Lower Bounds and Open Problems in Streams

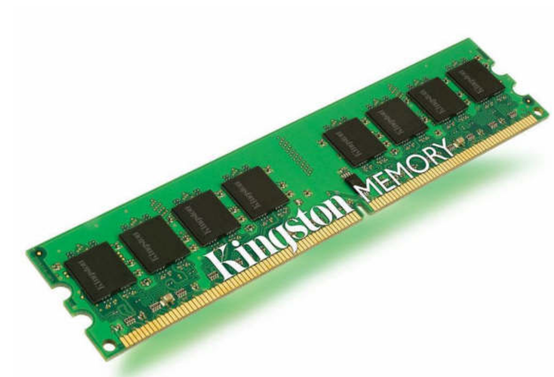
Raphaël Clifford

Joint work with

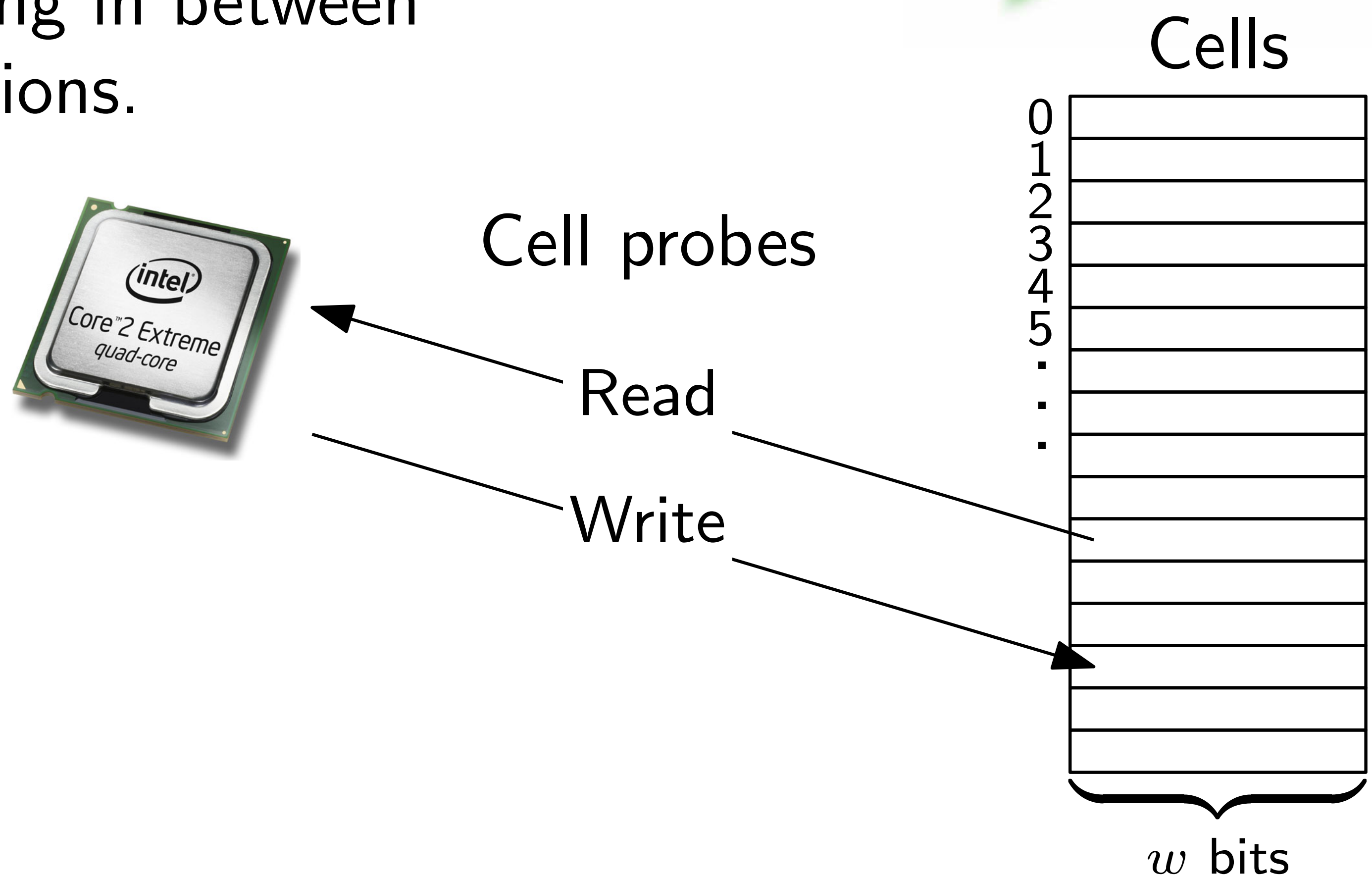
Markus Jalsenius and Benjamin Sach

University of BRISTOL

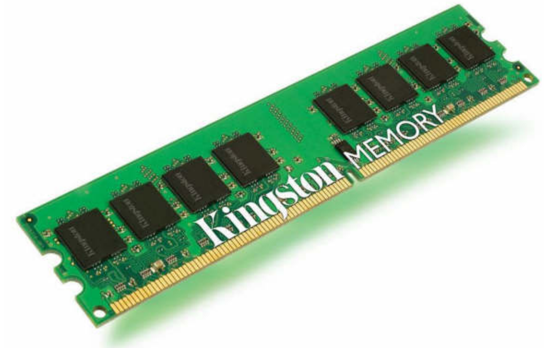
Cell-probe model



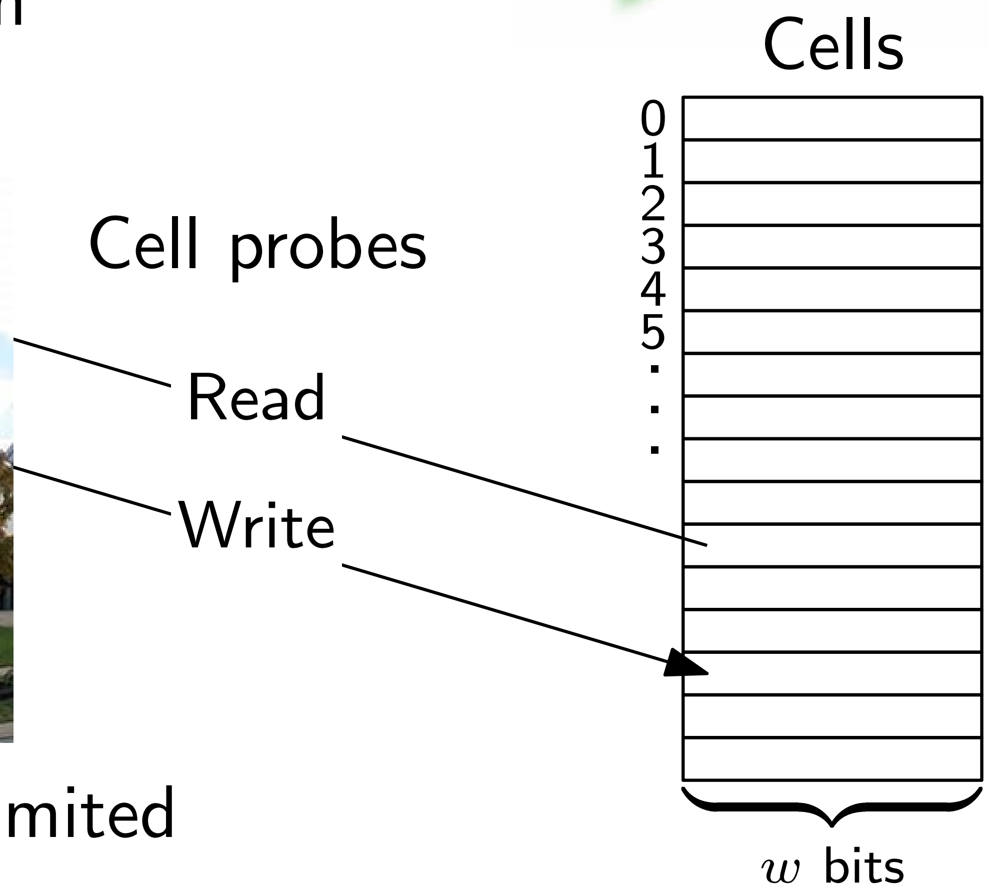
The CPU does not remember anything in between operations.



Cell-probe model



The CPU does not remember anything in between operations.



The CPU has unlimited computational power.

Data Structure Lower Bounds

Yao - FOCS '78

Predecessor (static)

- Ajtai - Combinatorica '88 (incorrect) (Communication complexity)
- Miltersen - STOC' 94
- Miltersen, Nisan, Safra, Wigdersen - STOC '95
- Beame, Fich - STOC '99
- Sen - ICALP '01

Dynamic problems (partial sums, union find)

- Fredman, Saks - STOC '89 (Chronogram technique)
- Ben-Amram, Galil - FOCS '91
- Miltersen, Subramanian, Vitter, Tamassia - TCS '94
- Husfeldt, Rauhe, Skyum - SWAT '96 (planar connectivity)
- Fredman, Henzinger - Algorithmica '98 (non-determinism)
- Alstrup, Husfeldt, Rauhe - FOCS '98 (marked ancestor)
- Alstrup, Husfeldt, Rauhe - SODA '01 (2D NN)
- Alstrup, Ben-Amram, Rauhe - STOC '99 (union-find)

Data Structure Lower Bounds

Yao - FOCS '78

Predecessor (static)

- Ajtai - Combinatorica '88 (incorrect) (Communication complexity)
- Miltersen - STOC' 94
- Miltersen, Nisan, Safra, Wigdersen - STOC '95
- Beame, Fich - STOC '99
- Sen - ICALP '01

Dynamic problems (partial sums, union find)

- Fredman, Saks - STOC '89 (Chronogram technique)
- Ben-Amram, Galil - FC
- Miltersen, Subramania '94
- Husfeldt, Rauhe, Skyu nectivity)
- Fredman, Henzinger - rminism)
- Alstrup, Husfeldt, Rau cestor)
- Alstrup, Husfeldt, Rauhe - SODA '01 (2D NN)
- Alstrup, Ben-Amram, Rauhe - STOC '99 (union-find)

Best lower bound

$$\Omega\left(\frac{\log n}{\log \log n}\right)$$

Data Structure Lower Bounds

Yao - FOCS '78

Predecessor (static)

- Ajtai - Combinatorica '88 (incorrect) (Communication complexity)
- Miltersen - STOC' 94
- Miltersen, Nisan, Safra, Wigdersen - STOC '95
- Be
- Se

Dynam

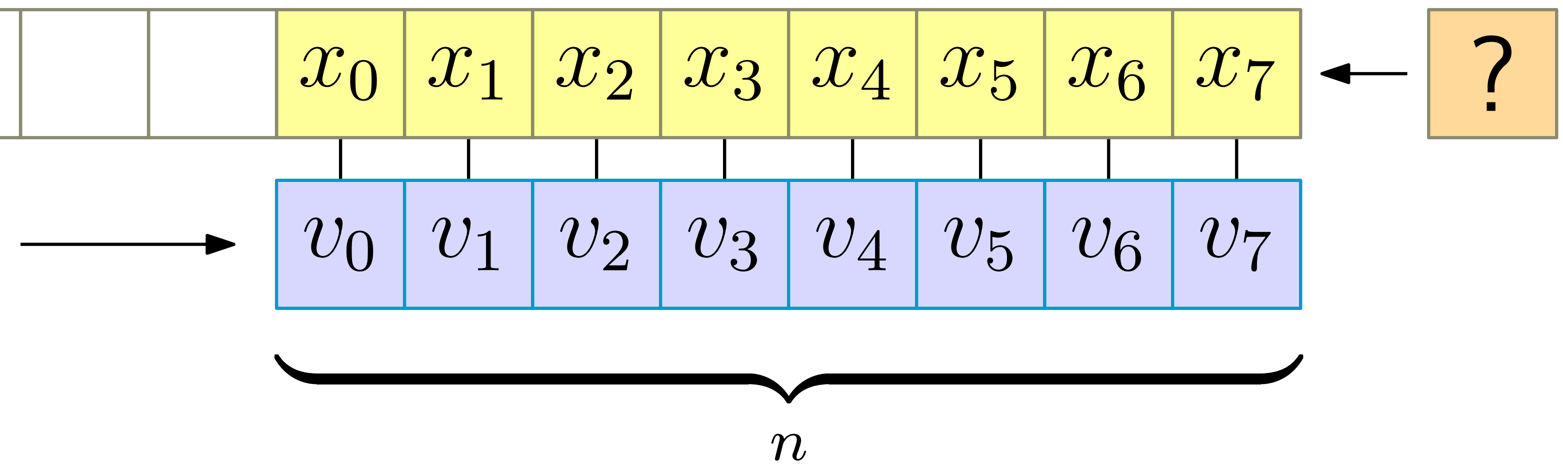
- Fre
- Be
- Mi
- Hu
- Fredman, Henzinger - Algorithmica '98 (non-determinism)
- Alstrup, Husfeldt, Rauhe - FOCS '98 (marked ancestor)
- Alstrup, Husfeldt, Rauhe - SODA '01 (2D NN)
- Alstrup, Ben-Amram, Rauhe - STOC '99 (union-find)

First $\Omega(\log n)$ lower bound using *information transfer*.

M. Pătrașcu and E. Demaine
Tight bounds for the partial-sums problem
SODA 2004

Convolution

Stream of numbers from $[q]$



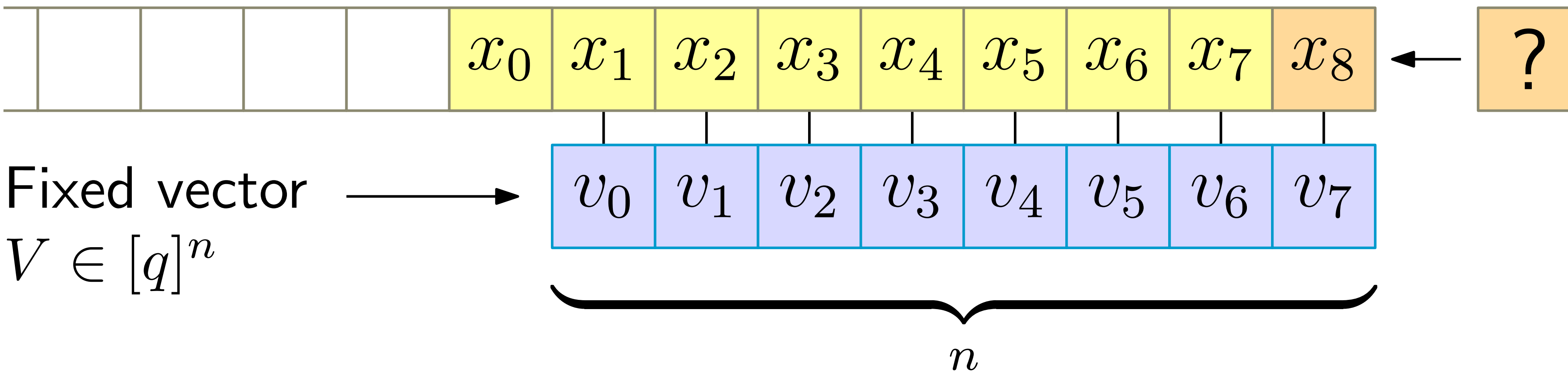
Fixed vector
$V \in [q]^n$

Output dot product (modulo q):

$$V \cdot (\text{last } n \text{ digits of stream}) = \sum_{i=0}^{n-1} v_i x_{(i + \text{leftmost aligned index})}$$

Convolution

Stream of numbers from $[q]$



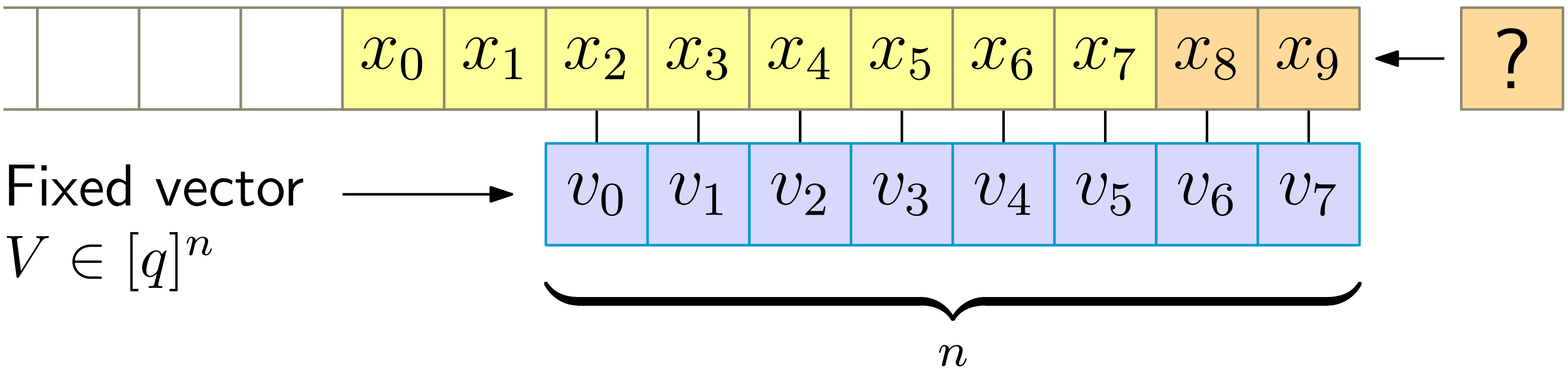
Fixed vector
$V \in [q]^n$

Output dot product (modulo q):

$$V \cdot (\text{last } n \text{ digits of stream}) = \sum_{i=0}^{n-1} v_i x_{(i + \text{leftmost aligned index})}$$

Convolution

Stream of numbers from $[q]$



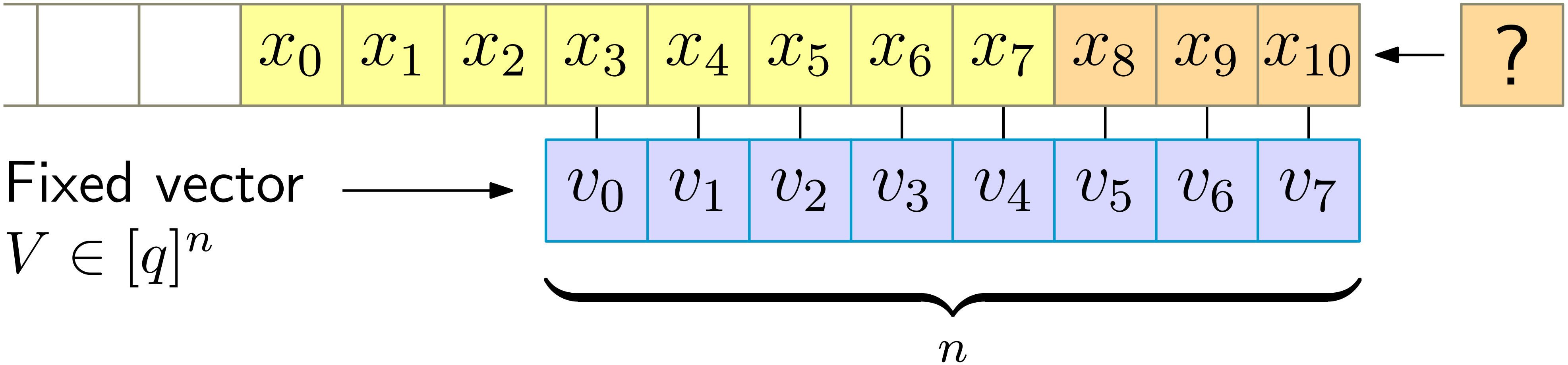
Fixed vector
$V \in [q]^n$

Output dot product (modulo q):

$$V \cdot (\text{last } n \text{ digits of stream}) = \sum_{i=0}^{n-1} v_i x_{(i\,+\,\text{leftmost aligned index})}$$

Convolution

Stream of numbers from $[q]$



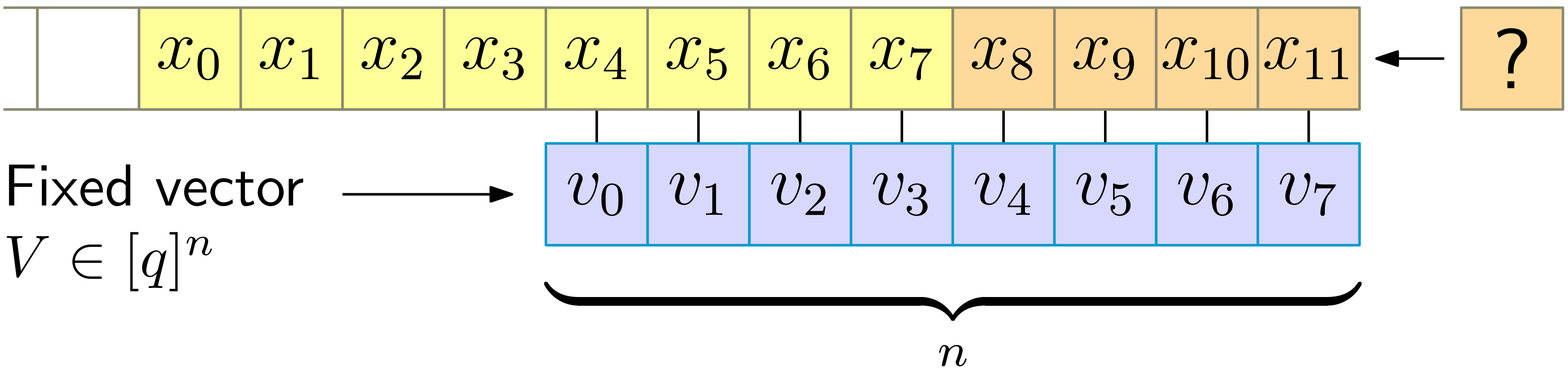
Fixed vector
$V \in [q]^n$

Output dot product (modulo q):

$$V \cdot (\text{last } n \text{ digits of stream}) = \sum_{i=0}^{n-1} v_i x_{(i + \text{leftmost aligned index})}$$

Convolution

Stream of numbers from $[q]$



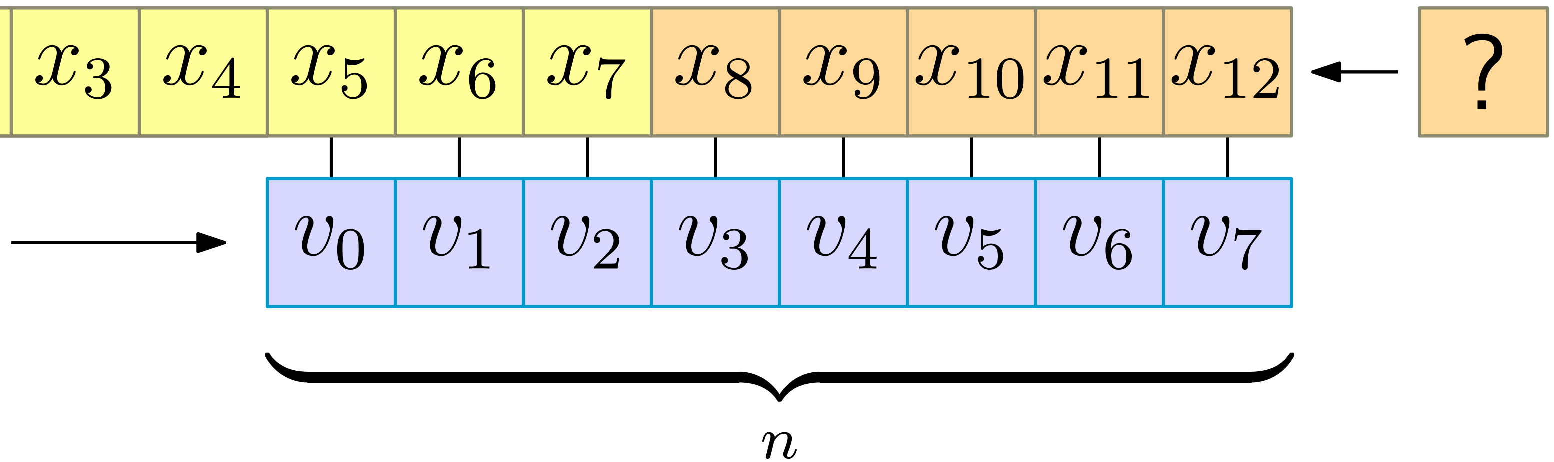
Fixed vector
$V \in [q]^n$

Output dot product (modulo q):

$$V \cdot (\text{last } n \text{ digits of stream}) = \sum_{i=0}^{n-1} v_i x_{(i + \text{leftmost aligned index})}$$

Convolution

Stream of numbers from $[q]$



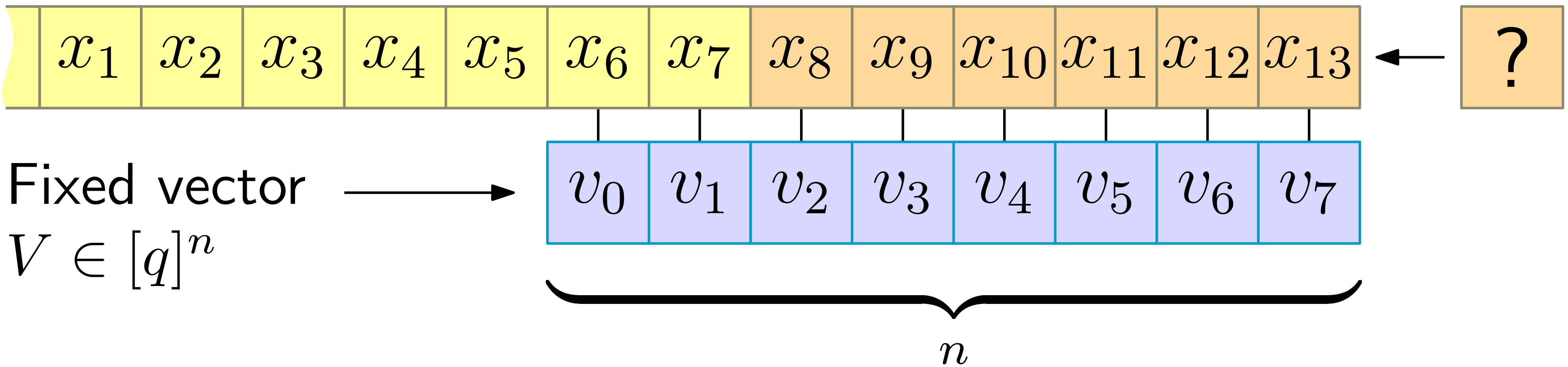
Fixed vector
$V \in [q]^n$

Output dot product (modulo q):

$$V \cdot (\text{last } n \text{ digits of stream}) = \sum_{i=0}^{n-1} v_i x_{(i + \text{leftmost aligned index})}$$

Convolution

Stream of numbers from $[q]$

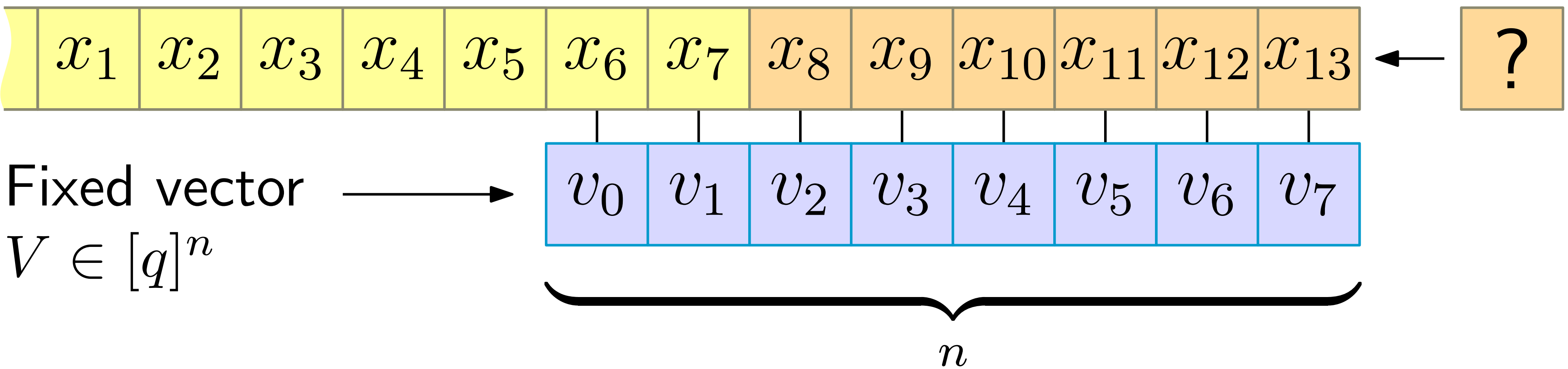


Output dot product (modulo q):

$$V \cdot (\text{last } n \text{ digits of stream}) = \sum_{i=0}^{n-1} v_i x_{(i + \text{leftmost aligned index})}$$

Convolution

Stream of numbers from $[q]$



Output dot product (modulo q):

$$V \cdot (\text{last } n \text{ digits of stream}) = \sum_{i=0}^{n-1} v_i x_{(i + \text{leftmost aligned index})}$$

Lower bound: $\Omega\left(\frac{\delta}{w} \log n\right)$

$\delta = \log q$, word size w.

C., Jalsenius. Lower Bounds for Online Integer Multiplication and Convolution in the Cell-Probe Mode. ICALP 2011

Previous bounds

M. J. Fischer and L. J. Stockmeyer
Fast on-line integer multiplication
STOC '73

C., K. Efremenko, B. Porat and E. Porat
A black box for online approximate pattern matching
Information and Computation 209(4):731–736, 2011

- $O(\log^2 n)$ time per arriving symbol (pair)

Previous bounds

M. J. Fischer and L. J. Stockmeyer
Fast on-line integer multiplication
STOC '73

C., K. Efremenko, B. Porat and E. Porat
A black box for online approximate pattern matching
Information and Computation 209(4):731–736, 2011

- $O(\log^2 n)$ time per arriving symbol (pair)

Offline cell probe complexity is linear!

$\Rightarrow$

online upper bound of $O(\log n)$

Previous bounds

M. J. Fischer and L. J. Stockmeyer
Fast on-line integer multiplication
STOC '73

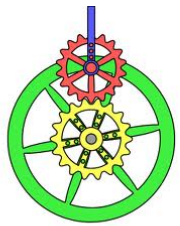
C., K. Efremenko, B. Porat and E. Porat
A black box for online approximate pattern matching
Information and Computation 209(4):731–736, 2011

- $O(\log^2 n)$ time per arriving symbol (pair)

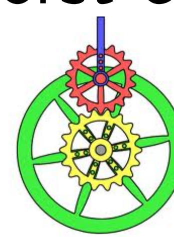
Better online lower bound
$\Rightarrow$
super linear lower bound for
offline convolution and multiplication

Yao's minimax principle

A lower bound on the expected running time for

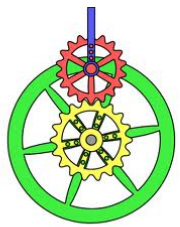
Algorithm		Input
Deterministic 	←	Random 

implies that the same lower bound holds for

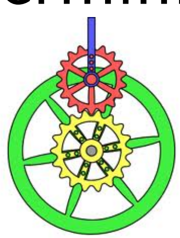
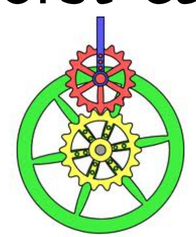
Algorithm		Input
Randomised 	←	Worst case 

Yao's minimax principle

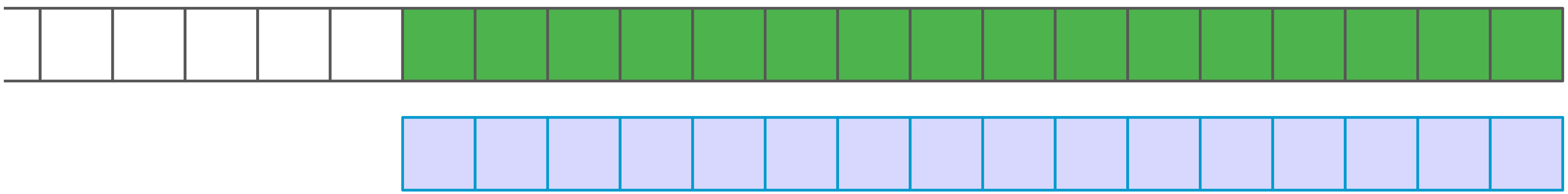
A lower bound on the expected running time for

Algorithm		Input
Deterministic 	←	Random 

implies that the same lower bound holds for

Algorithm		Input
Deterministic 	←	Worst case 

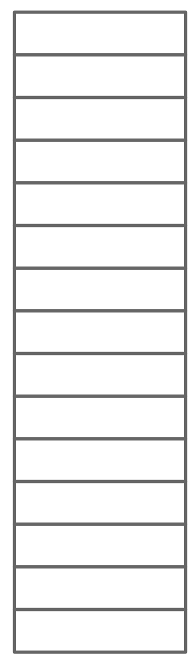
Information transfer



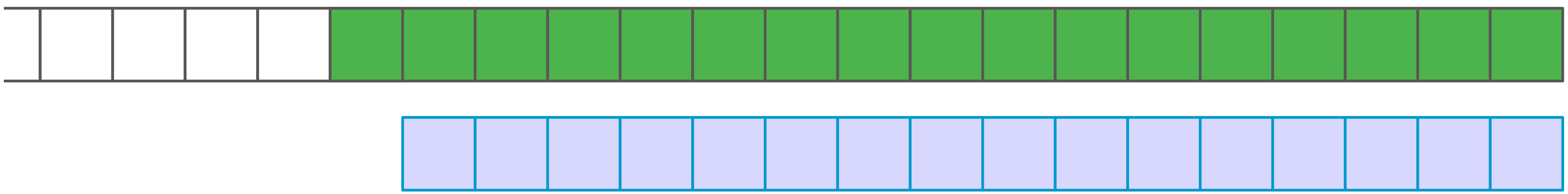
Fixed value

? Unknown value
chosen uniformly
at random from $[q]$

Memory cells



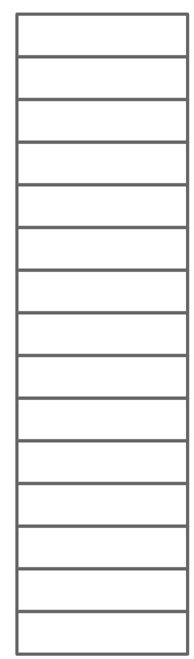
Information transfer



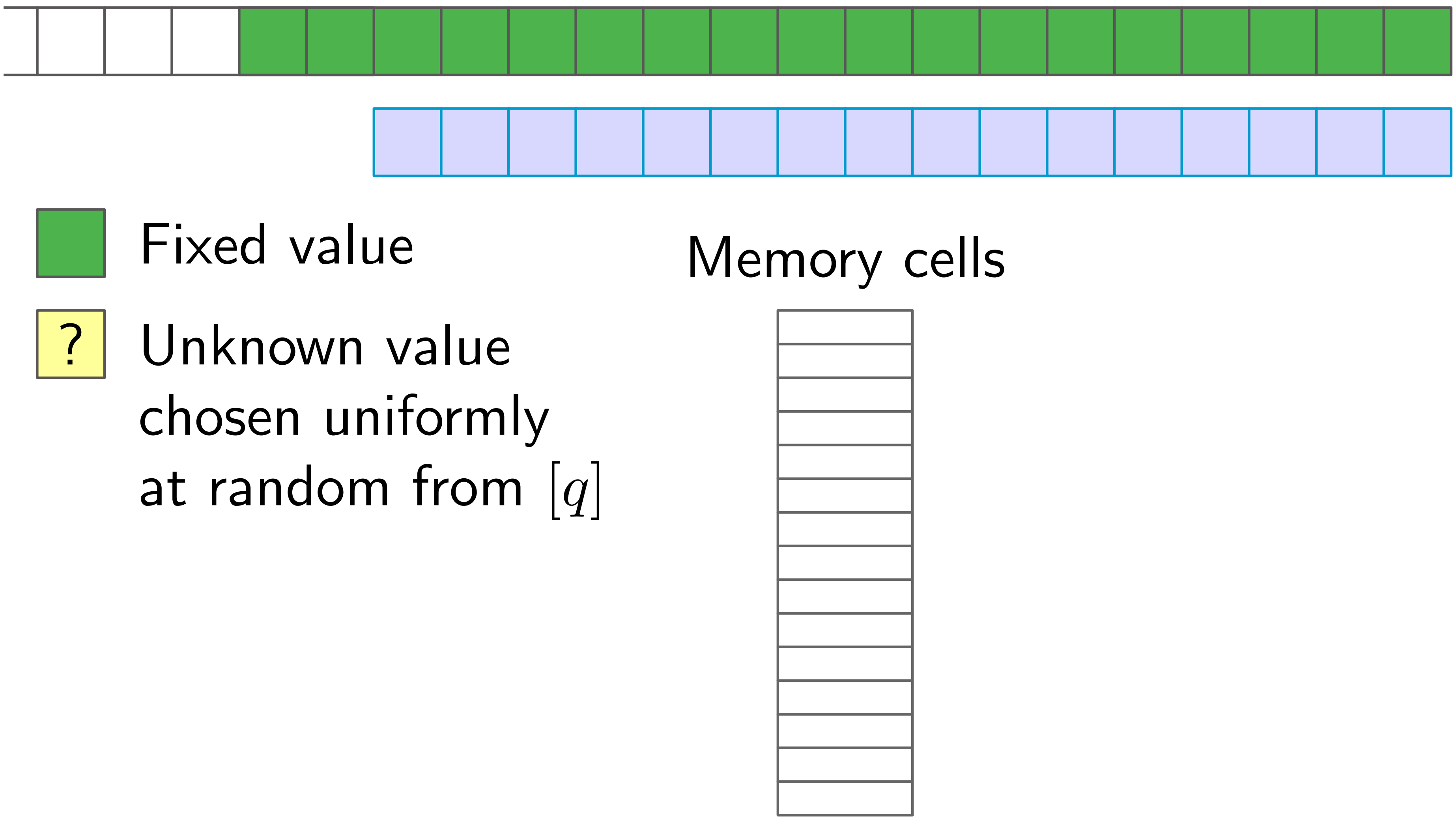
Fixed value

? Unknown value chosen uniformly at random from $[q]$

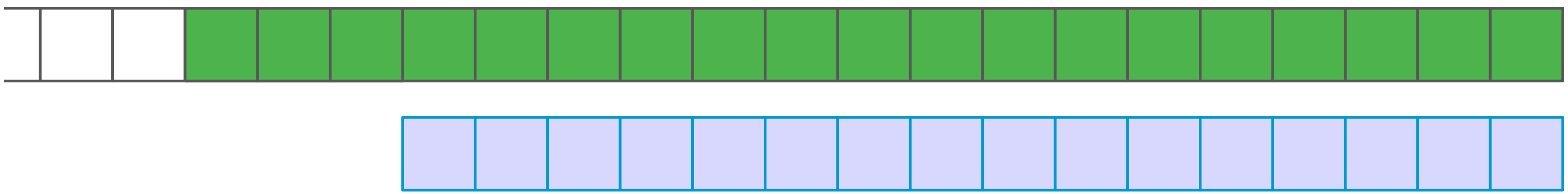
Memory cells



Information transfer



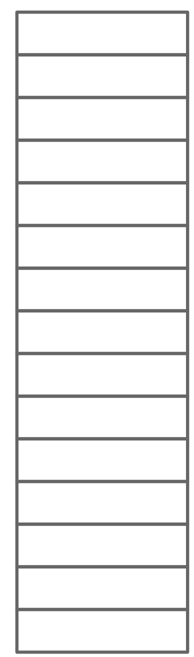
Information transfer



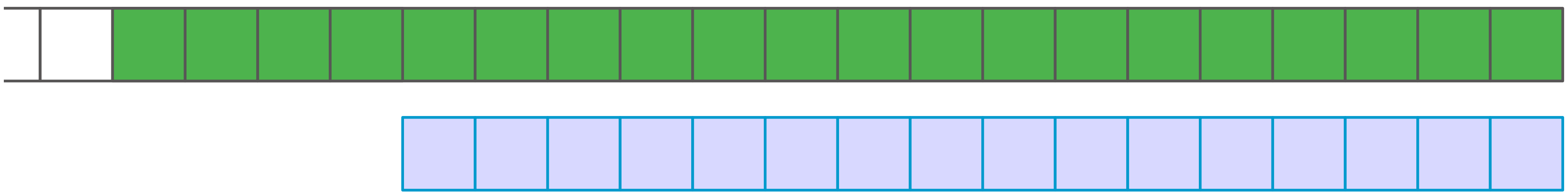
Fixed value

? Unknown value chosen uniformly at random from $[q]$

Memory cells



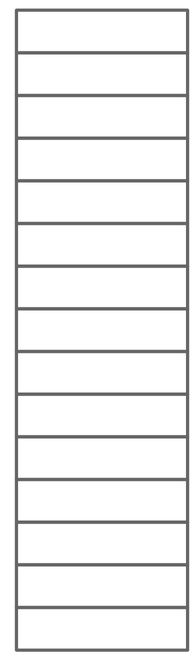
Information transfer



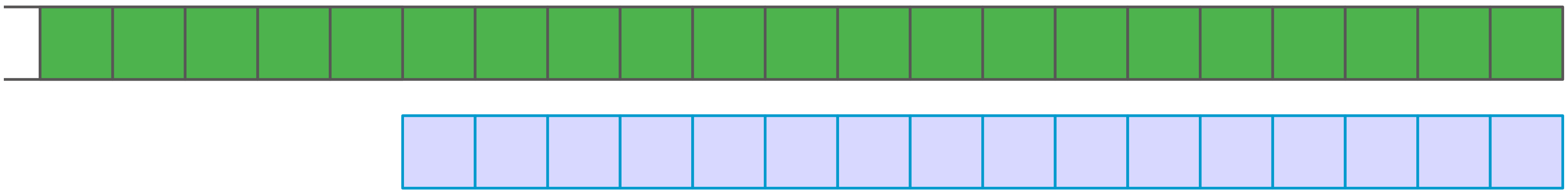
Fixed value

? Unknown value chosen uniformly at random from $[q]$

Memory cells



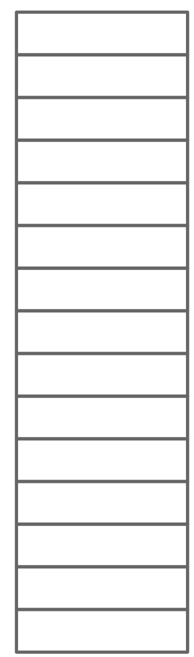
Information transfer



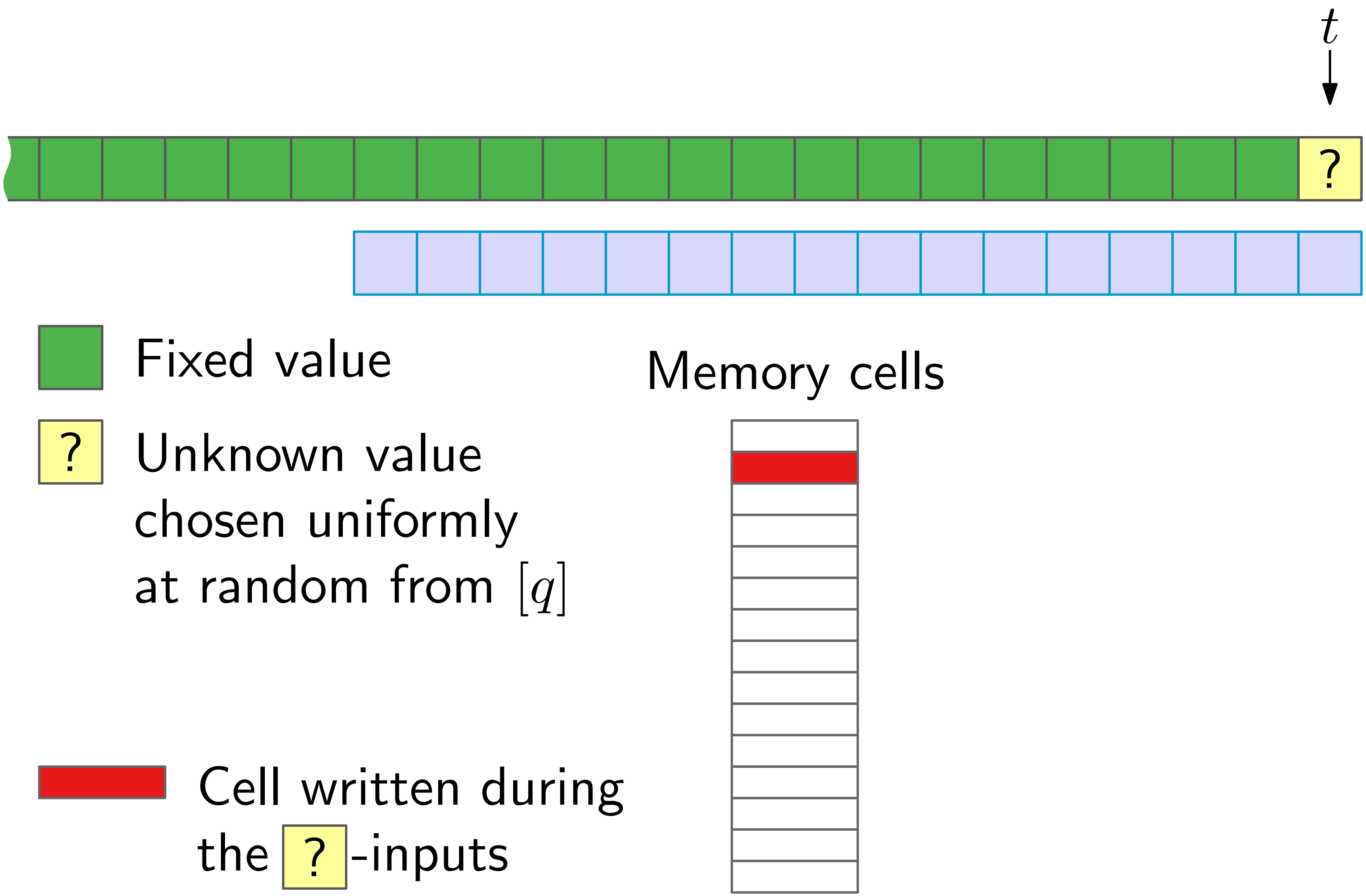
Fixed value

? Unknown value chosen uniformly at random from $[q]$

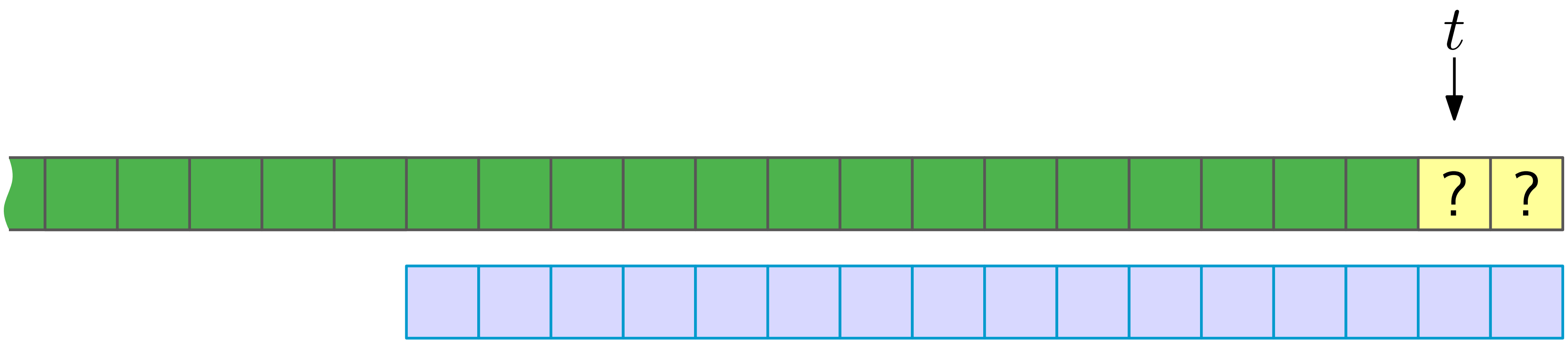
Memory cells



Information transfer



Information transfer

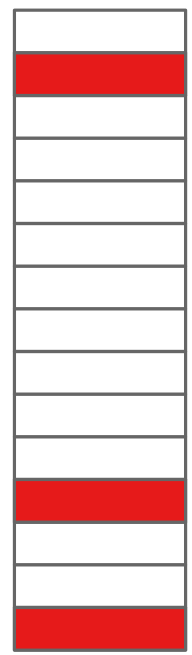


Fixed value

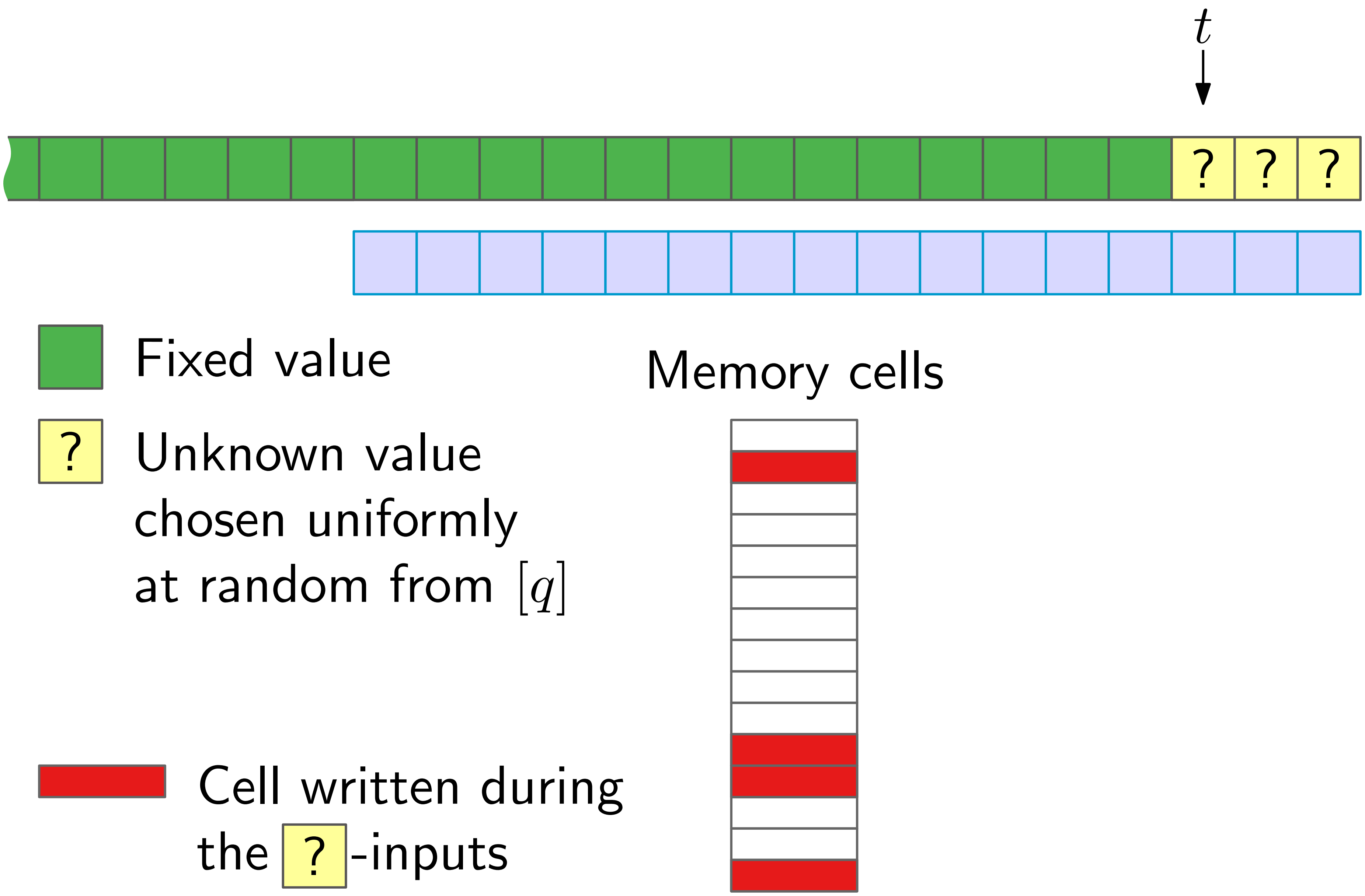
? Unknown value chosen uniformly at random from $[q]$

Cell written during the ? -inputs

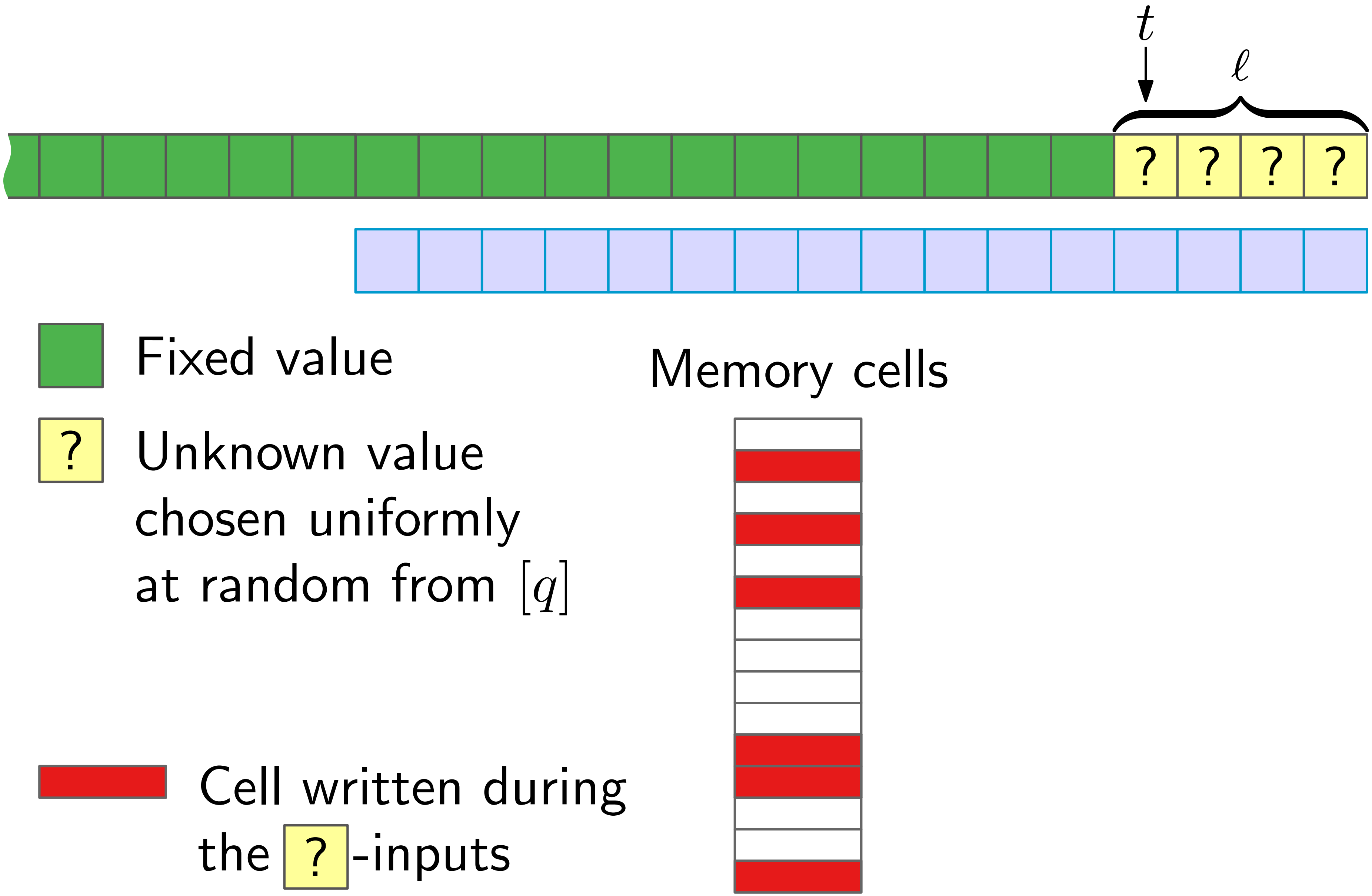
Memory cells



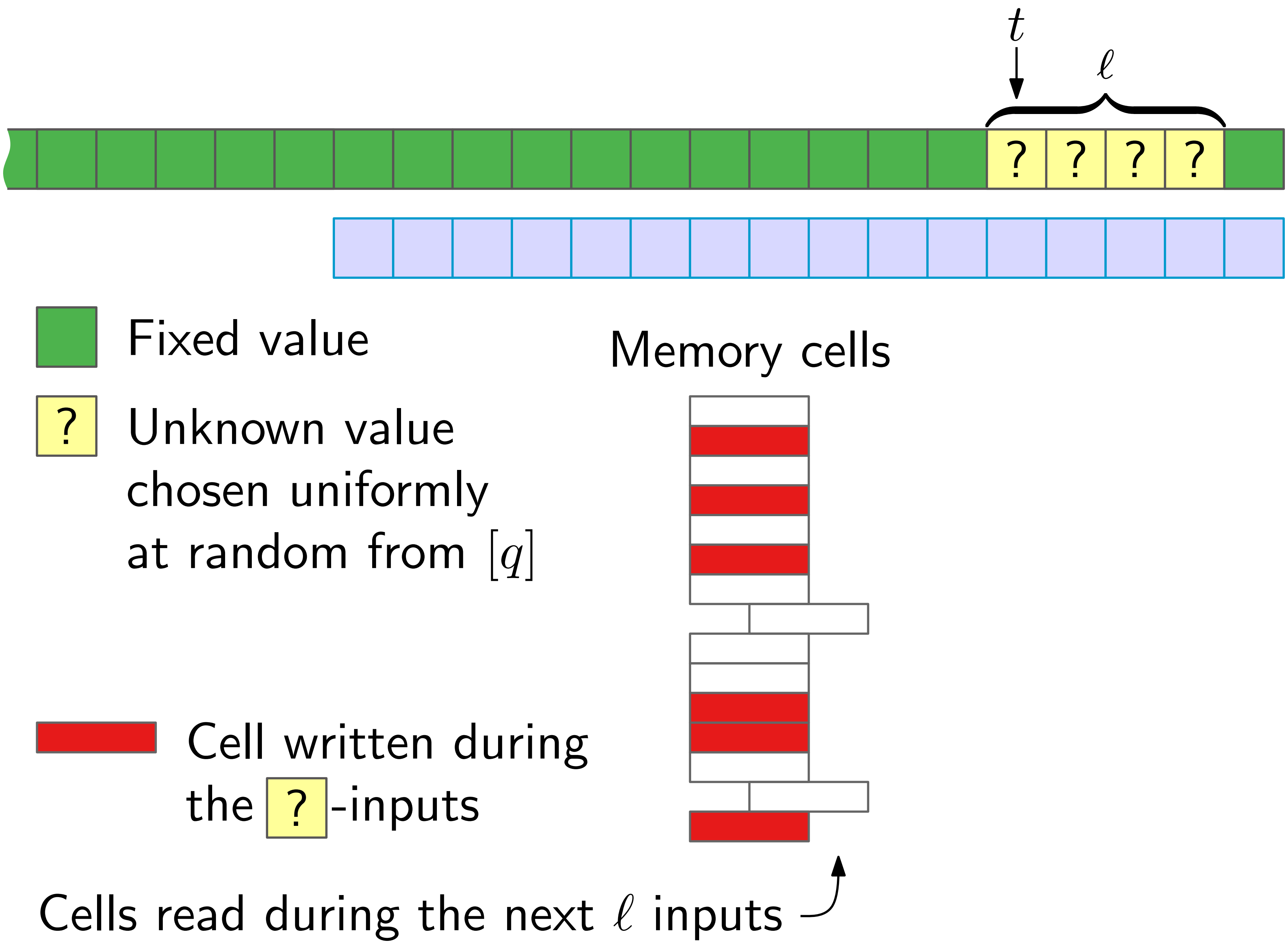
Information transfer



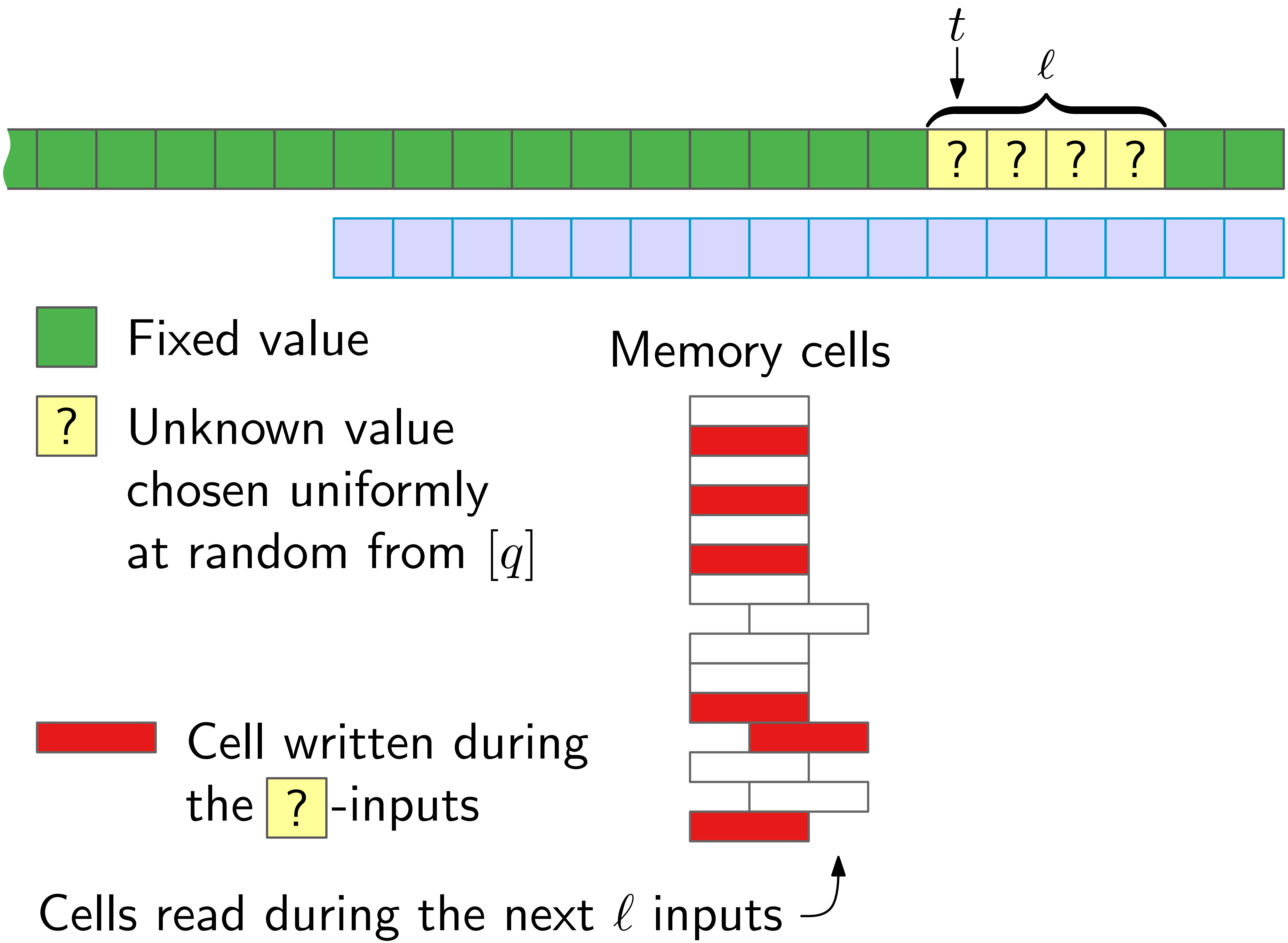
Information transfer



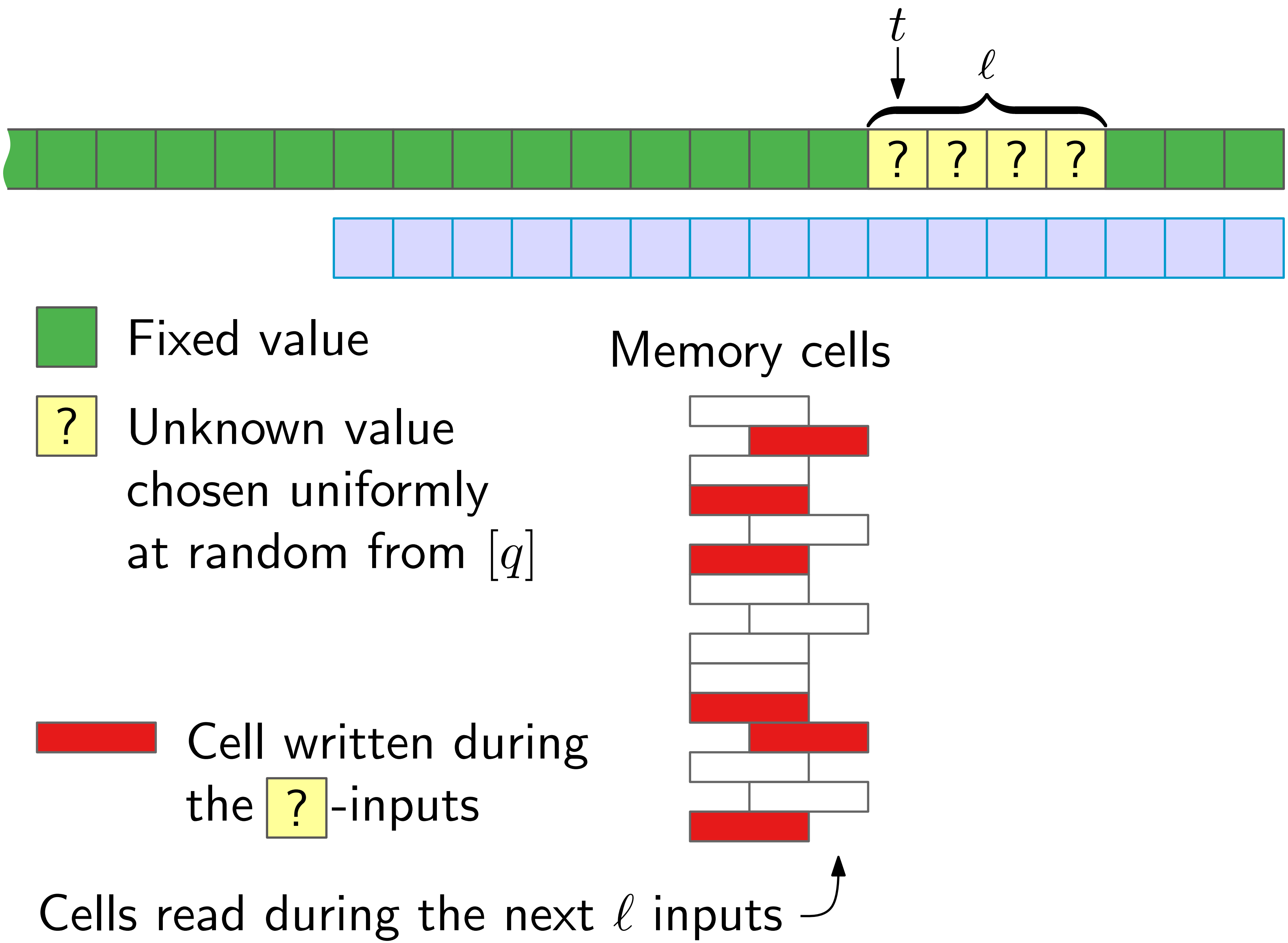
Information transfer



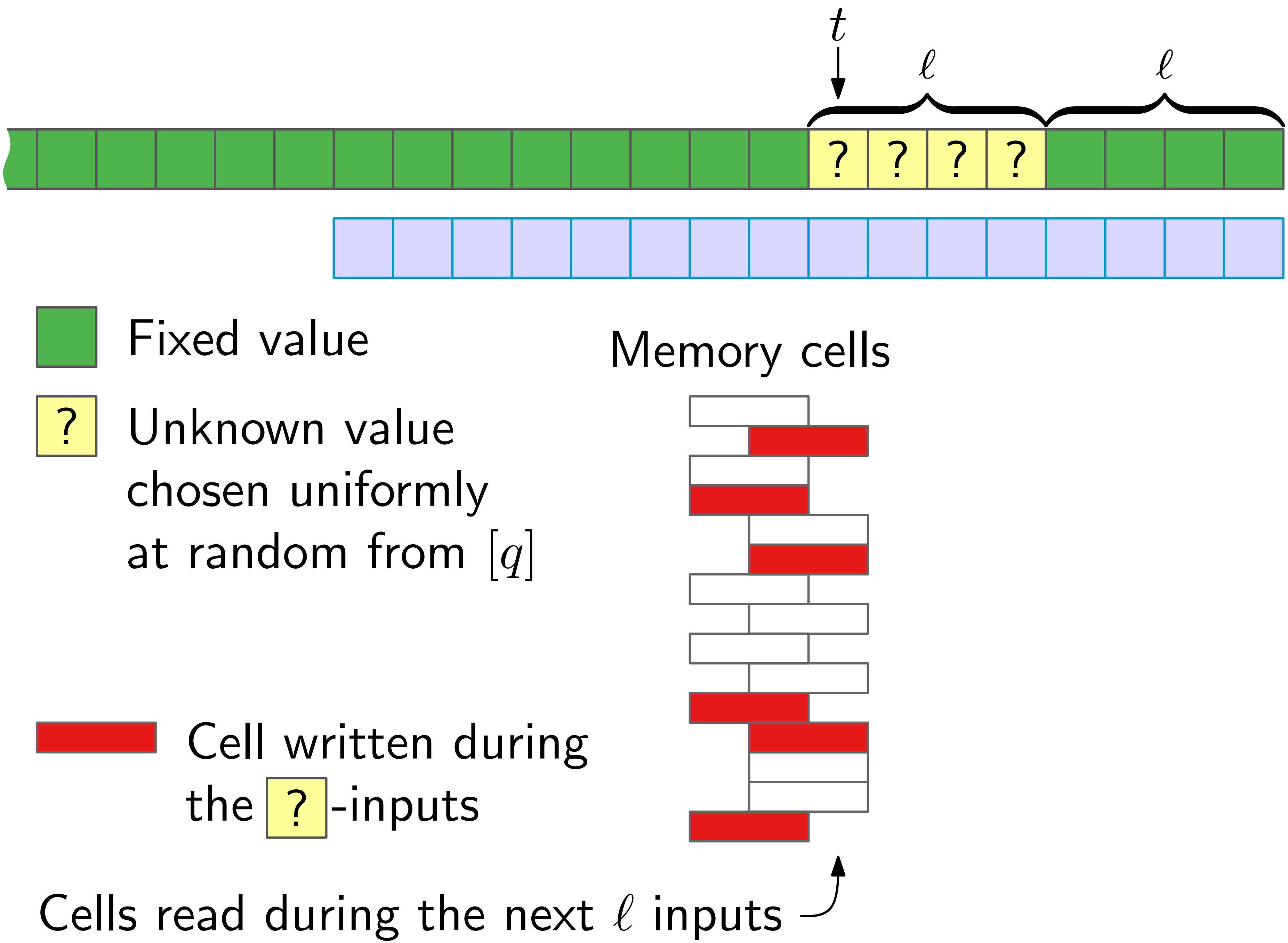
Information transfer



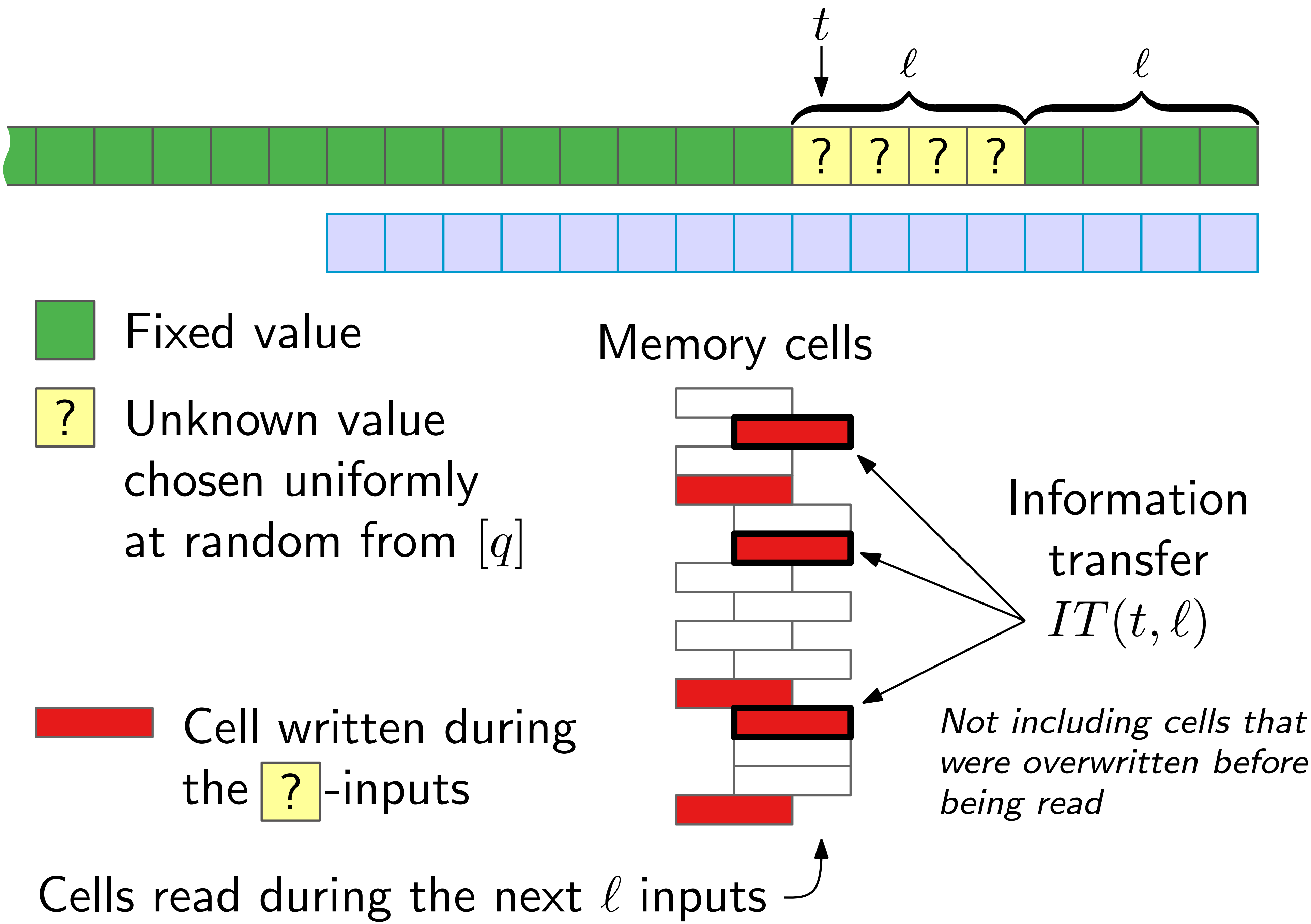
Information transfer



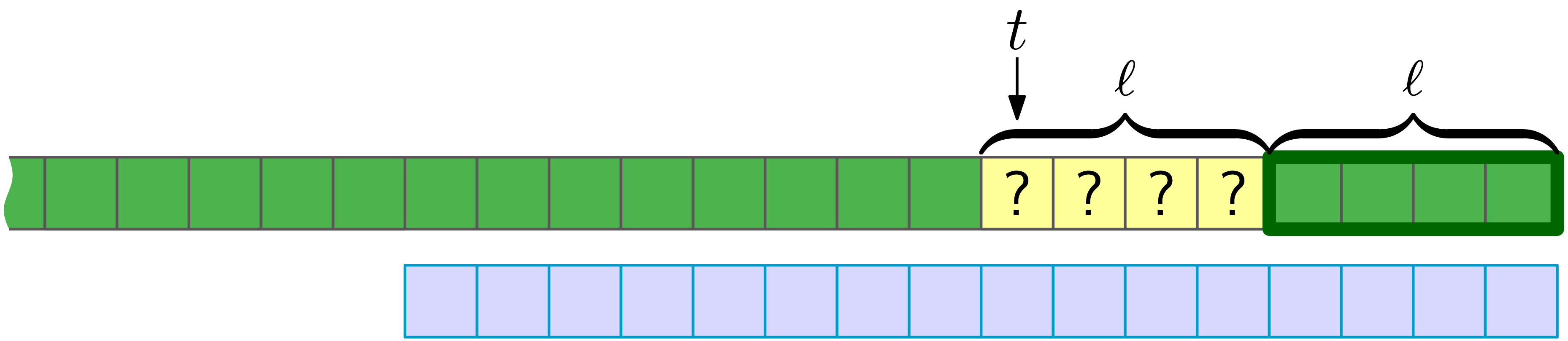
Information transfer



Information transfer



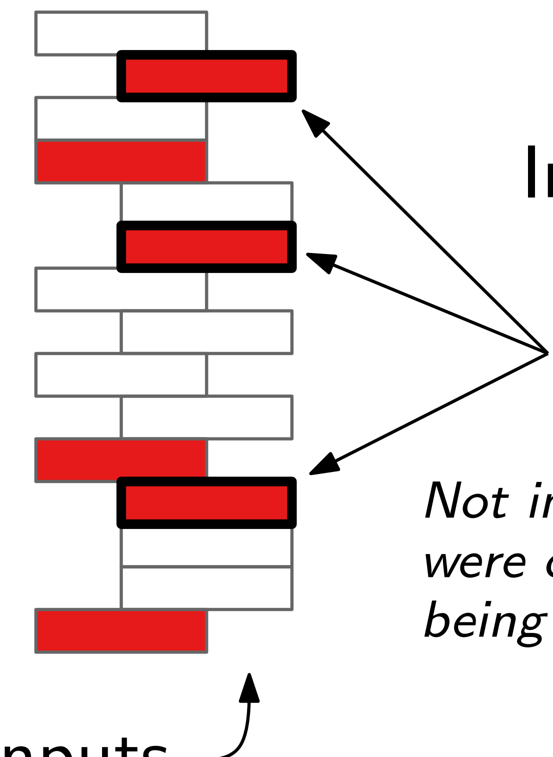
Information transfer



Fixed value

? Unknown value chosen uniformly at random from $[q]$

Memory cells

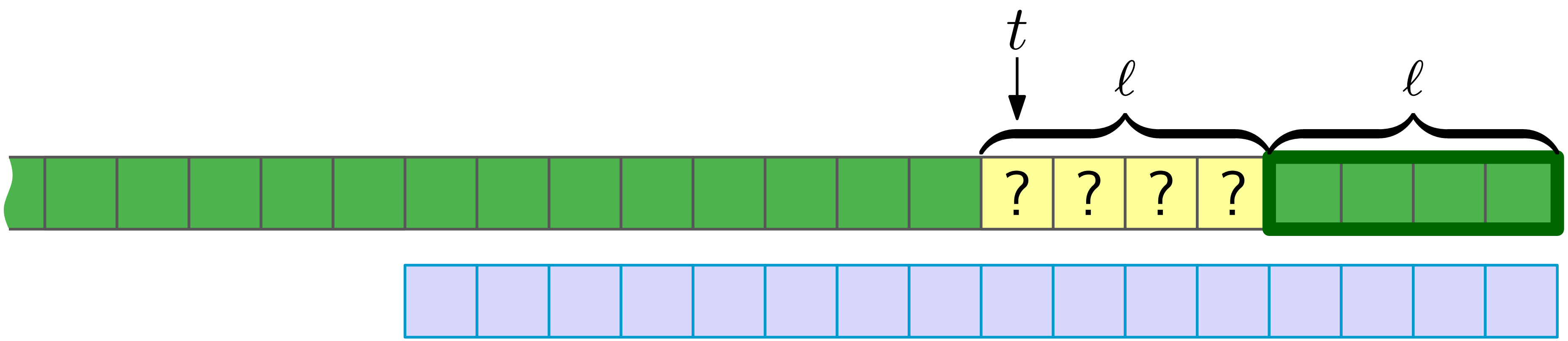


Information transfer $IT(t, \ell)$

Not including cells that were overwritten before being read

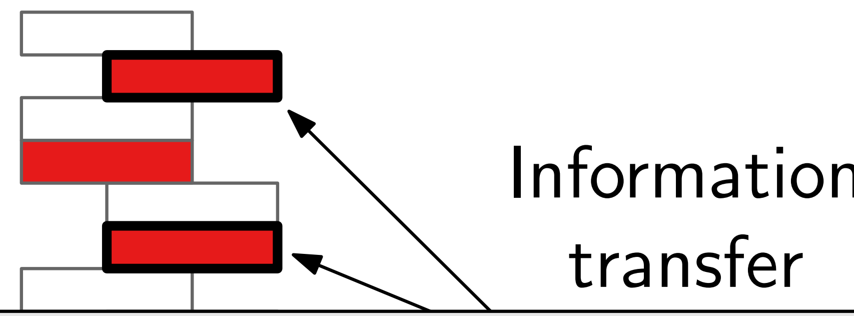
The cells in $IT(t, \ell)$ provide sufficient information in order to give correct output during

Information transfer



Fixed value

? Unknown value chosen uniformly at random from $[q]$



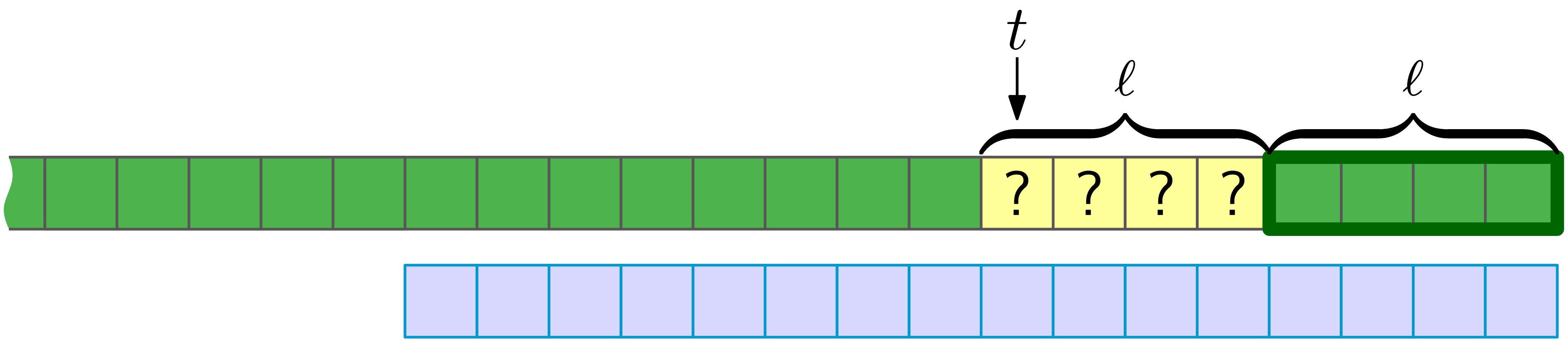
The conditional entropy

H(the outputs during | all fixed)

$$\leqslant w + 2w \cdot \mathbb{E}\left[|IT(t,\ell)| \mid \text{all fixed}\right]$$

w bits per cell

Information transfer



Fixed value

? Unknown value chosen uniformly at random from $[q]$

$|IT(t,\ell)|$

Cell	Address	Contents
	00124	76112
	34123	88819
	92540	01882

w bits to encode $|IT(t,\ell)|$

w bits

w bits

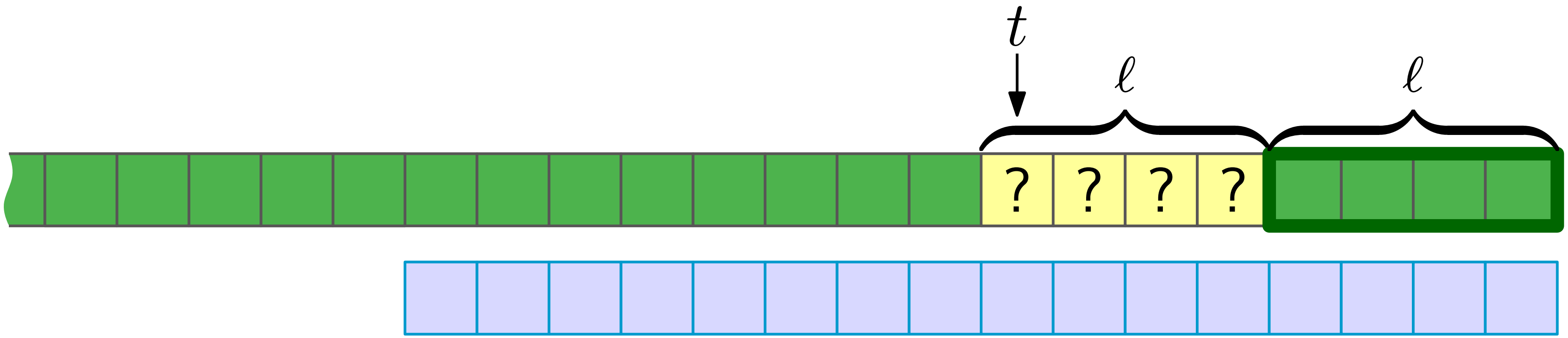
The conditional entropy

$H($the outputs during $\;|$ all fixed$)$

$\leqslant w + 2w \cdot \mathbb{E}\left[|IT(t,\ell)| \;|\; \text{all fixed}\right]$

w bits per cell

Information transfer



Fixed value

? Unknown value chosen uniformly at random from $[q]$

$|IT(t,\ell)|$

Cell	Address	Contents
	00124	76112
	00000	00000
	92540	01882

w bits to encode $|IT(t,\ell)|$

w bits w bits

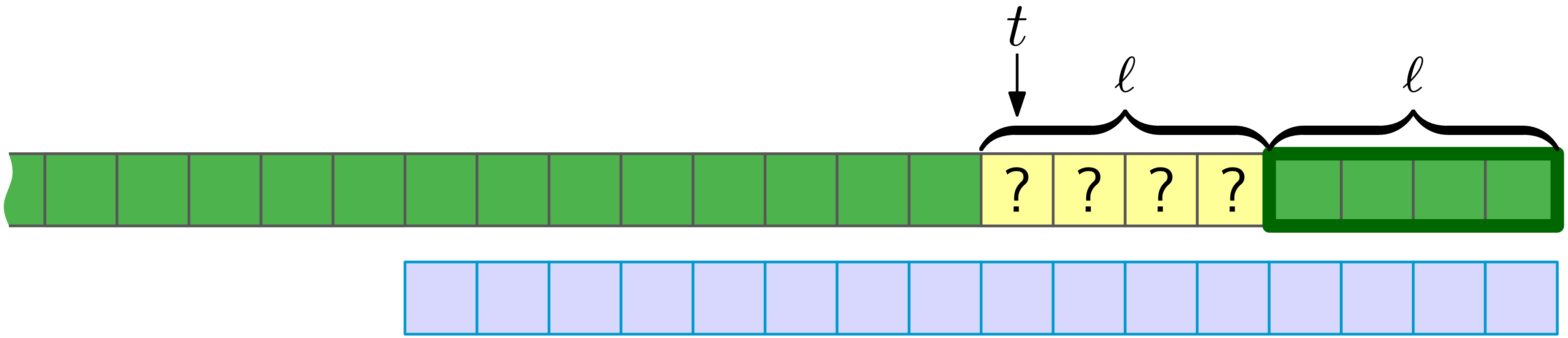
The conditional entropy

$H($the outputs during $|$ all fixed$)$

$$\leqslant w + 2w \cdot \mathbb{E}\left[|IT(t,\ell)| \mid \text{all fixed}\right]$$

w bits per cell

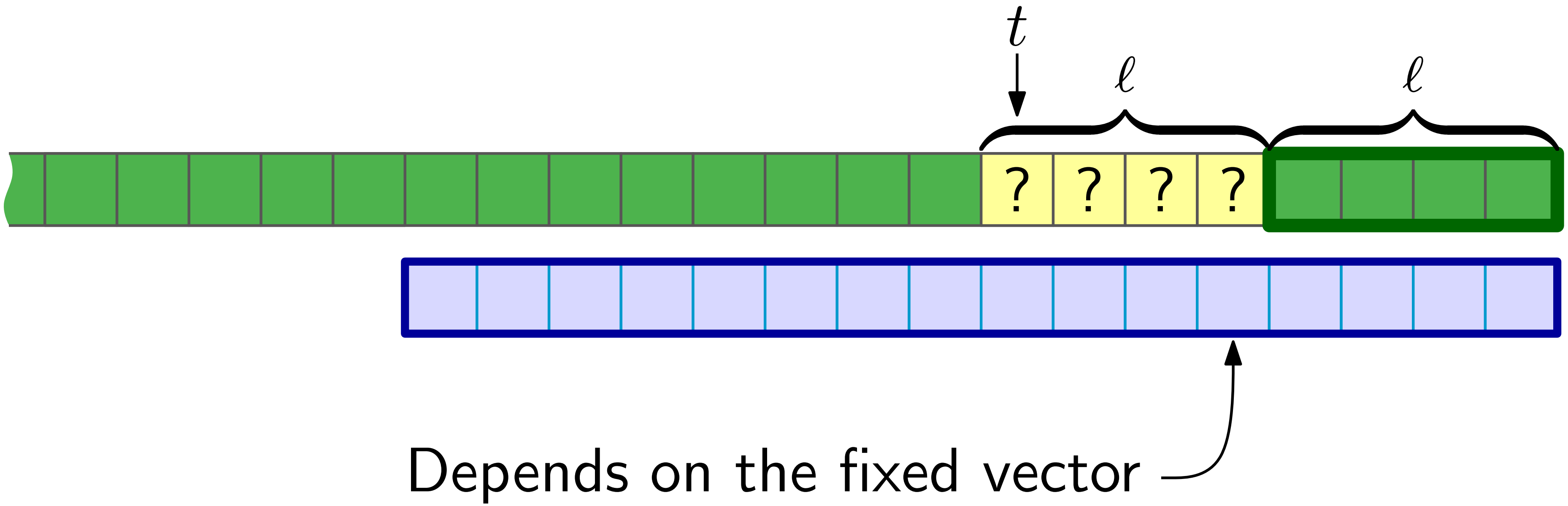
Information transfer



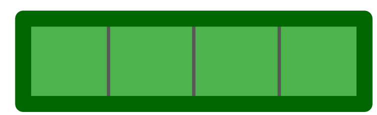
How much information about ? ? ? ? do we **need**

in order to give correct outputs during ?

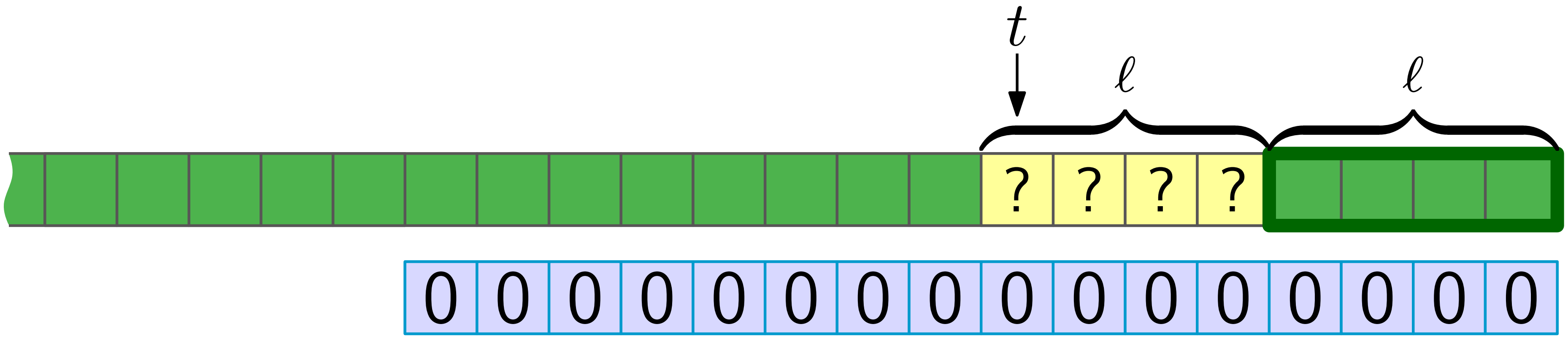
Information transfer



How much information about | ? | ? | ? | ? | do we **need**

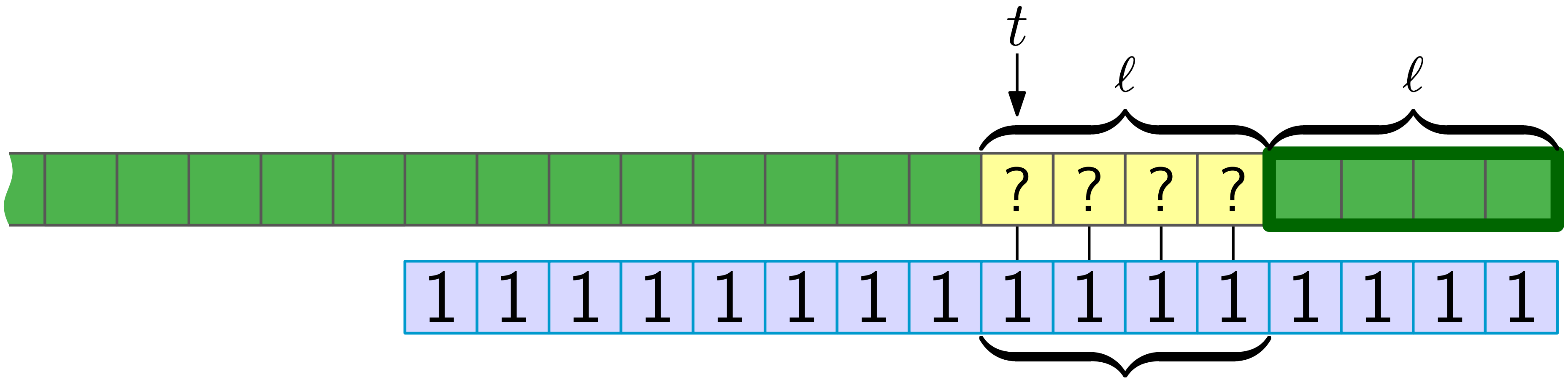
in order to give correct outputs during  ?

Information transfer



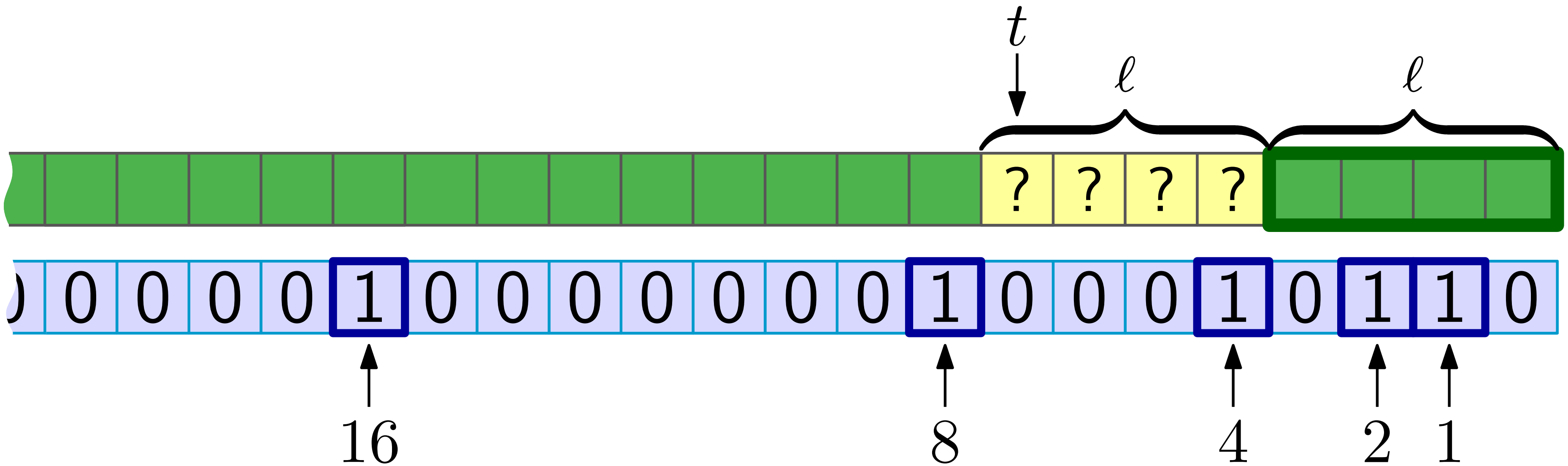
Output is always 0 (no information)

Information transfer



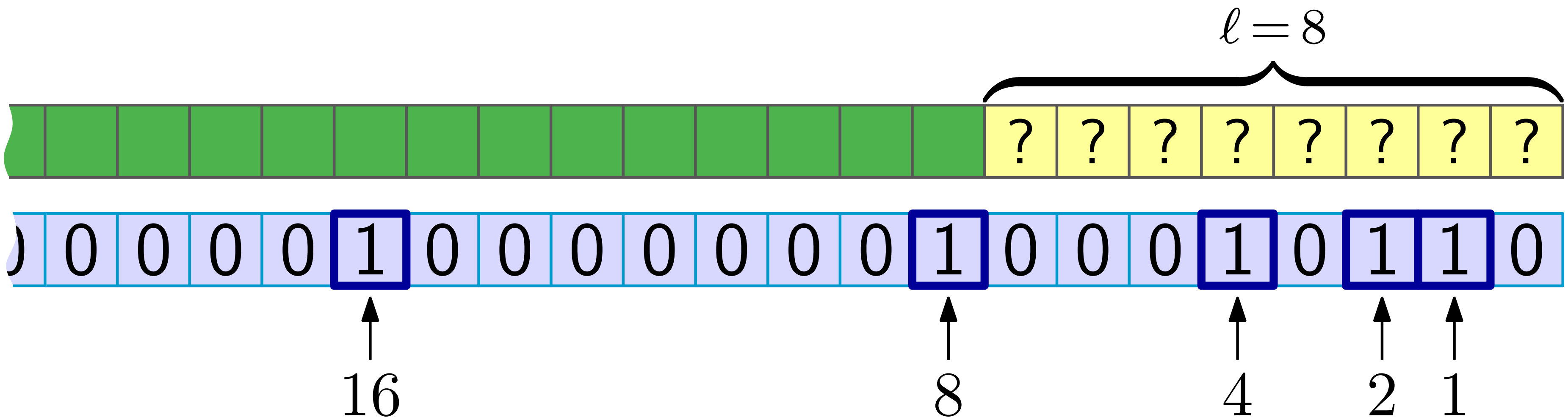
Contributes to the dot product with the same value at each alignment
($\delta = \log q$ bits of information)

Information transfer



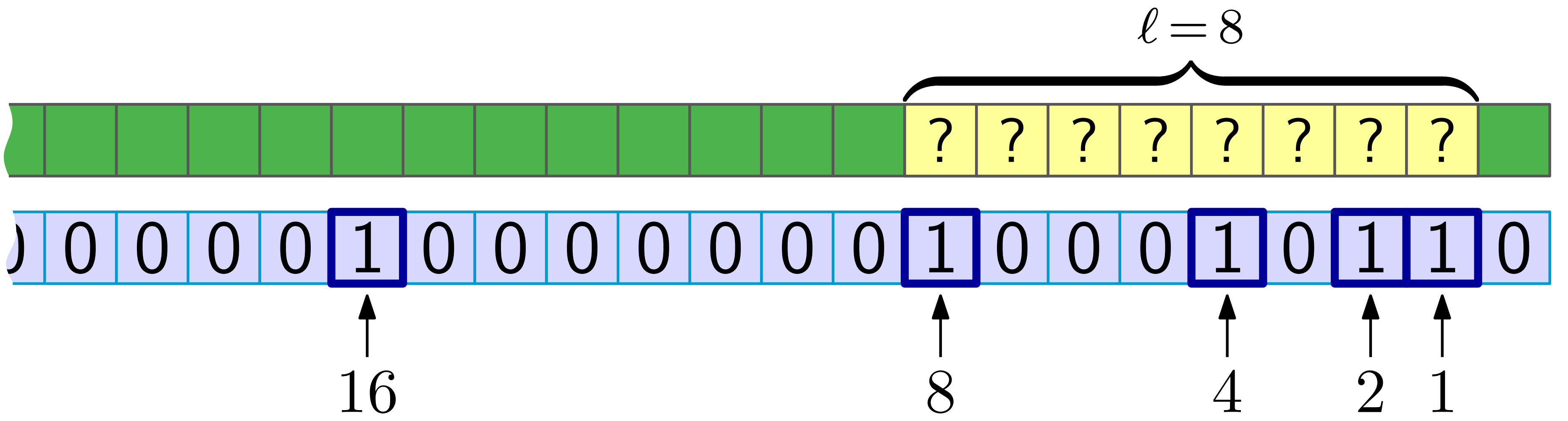
$\boxed{1}$ if the position is a power of 2

Information transfer



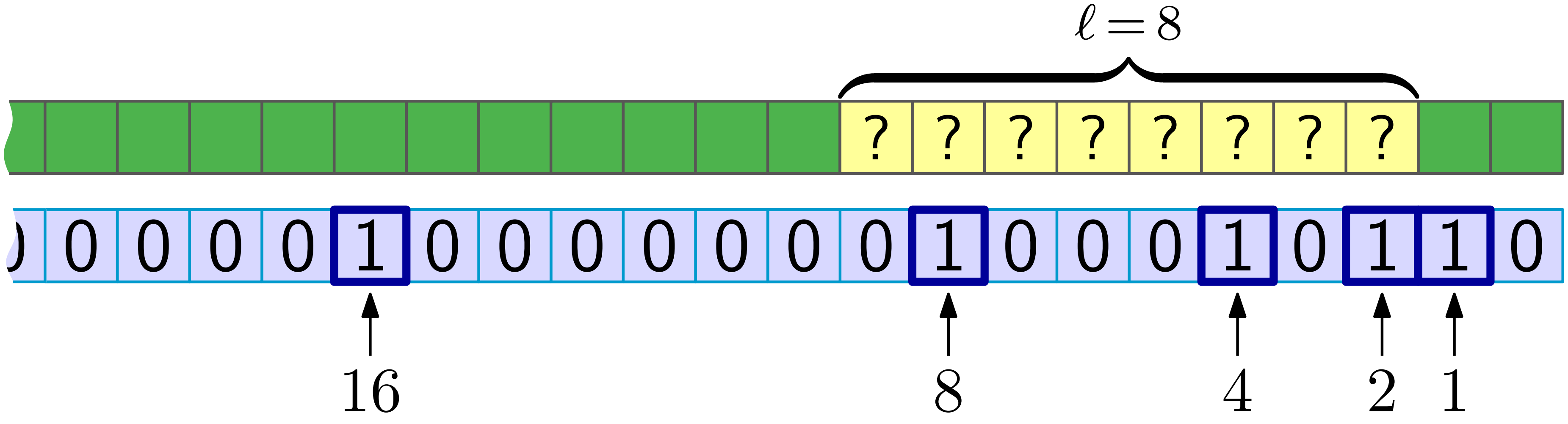
1 if the position is a power of 2

Information transfer



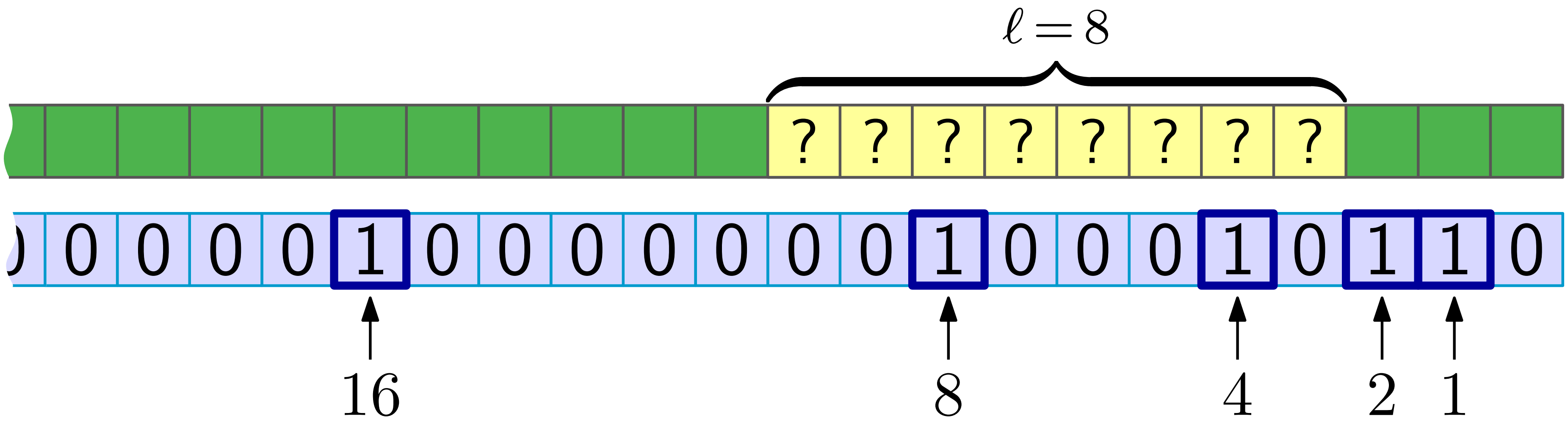
$\boxed{1}$ if the position is a power of 2

Information transfer



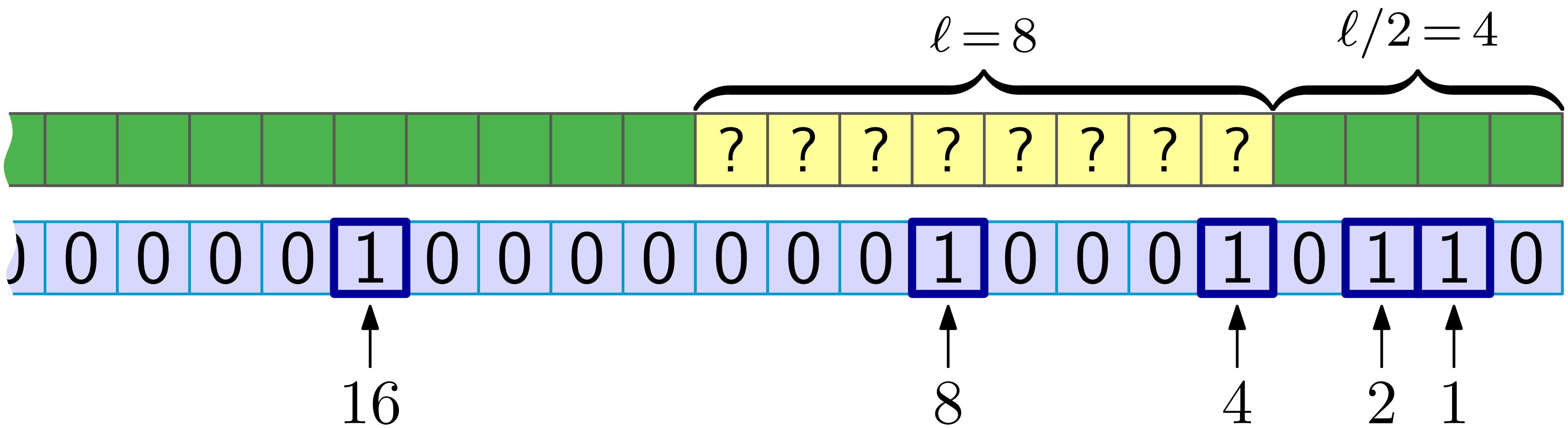
$\boxed{1}$ if the position is a power of 2

Information transfer



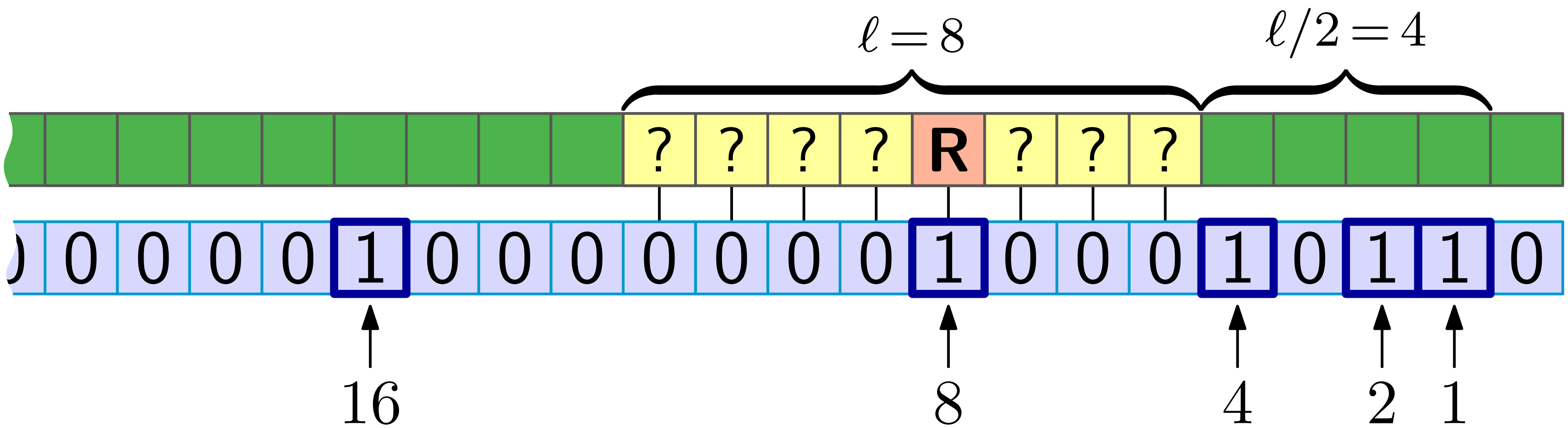
$\boxed{1}$ if the position is a power of 2

Information transfer



$\boxed{1}$ if the position is a power of 2

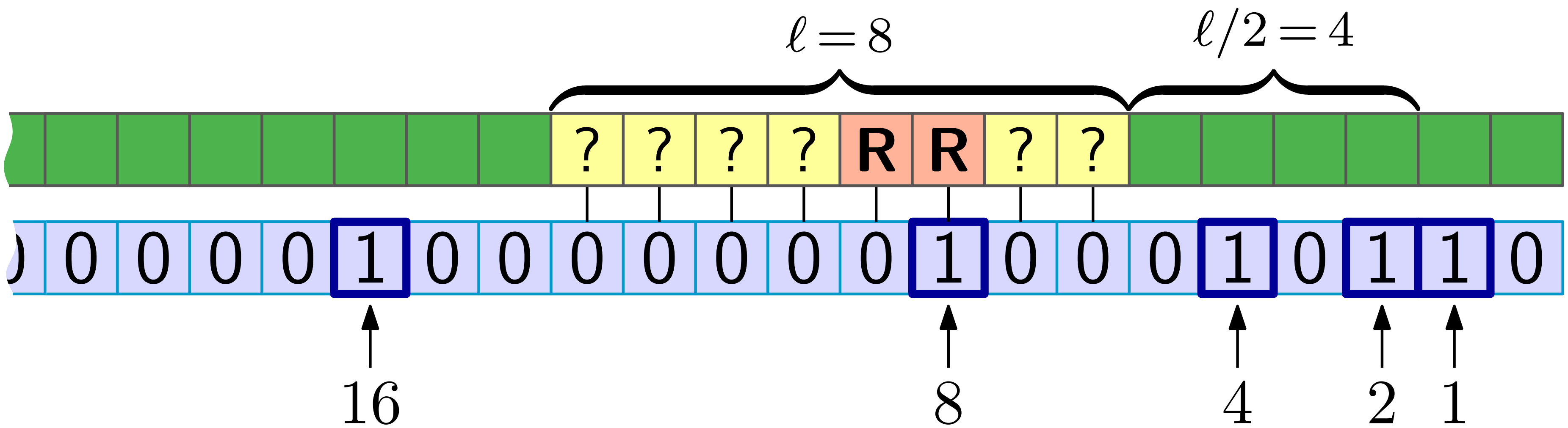
Information transfer



$\boxed{1}$ if the position is a power of 2

R = a recovered value
(recall that $\boxed{?}$ is chosen uniformly at random from $[q]$, hence contributes with $\delta = \log q$ bits to the entropy)

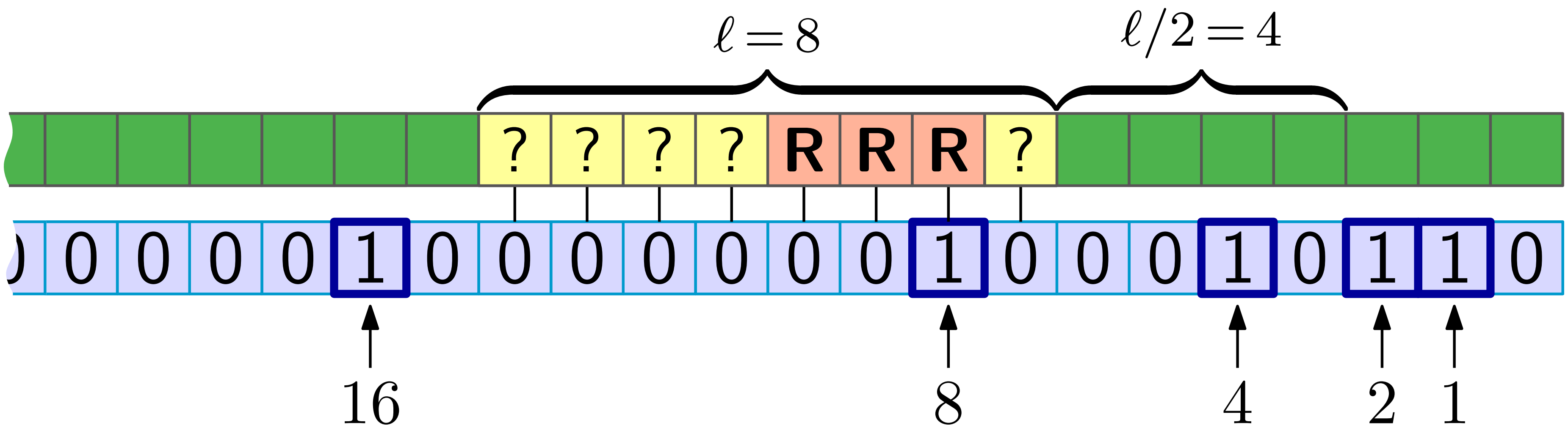
Information transfer



1 if the position is a power of 2

R = a recovered value
(recall that ? is chosen uniformly at random from $[q]$, hence contributes with $\delta = \log q$ bits to the entropy)

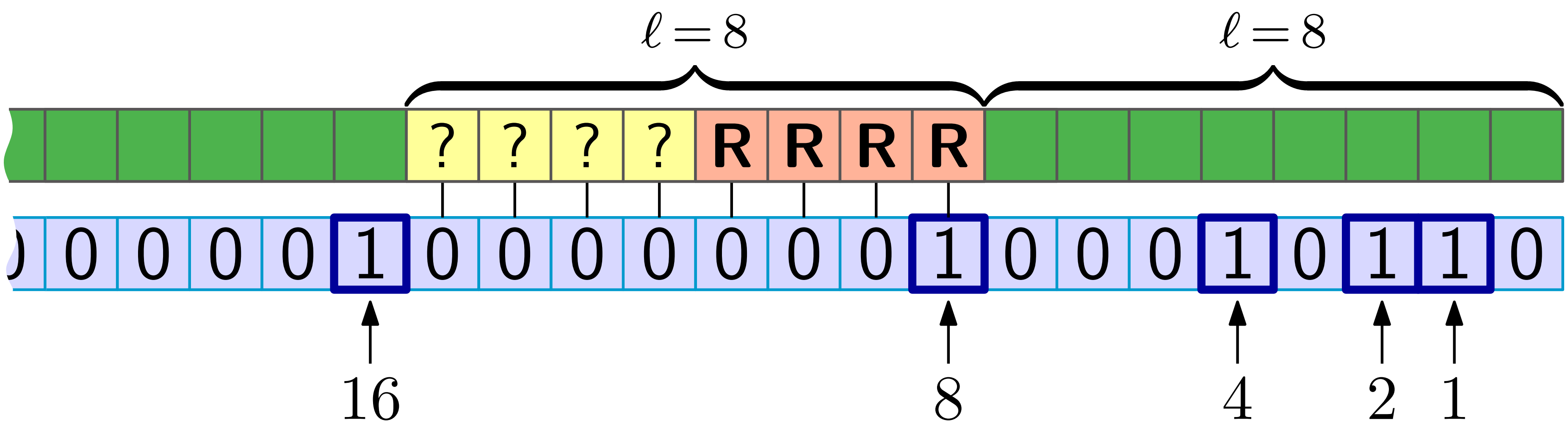
Information transfer



1 if the position is a power of 2

R = a recovered value
(recall that ? is chosen uniformly at random from $[q]$, hence contributes with $\delta = \log q$ bits to the entropy)

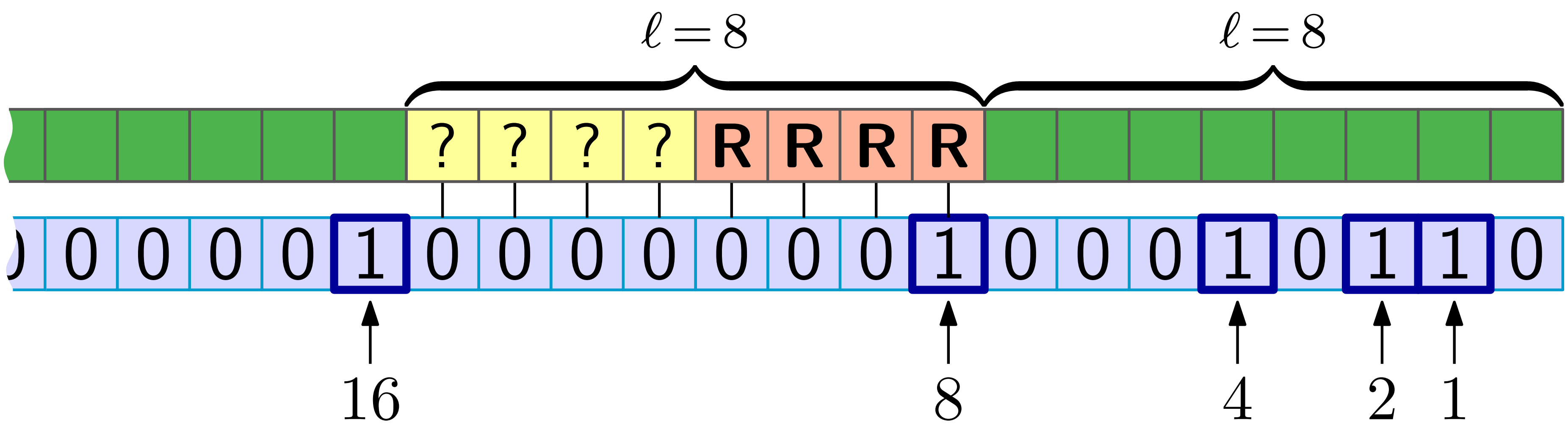
Information transfer



1 if the position is a power of 2

R = a recovered value
(recall that ? is chosen uniformly at random from $[q]$, hence contributes with $\delta = \log q$ bits to the entropy)

Information transfer

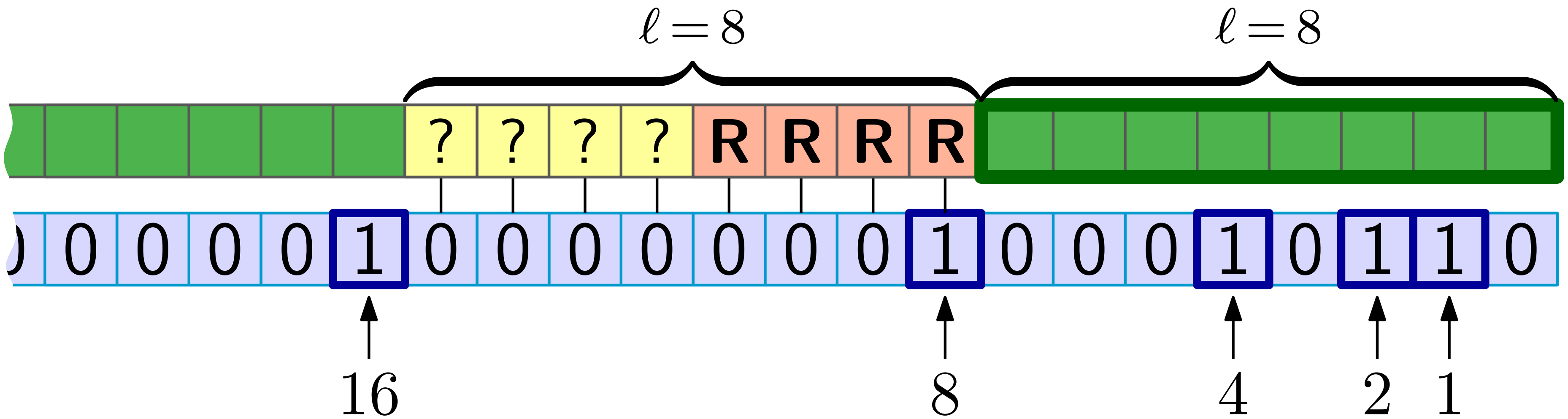


$\boxed{1}$ if the position is a power of 2

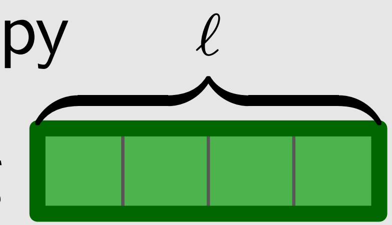
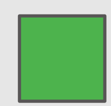
$\boxed{\textbf{R}}$ $=$ a recovered value
(recall that $\boxed{?}$ is chosen uniformly at random from $[q]$, hence contributes with $\delta = \log q$ bits to the entropy)

Conclusion: If ℓ is a power of 2 then we recover $\frac{\ell}{2}$ values

Information transfer

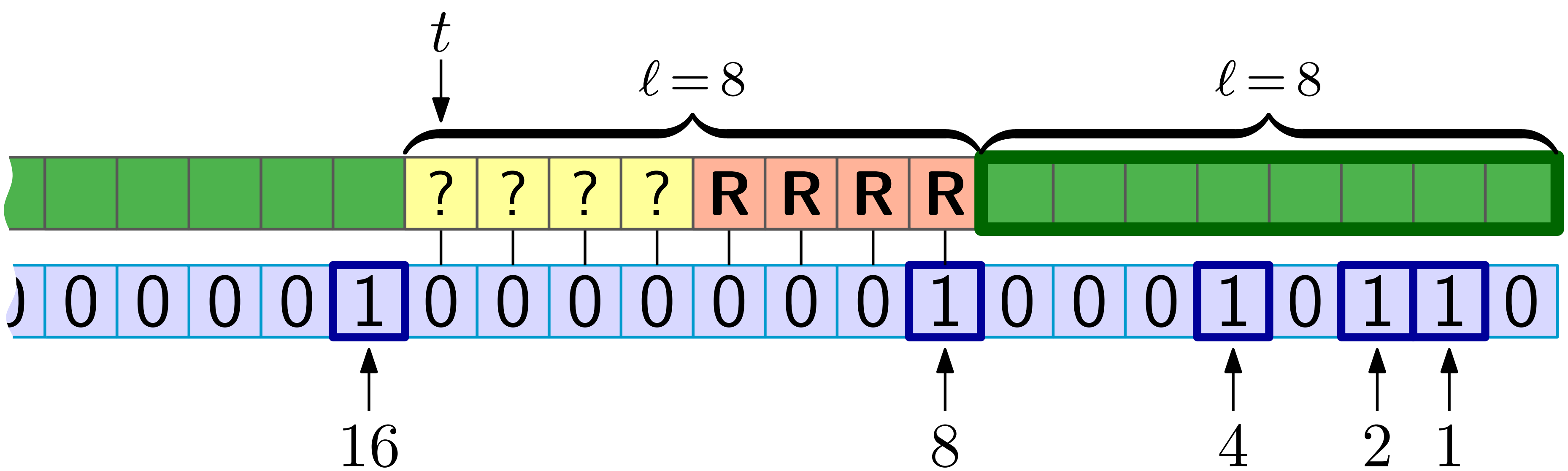


The conditional entropy

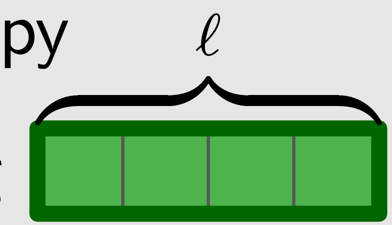
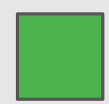
H(the outputs during  | all  fixed) $\geqslant \frac{\ell}{2}\delta$

Conclusion: If ℓ is a power of 2 then we recover $\frac{\ell}{2}$ values

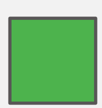
Information transfer



The conditional entropy

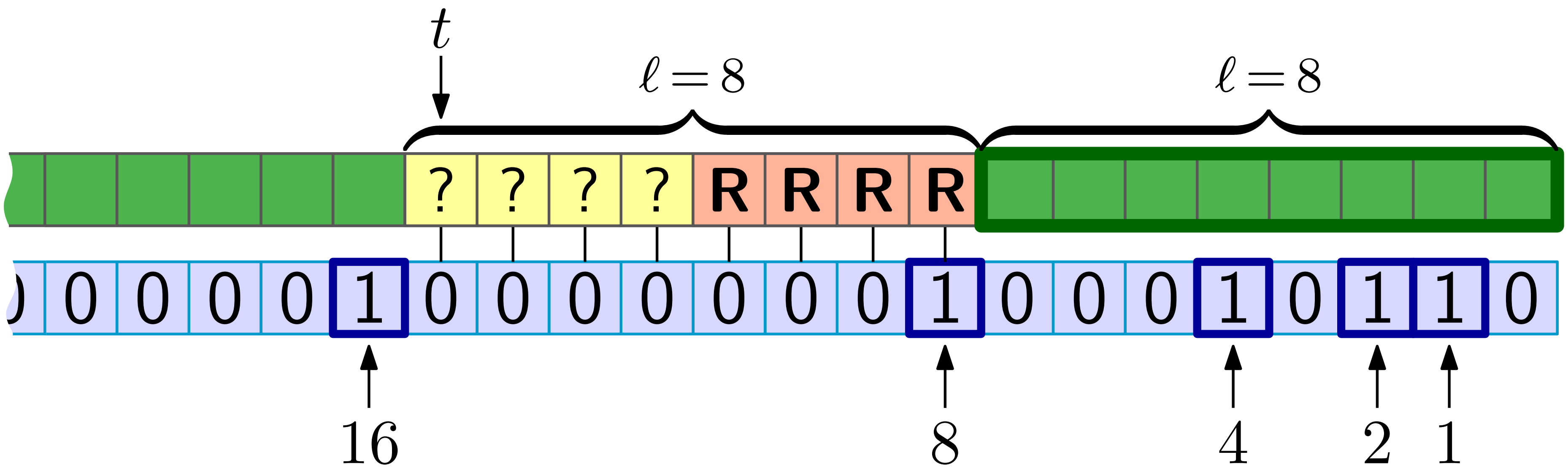
H(the outputs during  | all  fixed) $\geqslant \frac{\ell}{2}\delta$

The conditional information transfer

$\mathbb{E}\,[|IT(t,\ell)|$ | all  fixed] $\geqslant \frac{\delta}{4w}\ell - \frac{1}{2}$

w bits per cell

Information transfer



Suppose that all values (and ?) from the stream are chosen uniformly at random from $[q]$.

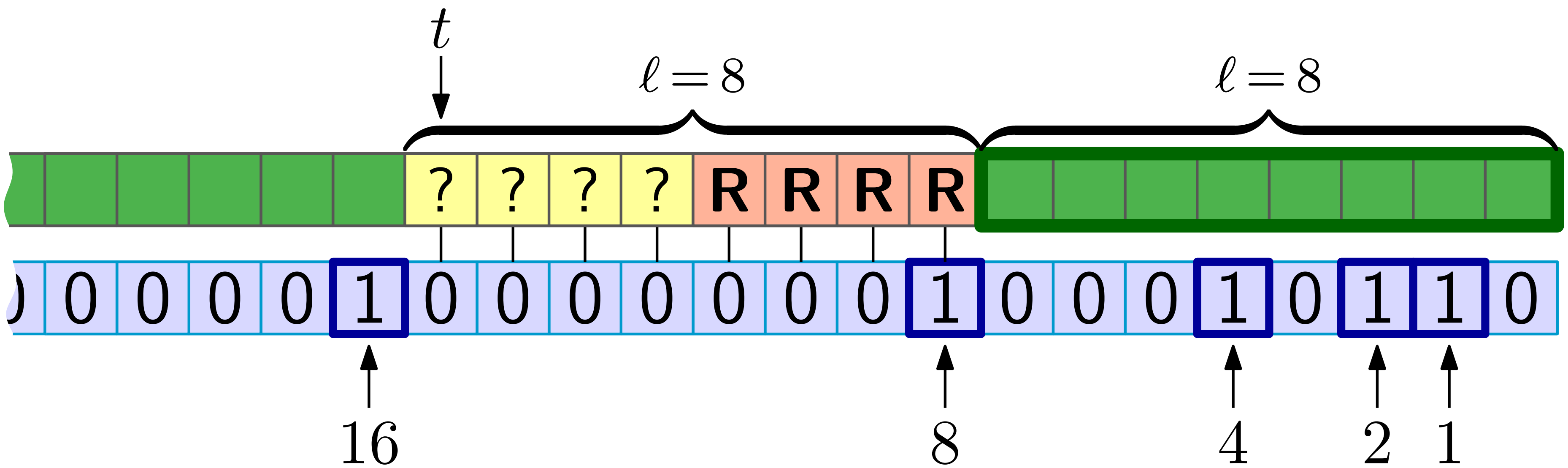
By linearity of expectation. . .

The conditional information transfer

$$\mathbb{E}\left[|IT(t,\ell)| \;\middle|\; \text{all } \text{ fixed}\right] \;\geqslant\; \frac{\delta}{4w}\ell - \frac{1}{2}$$

w bits per cell

Information transfer



Suppose that all values (and ?) from the stream are chosen uniformly at random from $[q]$.

By linearity of expectation. . .

The conditional information transfer

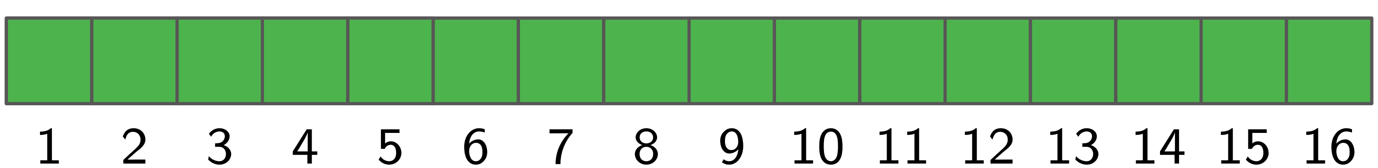
$$\mathbb{E}\left[|IT(t,\ell)| \mid \text{all} \quad \text{fixed}\right] \geqslant \frac{\delta}{4w}\ell - \frac{1}{2}$$

w bits per cell

Total number of cell reads

$$\underbrace{0\;0\;0\;0\;0\;0\;0\;\boxed{1}\;0\;0\;0\;\boxed{1}\;0\;\boxed{1}\,\boxed{1}\;0}_{n}$$

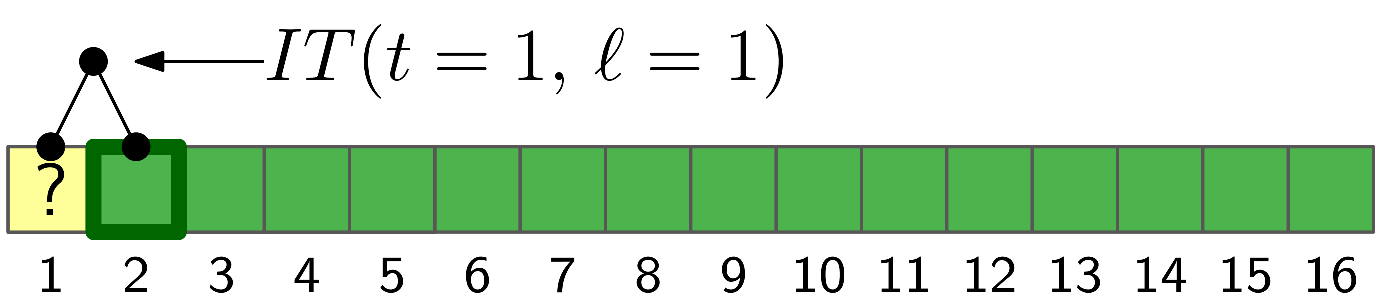
Feed the algorithm with n values chosen uniformly at random from $[q]$.



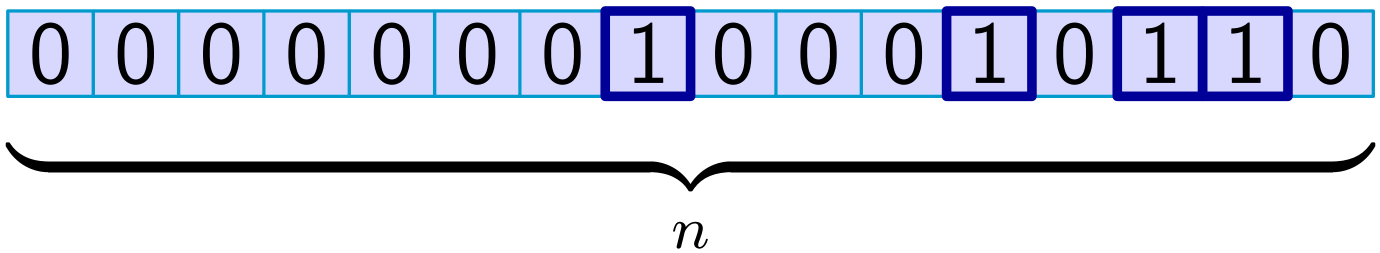
Total number of cell reads

$$\underbrace{0\ 0\ 0\ 0\ 0\ 0\ 0\ \boxed{1}\ 0\ 0\ 0\ \boxed{1}\ 0\ \boxed{1}\ \boxed{1}\ 0}_{n}$$

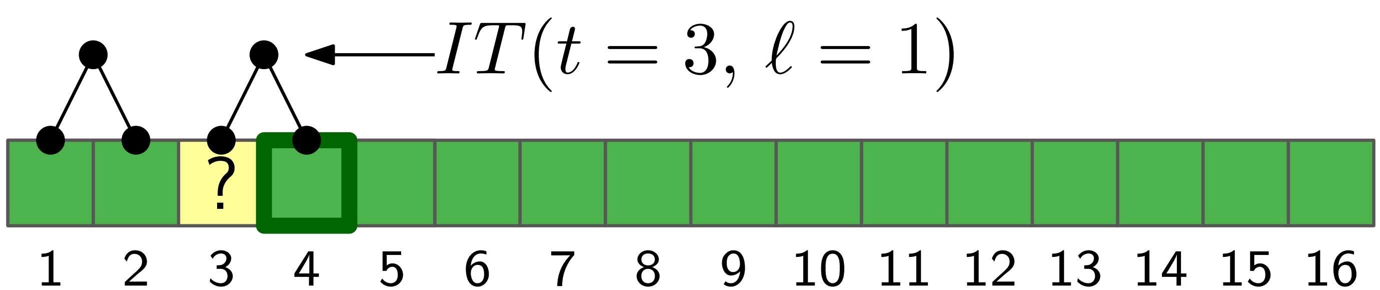
Feed the algorithm with n values chosen uniformly at random from $[q]$.



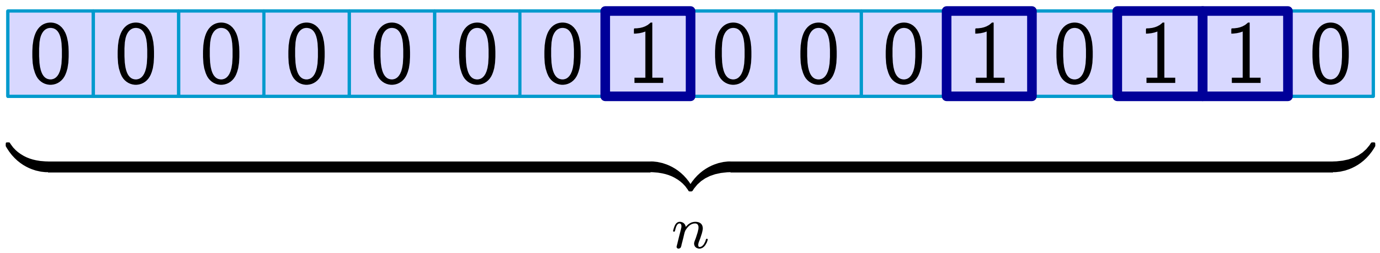
Total number of cell reads



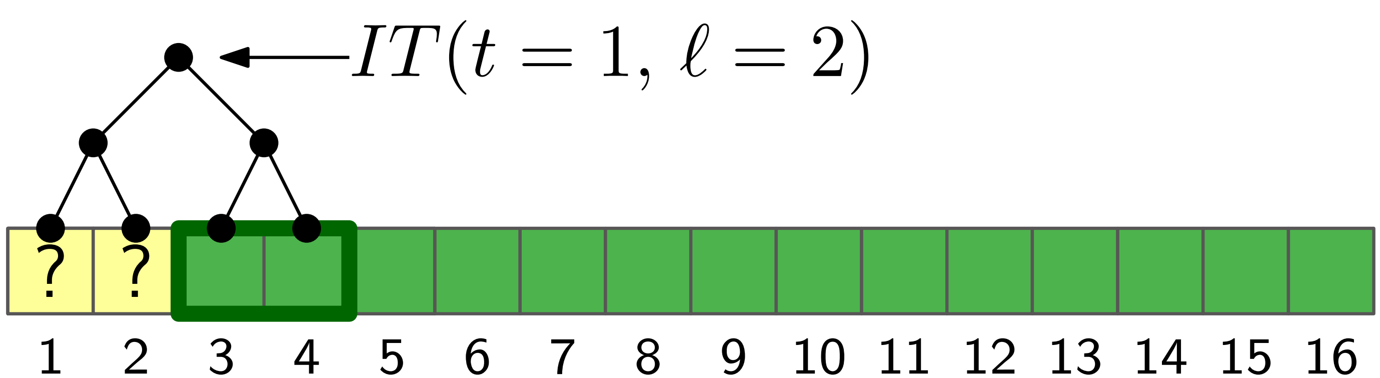
Feed the algorithm with n values chosen uniformly at random from $[q]$.



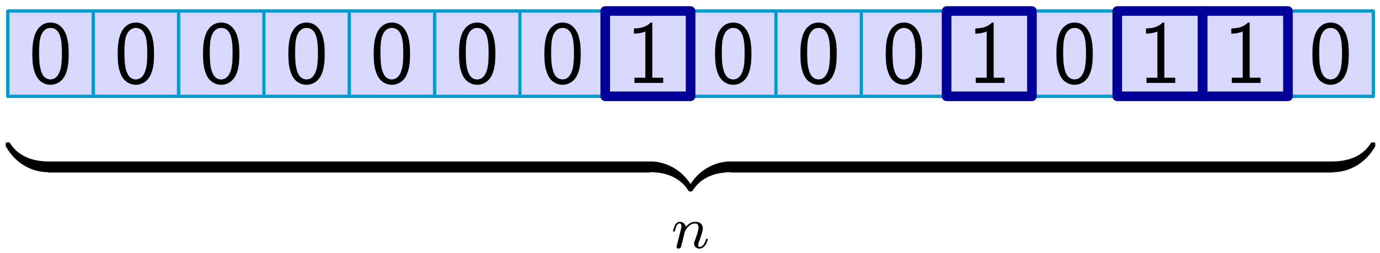
Total number of cell reads



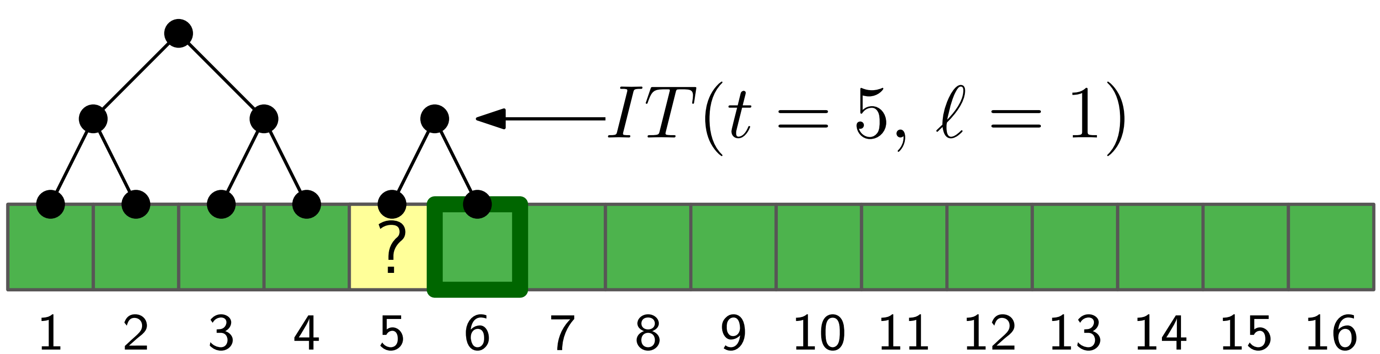
Feed the algorithm with n values chosen uniformly at random from $[q]$.



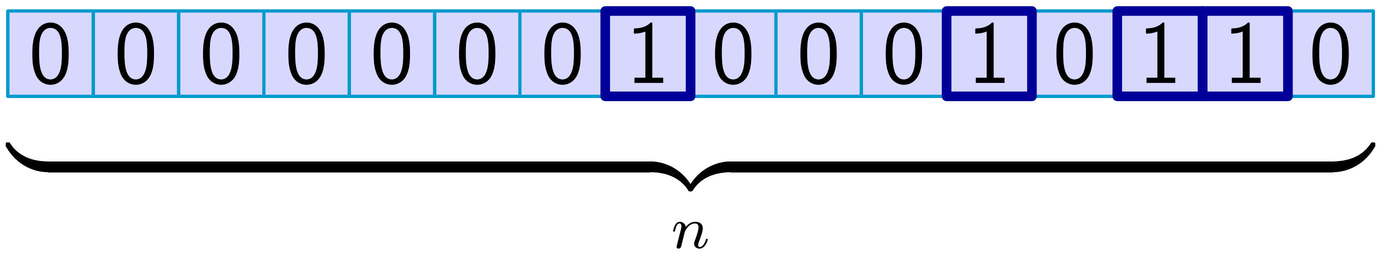
Total number of cell reads



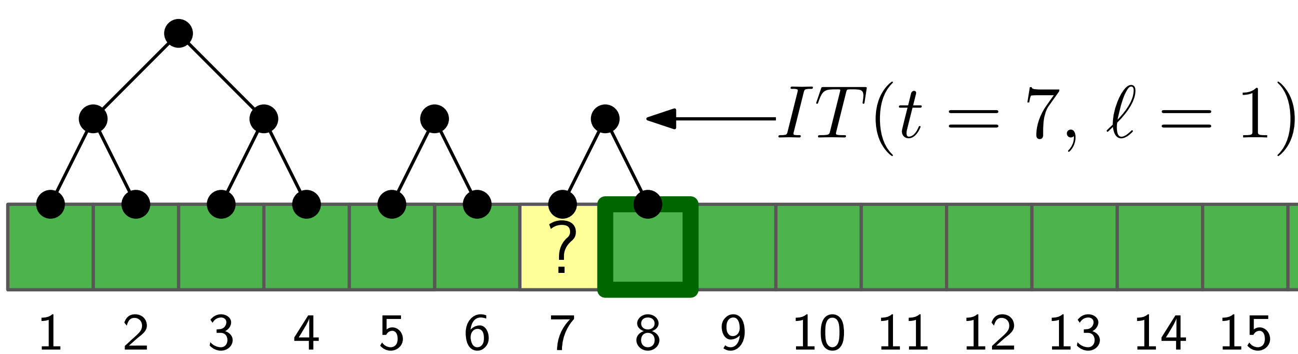
Feed the algorithm with n values chosen uniformly at random from $[q]$.



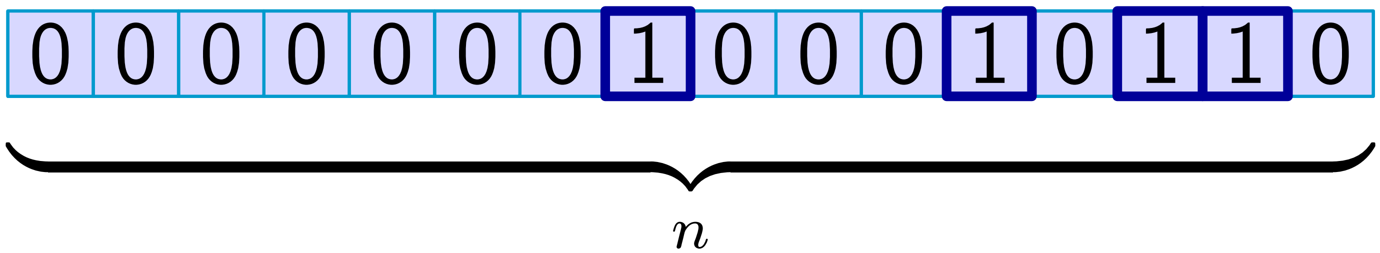
Total number of cell reads



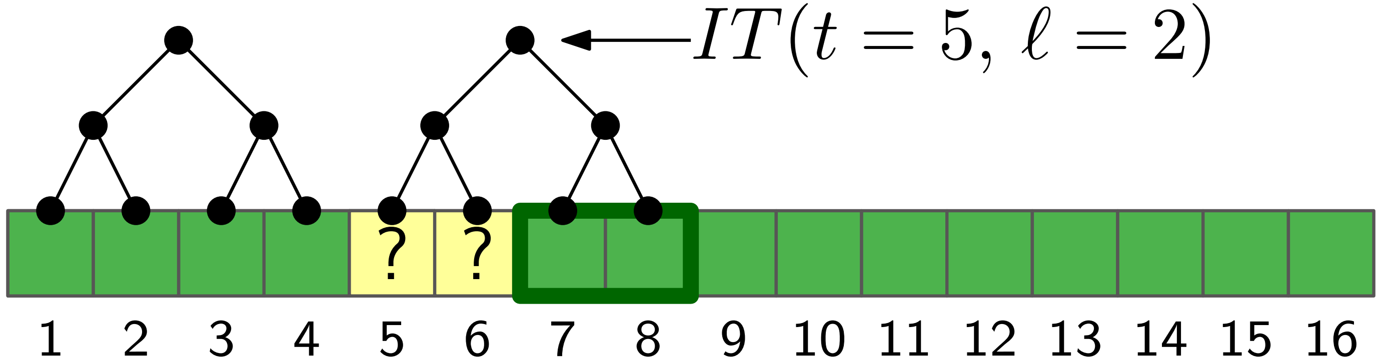
Feed the algorithm with n values chosen uniformly at random from $[q]$.



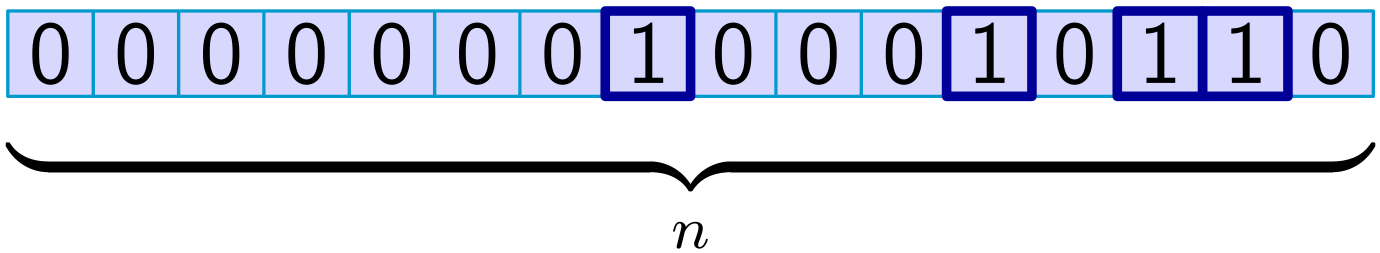
Total number of cell reads



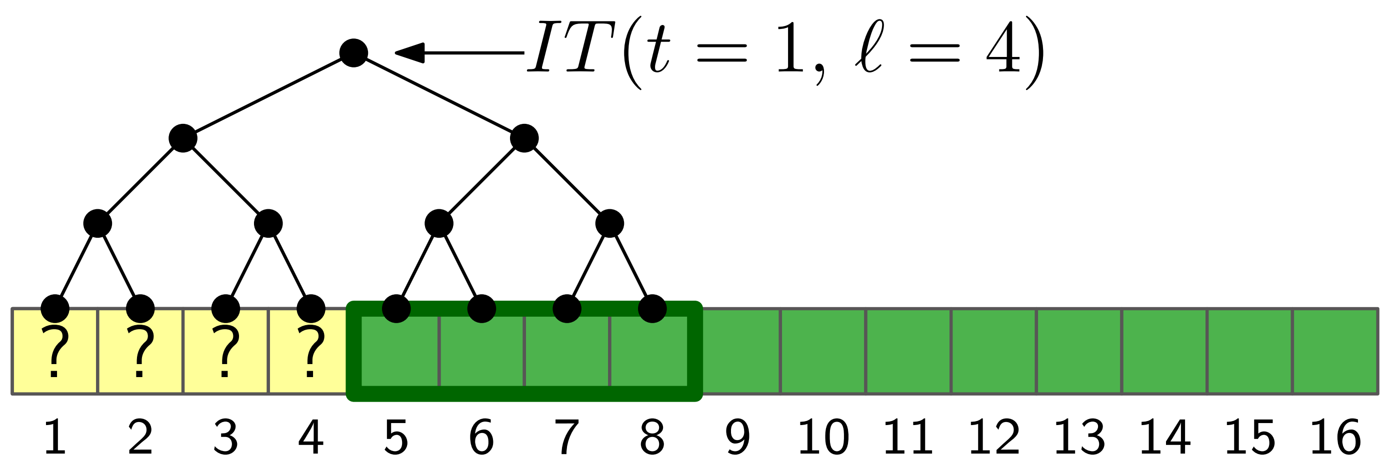
Feed the algorithm with n values chosen uniformly at random from $[q]$.



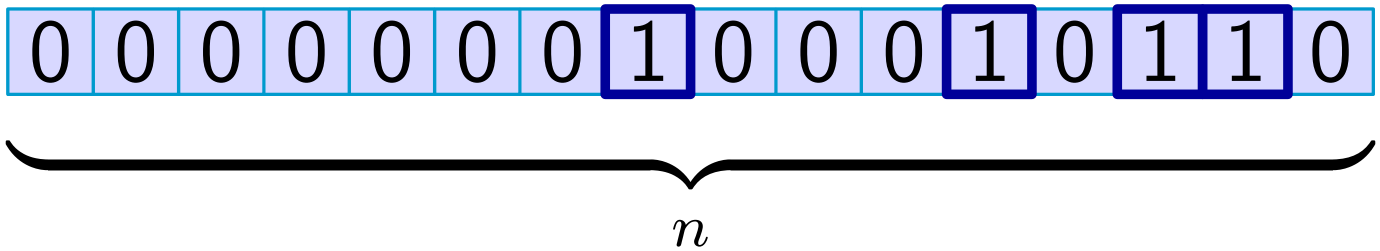
Total number of cell reads



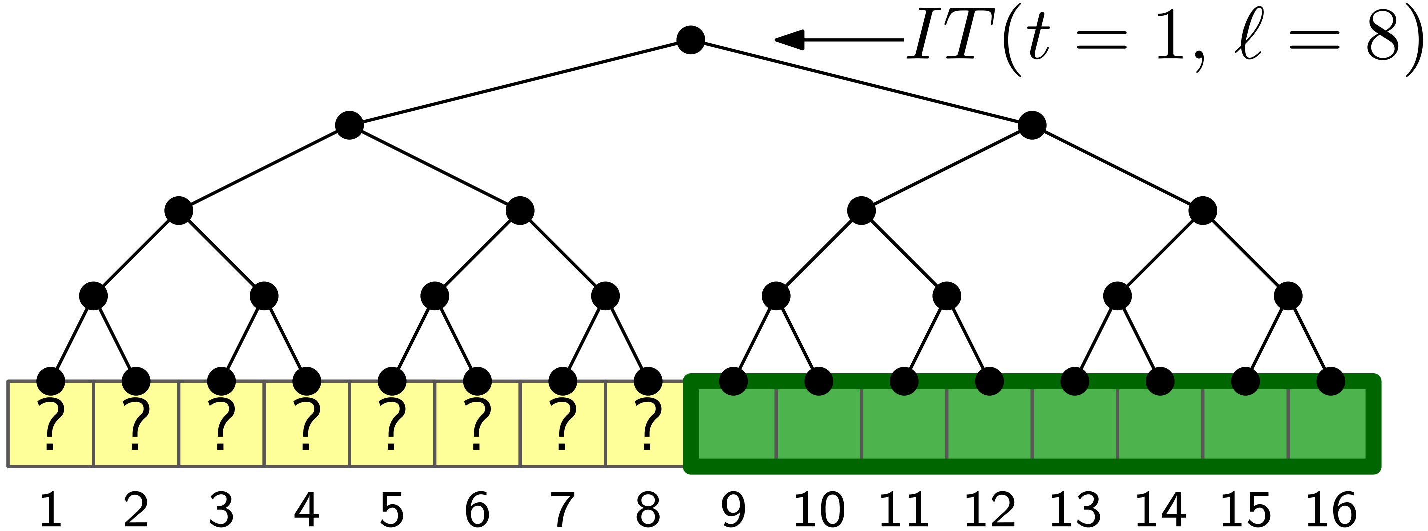
Feed the algorithm with n values chosen uniformly at random from $[q]$.



Total number of cell reads



Feed the algorithm with n values chosen uniformly at random from $[q]$.

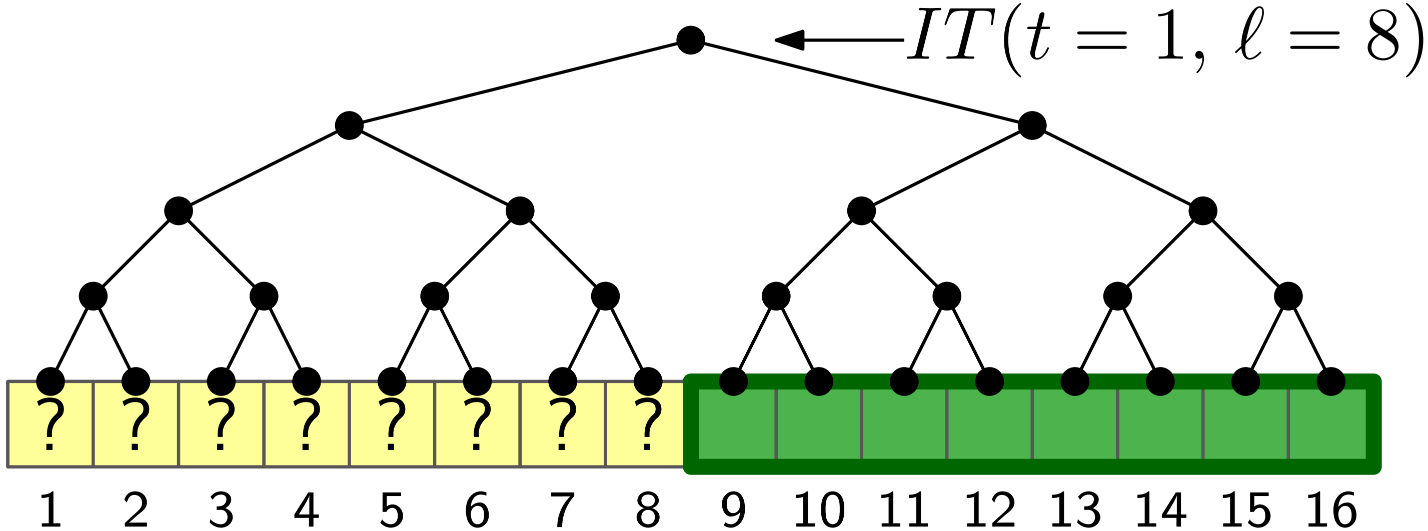


Total number of cell reads

The number of cell reads during the n inputs is at least

$$\sum_{\text{internal node } v} |IT(t_v, \ell_v)|$$

random from $[q]$.



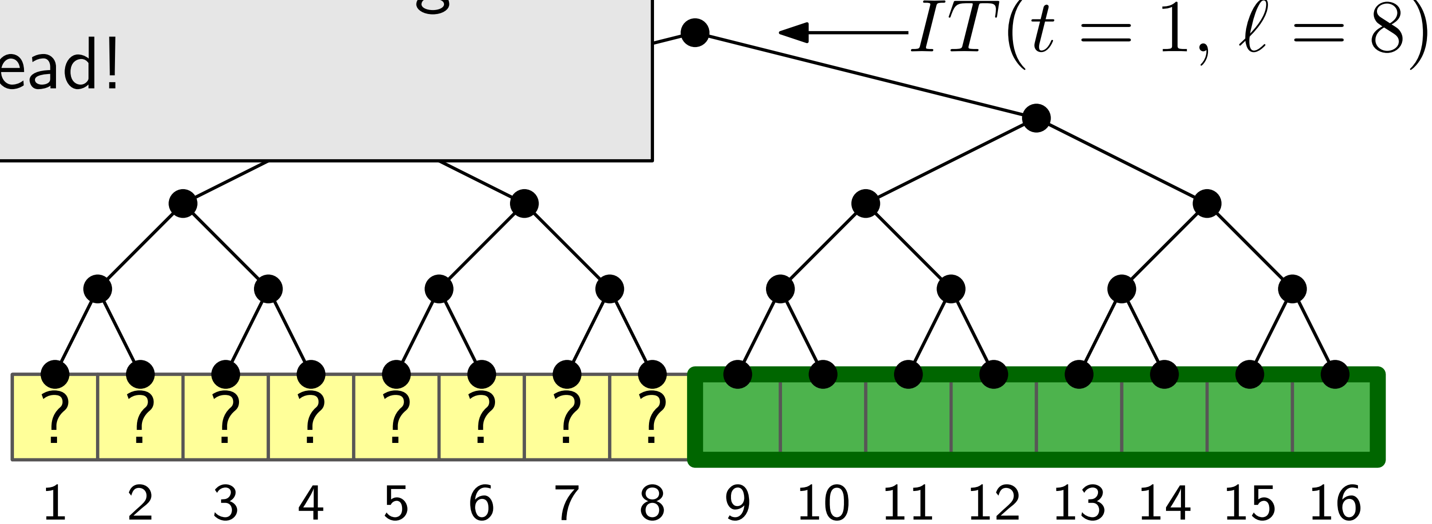
Total number of cell reads

The number of cell reads during the n inputs is at least

$$\sum_{\text{internal node } v} |IT(t_v, \ell_v)|$$

random from $[q]$.

No double counting of a cell read!



Total number of cell reads

The number of cell reads during the n inputs is at least

$$\sum_{\text{internal node } v} |IT(t_v, \ell_v)|$$

The expected number of cell reads is at least

$$\begin{aligned}
\mathbb{E}\left[\sum_{\text{internal node } v} |IT(t_v, \ell_v)|\right] &= \sum_{\text{internal node } v} \mathbb{E}\left[|IT(t_v, \ell_v)|\right] \\
&\geqslant \sum_{\text{internal node } v} \frac{\delta}{4w}\ell_v - \frac{1}{2} \\
&= \Omega\left(\frac{\delta}{w} \cdot n \log n\right)
\end{aligned}$$

Total number of cell reads

The number of cell reads during the n inputs is at least

$$\sum_{\text{internal node } v} |IT(t_v, \ell_v)|$$

The expected number of cell reads is at least

$$\mathbb{E}\left[\sum_{\text{internal node } v} |IT(t_v, \ell_v)|\right] = \sum_{\text{internal node } v} \mathbb{E}\left[|IT(t_v, \ell_v)|\right]$$

$$\geqslant \sum_{\text{internal node } v} \frac{\delta}{4w}\ell_v - \frac{1}{2}$$

$$= \Omega\left(\frac{\delta}{w} \cdot n \log n\right)$$

So...
The amortised time lower bound per output is $\Omega\left(\frac{\delta}{w} \log n\right)$

What happens if the alphabet is binary?

For binary alphabet and sensible word size, we get useless

$$\Omega\left(\frac{\log n}{w}\right) = \Omega(1).$$

What happens if the alphabet is binary?

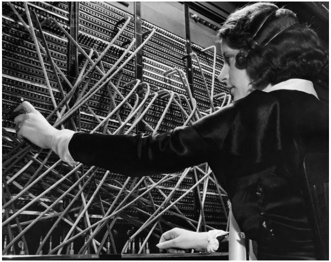
For binary alphabet and sensible word size, we get useless

$$\Omega\left(\frac{\log n}{w}\right) = \Omega(1).$$

But...

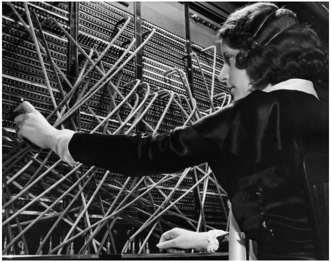
- What if each output is in $\{0, \ldots, n\}$?
- Total entropy of $n/\log n$ outputs *could* therefore be $\Omega(n)$.
- We could then use a new *lop-sided information transfer* technique instead.

Pattern matching with address errors



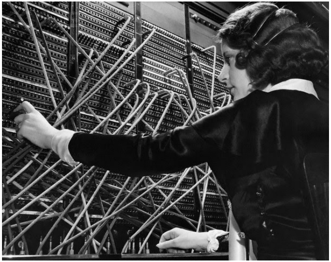
Message sent: *eleven plus two*

Pattern matching with address errors



Message sent: *eleven plus two*
Message received: *twelve plus one*

Pattern matching with address errors



Message sent: *eleven plus two*
Message received: *twelve plus one*

- The L_2-rearrangement distance defined to be $\min_{\pi \in \Pi} \sum_{j=0}^{n-1} (j - \pi(j))^2$ (AABLLPSV:2009)
- Online: $O(\log^2 n)$ time per arriving symbol (CS:2011).

Example

The L_2-rearrangement distance of 11100 and 10110 is $0^2 + 1^2 + 1^2 + 2^2 + 0^2 = 6$.

Pattern matching with address errors

For binary inputs, our new lower bound is:

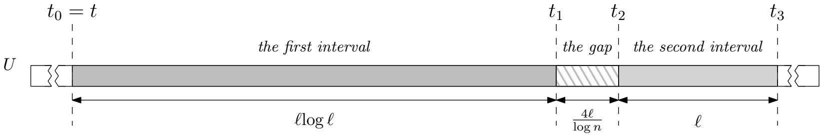
$$\Omega\left(\frac{\lg^2 n}{w \cdot \lg \lg n}\right)$$

To do this we must find an input distribution such that:

- The conditional entropy of the outputs is high.
- It is possible to sum the contributions from many interval lengths without double counting.

Lop-sided information transfer - Mind the gap

To sum contributions, we introduce a gap:



The lengths ℓ are taken from:

$$\left\{ n^{1/4} \cdot (\lg n)^{2i} \;\middle|\; i \in \left\{0, 1, 2, \ldots, \frac{\lg n}{4 \lg \lg n}\right\} \right\}.$$

Lop-sided information transfer - Mind the gap

Upper bound on entropy

$$H(A_{\ell,t} \mid \widetilde{U}_{\ell,t} = \widetilde{u}_{\ell,t}) \;\leq\; 2w + 2w \cdot \mathbb{E}[I_{\ell,t} + G_{\ell,t} \mid \widetilde{U}_{\ell,t} = \widetilde{u}_{\ell,t}].$$

Lop-sided information transfer - Mind the gap

Upper bound on entropy

$$H(A_{\ell,t} \mid \widetilde{U}_{\ell,t} = \widetilde{u}_{\ell,t}) \;\leq\; 2w + 2w \cdot \mathbb{E}[I_{\ell,t} + G_{\ell,t} \mid \widetilde{U}_{\ell,t} = \widetilde{u}_{\ell,t}].$$

Lower bound on entropy

Lemma

For the L_2-rearrangement distance problem there exists a hard input distribution such that

$$H(A_{\ell,t} \mid \widetilde{U}_{\ell,t} = \widetilde{u}_{\ell,t}) \;\geq\; \kappa \cdot \ell \cdot \lg n,$$

for any fixed $\widetilde{u}_{\ell,t}$.

Lop-sided information transfer - Mind the gap

We remove the conditioning by taking expectation over $\widetilde{U}_{\ell,t}$ under random U giving:

$$\mathbb{E}[I_{\ell,t}] \ \geq \ \frac{\kappa \cdot \ell \cdot \lg n}{2w} - 1 - \mathbb{E}[G_{\ell,t}].$$

By carefully choosing T_ℓ we get:

$$\mathbb{E}\left[\sum_{\ell \in L} \sum_{t \in T_\ell} I_{\ell,t}\right] \in \Omega\left(\frac{n \cdot \lg^2 n}{w \cdot \lg \lg n}\right).$$

The hard distribution for L_2-rearrangement

We let the incoming streaming be randomly sampled from:

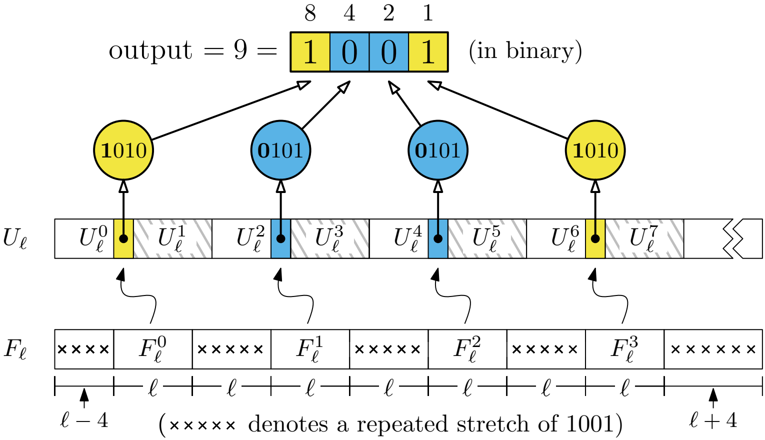
$$\{0101, 1010\}^*$$

The hard distribution for L_2-rearrangement

We let the incoming streaming be randomly sampled from:

$$\{0101, 1010\}^*$$

Different bits of the output give different bits of the stream.



A lower bound for convolution?

For convolution we hit a tricky mathematical hurdle.

- What is the entropy of $n/\log n$ consecutive overlapping inner products?

A lower bound for convolution?

For convolution we hit a tricky mathematical hurdle.

- What is the entropy of $n/\log n$ consecutive overlapping inner products?

$$111011 \longleftrightarrow \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Conjecture

Let $x \in \{0,1\}^\ell$ be sampled at random. There exist $\ell/\log \ell$ by ℓ Toeplitz matrices M such such that $H(Mx) \in \Omega(\ell)$.

A lower bound for convolution?

For convolution we hit a tricky mathematical hurdle.

- What is the entropy of $n/\log n$ consecutive overlapping inner products?

$$111011 \longleftrightarrow \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Conjecture

Let $x \in \{0,1\}^\ell$ be sampled at random. There exist $\ell/\log \ell$ by ℓ Toeplitz matrices M such such that $H(Mx) \in \Omega(\ell)$.

A lower bound for convolution?

For convolution we hit a tricky mathematical hurdle.

- What is the entropy of $n/\log n$ consecutive overlapping inner products?

$$111011 \longleftrightarrow \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Conjecture

Let $x \in \{0,1\}^\ell$ be sampled at random. There exist $\ell/\log \ell$ by ℓ Toeplitz matrices M such such that $H(Mx) \in \Omega(\ell)$.

Thank you!