A, b \in B : \langle a,b \rangle \neq 0 \implies LCS(x_A, y_B) < \tau$ (Small LCS)

Proof:

Claim 1: $\exists a, b : \langle a,b \rangle = 0 \implies LCS(x_A, y_B) \geq \tau$

$VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$ 4^γ $VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$

All 4's in x get matched: $+(2n-1)\cdot\gamma$

$(4^\gamma)^n$ $VG(b_1)$ 4^γ $\cdots$ 4^γ $VG(b_n)$ $(4^\gamma)^n$

Set: $\gamma := 6d^2 - d - 2 = |VG_B(b)|$, and $\tau := (2n-1)\cdot\gamma + n\beta_d + 2$.

Step 2: **Outer-OR Gadget**

Thm: Given $A, B \subseteq \{0,1\}^d$ can construct x_A , y_B $\in \{0,1,2,3,4\}^{O(nd^2)}$ and an integer τ, in $O(nd^2)$ time s.t.

$\exists a \in A, b \in B : \langle a, b \rangle = 0 \implies LCS(x_A, y_B) \geq \tau$ (Large LCS)

$\forall a \in A, b \in B : \langle a, b \rangle \neq 0 \implies LCS(x_A, y_B) < \tau$ (Small LCS)

Proof:

Claim 1: $\exists a, b : \langle a, b \rangle = 0 \implies LCS(x_A, y_B) \geq \tau$

$VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$ 4^γ $VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$

All 4's in x get matched: $+(2n-1)\cdot\gamma$

All VG(b) gadgets get matched: $+n \cdot \beta$ at least one of them gives +2

$(4^\gamma)^n$ $VG(b_1)$ 4^γ $\cdots$ 4^γ $VG(b_n)$ $(4^\gamma)^n$

Set: $\gamma := 6d^2 - d - 2 = |VG_B(b)|$, and $\tau := (2n-1)\cdot\gamma + n\beta_d + 2$.

Step 2: **Outer-OR Gadget**

Thm: Given $A, B \subseteq \{0,1\}^d$ can construct x_A , y_B $\in \{0,1,2,3,4\}^{O(nd^2)}$ and an integer τ, in $O(nd^2)$ time s.t.

$\exists a \in A, b \in B : \langle a,b \rangle = 0 \implies LCS(x_A, y_B) \geq \tau$ (Large LCS)

$\forall a \in A, b \in B : \langle a,b \rangle \neq 0 \implies LCS(x_A, y_B) < \tau$ (Small LCS)

Proof:

Claim 2: $\forall a, b : \langle a,b \rangle \neq 0 \implies LCS(x_A, y_B) \leq \tau - 2$

$VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$ 4^γ $VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$

$(4^\gamma)^n$ $VG(b_1)$ 4^γ $\cdots$ 4^γ $VG(b_n)$ $(4^\gamma)^n$

Set: $\gamma := 6d^2 - d - 2 = |VG_B(b)|$, and $\tau := (2n-1)\cdot\gamma + n\beta_d + 2$.

Step 2: **Outer-OR Gadget**

Thm: Given $A, B \subseteq \{0,1\}^d$ can construct x_A , y_B $\in \{0,1,2,3,4\}^{O(nd^2)}$ and an integer τ, in $O(nd^2)$ time s.t.

$\exists a \in A, b \in B : \langle a, b \rangle = 0 \implies LCS(x_A, y_B) \geq \tau$ (Large LCS)

$\forall a \in A, b \in B : \langle a, b \rangle \neq 0 \implies LCS(x_A, y_B) < \tau$ (Small LCS)

Proof:

Claim 2: $\forall a, b : \langle a, b \rangle \neq 0 \implies LCS(x_A, y_B) \leq \tau - 2$

$VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$ 4^γ $VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$

Contribution from 4's:

$\leq (2n-1) \cdot \gamma$

$(4^\gamma)^n$ $VG(b_1)$ 4^γ $\cdots$ 4^γ $VG(b_n)$ $(4^\gamma)^n$

Set: $\gamma := 6d^2 - d - 2 = |VG_B(b)|$, and $\tau := (2n-1) \cdot \gamma + n\beta_d + 2$.

Step 2: **Outer-OR Gadget**

Thm: Given $A, B \subseteq \{0,1\}^d$ can construct x_A , y_B $\in \{0,1,2,3,4\}^{O(nd^2)}$ and an integer τ, in $O(nd^2)$ time s.t.

$\exists a \in A, b \in B : \langle a, b \rangle = 0 \implies LCS(x_A, y_B) \geq \tau$ (Large LCS)

$\forall a \in A, b \in B : \langle a, b \rangle \neq 0 \implies LCS(x_A, y_B) < \tau$ (Small LCS)

Proof:

Claim 2: $\forall a, b : \langle a, b \rangle \neq 0 \implies LCS(x_A, y_B) \leq \tau - 2$

$VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$ 4^γ $VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$

Contribution from 4's:

$\leq (2n-1) \cdot \gamma$

Contribution of each $VG_B(b)$:

$(4^\gamma)^n$ $VG(b_1)$ 4^γ $\cdots$ 4^γ $VG(b_n)$ $(4^\gamma)^n$

Set: $\gamma := 6d^2 - d - 2 = |VG_B(b)|$, and $\tau := (2n-1) \cdot \gamma + n\beta_d + 2$.

Step 2: **Outer-OR Gadget**

Thm: Given $A, B \subseteq \{0,1\}^d$ can construct x_A , y_B $\in \{0,1,2,3,4\}^{O(nd^2)}$ and an integer τ, in $O(nd^2)$ time s.t.

$\exists a \in A, b \in B : \langle a, b \rangle = 0 \implies LCS(x_A, y_B) \geq \tau$ (Large LCS)

$\forall a \in A, b \in B : \langle a, b \rangle \neq 0 \implies LCS(x_A, y_B) < \tau$ (Small LCS)

Proof:

Claim 2: $\forall a, b : \langle a, b \rangle \neq 0 \implies LCS(x_A, y_B) \leq \tau - 2$

$VG(a_1)\ 4^\gamma\ \cdots\ 4^\gamma\ VG(a_n)\ 4^\gamma\ VG(a_1)\ 4^\gamma\ \cdots\ 4^\gamma\ VG(a_n)$

Contribution from 4's:

$\leq (2n-1) \cdot \gamma$

Contribution of each $VG_B(b)$:

$= 0$

unmatched

$(4^\gamma)^n\ VG(b_1)\ 4^\gamma\ \cdots\ 4^\gamma\ VG(b_n)\ (4^\gamma)^n$

Set: $\gamma := 6d^2 - d - 2 = |VG_B(b)|$, and $\tau := (2n-1) \cdot \gamma + n\beta_d + 2$.

Step 2: **Outer-OR Gadget**

Thm: Given $A, B \subseteq \{0,1\}^d$ can construct x_A , y_B $\in \{0,1,2,3,4\}^{O(nd^2)}$ and an integer τ, in $O(nd^2)$ time s.t.

$\exists a \in A, b \in B : \langle a, b \rangle = 0 \implies LCS(x_A, y_B) \geq \tau$ (Large LCS)

$\forall a \in A, b \in B : \langle a, b \rangle \neq 0 \implies LCS(x_A, y_B) < \tau$ (Small LCS)

Proof:

Claim 2: $\forall a, b : \langle a, b \rangle \neq 0 \implies LCS(x_A, y_B) \leq \tau - 2$

$VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$ 4^γ $VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$

Contribution from 4's:
$\leq (2n-1) \cdot \gamma$

Contribution of each $VG_B(b)$:
$= 0$ *unmatched*
$\leq \beta_d$ *matched to one* $VG_A(a)$

$(4^\gamma)^n$ $VG(b_1)$ 4^γ $\cdots$ 4^γ $VG(b_n)$ $(4^\gamma)^n$

Set: $\gamma := 6d^2 - d - 2 = |VG_B(b)|$, and $\tau := (2n-1) \cdot \gamma + n\beta_d + 2$.

Step 2: **Outer-OR Gadget**

Thm: Given $A, B \subseteq \{0,1\}^d$ can construct x_A , y_B $\in \{0,1,2,3,4\}^{O(nd^2)}$ and an integer τ, in $O(nd^2)$ time s.t.

$\exists a \in A, b \in B : \langle a, b \rangle = 0 \implies LCS(x_A, y_B) \geq \tau$ (Large LCS)

$\forall a \in A, b \in B : \langle a, b \rangle \neq 0 \implies LCS(x_A, y_B) < \tau$ (Small LCS)

Proof:

Claim 2: $\forall a, b : \langle a, b \rangle \neq 0 \implies LCS(x_A, y_B) \leq \tau - 2$

$VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$ 4^γ $VG(a_1)$ 4^γ $\cdots$ 4^γ $VG(a_n)$

Contribution from 4's:
$\leq (2n-1) \cdot \gamma$

Contribution of each $VG_B(b)$:

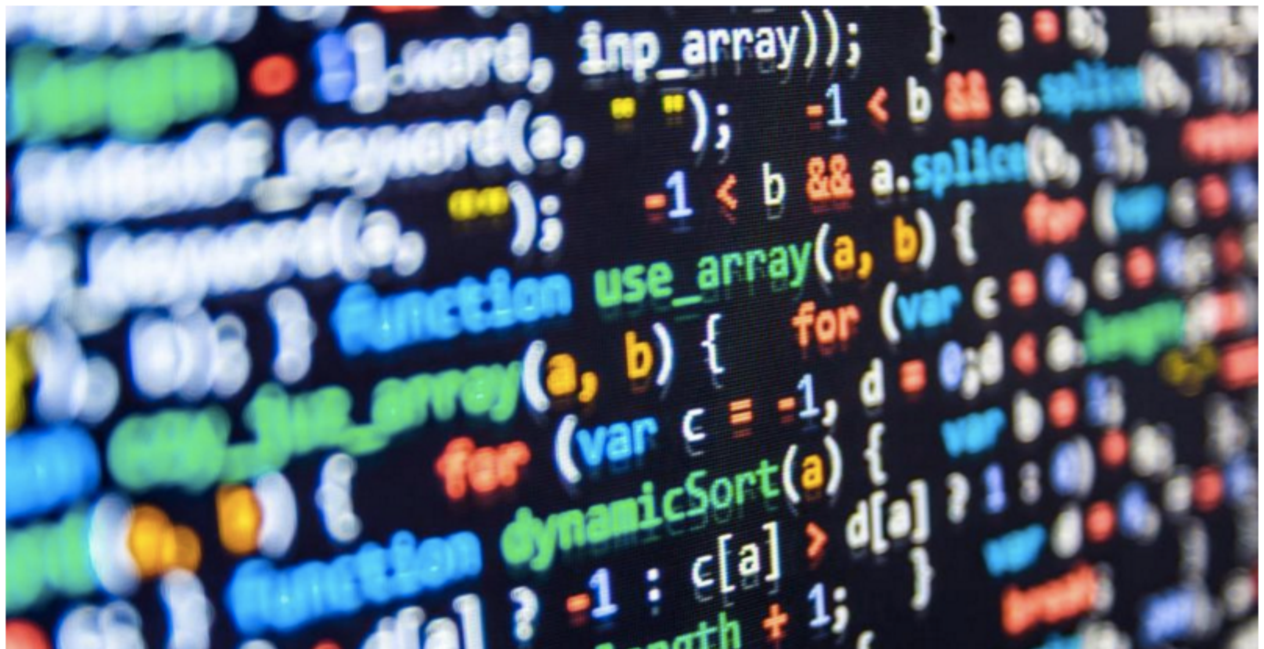
- $= 0$ *unmatched*
- $\leq \beta_d$ *matched to one* $VG_A(a)$
- $\leq |VG_B(b)| - \gamma = 0$ *matched to >1* $VG_A(a)$

$(4^\gamma)^n$ $VG(b_1)$ 4^γ $\cdots$ 4^γ $VG(b_n)$ $(4^\gamma)^n$

Set: $\gamma := 6d^2 - d - 2 = |VG_B(b)|$, and $\tau := (2n-1) \cdot \gamma + n\beta_d + 2$.

BRAINIAC

For 40 years, computer scientists looked for a solution that doesn't exist



Top 10 Trending Articles

Most Viewed | Most Commented | Most Shared

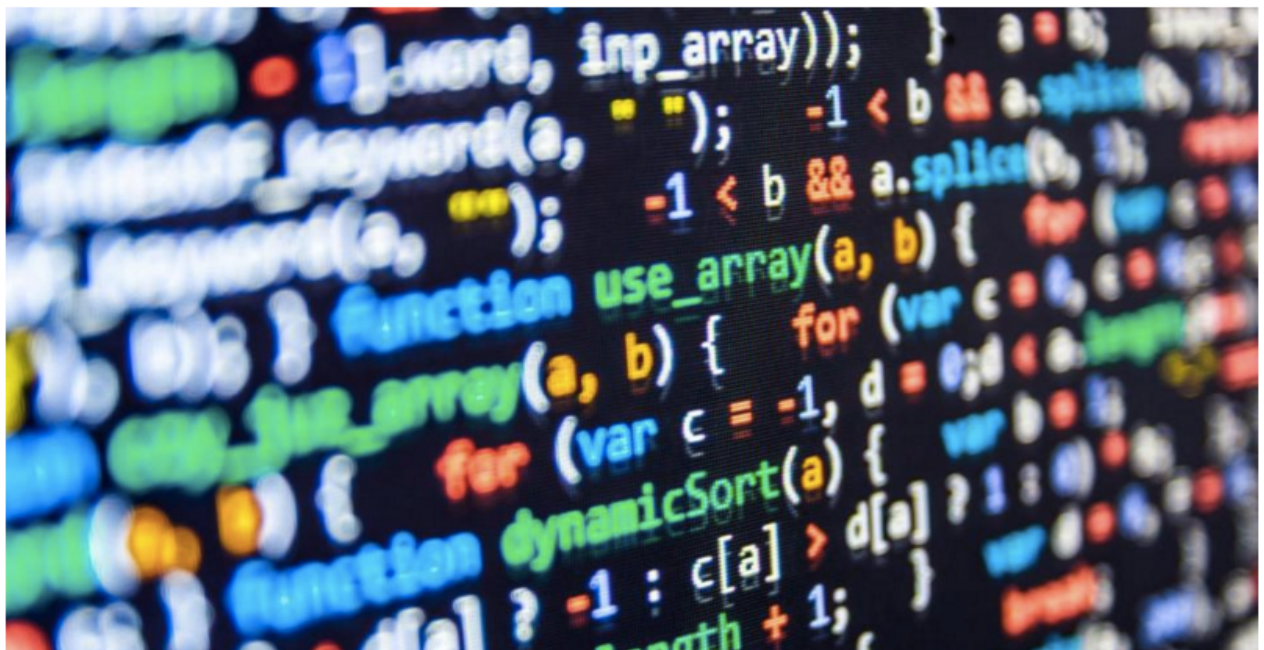
- Words matter in 'ISIS' war, so use 'Daesh'
- Governor Charlie Baker is opposed to allowing Syrian refugees in Mass. for now
- Paris and the fall of Rome
- Chris Kimball to leave America's Test Kitchen after contract dispute
- Paris attacks: Baker, others wrong to reject Syrian refugees
- Paris attacks | Andrew J. Bacevich: A war the West cannot win
- A Patriots-Giants rematch in February: You can't beat that
- How will Patriots fill void while Julian Edelman is out?

Menu | Ideas | SPOTLIGHT The story behind Spotlight in our free e-book | Subscribe Starting at 99 cents | Members Sign In

BRAINIAC

For 40 years, computer scientists looked for a solution that doesn't exist

24



Top 10 Trending Articles

Most Viewed | Most Commented | Most Shared

- Words matter in 'ISIS' war, so use 'Daesh'
- Governor Charlie Baker is opposed to allowing Syrian refugees in Mass. for now
- Paris and the fall of Rome
- Chris Kimball to leave America's Test Kitchen after contract dispute
- Paris attacks: Baker, others wrong to reject Syrian refugees
- Paris attacks | Andrew J. Bacevich: A war the West cannot win
- A Patriots-Giants rematch in February: You can't beat that
- How will Patriots fill void while Julian Edelman is out?

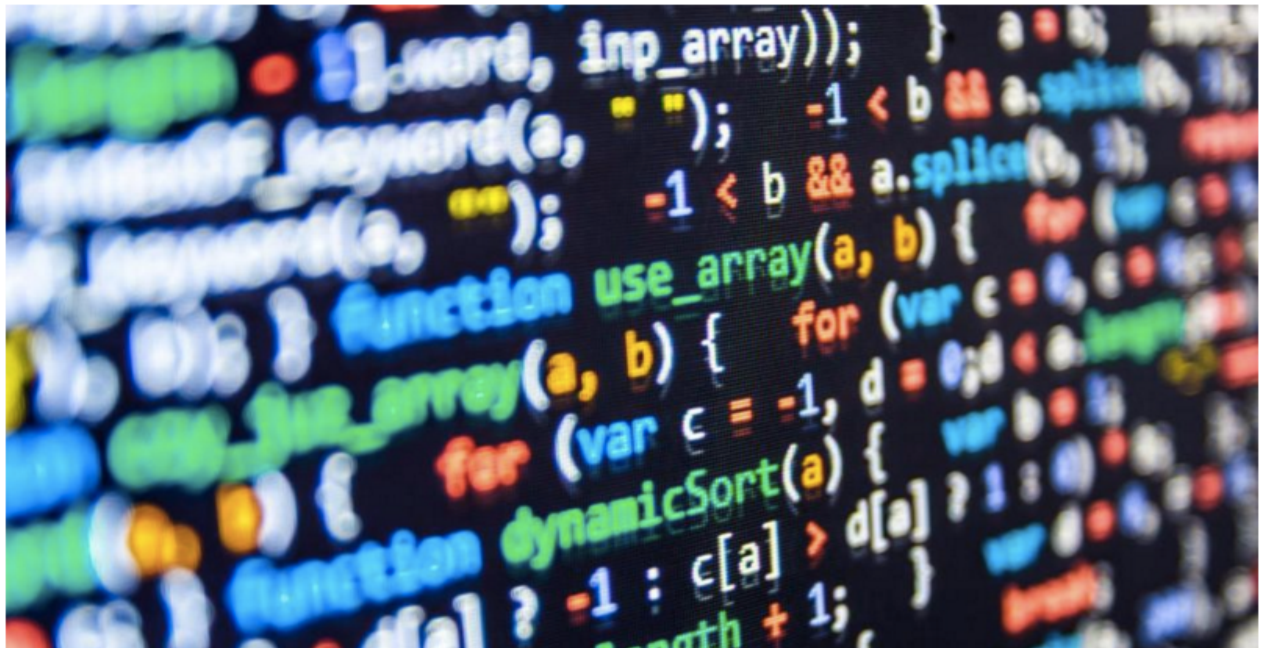
Because it's computationally infeasible to compute the edit distance between two human genomes, biologists have to rely on approximations. They'd love a faster method, but Indyk and his coauthor, MIT graduate student Arturs Backurs, recently demonstrated a faster method is impossible to create. And by impossible they don't mean "very hard" or "our technology has to improve first." They mean something like, "by the laws of the mathematics, it can't be done."

Menu | Ideas | SPOTLIGHT The story behind Spotlight in our free e-book | Subscribe Starting at 99 cents | Members Sign In

BRAINIAC

For 40 years, computer scientists looked for a solution that doesn’t exist

24



Top 10 Trending Articles

Most Viewed | Most Commented | Most Shared

- Words matter in ‘ISIS’ war, so use ‘Daesh’
- Governor Charlie Baker is opposed to allowing Syrian refugees in Mass. for now
- Paris and the fall of Rome
- Chris Kimball to leave America’s Test Kitchen after contract dispute
- Paris attacks: Baker, others wrong to reject Syrian refugees
- Paris attacks | Andrew J. Bacevich: A war the West cannot win
- A Patriots-Giants rematch in February: You can’t beat that
- How will Patriots fill void while Julian Edelman is out?

Because it’s computationally infeasible to compute the edit distance between two human genomes, biologists have to rely on approximations. They’d love a faster method, but Indyk and his coauthor, MIT graduate student Arturs Backurs, recently demonstrated a faster method is impossible to create. And by impossible they don’t mean “very hard” or “our technology has to improve first.” They mean something like, “by the laws of the mathematics, it can’t be done.”

Next: How Strong is SETH?