Lower Bounds for Online Integer Multiplication and Convolution in the Cell-Probe Model

ICALP, 4–8 July 2011

Markus Jalsenius

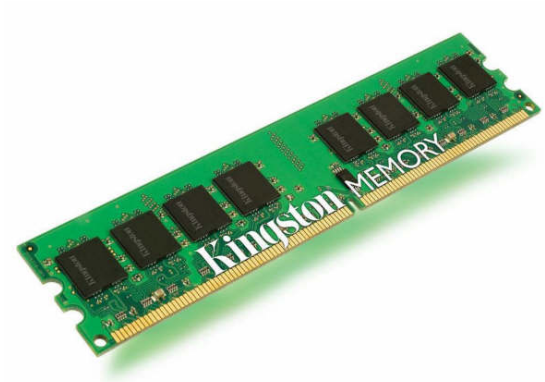
Joint work with

Raphaël Clifford

University of BRISTOL

Cell-probe model

The CPU does not remember anything in between operations.



Cell probes

Read

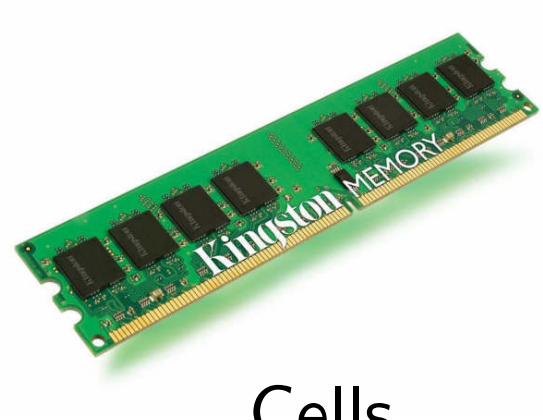
Write

Cells

0
1
2
3
4
5
. . .

w bits

Cell-probe model



The CPU does not remember anything in between operations.

Cell probes

Read

Write

Cells

0
1
2
3
4
5
...

w bits

The CPU has unlimited computational power.

Multiplication (problem 1)

$\times$

Digits from the set $[q]$ (base q)

Notation: $[q] = \{0, \ldots, q-1\}$

Multiplication (problem 1)

$$\begin{array}{rr} & 2 \\ \times & 3 \\ \hline \end{array}$$

Digits from the set $[q]$ (base q)

Notation: $[q] = \{0, \ldots, q-1\}$

Multiplication (problem 1)

$$\begin{array}{r} 2 \\ \times \quad 3 \\ \hline 6 \end{array}$$

Digits from the set $[q]$ (base q)

Notation: $[q] = \{0, \ldots, q-1\}$

Multiplication (problem 1)

$$\begin{array}{rr} & 4\ 2 \\ \times & 8\ 3 \\ \hline & 6 \end{array}$$

Digits from the set $[q]$ (base q)

Notation: $[q] = \{0, \ldots, q-1\}$

Multiplication (problem 1)

$$\begin{array}{rr} & 4\ 2 \\ \times & 8\ 3 \\ \hline & 8\ 6 \end{array}$$

Digits from the set $[q]$ (base q)

Notation: $[q] = \{0, \ldots, q-1\}$

Multiplication (problem 1)

$$
\begin{array}{rr}
 & 0\ 4\ 2 \\
\times & 9\ 8\ 3 \\
\hline
 & 2\ 8\ 6
\end{array}
$$

Digits from the set $[q]$ (base q)

Notation: $[q] = \{0, \ldots, q-1\}$

Multiplication (problem 1)

$$
\begin{array}{rr}
 & 1\ 0\ 4\ 2 \\
\times & 2\ 9\ 8\ 3 \\
\hline
 & 8\ 2\ 8\ 6
\end{array}
$$

Digits from the set $[q]$ (base q)

Notation: $[q] = \{0, \ldots, q-1\}$

Multiplication (problem 1)

$$
\begin{array}{rr}
 & 4\ 1\ 0\ 4\ 2 \\
\times & 4\ 2\ 9\ 8\ 3 \\
\hline
 & 0\ 8\ 2\ 8\ 6
\end{array}
$$

Digits from the set $[q]$ (base q)

Notation: $[q] = \{0, \ldots, q-1\}$

Multiplication (problem 1)

$$
\begin{array}{rr}
 & 6\ 4\ 1\ 0\ 4\ 2 \\
\times & 0\ 4\ 2\ 9\ 8\ 3 \\
\hline
 & 9\ 0\ 8\ 2\ 8\ 6
\end{array}
$$

Digits from the set $[q]$ (base q)

Notation: $[q] = \{0, \ldots, q-1\}$

Multiplication (problem 1)

$$
\begin{array}{rr}
 & 2\ 6\ 4\ 1\ 0\ 4\ 2 \\
\times & 1\ 0\ 4\ 2\ 9\ 8\ 3 \\
\hline
 & 1\ 9\ 0\ 8\ 2\ 8\ 6
\end{array}
$$

Digits from the set $[q]$ (base q)

Notation: $[q] = \{0, \ldots, q-1\}$

Multiplication (problem 1)

$$
\begin{array}{rc}
 & \overbrace{}^{n} \\
 & 5\;2\;6\;4\;1\;0\;4\;2 \\
\times & 7\;1\;0\;4\;2\;9\;8\;3 \\
\hline
 & 5\;1\;9\;0\;8\;2\;8\;6
\end{array}
$$

Digits from the set $[q]$ (base q)

Notation: $[q] = \{0, \ldots, q-1\}$

Multiplication (problem 1)

$$
\begin{array}{rcccccccccccccccc}
 & & & & & & & & & \overbrace{}^{n} & & & & & & \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 2 & 6 & 4 & 1 & 0 & 4 & 2 \\
\times & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & 1 & 0 & 4 & 2 & 9 & 8 & 3 \\
\hline
 & 3 & 7 & 3 & 9 & 7 & 7 & 6 & 6 & 5 & 1 & 9 & 0 & 8 & 2 & 8 & 6
\end{array}
$$

Digits from the set $[q]$ (base q)

Notation: $[q] = \{0, \ldots, q-1\}$

Multiplication (problem 1)

$$
\begin{array}{rc}
 & \overbrace{}^{n} \\
 & 5\;2\;6\;4\;1\;0\;4\;2 \\
\times & 7\;1\;0\;4\;2\;9\;8\;3 \\
\hline
 & 5\;1\;9\;0\;8\;2\;8\;6
\end{array}
$$

Digits from the set $[q]$ (base q)

Notation: $[q] = \{0, \ldots, q-1\}$

Multiplication (problem 1)

$$
\begin{array}{rcccccccc}
 & \overbrace{}^{n} & & & & & & & \\
 & 5 & 2 & 6 & 4 & 1 & 0 & 4 & 2 \\
\times & 7 & 1 & 0 & 4 & 2 & 9 & 8 & 3 \\
\hline
 & 5 & 1 & 9 & 0 & 8 & 2 & 8 & 6
\end{array}
$$

Digits from the set $[q]$ (base q)

Notation: $[q] = \{0, \ldots, q-1\}$

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

$\delta = \log q$

Cell-probe model with word size w bits

Multiplication (problem 1)

M. S. Paterson, M. J. Fischer and A. R. Meyer
An Improved Overlap Argument for On-Line Multiplication
SIAM-AMS Proceedings, 1974

For binary numbers on

- Multitape Turing machine: $\Omega(n \log n)$
- Some "random access machines": $\Omega\left(\frac{n \log n}{\log \log n}\right)$

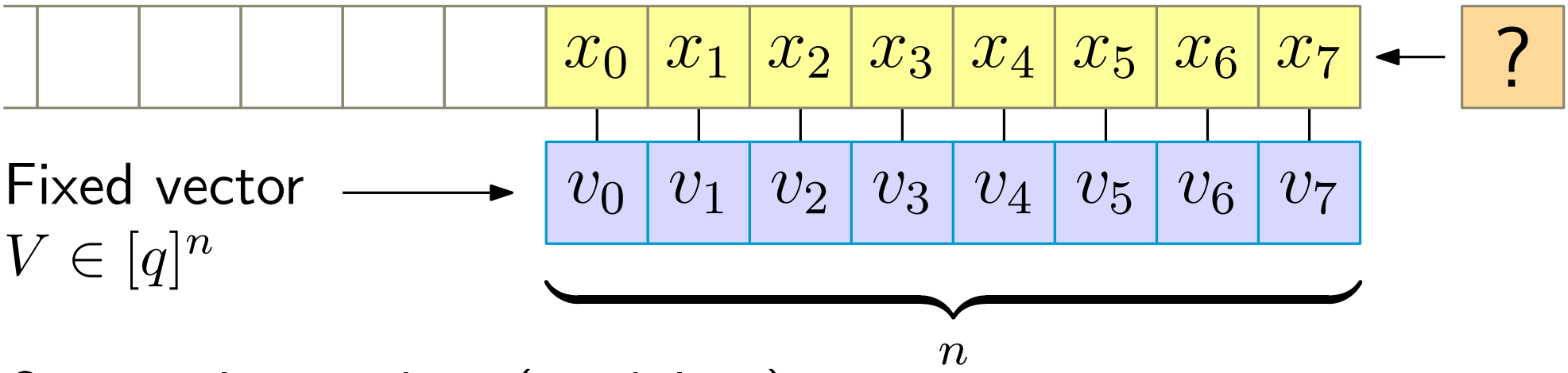
Time lower bound: $\Omega\left(\frac{\delta}{w} \cdot n \log n\right)$

$\delta = \log q$

Cell-probe model with word size w bits

Convolution (problem 2)

Stream of numbers from $[q]$

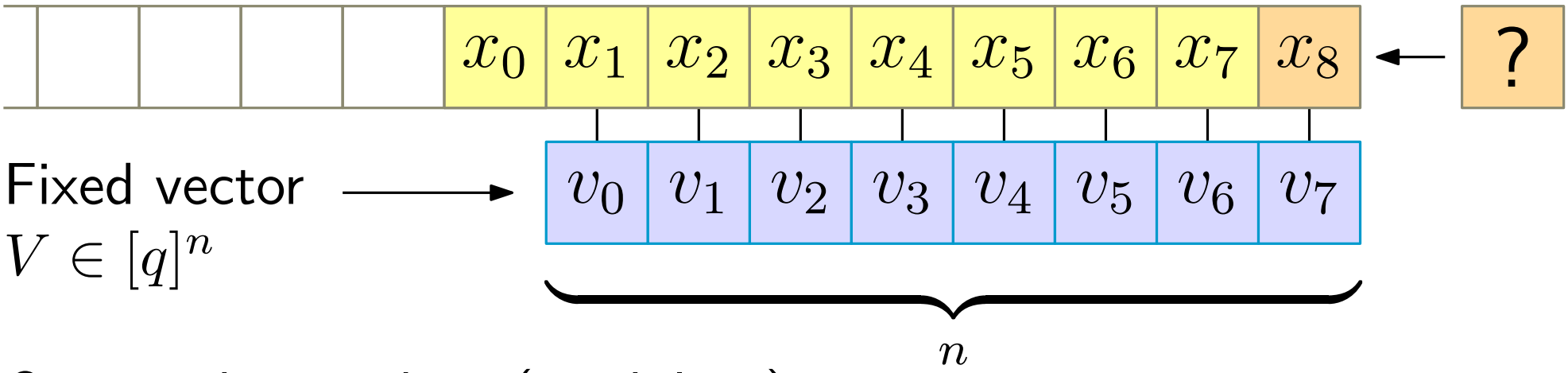


Output dot product (modulo q):

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

Convolution (problem 2)

Stream of numbers from $[q]$

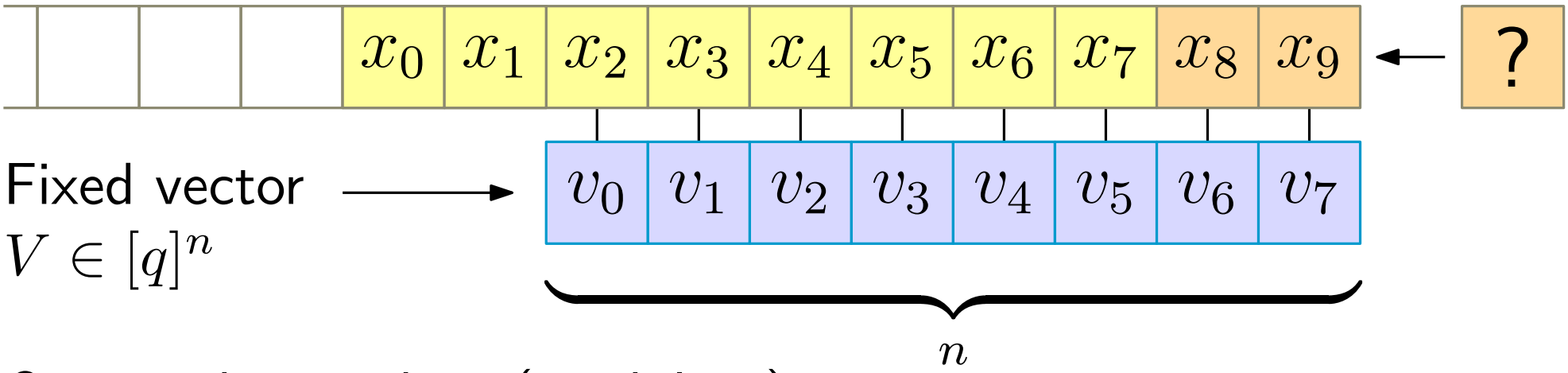


Output dot product (modulo q):

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

Convolution (problem 2)

Stream of numbers from $[q]$

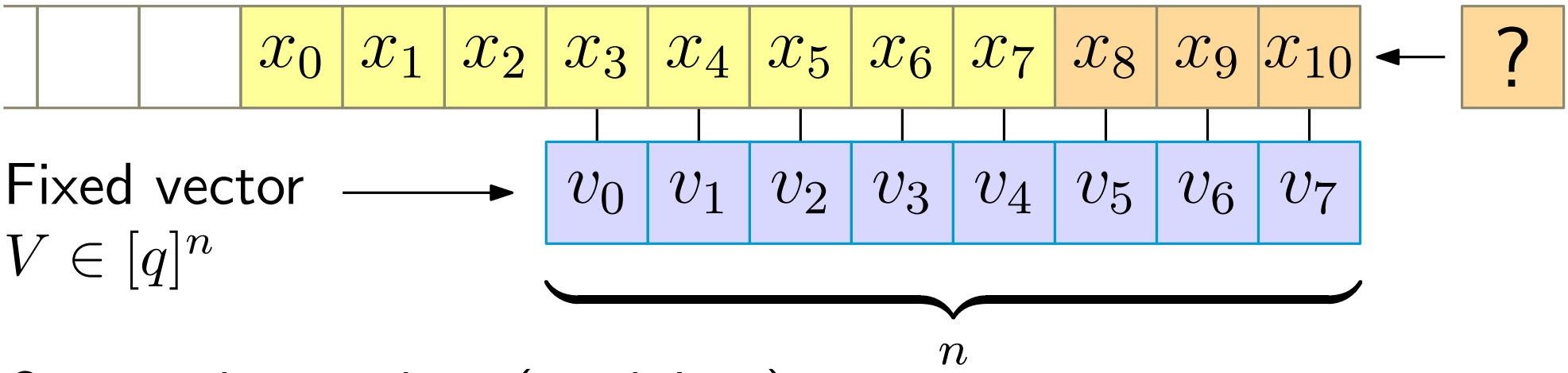


Output dot product (modulo q):

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

Convolution (problem 2)

Stream of numbers from $[q]$

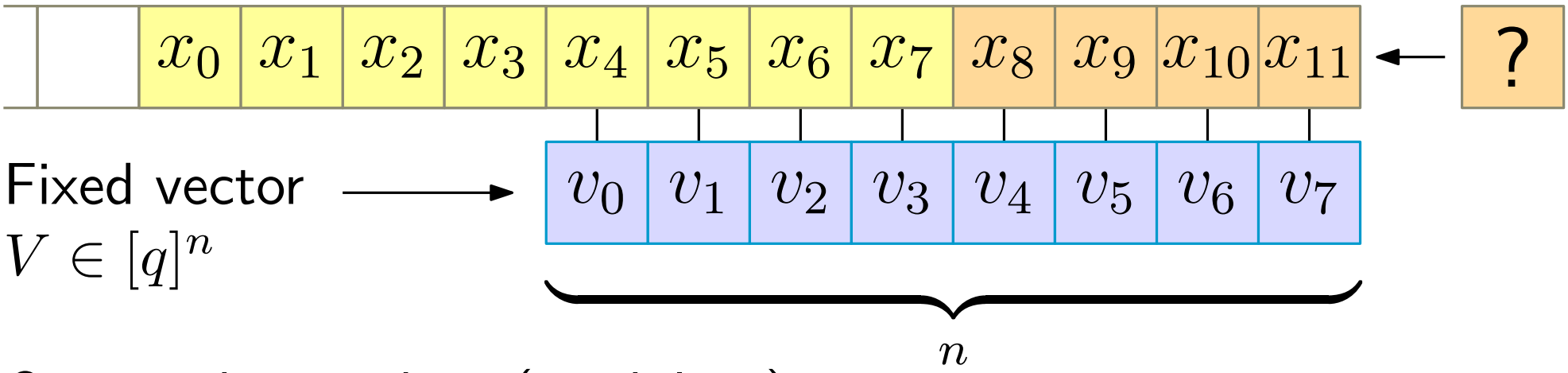


Output dot product (modulo q):

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

Convolution (problem 2)

Stream of numbers from $[q]$

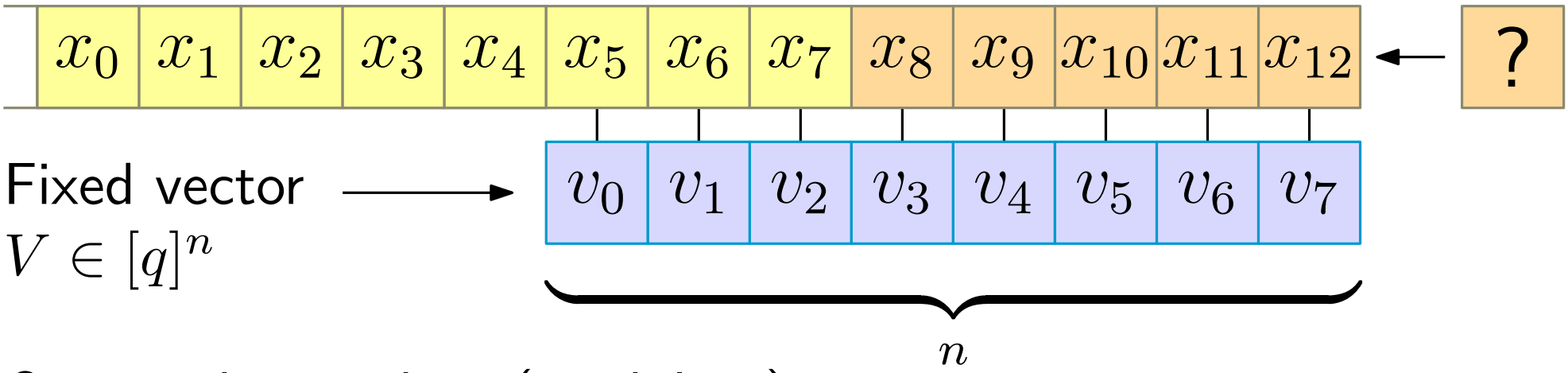


Output dot product (modulo q):

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

Convolution (problem 2)

Stream of numbers from $[q]$

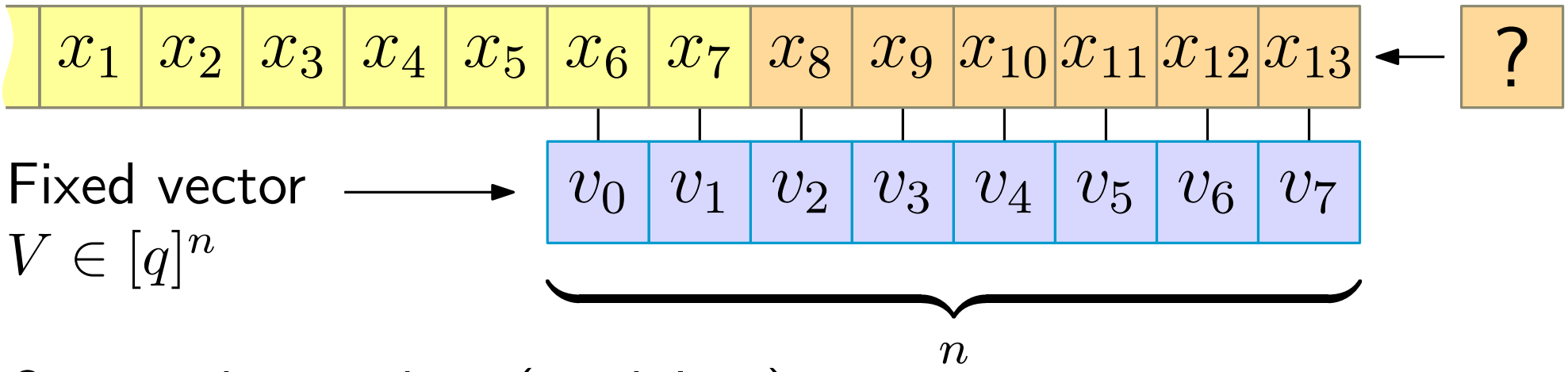


Output dot product (modulo q):

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

Convolution (problem 2)

Stream of numbers from $[q]$

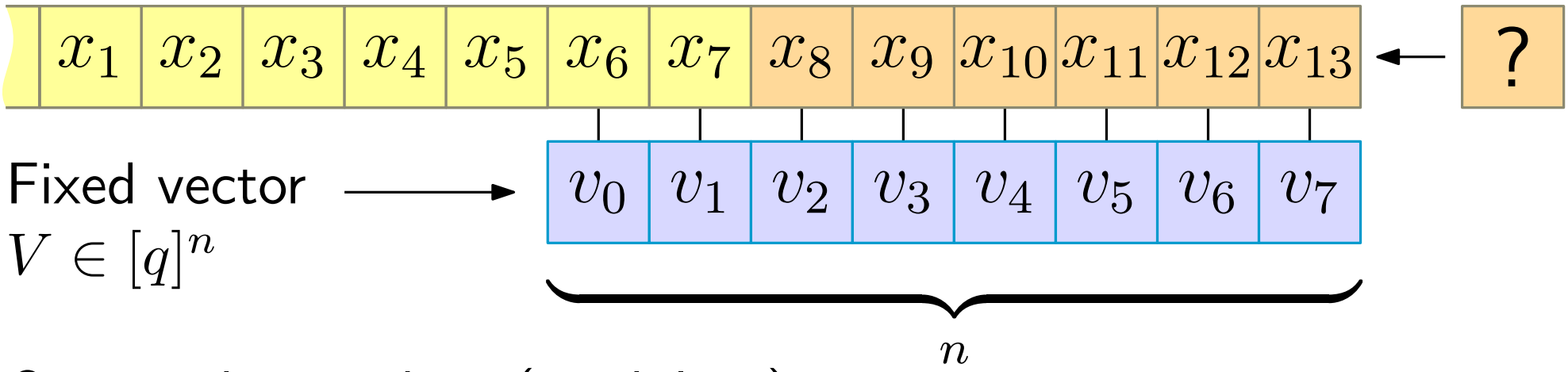


Output dot product (modulo q):

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

Convolution (problem 2)

Stream of numbers from $[q]$



Output dot product (modulo q):

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

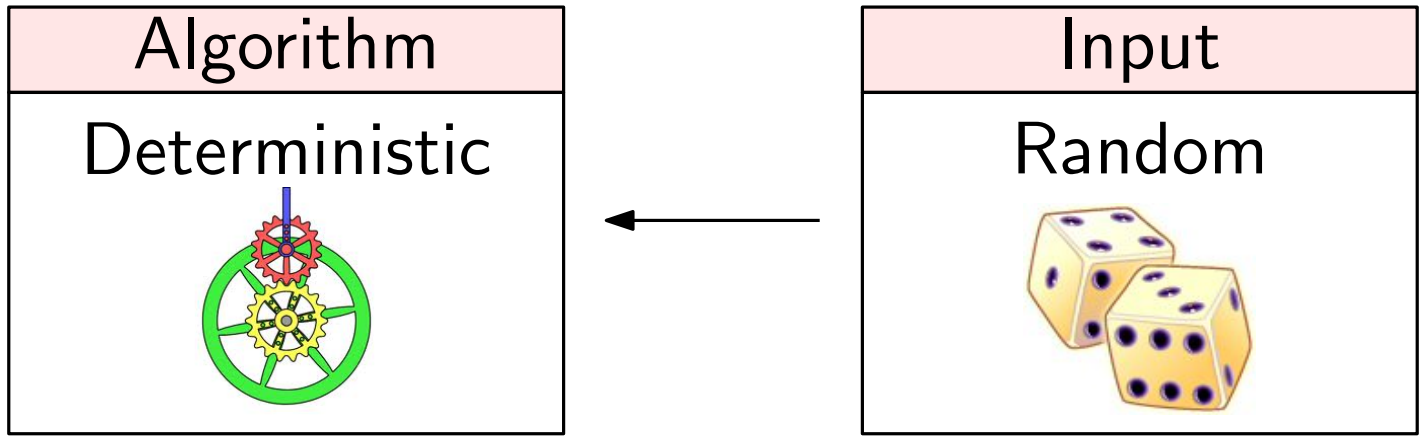
Amortised time lower bound per output: $\Omega\left(\frac{\delta}{w} \log n\right)$

$\delta = \log q$

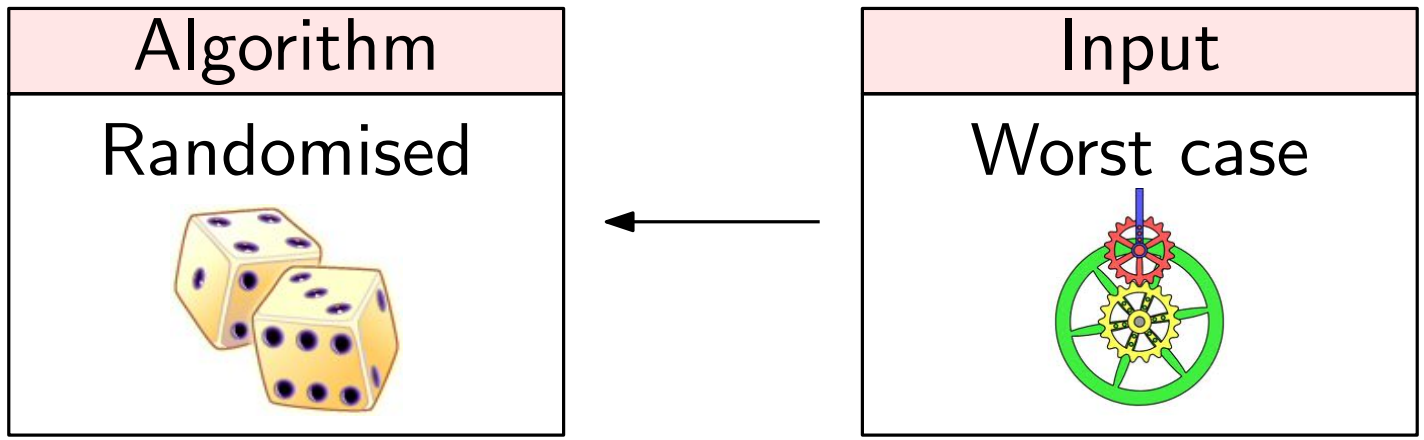
Cell-probe model with word size w bits

Yao's minimax principle

A lower bound on the expected running time for

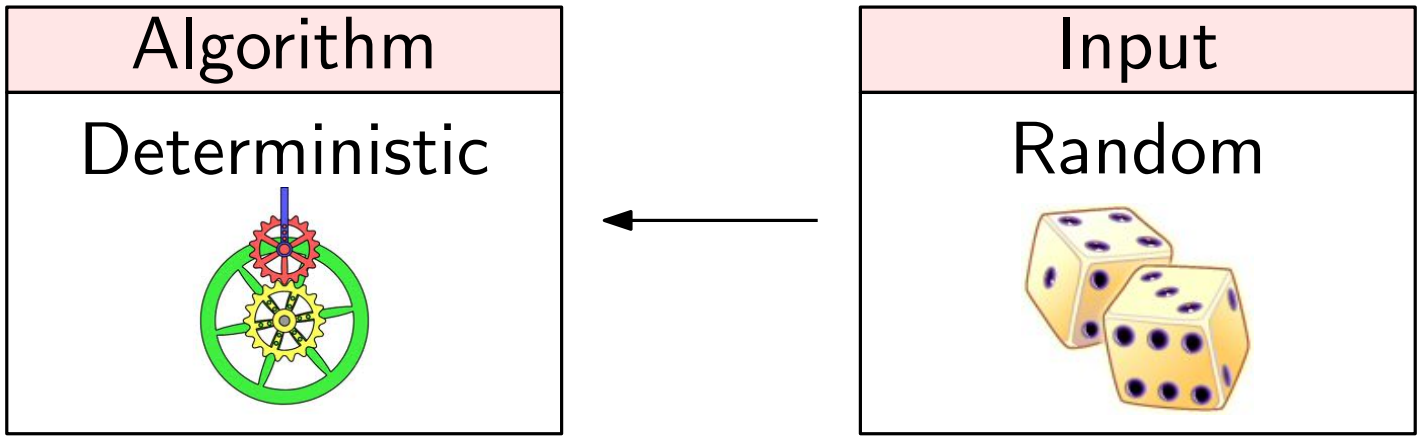


implies that the same lower bound holds for

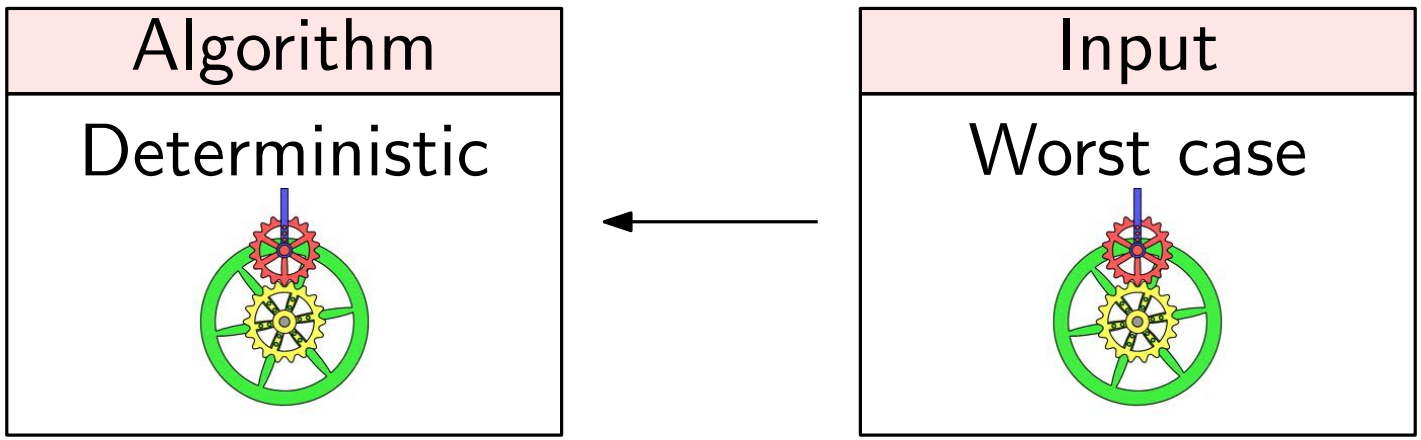


Yao's minimax principle

A lower bound on the expected running time for



implies that the same lower bound holds for

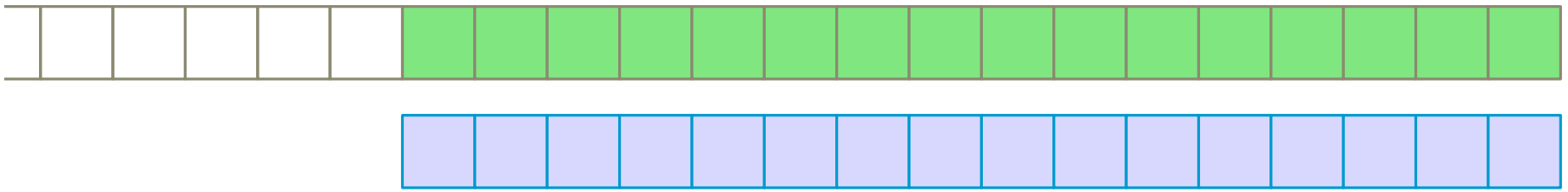


Information transfer

Will use information theoretic arguments from

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

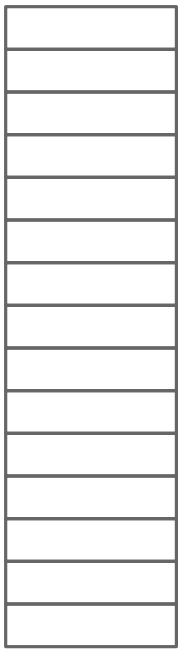
Information transfer



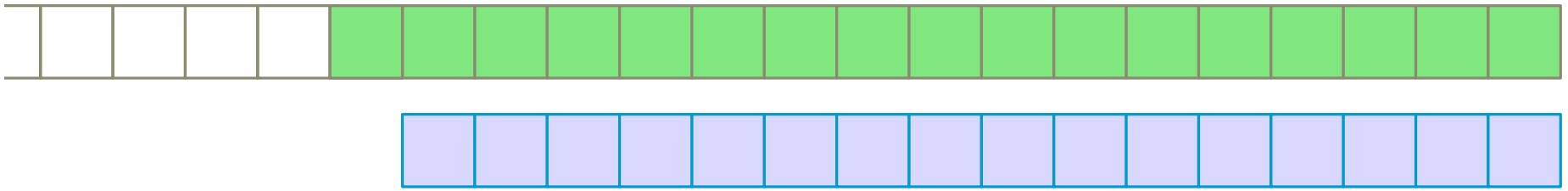
Fixed value

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

Memory cells



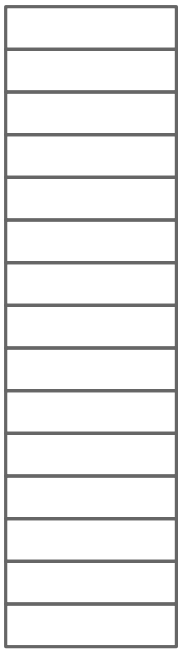
Information transfer



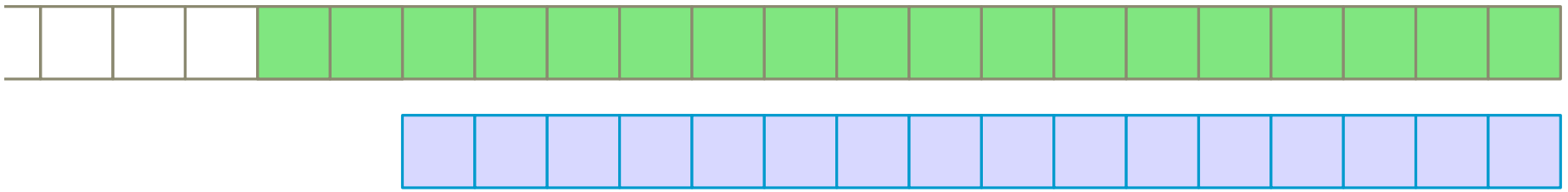
Fixed value

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

Memory cells



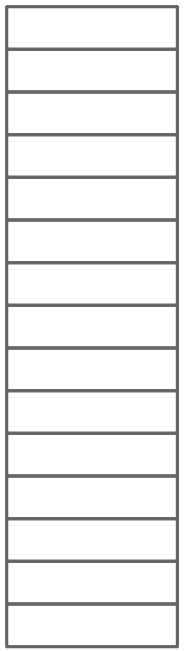
Information transfer



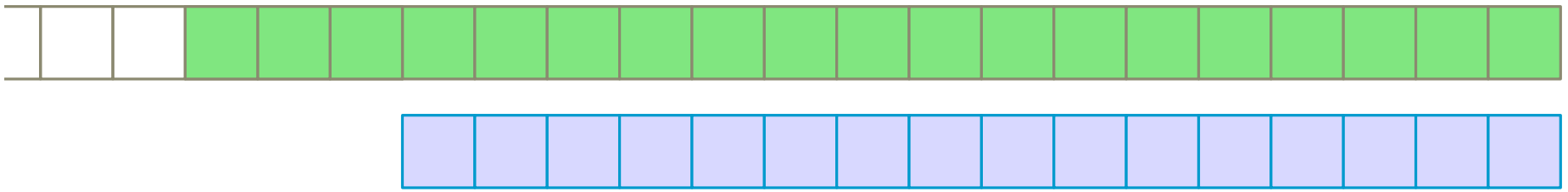
Fixed value

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

Memory cells



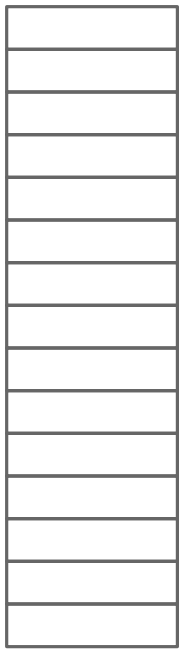
Information transfer



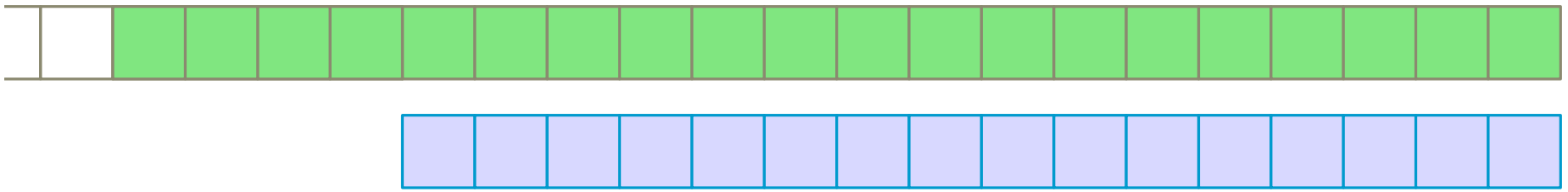
Fixed value

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

Memory cells



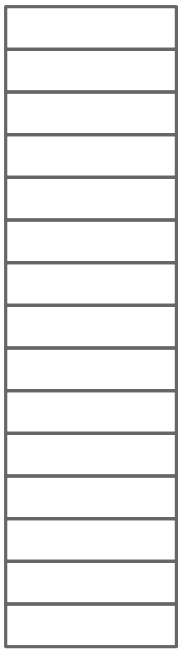
Information transfer



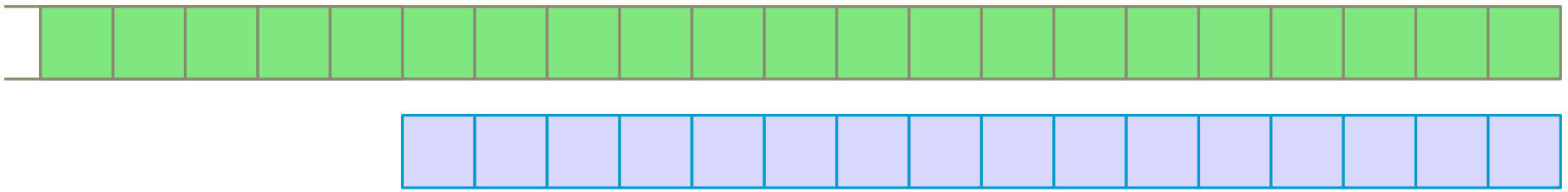
Fixed value

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

Memory cells



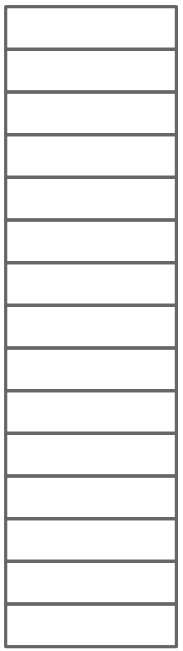
Information transfer



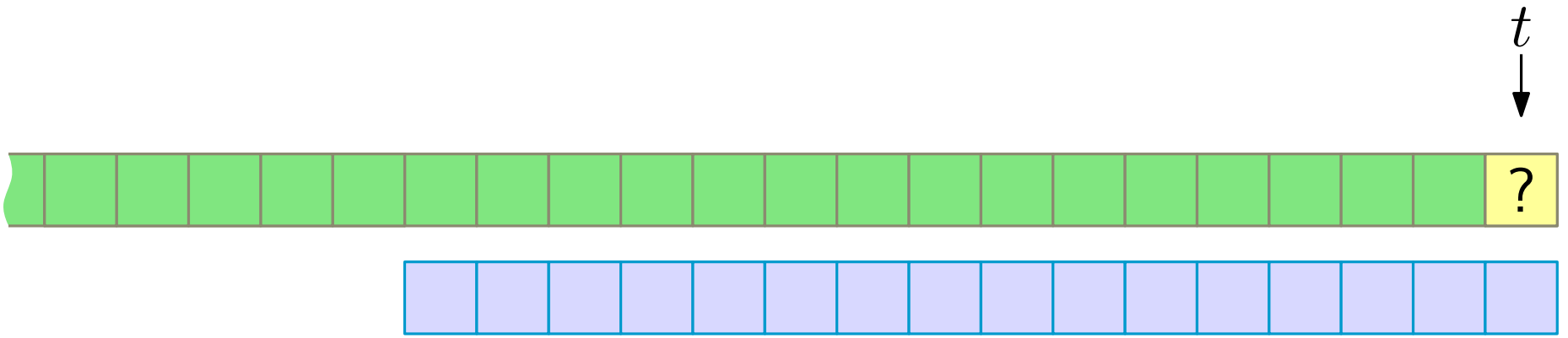
Fixed value

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

Memory cells



Information transfer

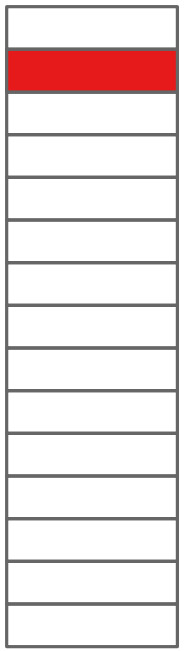


Fixed value

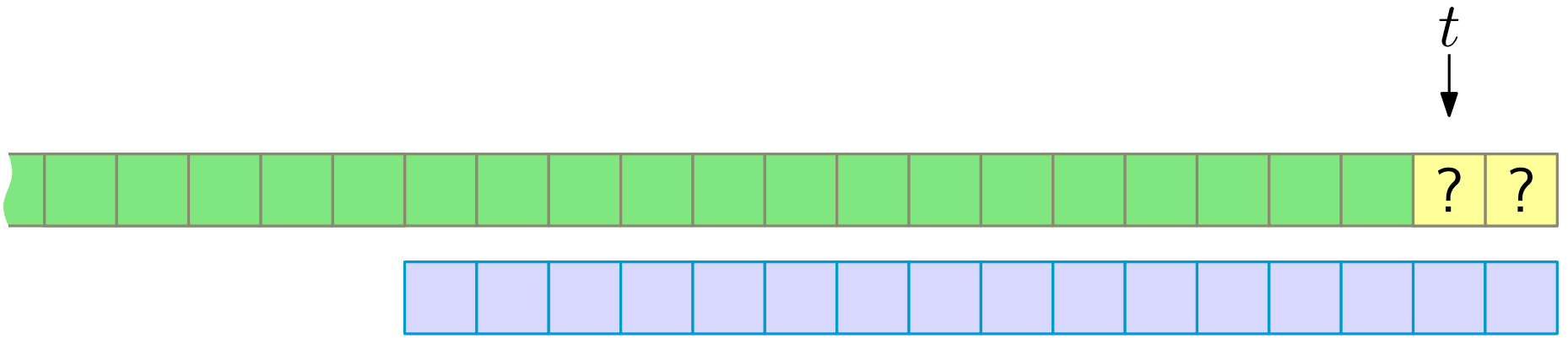
? Unknown value chosen uniformly at random from $[q]$

Cell written during the ? -inputs

Memory cells



Information transfer

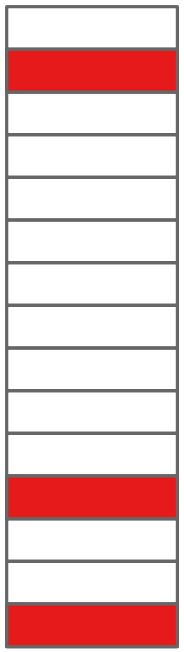


Fixed value

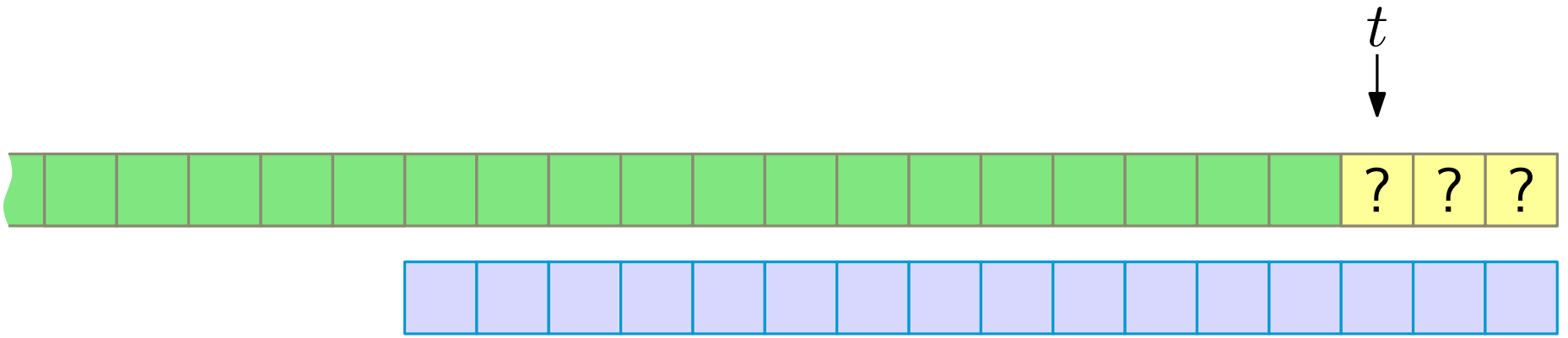
? Unknown value chosen uniformly at random from $[q]$

Cell written during the ? -inputs

Memory cells



Information transfer

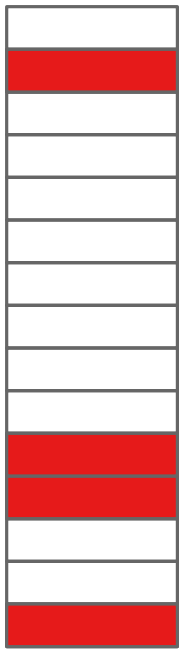


Fixed value

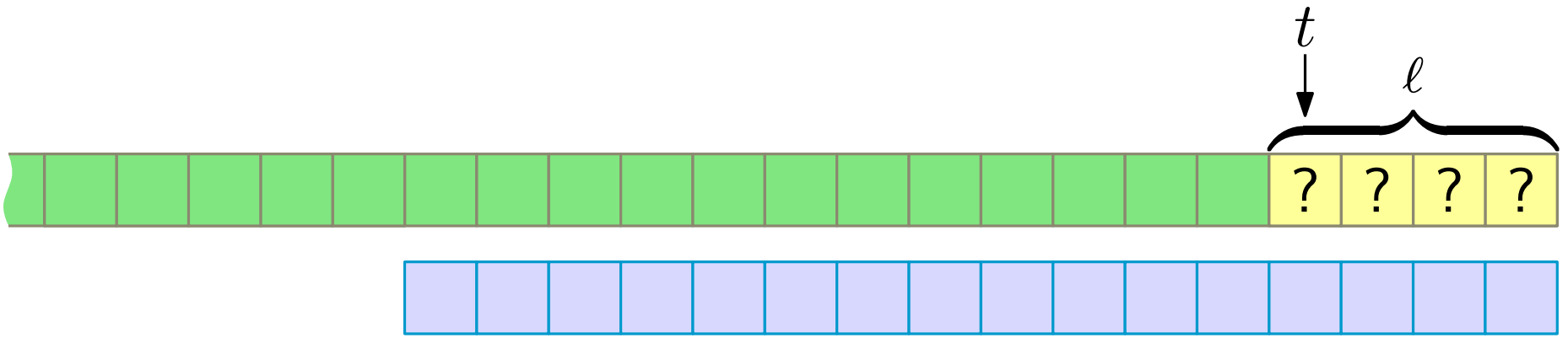
? Unknown value chosen uniformly at random from $[q]$

Cell written during the ? -inputs

Memory cells



Information transfer

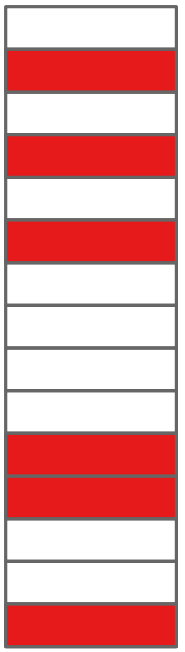


Fixed value

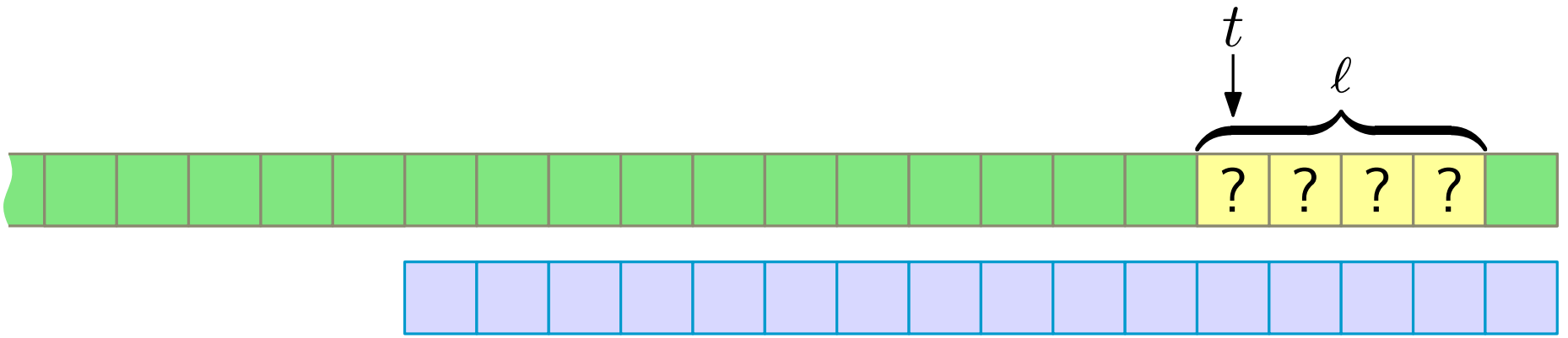
? Unknown value chosen uniformly at random from $[q]$

Cell written during the ? -inputs

Memory cells



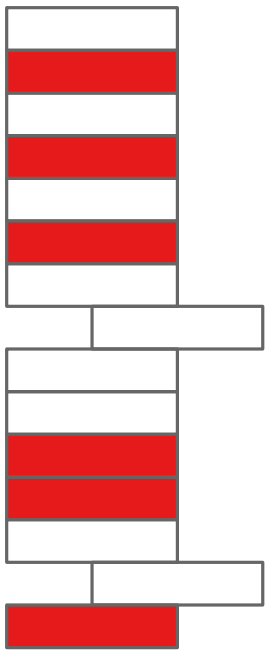
Information transfer



Fixed value

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

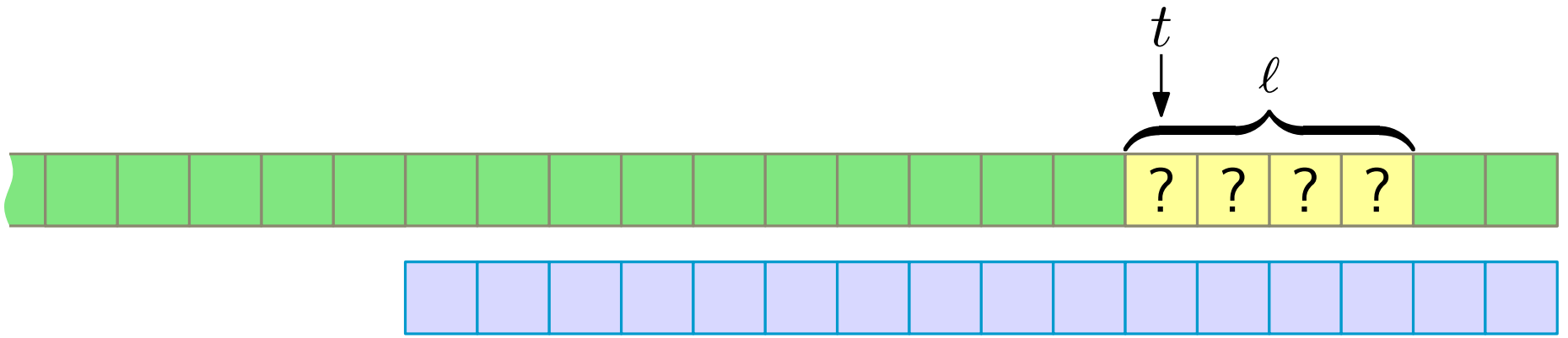
Memory cells



Cell written during the ? -inputs

Cells read during the next ℓ inputs

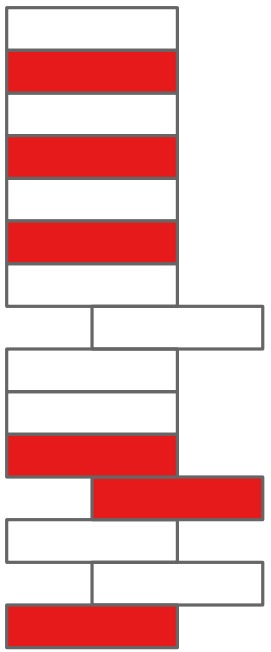
Information transfer



Fixed value

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

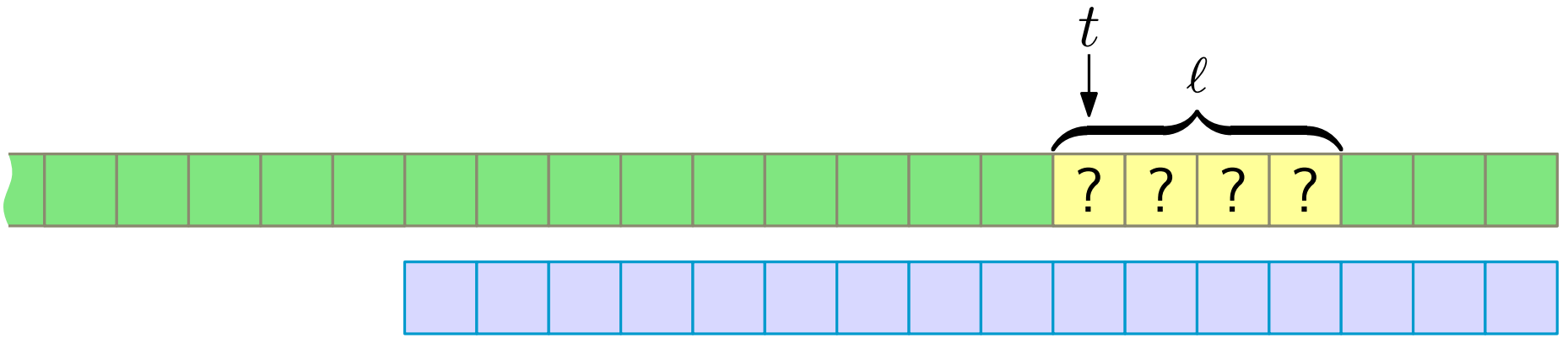
Memory cells



Cell written during the ? -inputs

Cells read during the next ℓ inputs

Information transfer

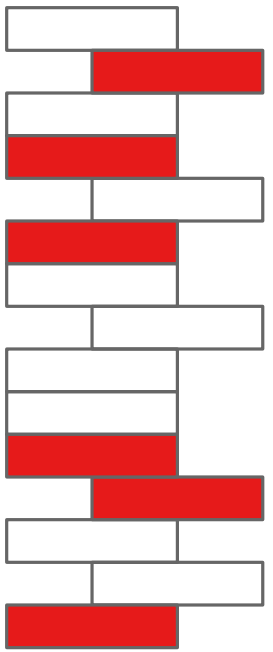


Fixed value

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

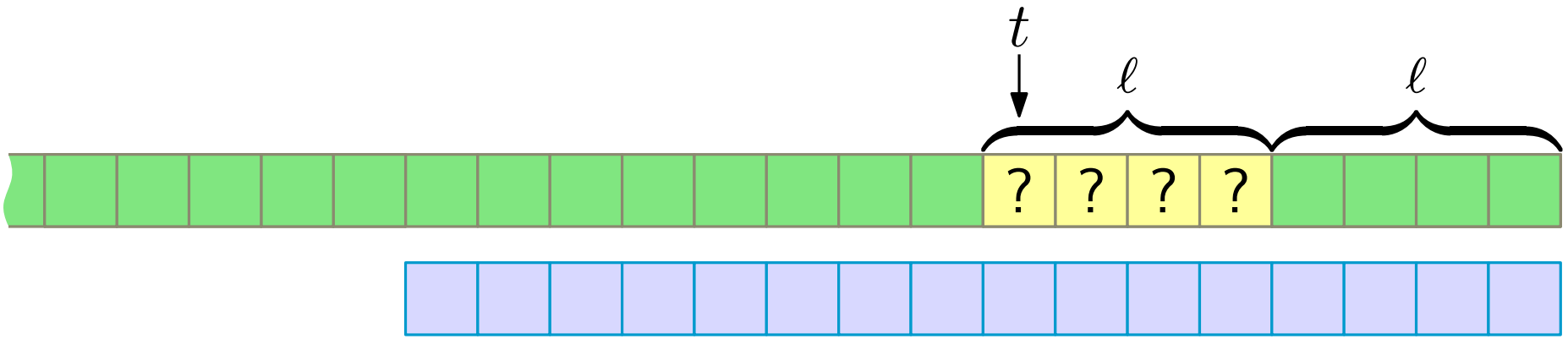
Cell written during the ? -inputs

Memory cells



Cells read during the next ℓ inputs

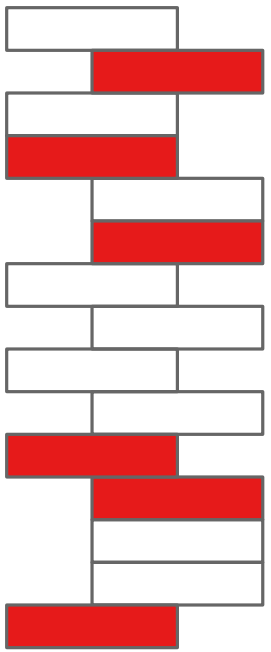
Information transfer



Fixed value

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

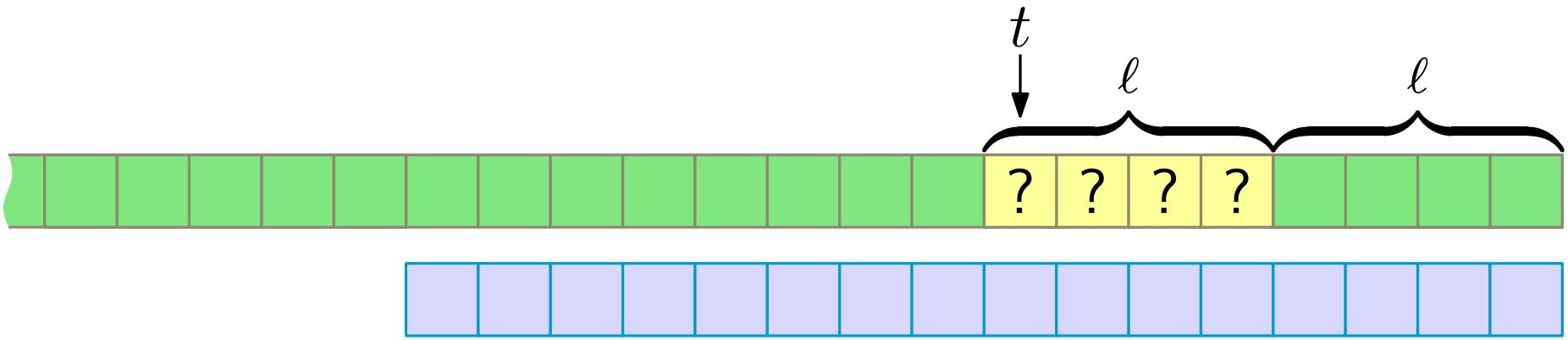
Memory cells



Cell written during the ? -inputs

Cells read during the next ℓ inputs

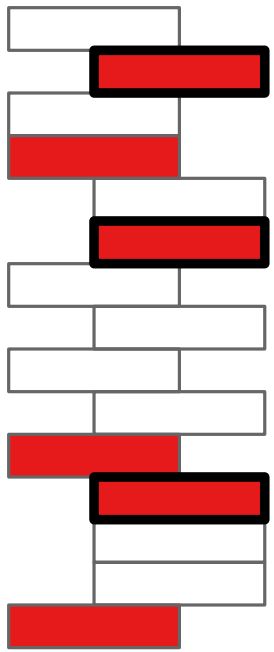
Information transfer



Fixed value

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

Memory cells



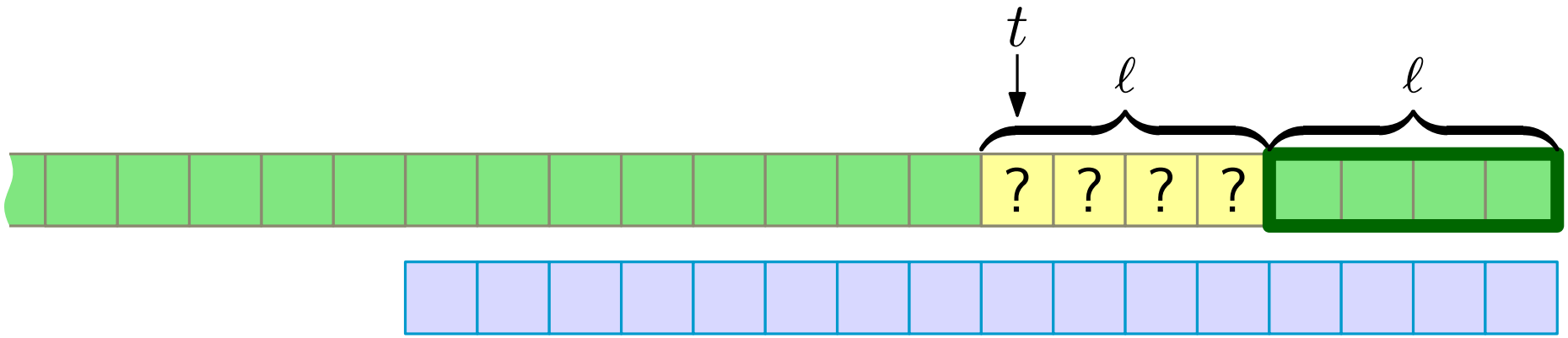
Information transfer $IT(t, \ell)$

Not including cells that were overwritten before being read

Cell written during the ? -inputs

Cells read during the next ℓ inputs

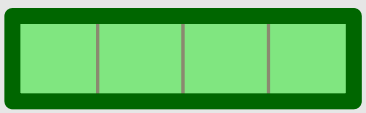
Information transfer



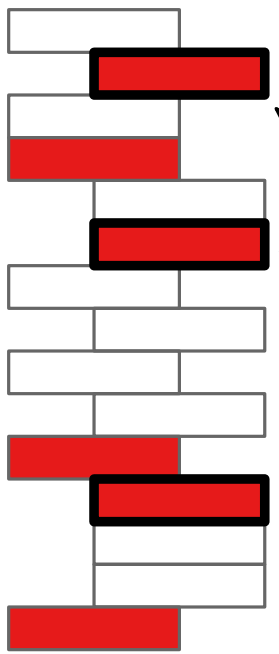
Fixed value

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

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



Memory cells

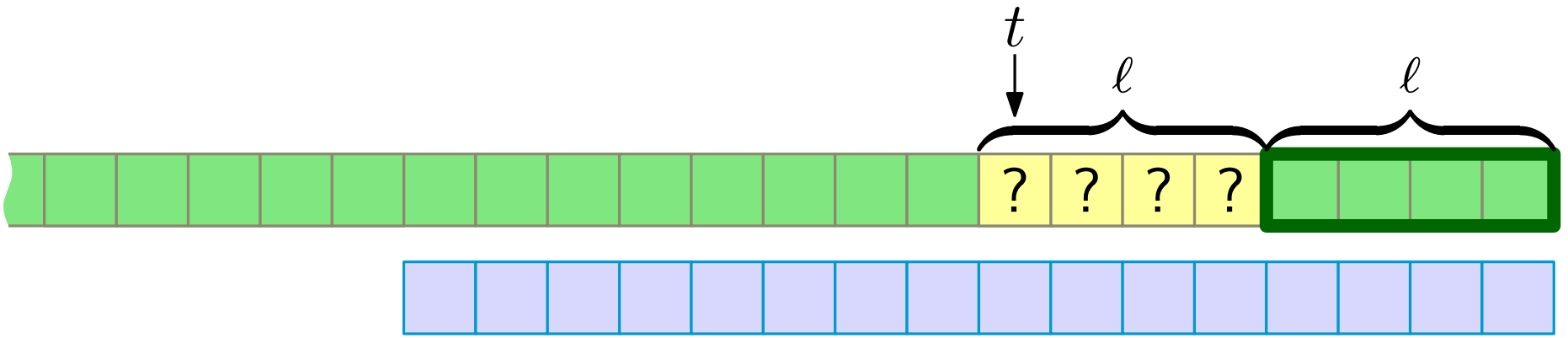


Information transfer $IT(t, \ell)$

Not including cells that were overwritten before being read

inputs

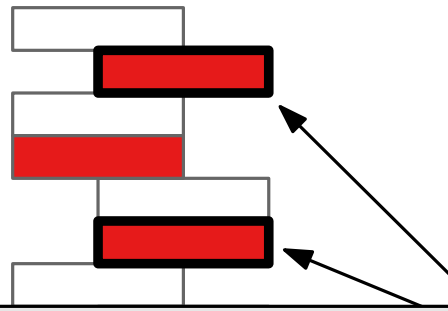
Information transfer



Fixed value

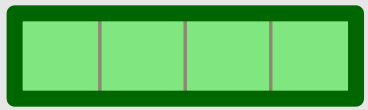
? Unknown value chosen uniformly at random from $[q]$

Memory cells



Information transfer

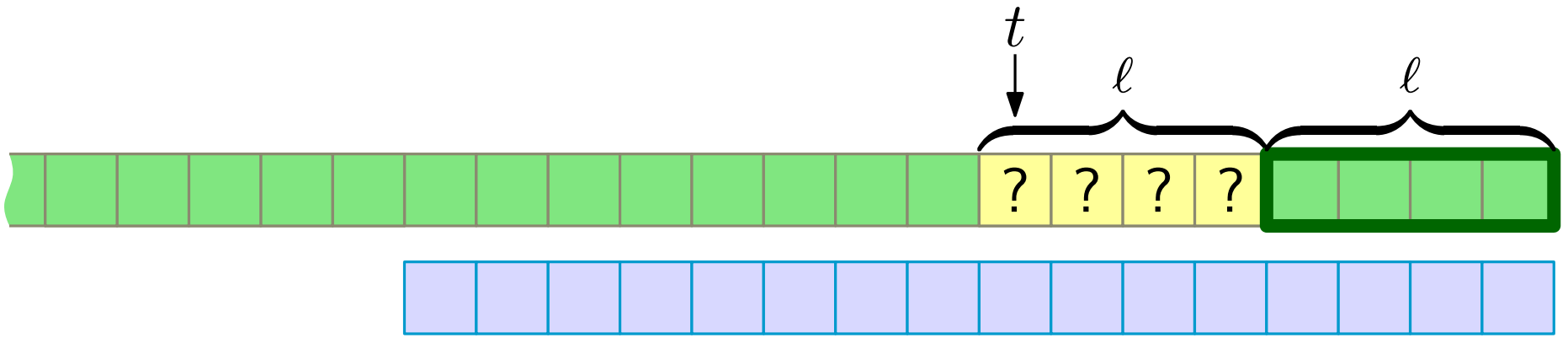
The conditional entropy

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

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

w bits per cell

Information transfer



Fixed value

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

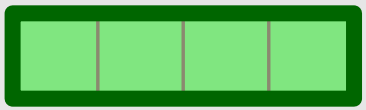
$|IT(t,\ell)|$

Cell	Address	Contents
	00124	76112
	34123	88819
	92540	01882

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

w bits w bits

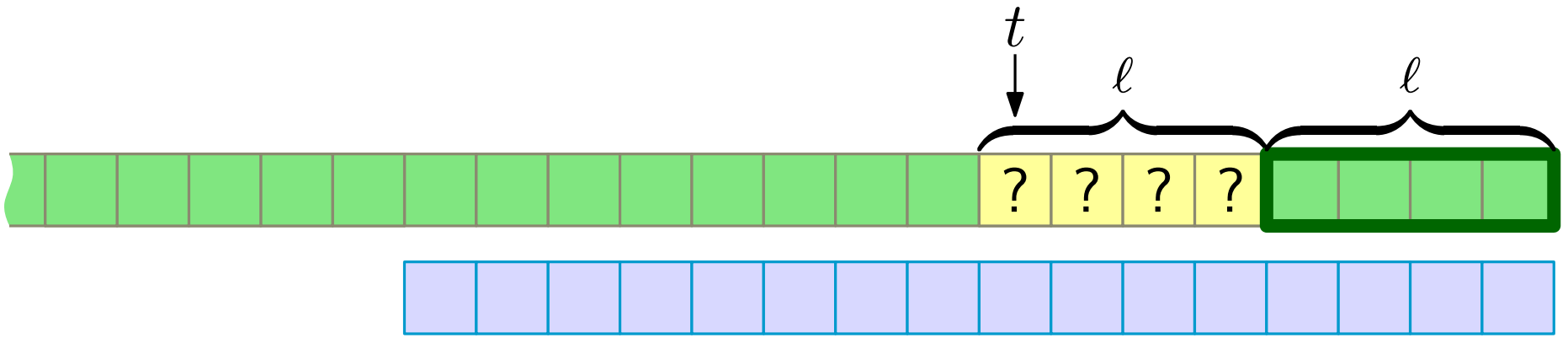
The conditional entropy

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

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

w bits per cell

Information transfer



Fixed value

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

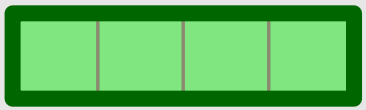
$|IT(t,\ell)|$

Cell	Address	Contents
	00124	76112
	00000	00000
	92540	01882

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

w bits w bits

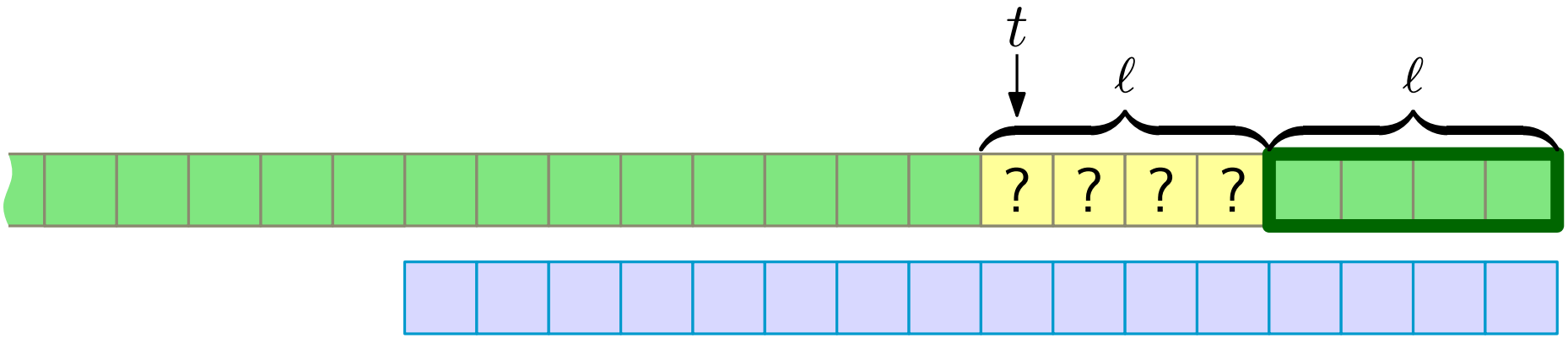
The conditional entropy

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

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

w bits per cell

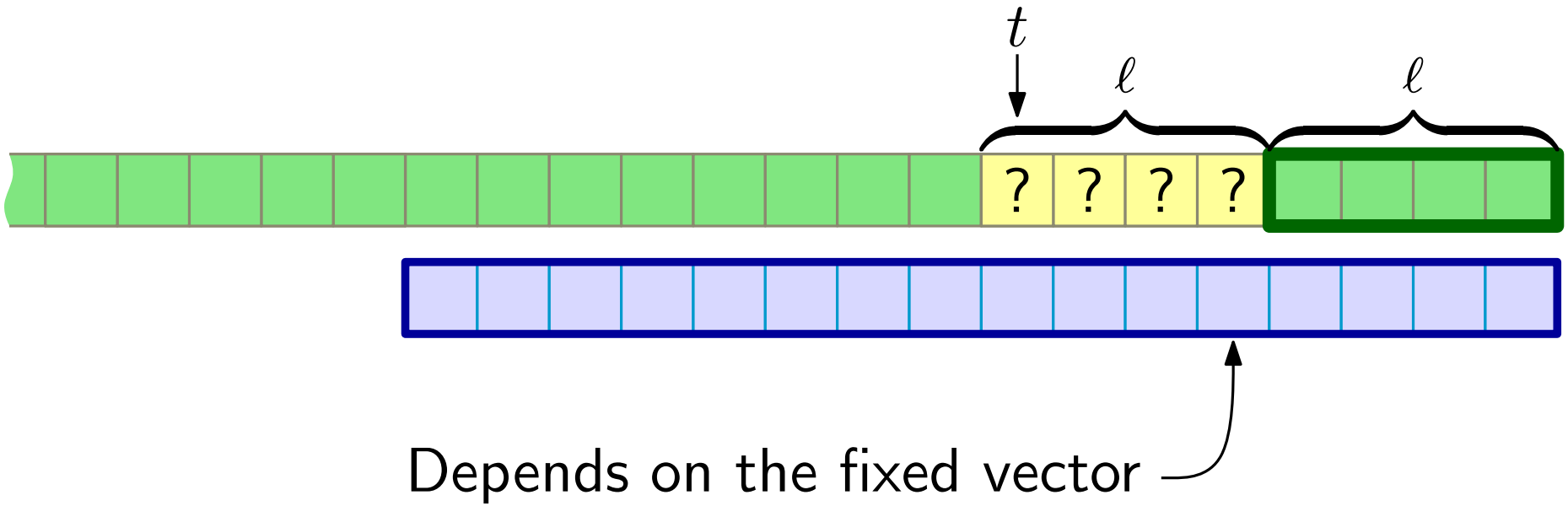
Information transfer



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

in order to give correct outputs during ?

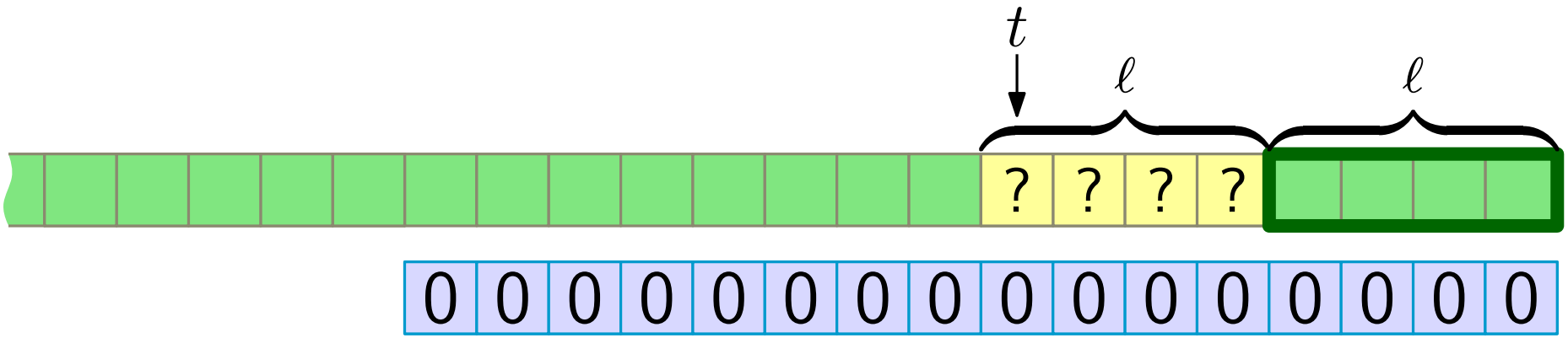
Information transfer



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

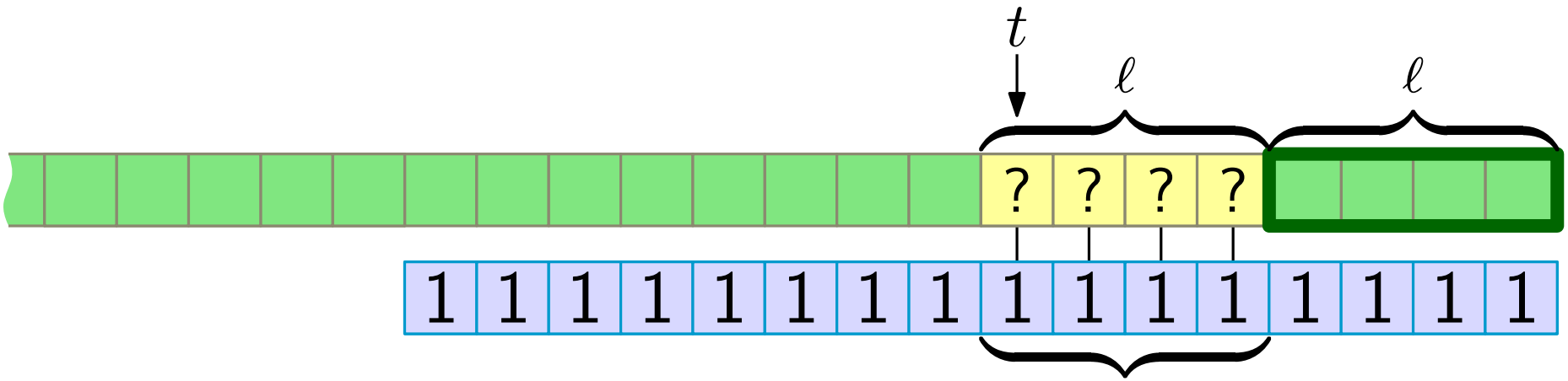
in order to give correct outputs during ?

Information transfer



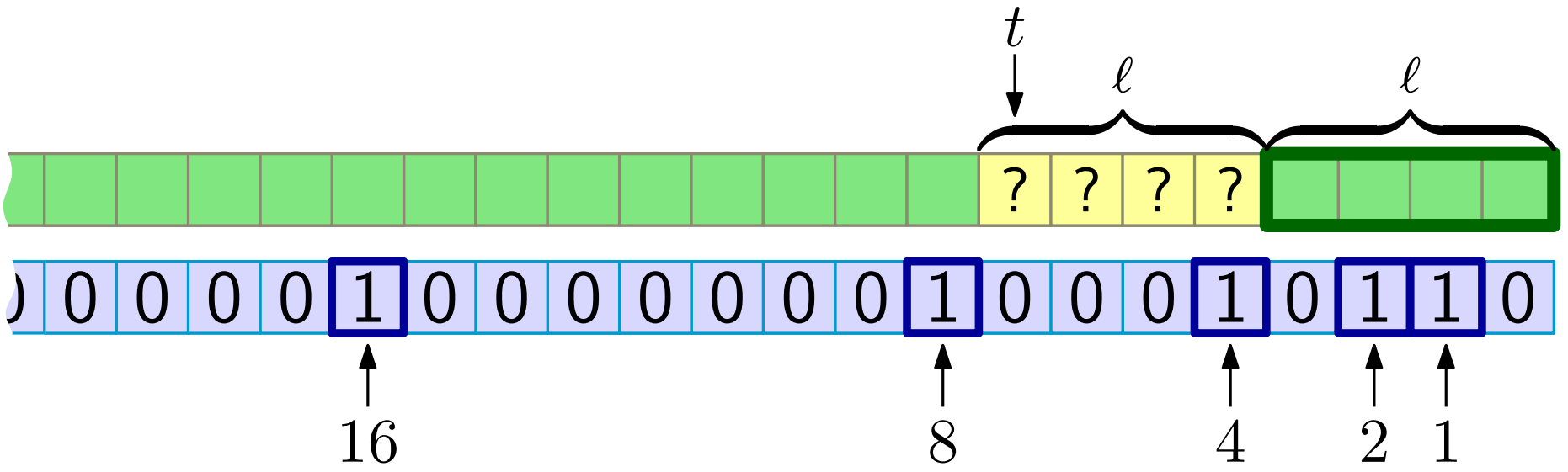
Output is always 0 (no information)

Information transfer



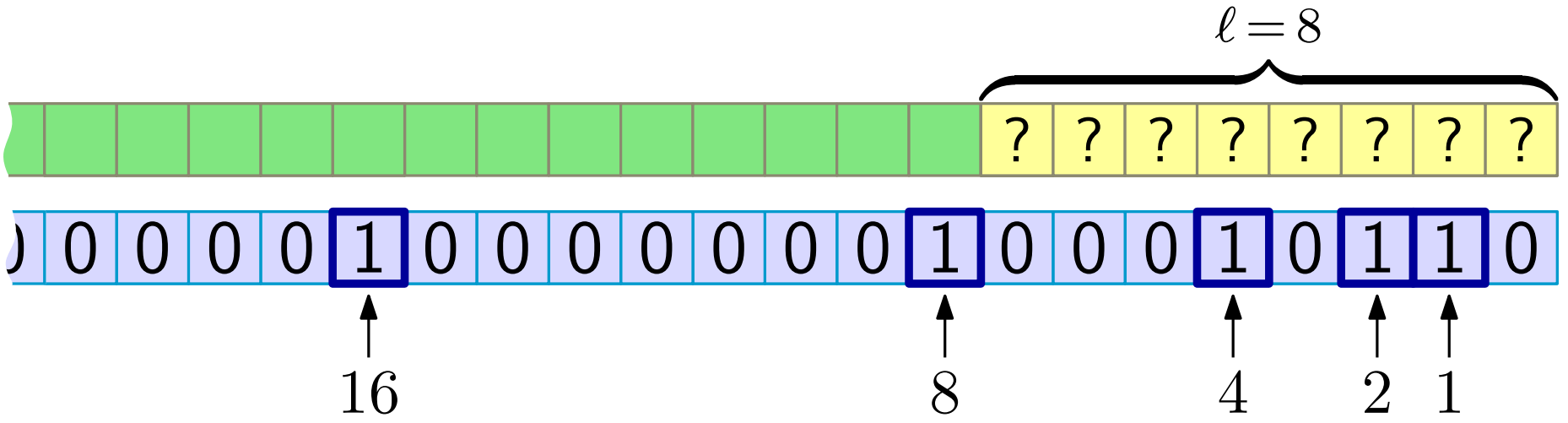
Contributes to the dot product with the same value at each alignment
($\delta = \log q$ bits of information)

Information transfer



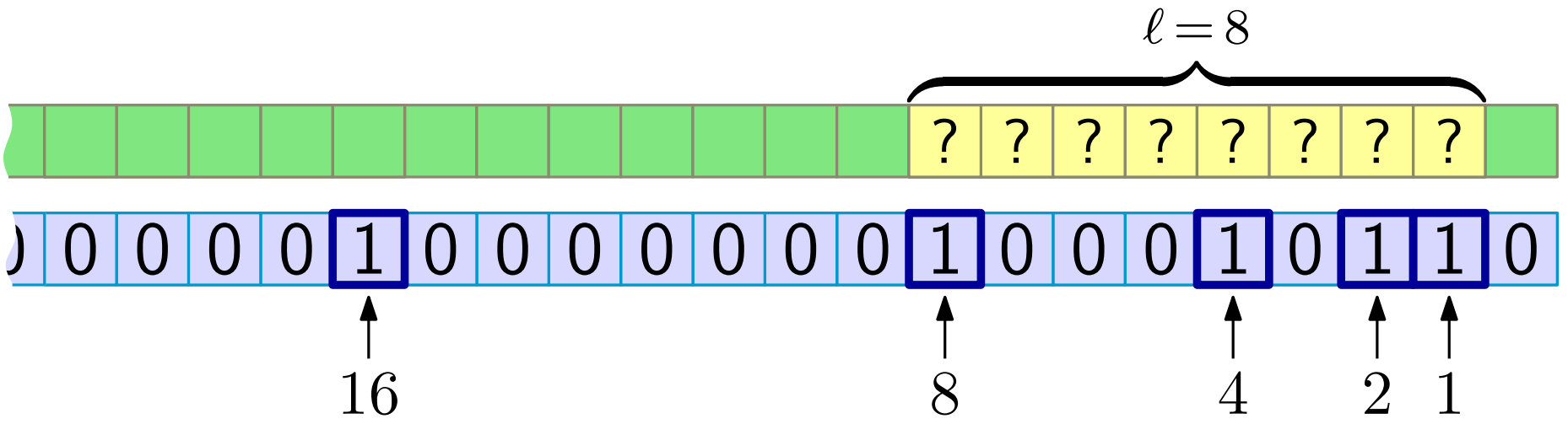
1 if the position is a power of 2

Information transfer



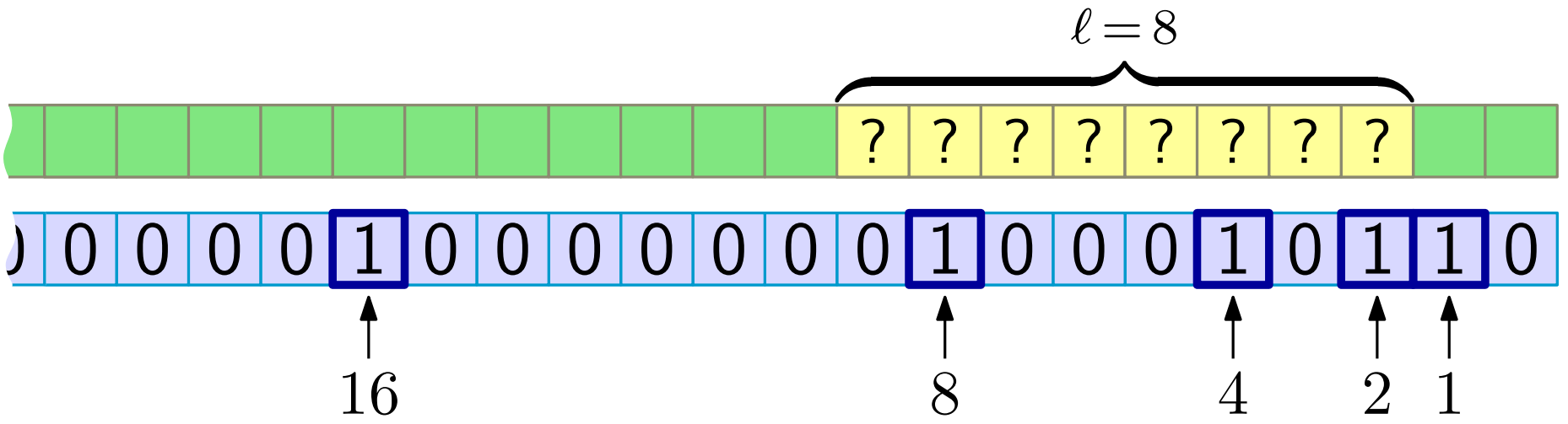
1 if the position is a power of 2

Information transfer



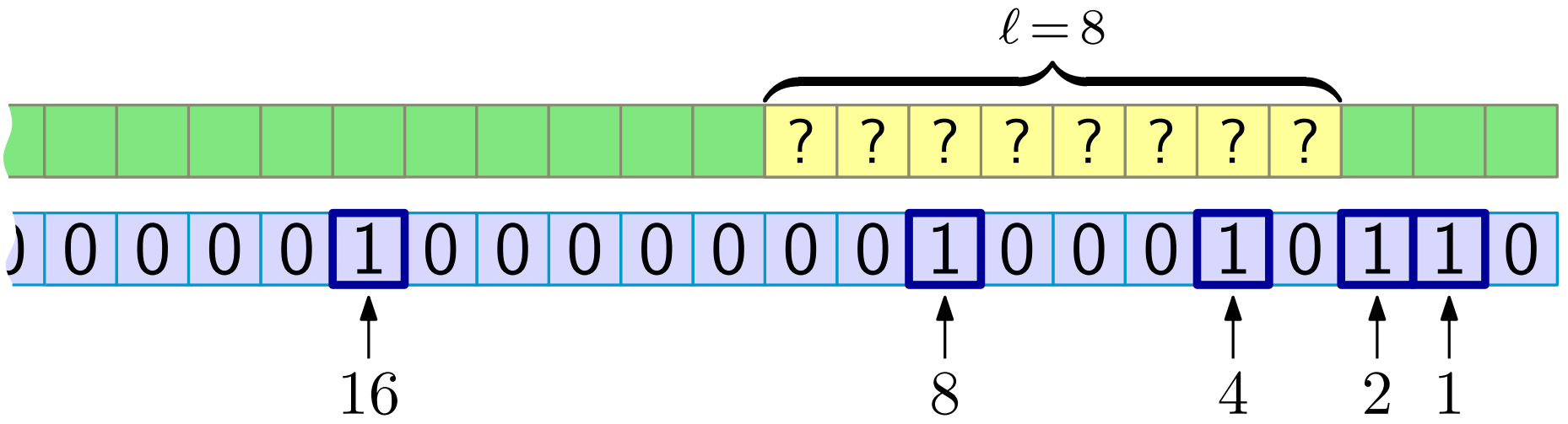
1 if the position is a power of 2

Information transfer



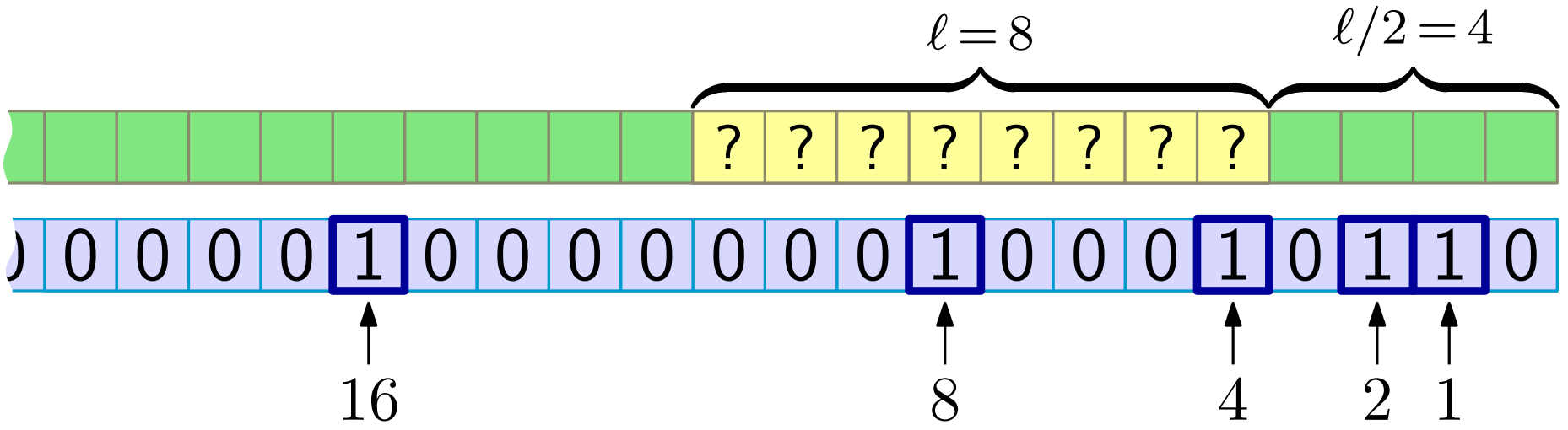
1 if the position is a power of 2

Information transfer



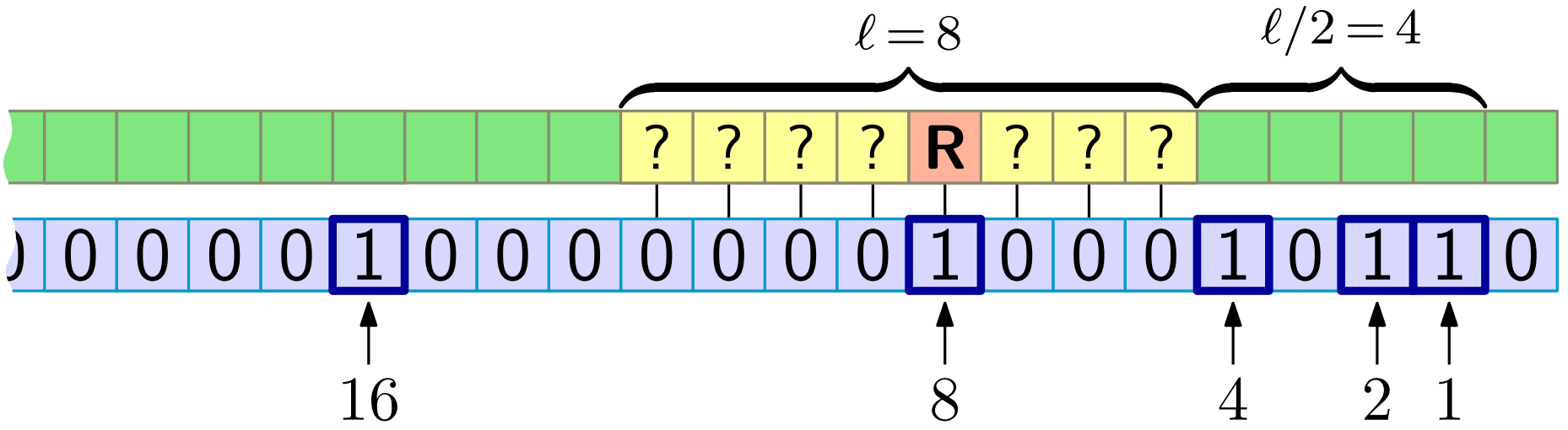
1 if the position is a power of 2

Information transfer



1 if the position is a power of 2

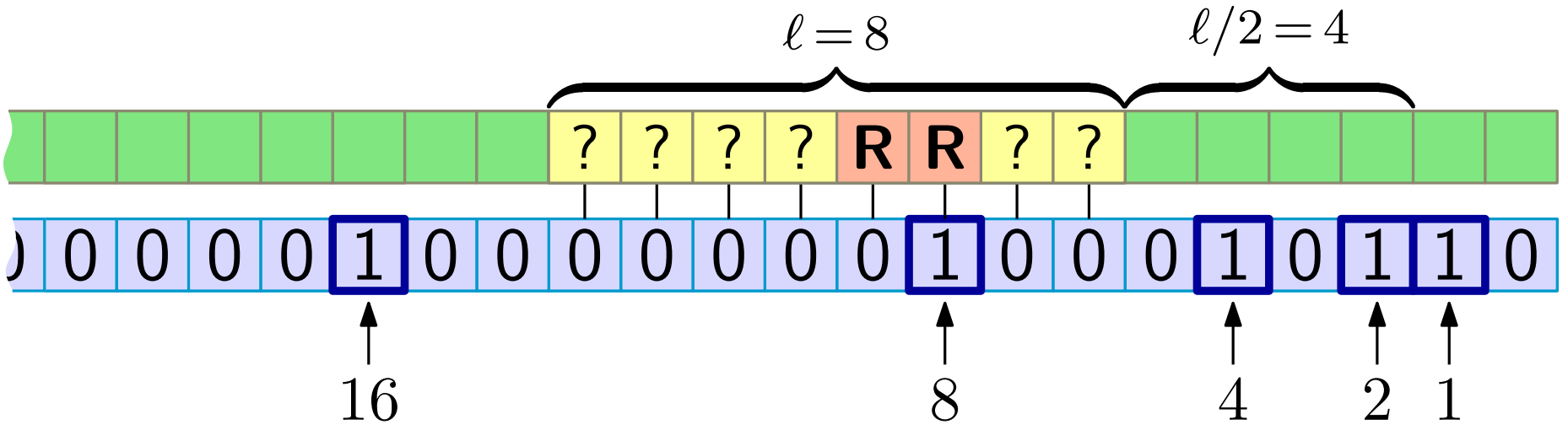
Information transfer



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)

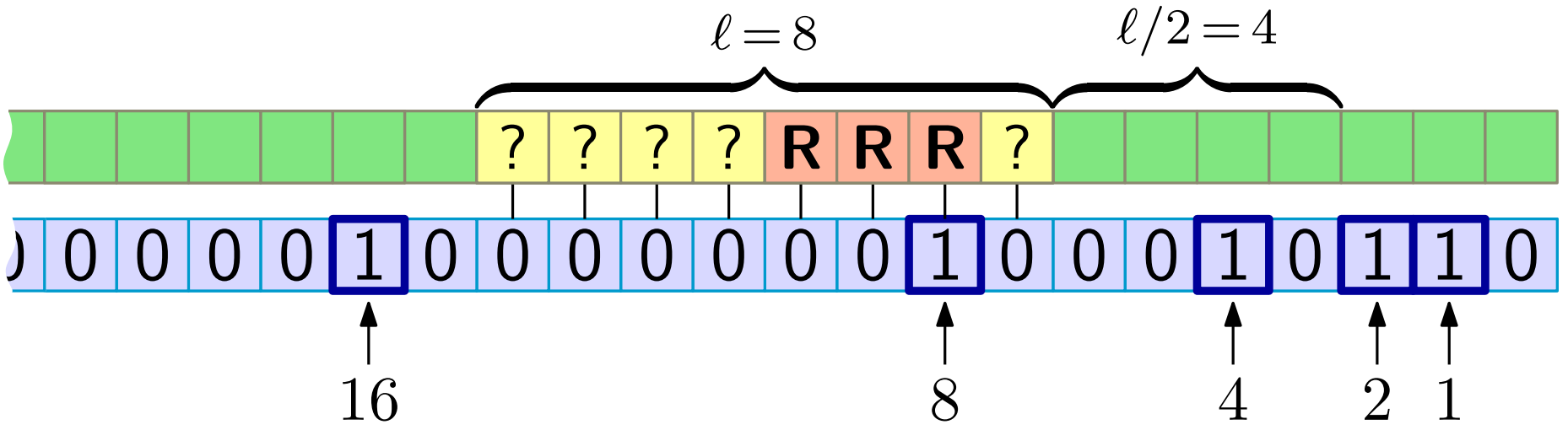
Information transfer



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)

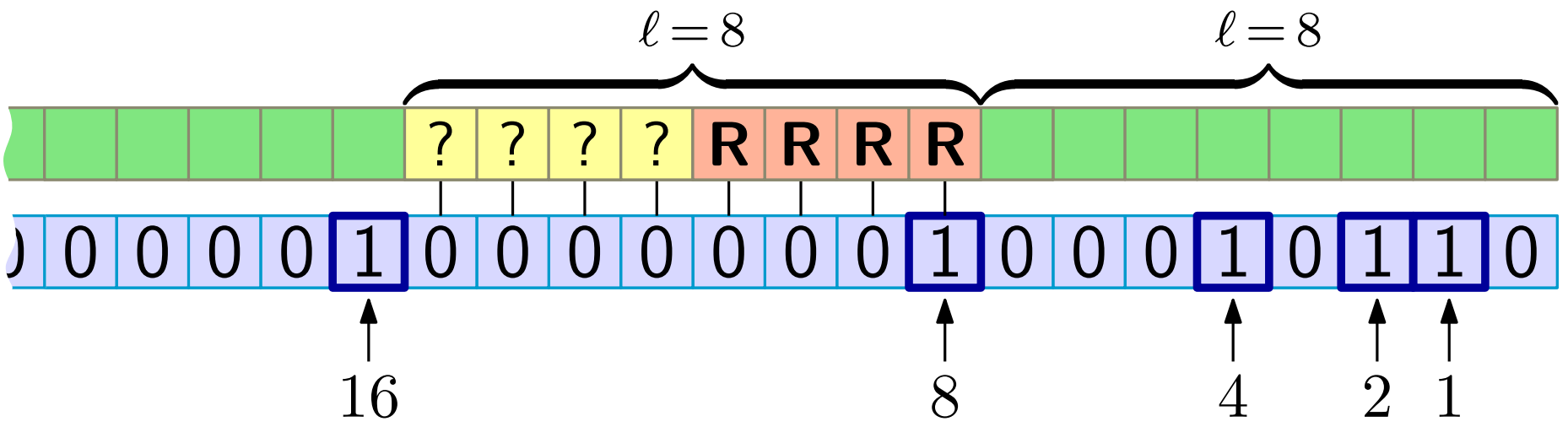
Information transfer



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)

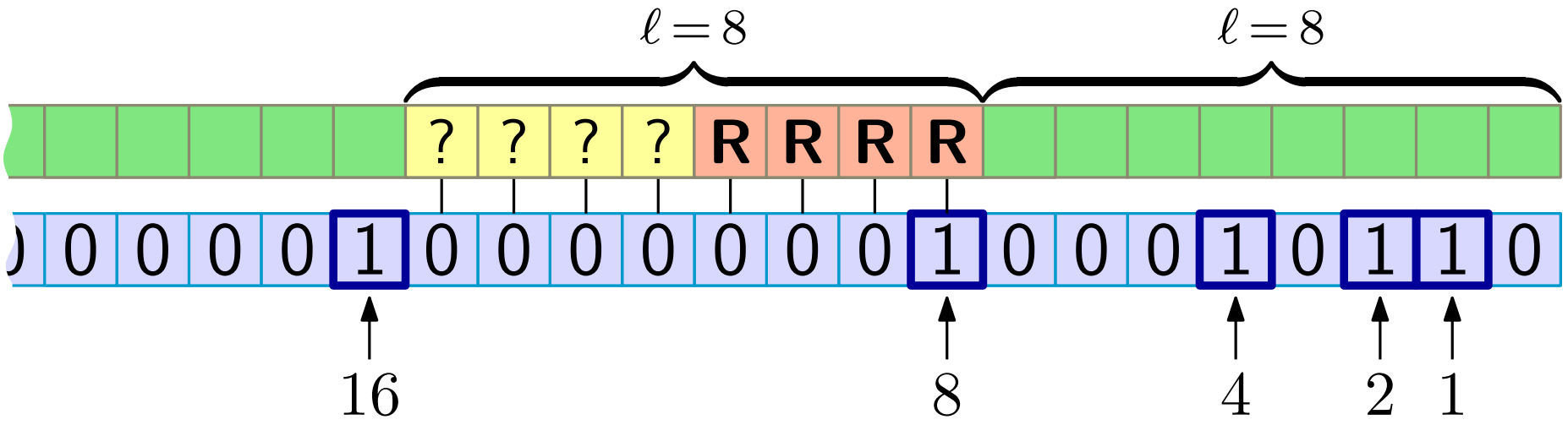
Information transfer



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

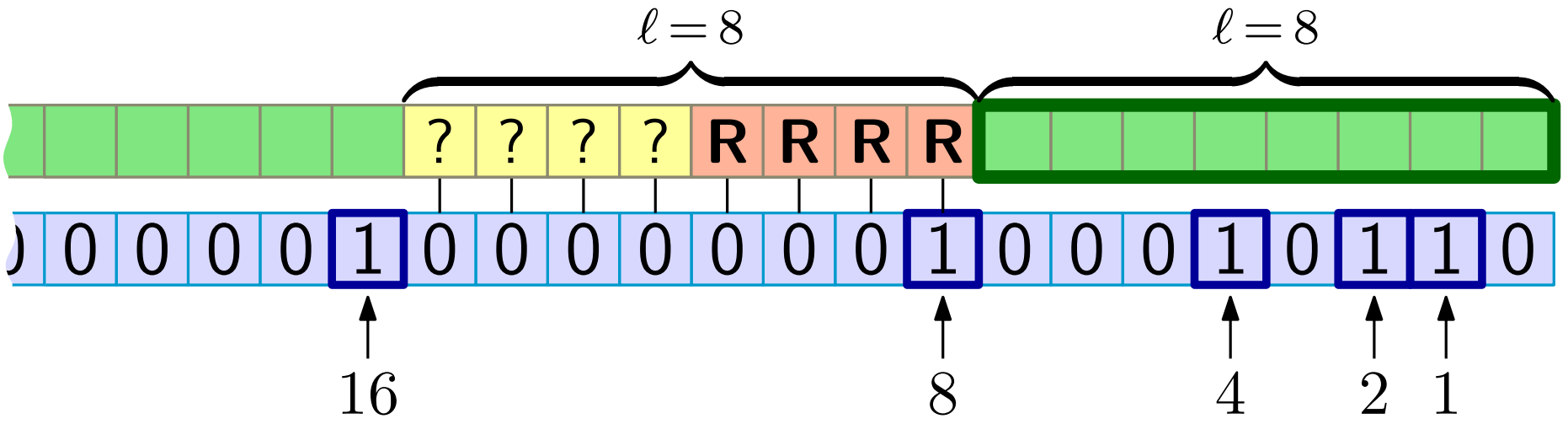


1 if the position is a power of 2

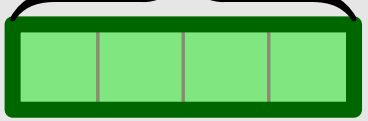
R = a recovered value
(recall that ? is chosen uniformly at random from $[q]$, hence contributes with $\delta = \log q$ bits to the entropy)

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

Information transfer

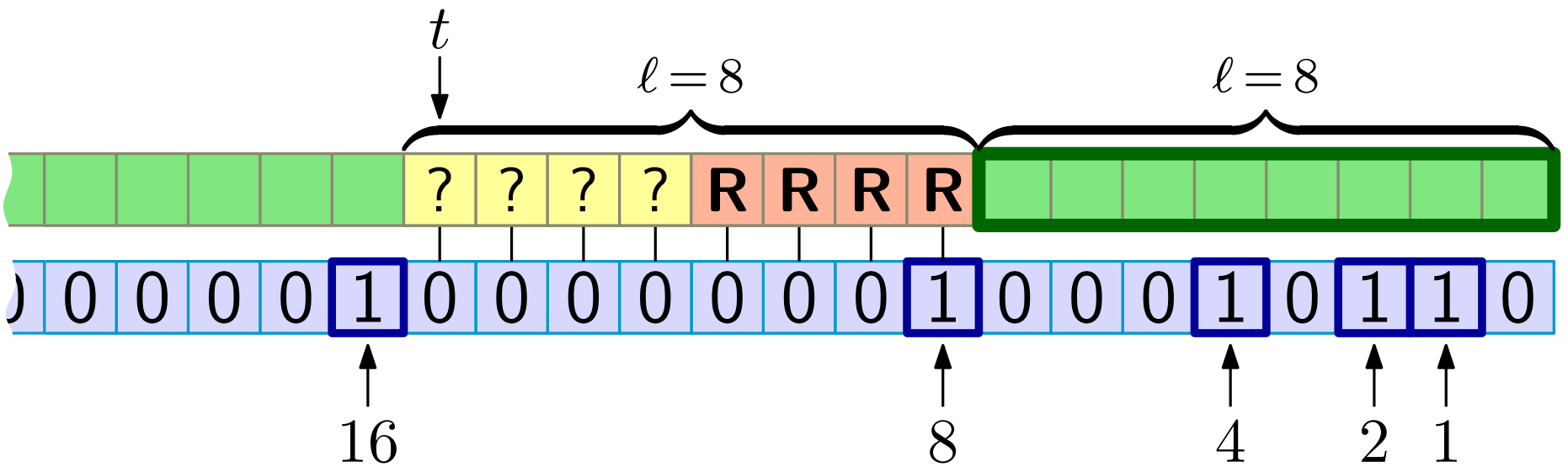


The conditional entropy

$H($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

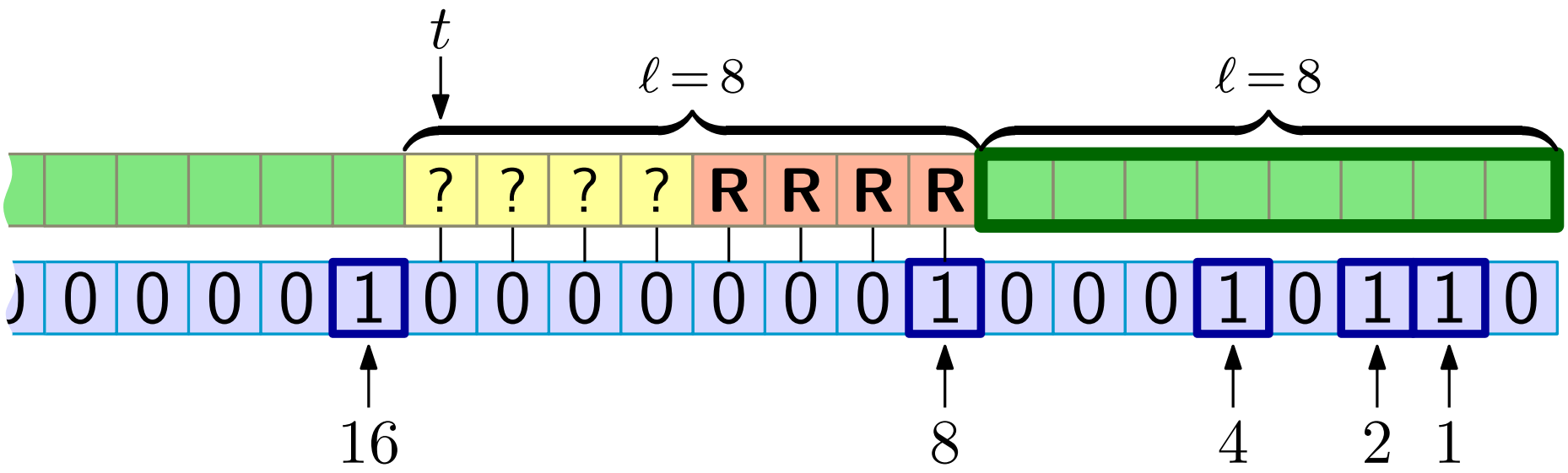
$H($the outputs during $\overbrace{\square\square\square\square}^{\ell}$ | all $\square$ fixed$) \geqslant \frac{\ell}{2}\delta$

The conditional information transfer

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

w bits per cell

Information transfer



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

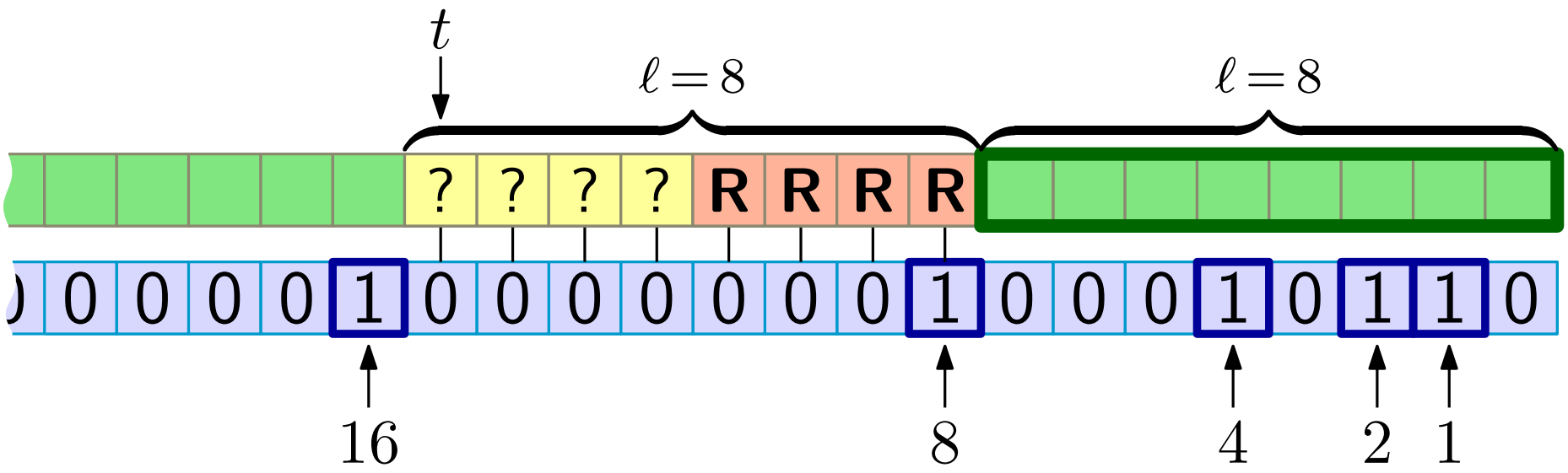
By linearity of expectation...

The conditional information transfer

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

w bits per cell

Information transfer



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

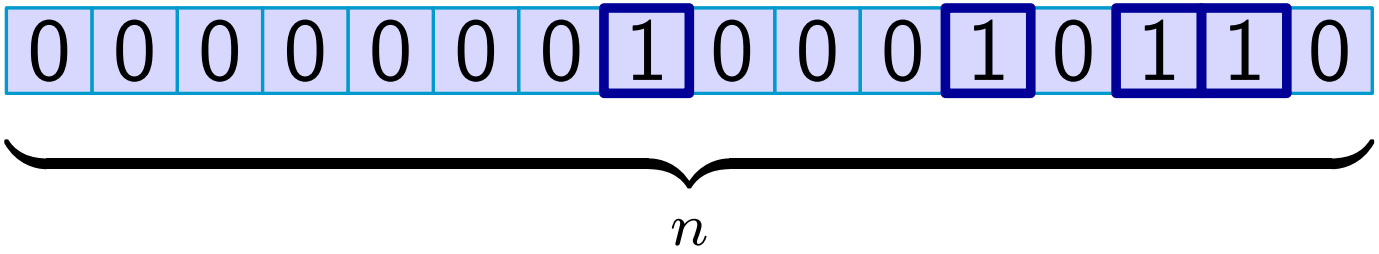
By linearity of expectation...

The conditional information transfer

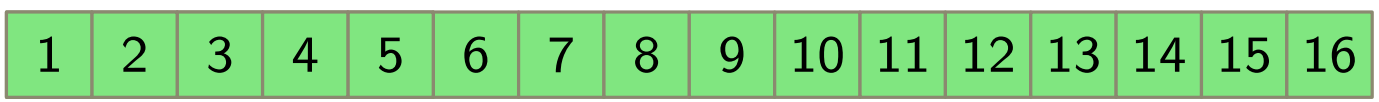
$$\mathbb{E}\left[|IT(t,\ell)| \mid \text{all fixed}\right] \geqslant \frac{\delta}{4w}\ell - \frac{1}{2}$$

w bits per cell

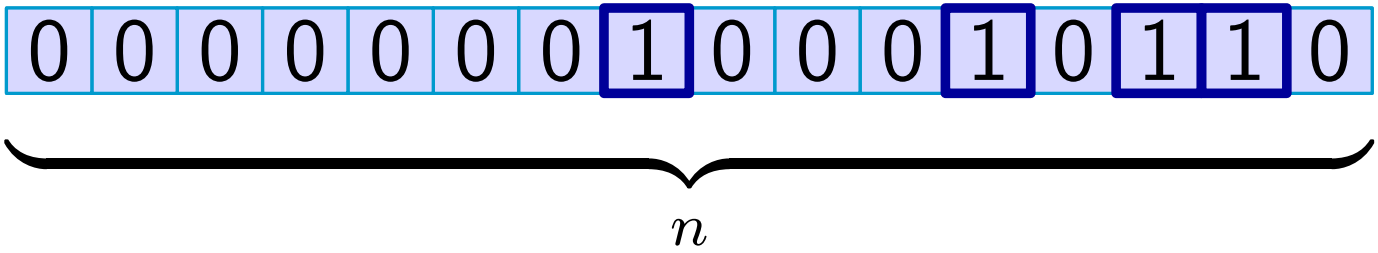
Total number of cell reads



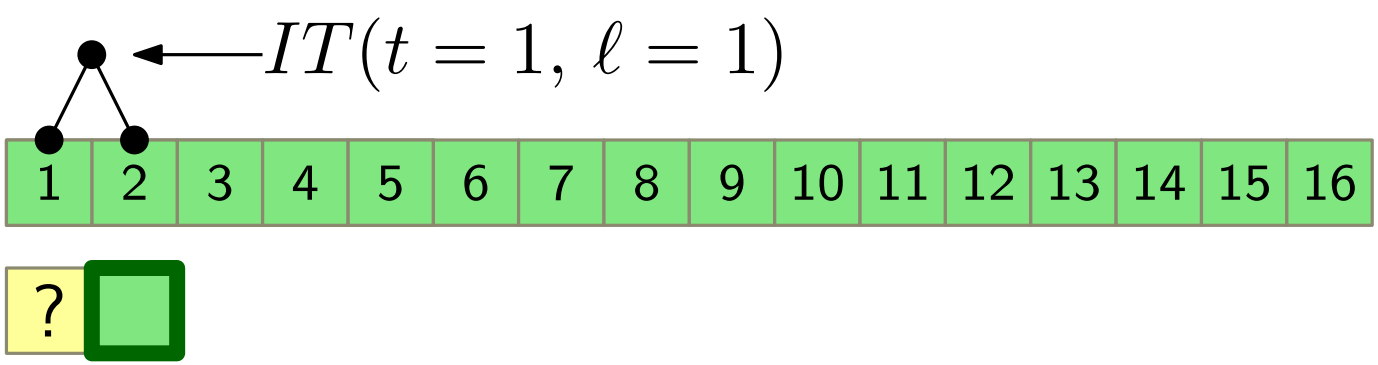
Feed the algorithm with n values chosen uniformly at random from $[q]$.



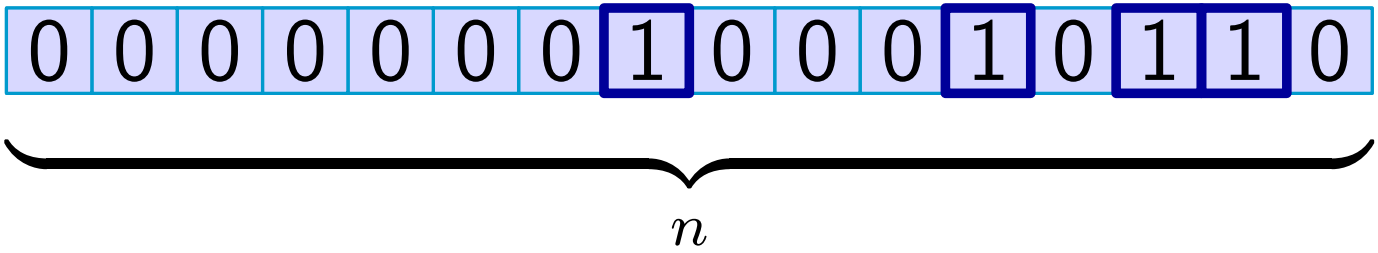
Total number of cell reads



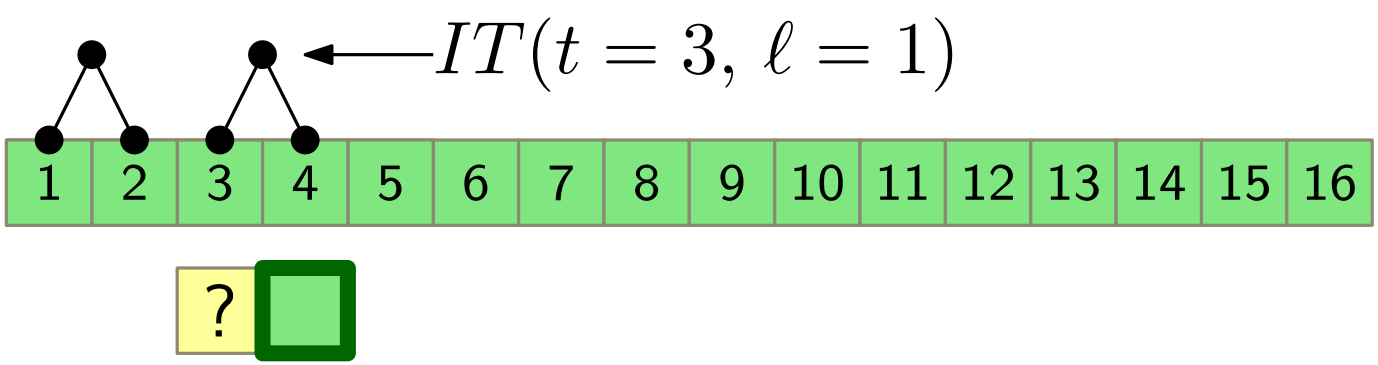
Feed the algorithm with n values chosen uniformly at random from $[q]$.



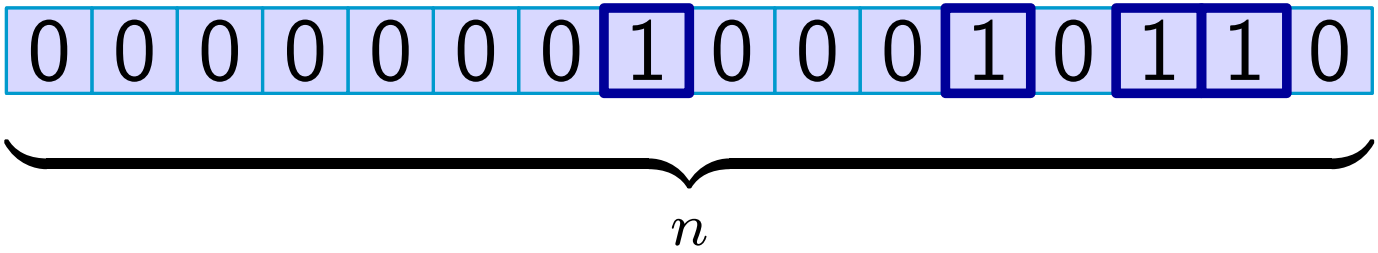
Total number of cell reads



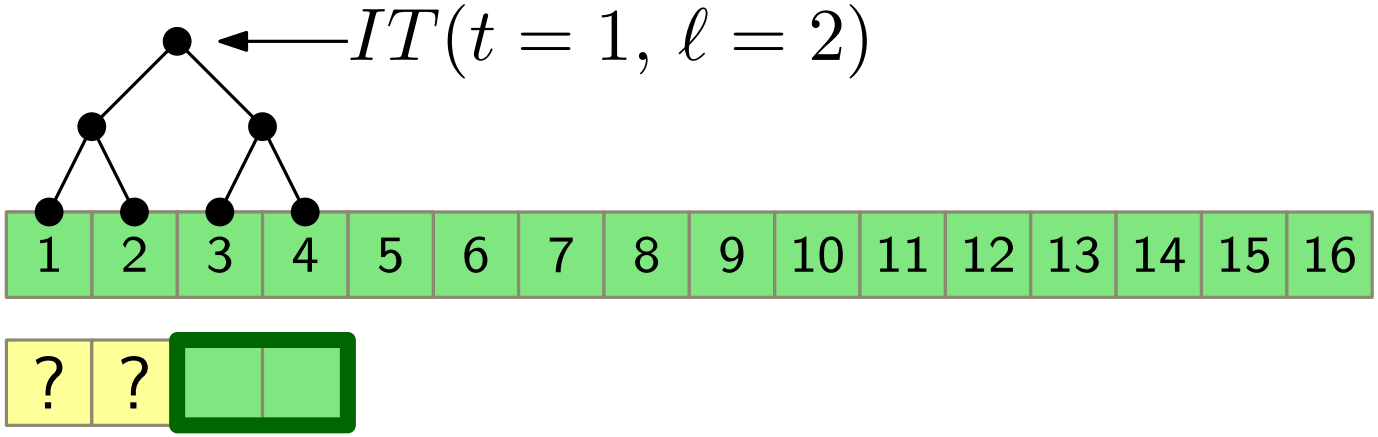
Feed the algorithm with n values chosen uniformly at random from $[q]$.



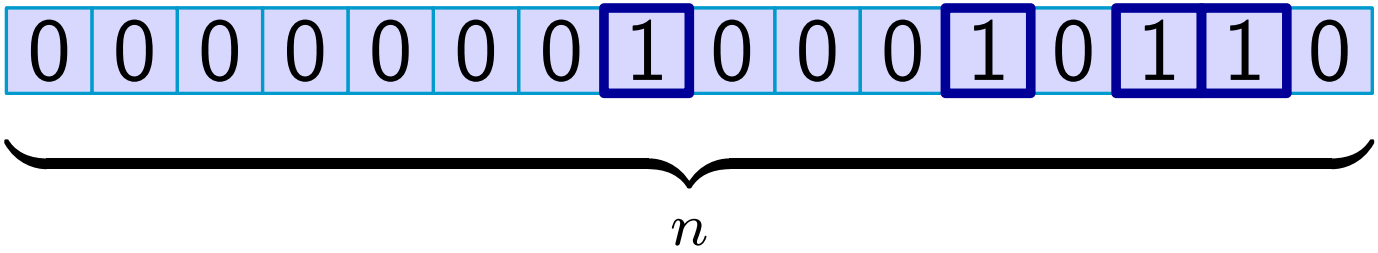
Total number of cell reads



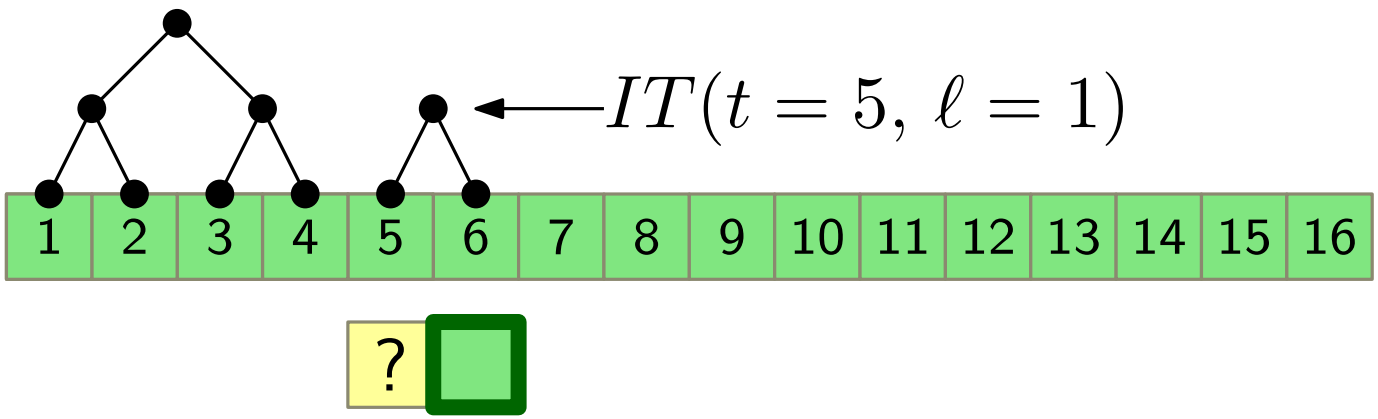
Feed the algorithm with n values chosen uniformly at random from $[q]$.



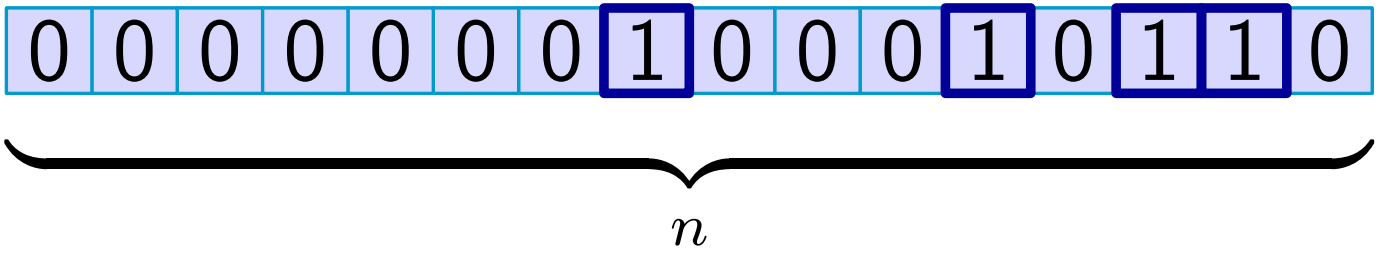
Total number of cell reads



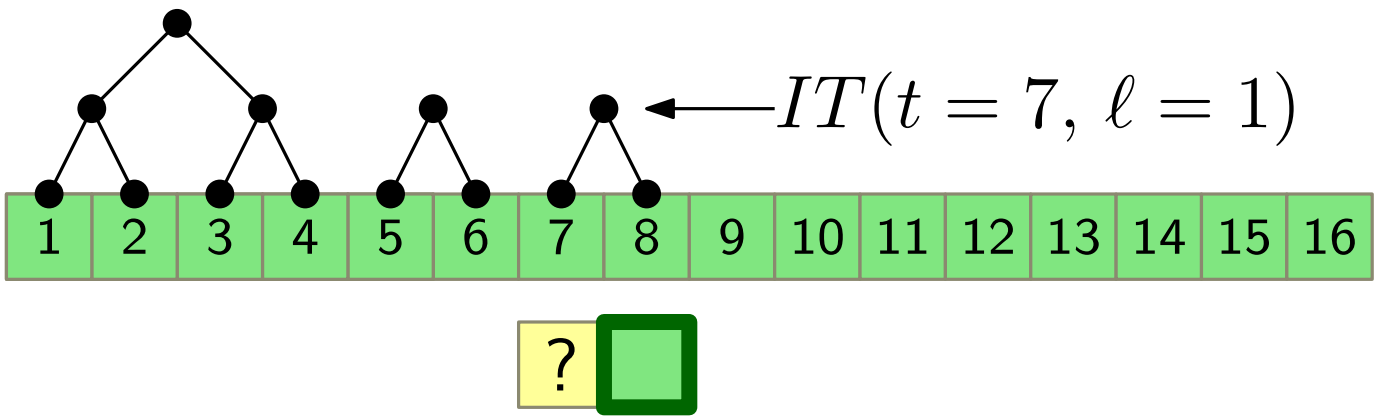
Feed the algorithm with n values chosen uniformly at random from $[q]$.



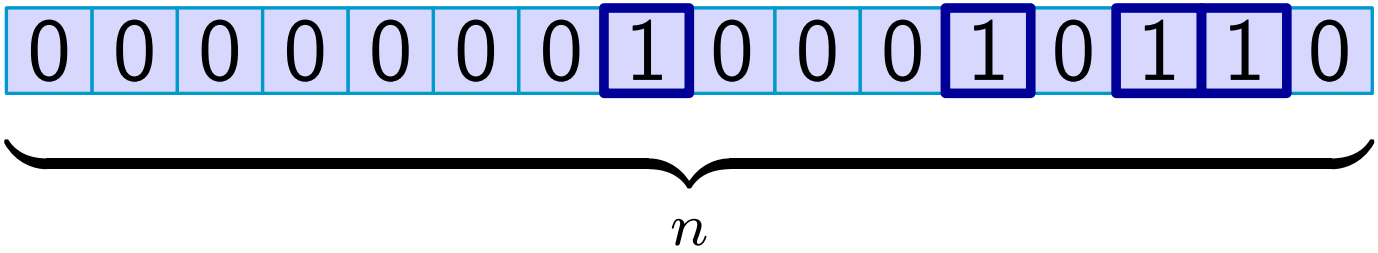
Total number of cell reads



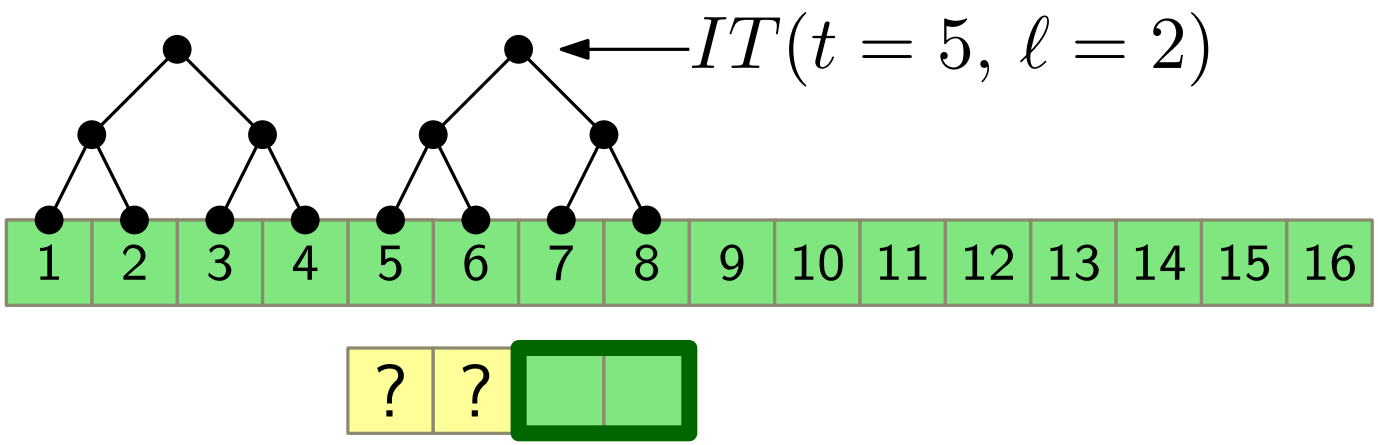
Feed the algorithm with n values chosen uniformly at random from $[q]$.



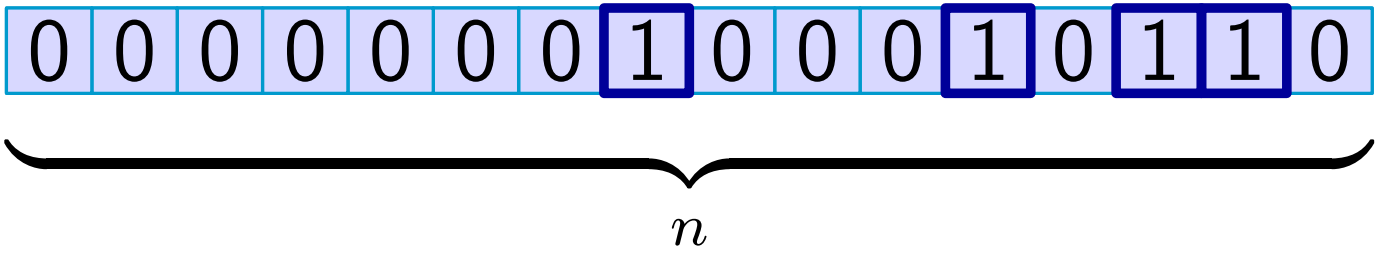
Total number of cell reads



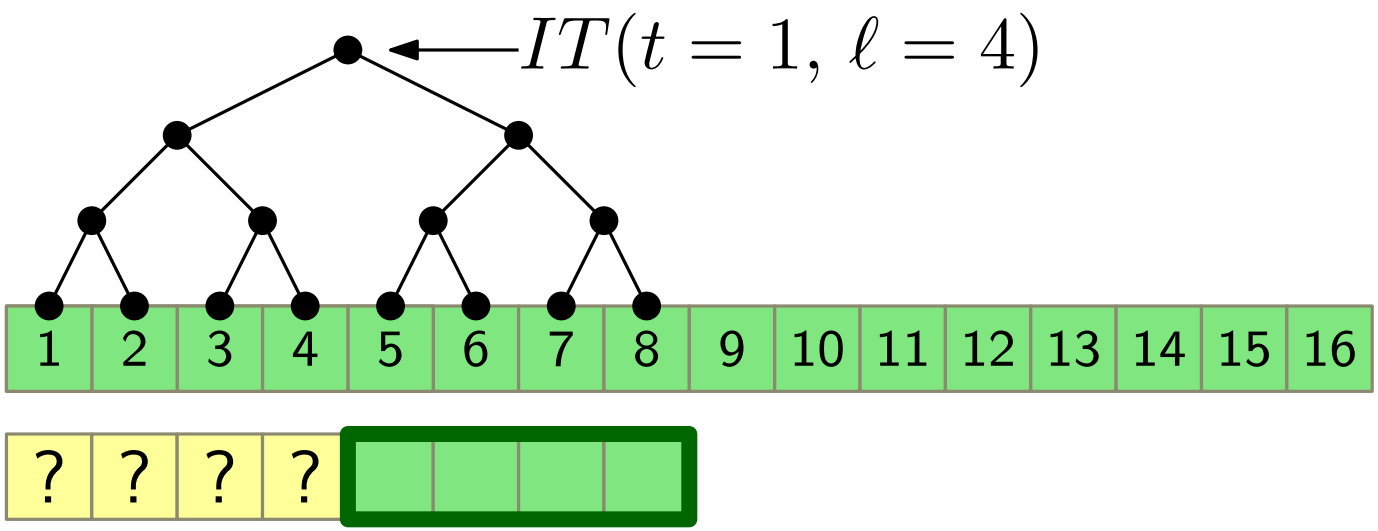
Feed the algorithm with n values chosen uniformly at random from $[q]$.



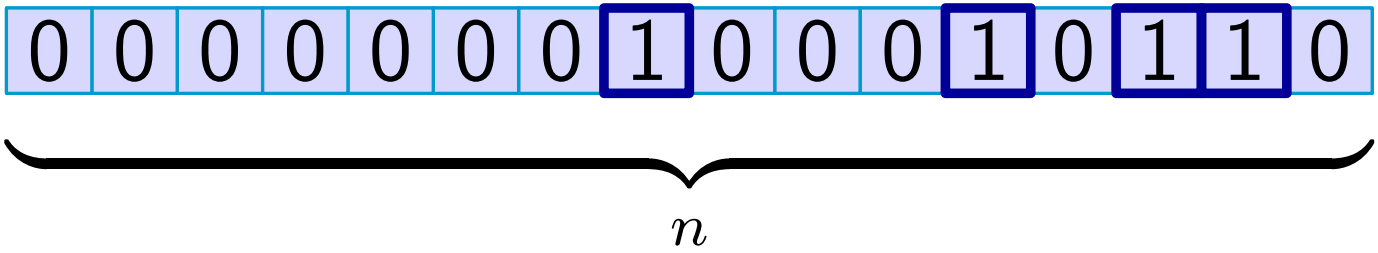
Total number of cell reads



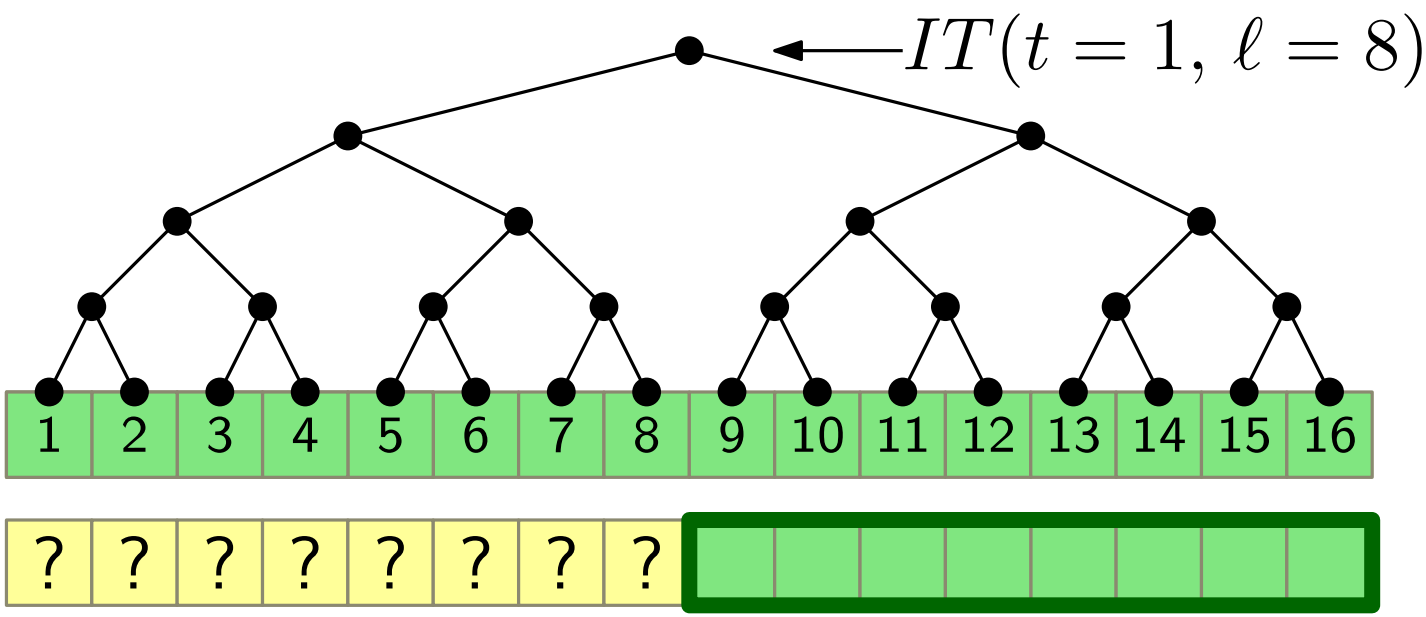
Feed the algorithm with n values chosen uniformly at random from $[q]$.



Total number of cell reads



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

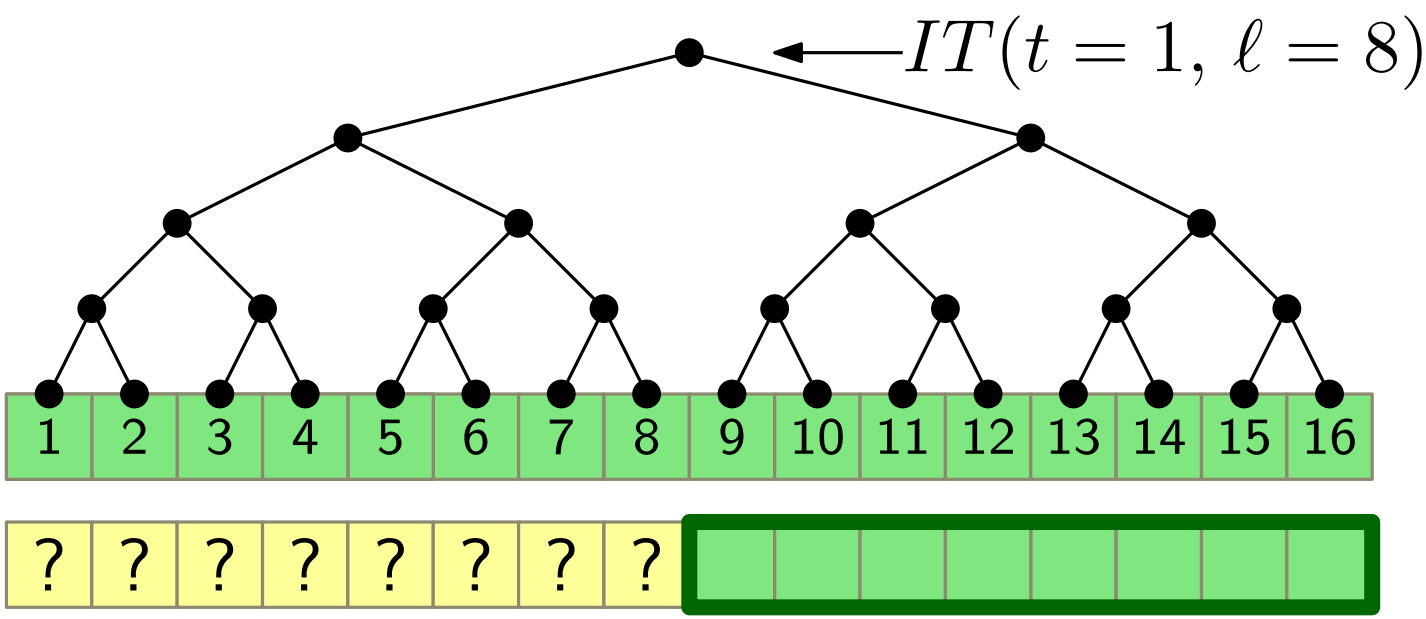


Total number of cell reads

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

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

random from $[q]$.



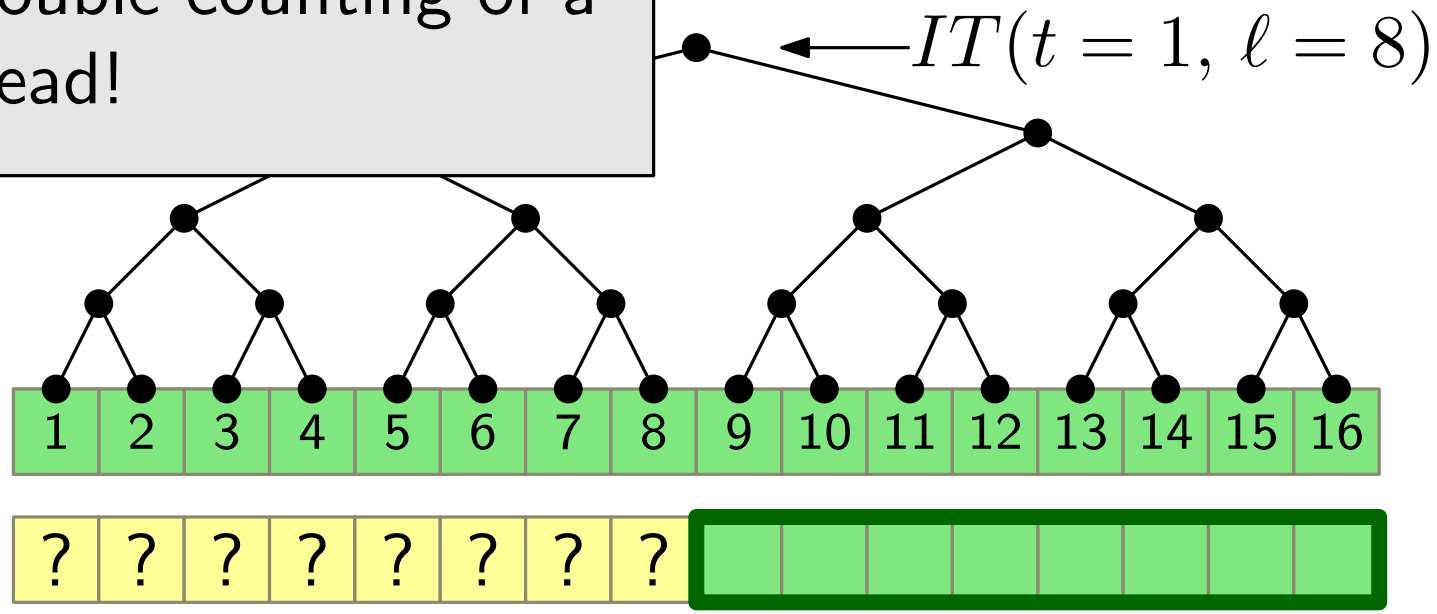
Total number of cell reads

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

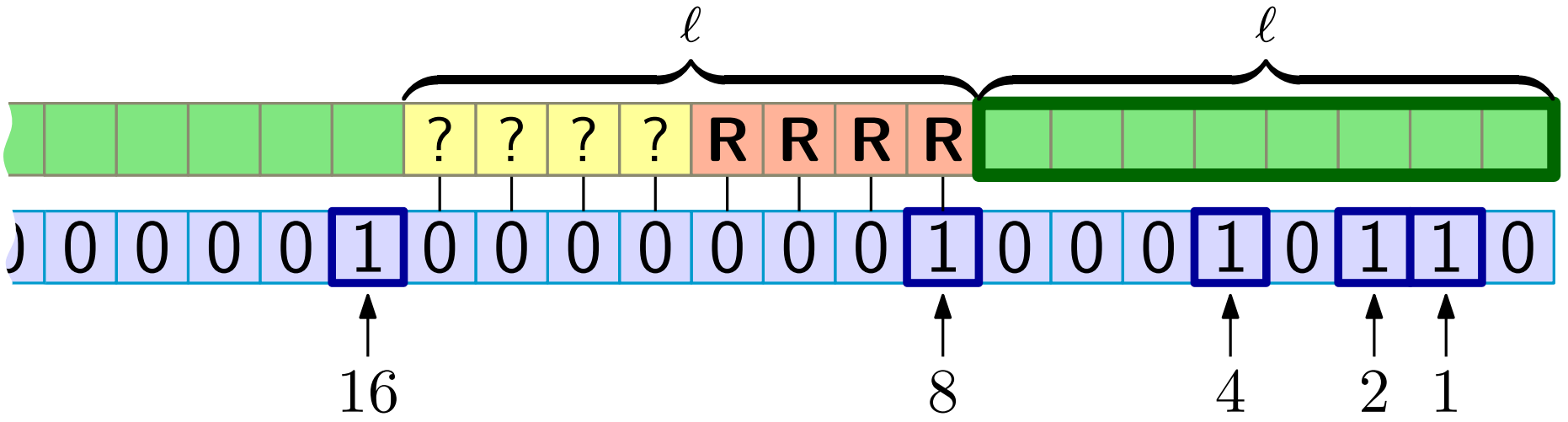
$$\mathbb{E}\left[\sum_{\text{internal node } v} |IT(t_v, \ell_v)|\right] = \sum_{\text{internal node } v} \mathbb{E}\left[|IT(t_v, \ell_v)|\right]$$

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

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

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

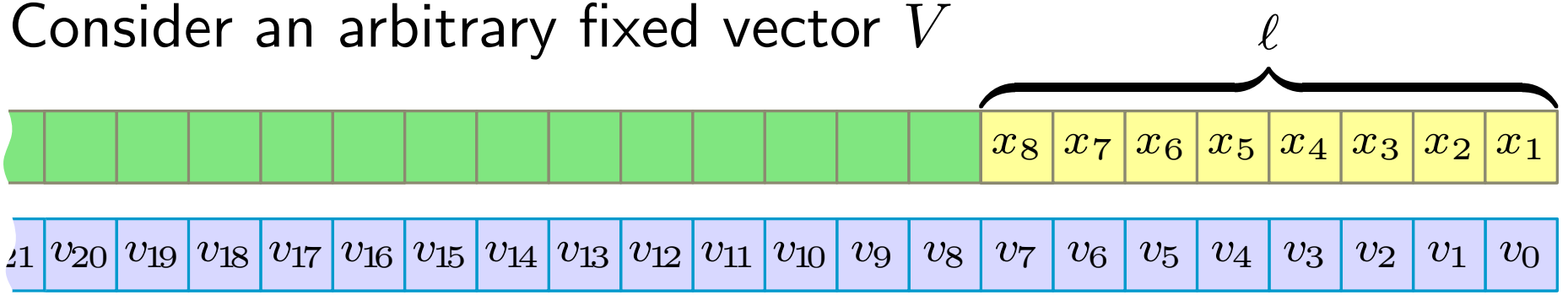
Random fixed vector



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

Random fixed vector

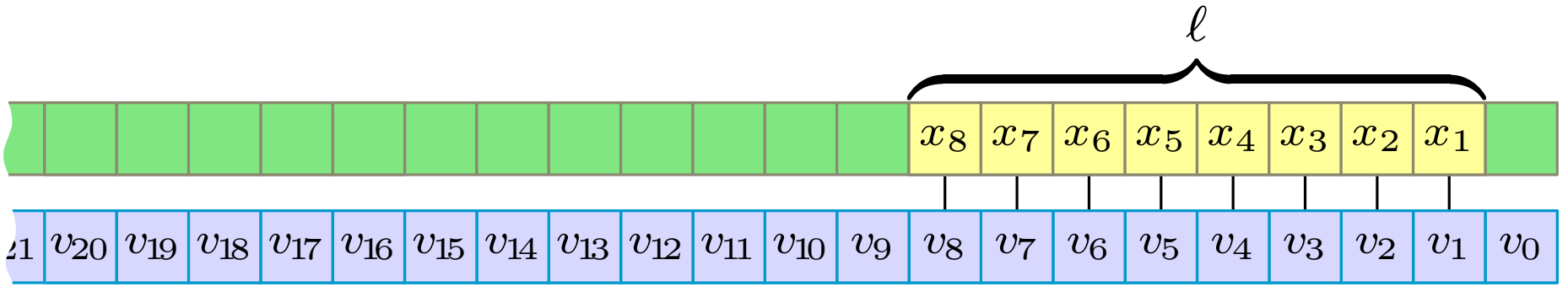
Consider an arbitrary fixed vector V



Fixed value

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

Random fixed vector



Fixed value

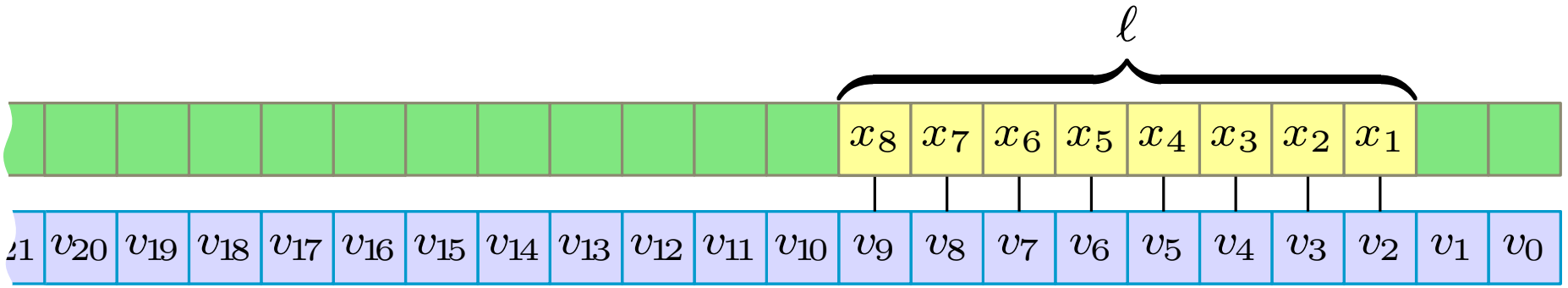
x_i Unknown value chosen uniformly at random from $[q]$

$$\begin{pmatrix} v_8 & v_7 & v_6 & v_5 & v_4 & v_3 & v_2 & v_1 \\ & & & & & & & \\ & & & & & & & \end{pmatrix} \times \begin{pmatrix} x_8 \\ x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} y_8 \\ \\ \end{pmatrix}$$

Contribution from alignment with

x_8	x_7	x_6	x_5	x_4	x_3	x_2	x_1

Random fixed vector



Fixed value

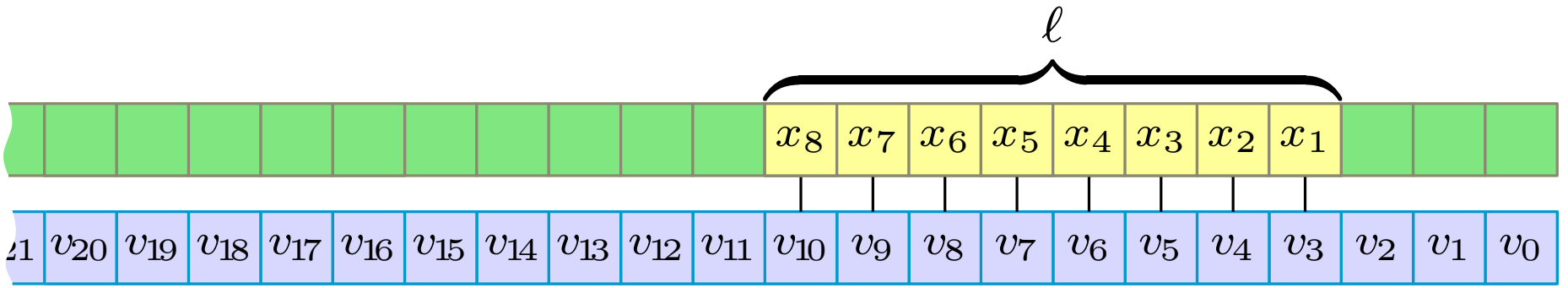
x_i Unknown value chosen uniformly at random from $[q]$

$$\begin{pmatrix} v_9 & v_8 & v_7 & v_6 & v_5 & v_4 & v_3 & v_2 \end{pmatrix} \times \begin{pmatrix} x_8 \\ x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} y_7 \end{pmatrix}$$

Contribution from alignment with

x_8 x_7 x_6 x_5 x_4 x_3 x_2 x_1

Random fixed vector

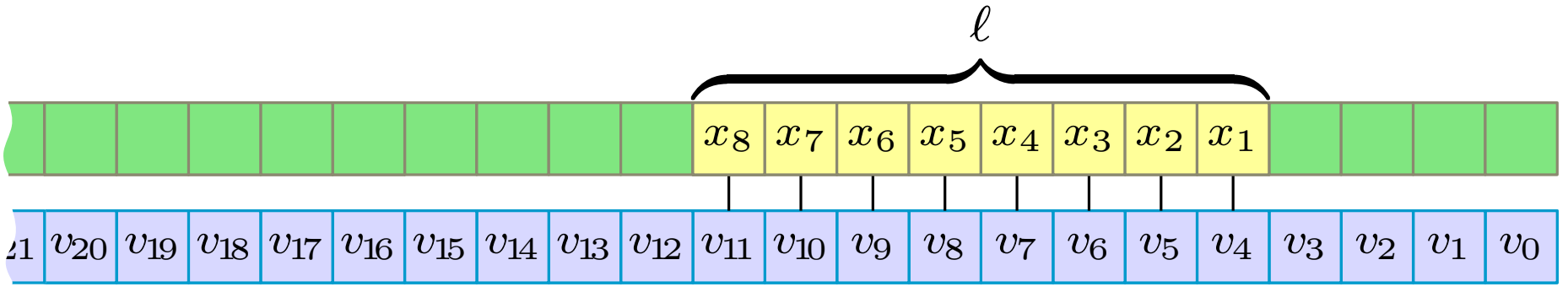


Fixed value

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

$$\begin{pmatrix} v_{10} & v_9 & v_8 & v_7 & v_6 & v_5 & v_4 & v_3 \end{pmatrix} \times \begin{pmatrix} x_8 \\ x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} \\ y_6 \\ \\ \end{pmatrix}$$

Random fixed vector

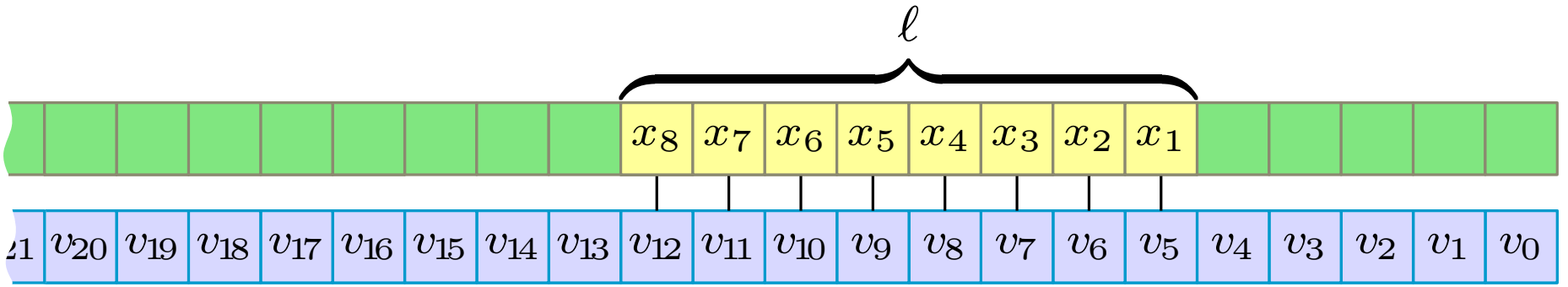


Fixed value

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

$$\begin{pmatrix} v_{11} & v_{10} & v_9 & v_8 & v_7 & v_6 & v_5 & v_4 \end{pmatrix} \times \begin{pmatrix} x_8 \\ x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} \\ \\ \\ \\ \\ \\ \\ \end{pmatrix} y_5$$

Random fixed vector

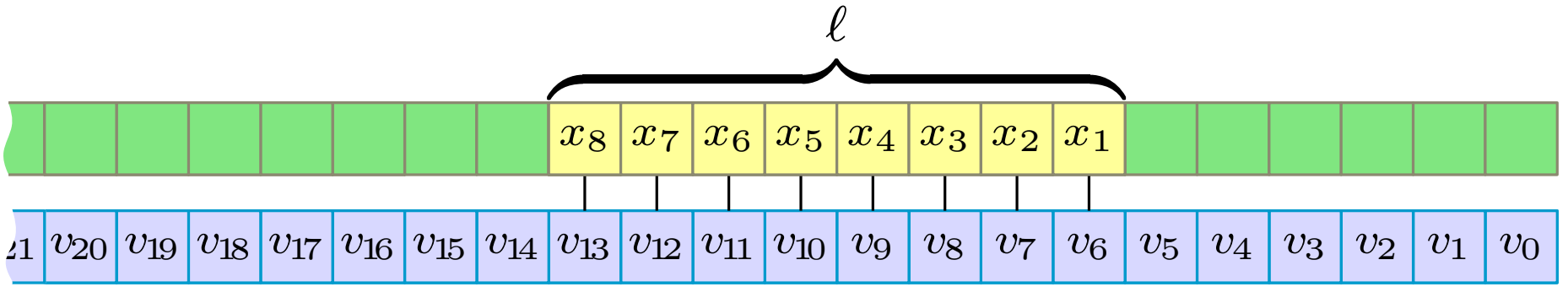


Fixed value

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

$$\begin{pmatrix} v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 & v_6 & v_5 \end{pmatrix} \times \begin{pmatrix} x_8 \\ x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} \\ y_4 \\ \\ \end{pmatrix}$$

Random fixed vector

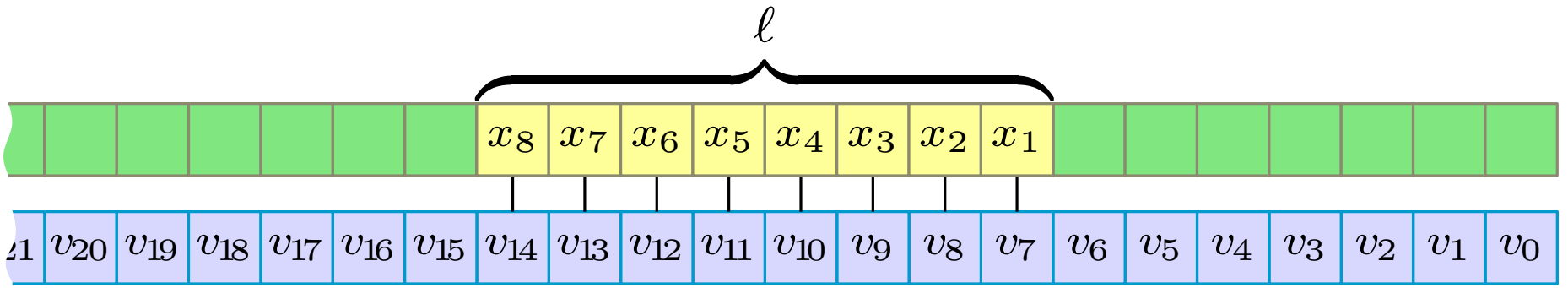


Fixed value

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

$$\begin{pmatrix} & & & & & & & \\ v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 & v_6 \\ & & & & & & & \end{pmatrix} \times \begin{pmatrix} x_8 \\ x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} \\ y_3 \\ \\ \end{pmatrix}$$

Random fixed vector

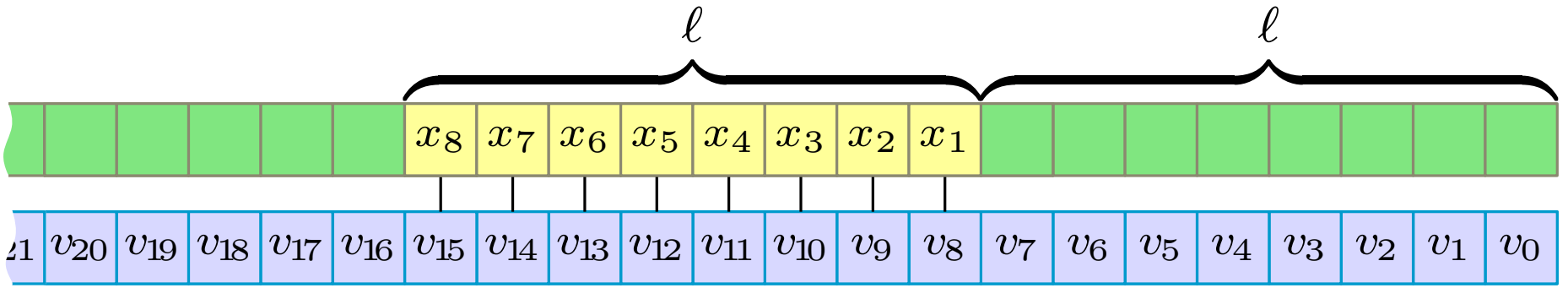


Fixed value

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

$$\begin{pmatrix} & & & & & & & \\ v_{14} & v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 \\ & & & & & & & \end{pmatrix} \times \begin{pmatrix} x_8 \\ x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} \\ \\ \\ \\ \\ \\ \\ y_2 \\ \\ \end{pmatrix}$$

Random fixed vector

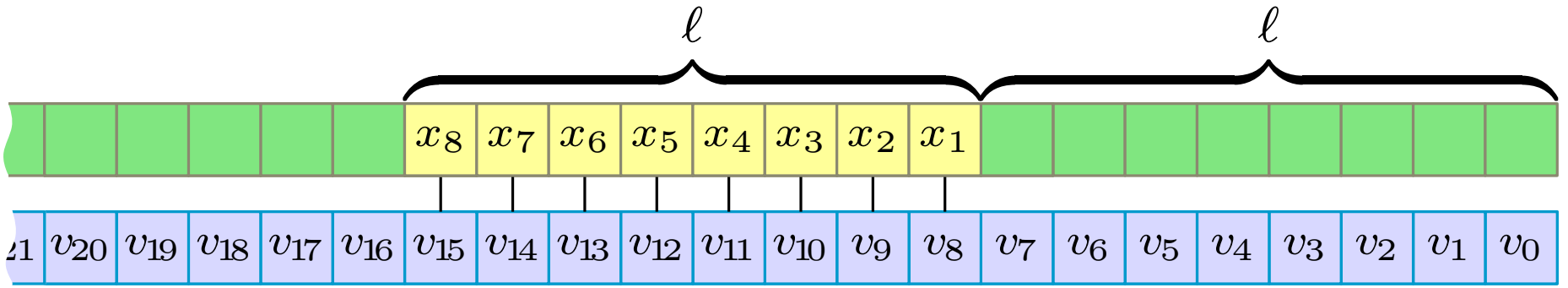


Fixed value

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

$$\begin{pmatrix} & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ v_{15} & v_{14} & v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 \end{pmatrix} \times \begin{pmatrix} x_8 \\ x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} \\ \\ \\ \\ \\ \\ \\ y_1 \end{pmatrix}$$

Random fixed vector



Fixed value

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

$$\begin{pmatrix} v_8 & v_7 & v_6 & v_5 & v_4 & v_3 & v_2 & v_1 \\ v_9 & v_8 & v_7 & v_6 & v_5 & v_4 & v_3 & v_2 \\ v_{10} & v_9 & v_8 & v_7 & v_6 & v_5 & v_4 & v_3 \\ v_{11} & v_{10} & v_9 & v_8 & v_7 & v_6 & v_5 & v_4 \\ v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 & v_6 & v_5 \\ v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 & v_6 \\ v_{14} & v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 \\ v_{15} & v_{14} & v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 \end{pmatrix} \times \begin{pmatrix} x_8 \\ x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} y_8 \\ y_7 \\ y_6 \\ y_5 \\ y_4 \\ y_3 \\ y_2 \\ y_1 \end{pmatrix}$$

Random fixed vector

The matrix is a Toeplitz matrix.
Suppose we choose the vector V randomly from $[q]^n$.

$$\begin{pmatrix} v_8 & v_7 & v_6 & v_5 & v_4 & v_3 & v_2 & v_1 \\ v_9 & v_8 & v_7 & v_6 & v_5 & v_4 & v_3 & v_2 \\ v_{10} & v_9 & v_8 & v_7 & v_6 & v_5 & v_4 & v_3 \\ v_{11} & v_{10} & v_9 & v_8 & v_7 & v_6 & v_5 & v_4 \\ v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 & v_6 & v_5 \\ v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 & v_6 \\ v_{14} & v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 \\ v_{15} & v_{14} & v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 \end{pmatrix} \times \begin{pmatrix} x_8 \\ x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} y_8 \\ y_7 \\ y_6 \\ y_5 \\ y_4 \\ y_3 \\ y_2 \\ y_1 \end{pmatrix}$$

Random fixed vector

The matrix is a Toeplitz matrix.
Suppose we choose the vector V randomly from $[q]^n$.

If q is a prime then we operate in the field $\mathbb{Z}/q\mathbb{Z}$, and the Toeplitz matrix is invertible with probability $1 - 1/q \geqslant 1/2$.

$$\begin{pmatrix} v_8 & v_7 & v_6 & v_5 & v_4 & v_3 & v_2 & v_1 \\ v_9 & v_8 & v_7 & v_6 & v_5 & v_4 & v_3 & v_2 \\ v_{10} & v_9 & v_8 & v_7 & v_6 & v_5 & v_4 & v_3 \\ v_{11} & v_{10} & v_9 & v_8 & v_7 & v_6 & v_5 & v_4 \\ v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 & v_6 & v_5 \\ v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 & v_6 \\ v_{14} & v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 \\ v_{15} & v_{14} & v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 \end{pmatrix} \times \begin{pmatrix} x_8 \\ x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} y_8 \\ y_7 \\ y_6 \\ y_5 \\ y_4 \\ y_3 \\ y_2 \\ y_1 \end{pmatrix}$$

Random fixed vector

The matrix is a Toeplitz matrix.
Suppose we choose the vector V randomly from $[q]^n$.

If q is a prime then we operate in the field $\mathbb{Z}/q\mathbb{Z}$, and the Toeplitz matrix is invertible with probability $1 - 1/q \geqslant 1/2$.

When invertible, we can recover $x_1, \ldots, x_\ell$ (ℓ values).

$$\begin{pmatrix} v_8 & v_7 & v_6 & v_5 & v_4 & v_3 & v_2 & v_1 \\ v_9 & v_8 & v_7 & v_6 & v_5 & v_4 & v_3 & v_2 \\ v_{10} & v_9 & v_8 & v_7 & v_6 & v_5 & v_4 & v_3 \\ v_{11} & v_{10} & v_9 & v_8 & v_7 & v_6 & v_5 & v_4 \\ v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 & v_6 & v_5 \\ v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 & v_6 \\ v_{14} & v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 \\ v_{15} & v_{14} & v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 \end{pmatrix} \times \begin{pmatrix} x_8 \\ x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} y_8 \\ y_7 \\ y_6 \\ y_5 \\ y_4 \\ y_3 \\ y_2 \\ y_1 \end{pmatrix}$$

Random fixed vector

The matrix is a Toeplitz matrix.
Suppose we choose the vector V randomly from $[q]^n$.

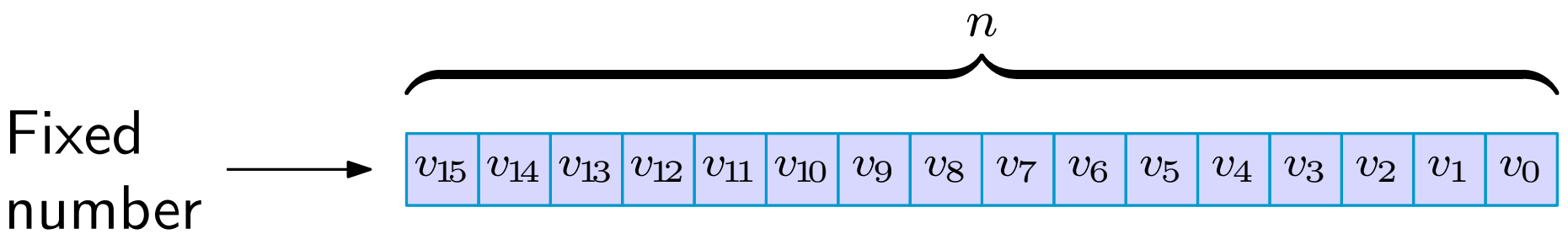
If q is a prime then we operate in the field $\mathbb{Z}/q\mathbb{Z}$, and the Toeplitz matrix is invertible with probability $1 - 1/q \geqslant 1/2$.

When invertible, we can recover $x_1, \ldots, x_\ell$ (ℓ values).

So...
For a random fixed vector V, the amortised time lower bound per output is also $\Omega\left(\frac{\delta}{w} \log n\right)$

$$\begin{pmatrix} v_8 & & & & & & & \\ v_9 & & & & & & & \\ v_{10} & & & & & & & \\ v_{11} & & & & & & & \\ v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 & v_6 & v_5 \\ v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 & v_6 \\ v_{14} & v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 & v_7 \\ v_{15} & v_{14} & v_{13} & v_{12} & v_{11} & v_{10} & v_9 & v_8 \end{pmatrix} \times \begin{pmatrix} x_4 \\ x_3 \\ x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} y_8 \\ y_7 \\ y_6 \\ y_5 \\ y_4 \\ y_3 \\ y_2 \\ y_1 \end{pmatrix}$$

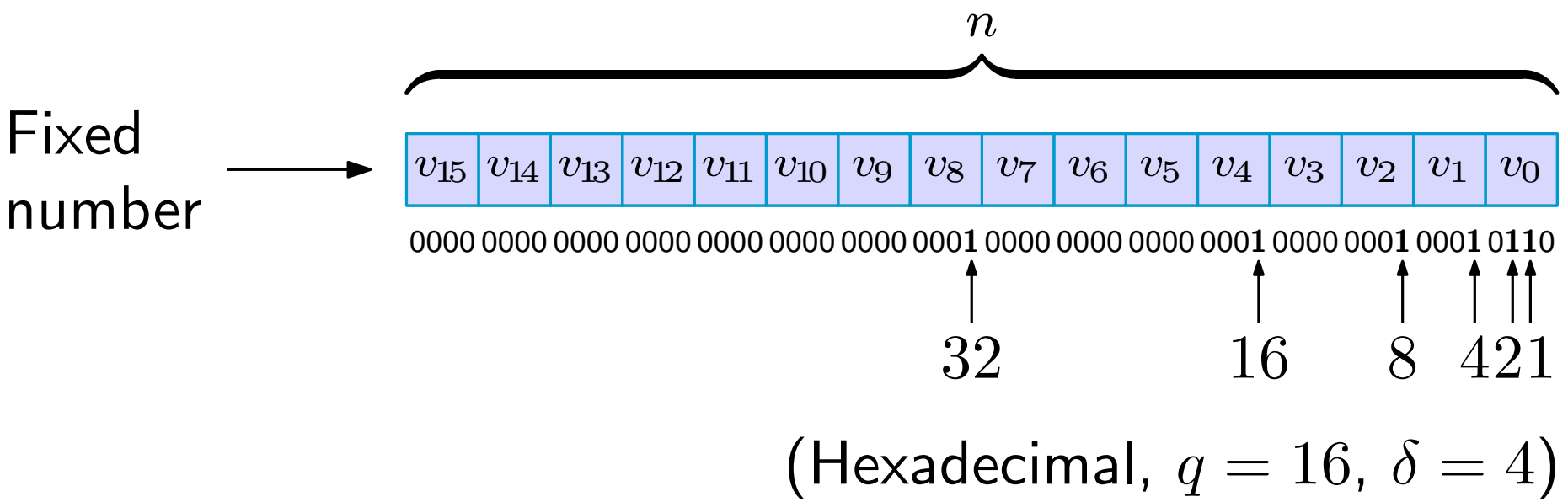
Multiplication



Digits from the set $[q]$ (base q)

$\delta = \log q$ bits per digit

Multiplication

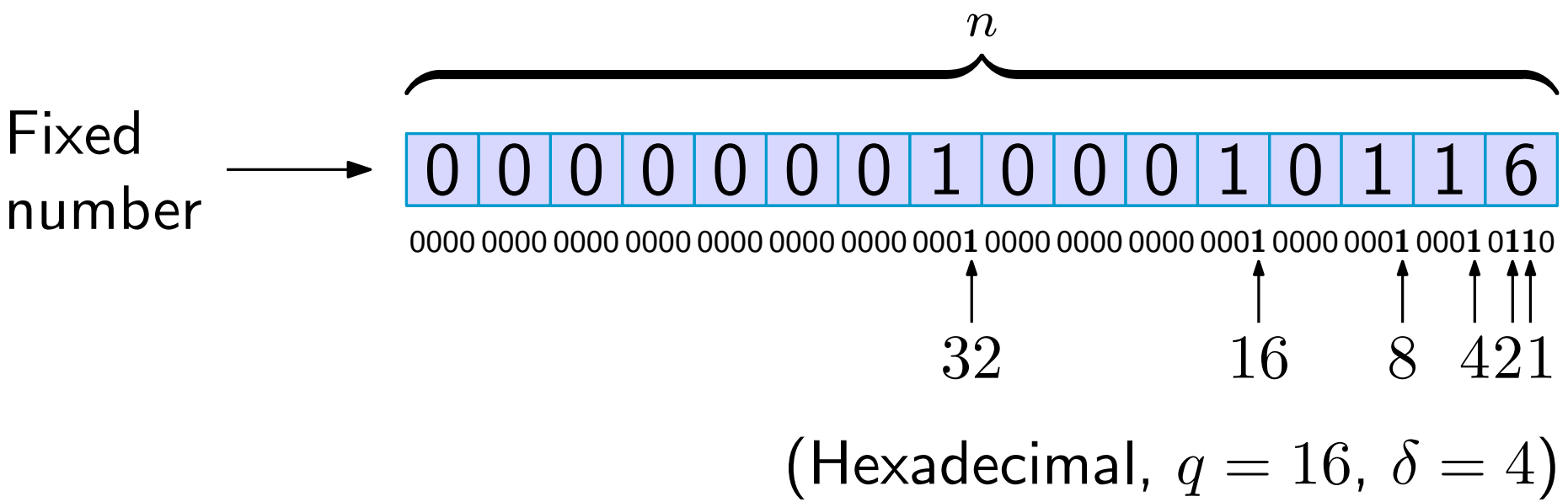


(Hexadecimal, $q = 16$, $\delta = 4$)

Digits from the set $[q]$ (base q)

$\delta = \log q$ bits per digit

Multiplication

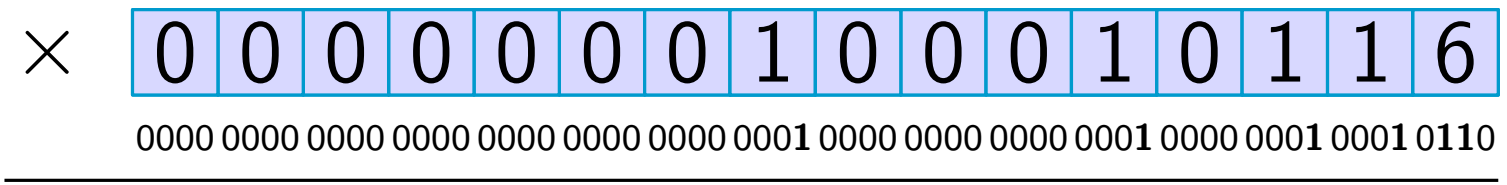


(Hexadecimal, $q = 16$, $\delta = 4$)

Digits from the set $[q]$ (base q)

$\delta = \log q$ bits per digit

Multiplication

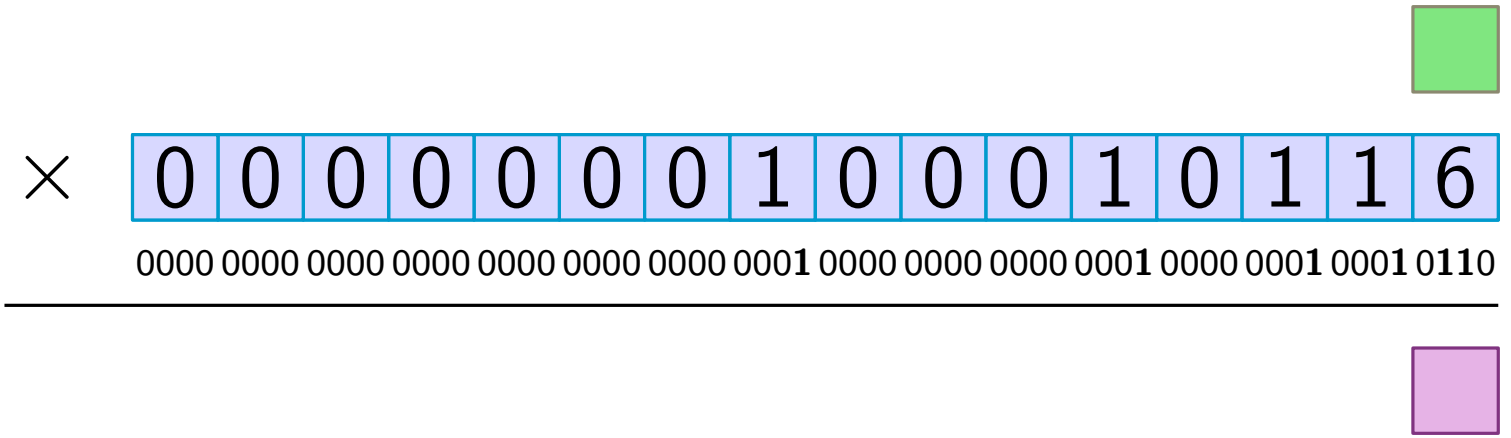


Arriving digit (fixed value)

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

Output digit

Multiplication

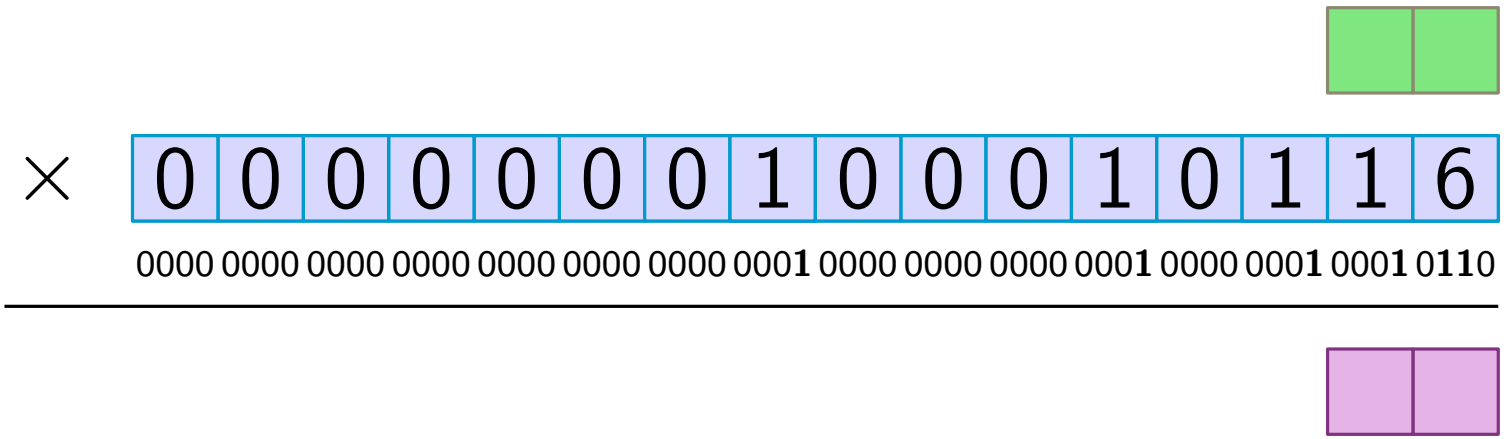


Arriving digit (fixed value)

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

Output digit

Multiplication

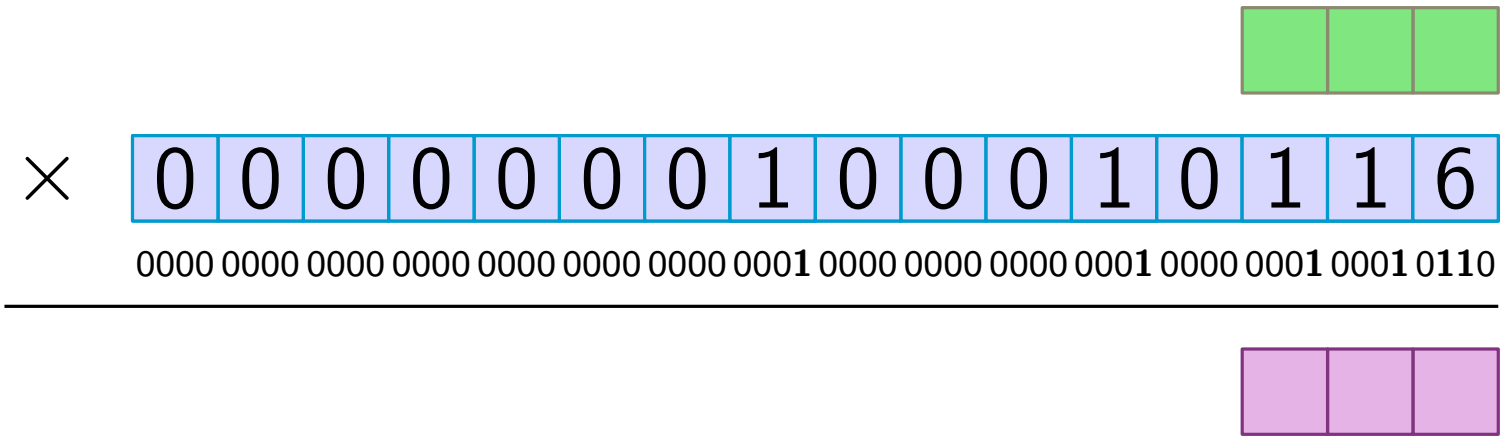


Arriving digit (fixed value)

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

Output digit

Multiplication

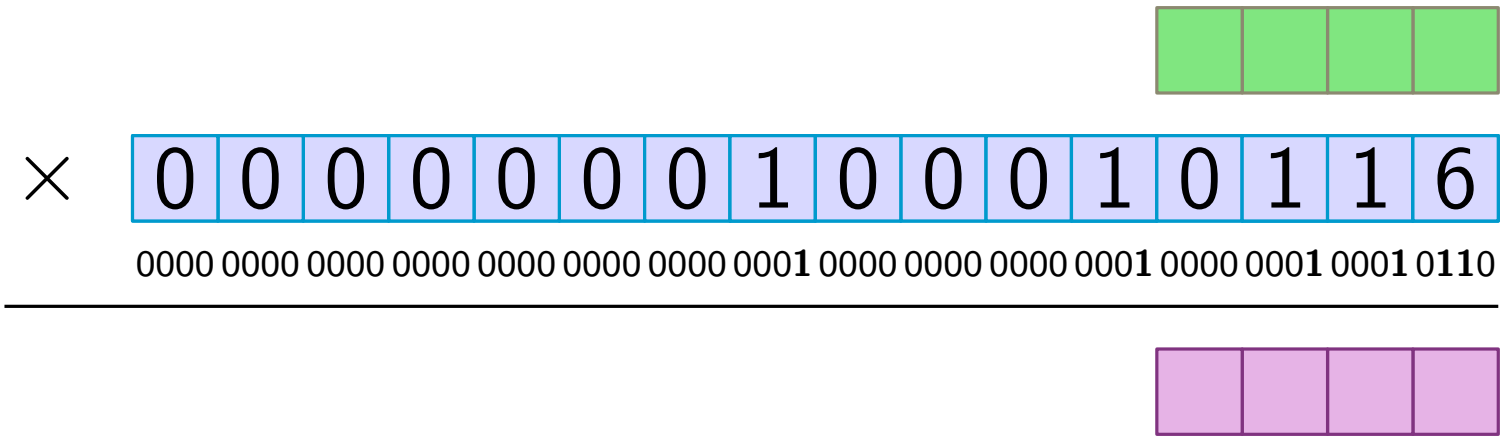


Arriving digit (fixed value)

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

Output digit

Multiplication

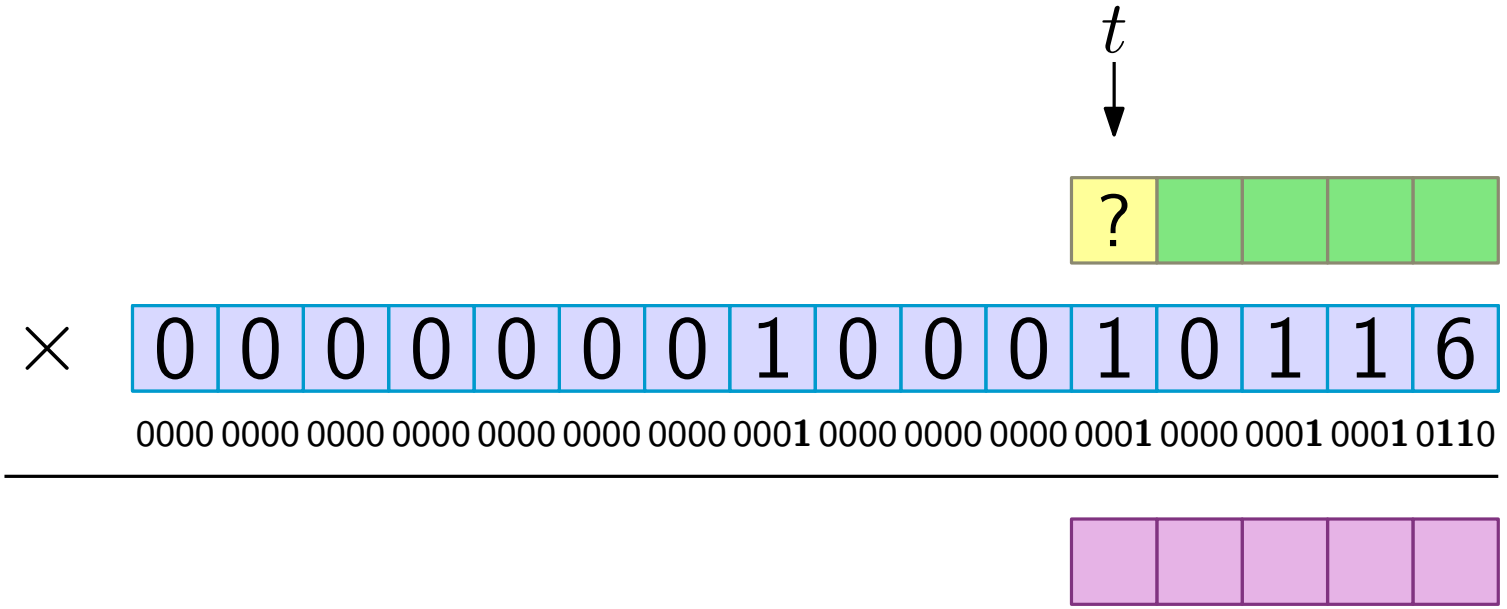


Arriving digit (fixed value)

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

Output digit

Multiplication

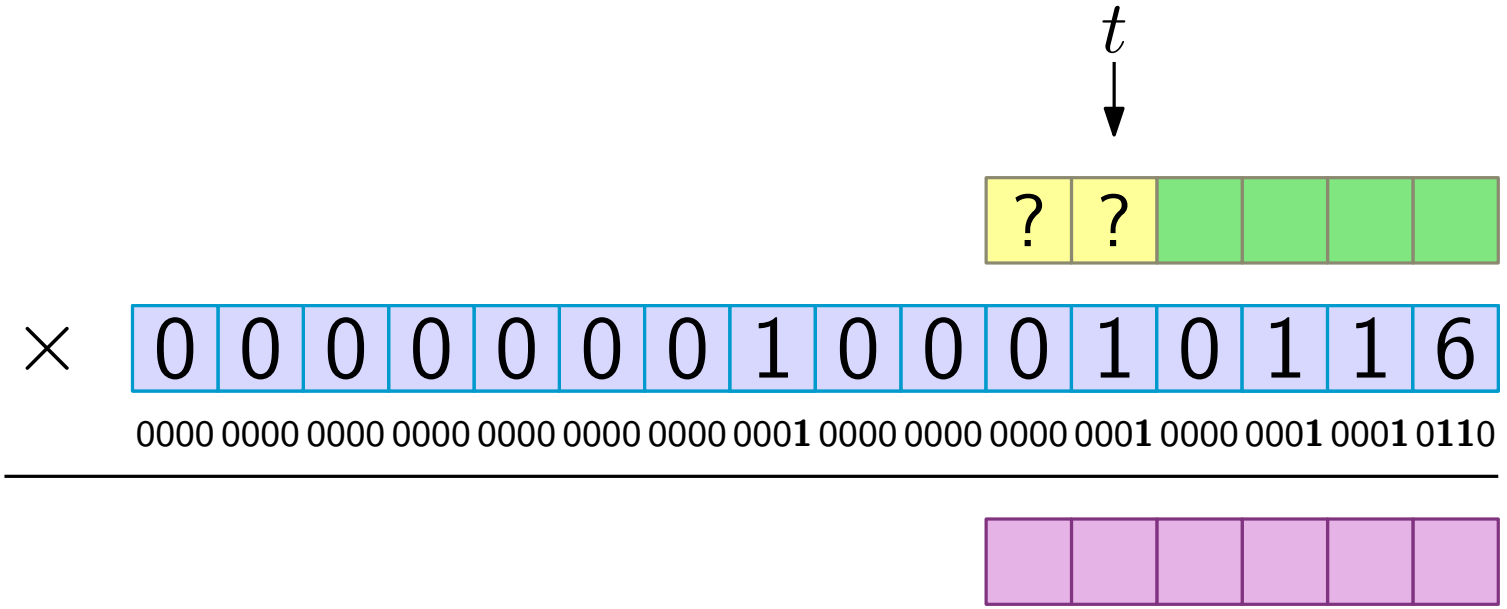


Arriving digit (fixed value)

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

Output digit

Multiplication

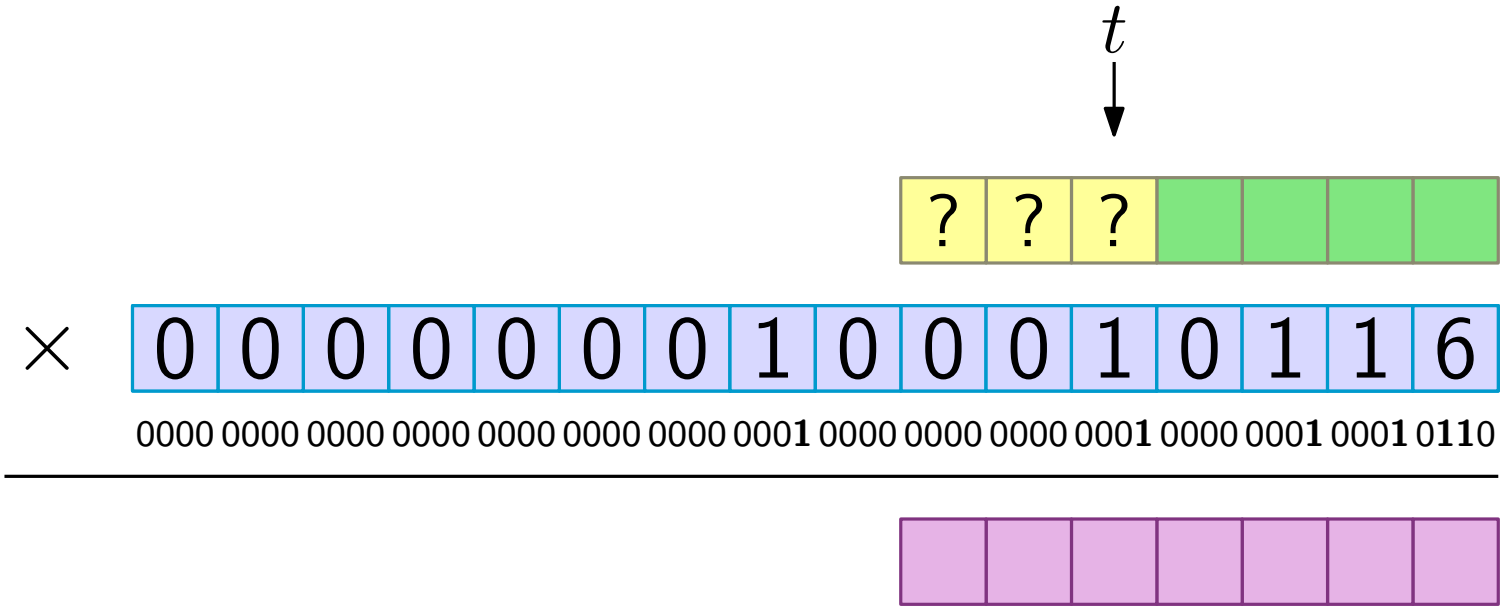


Arriving digit (fixed value)

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

Output digit

Multiplication

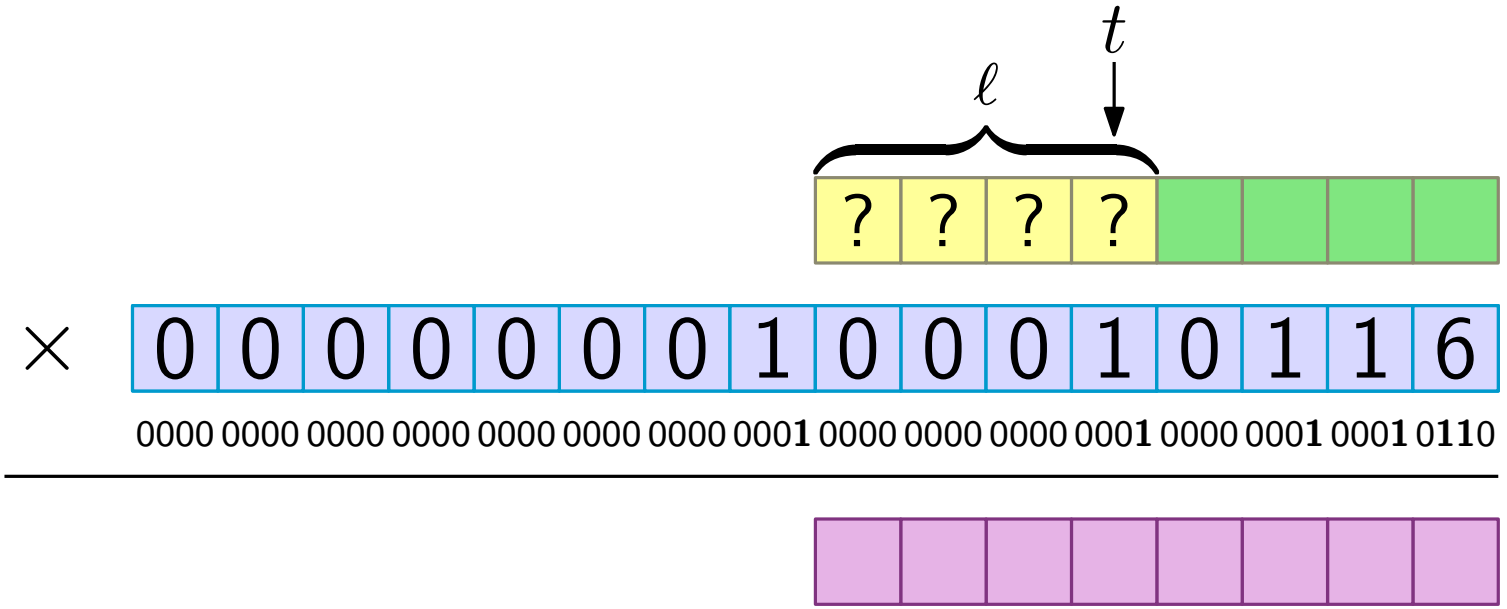


Arriving digit (fixed value)

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

Output digit

Multiplication

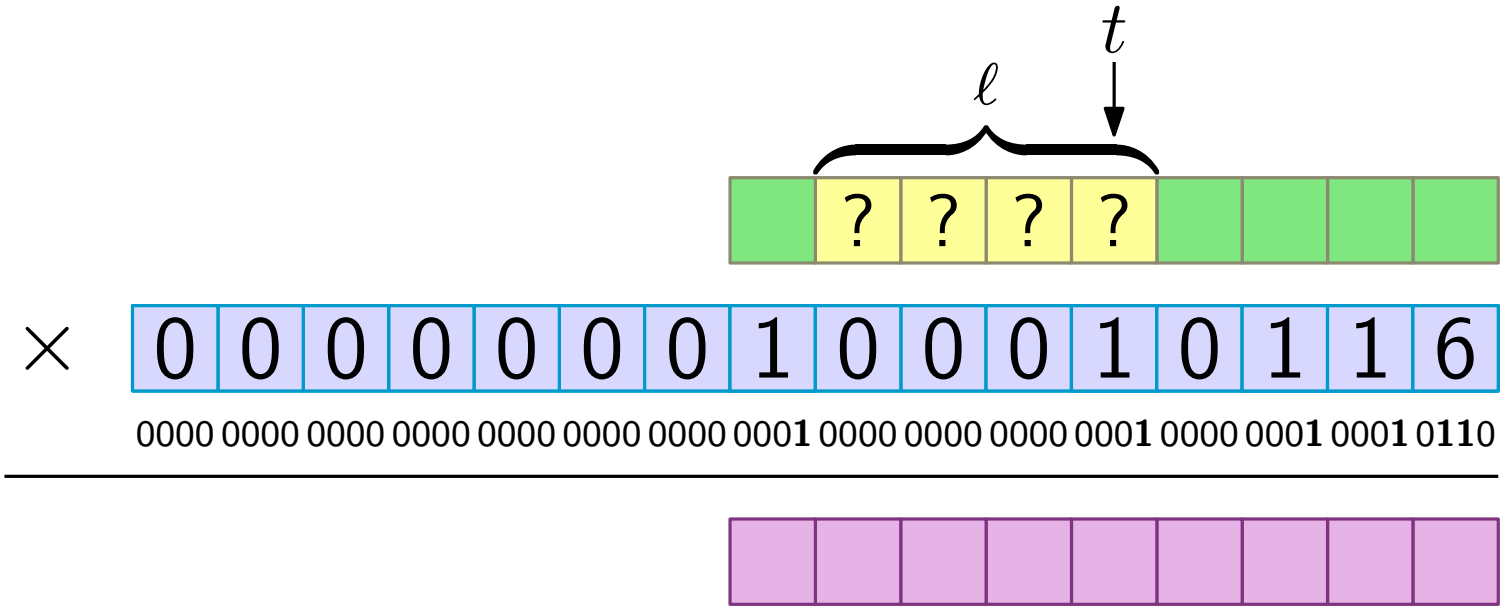


Arriving digit (fixed value)

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

Output digit

Multiplication

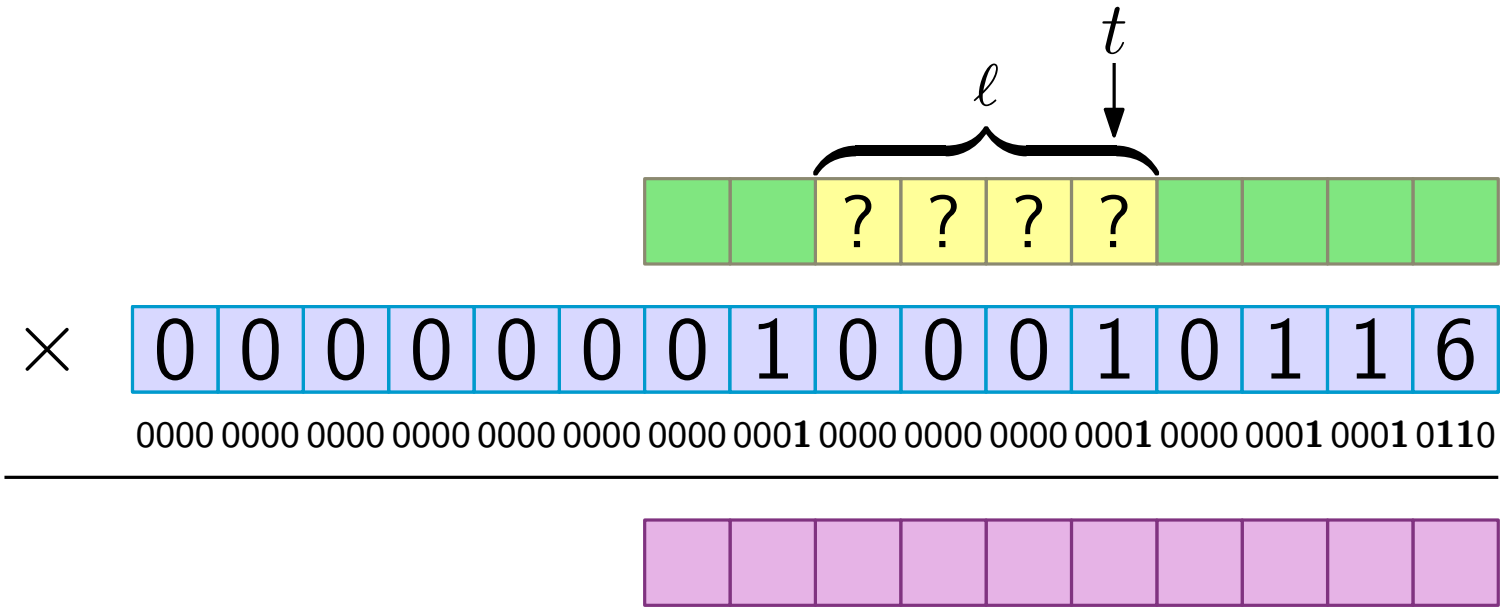


Arriving digit (fixed value)

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

Output digit

Multiplication

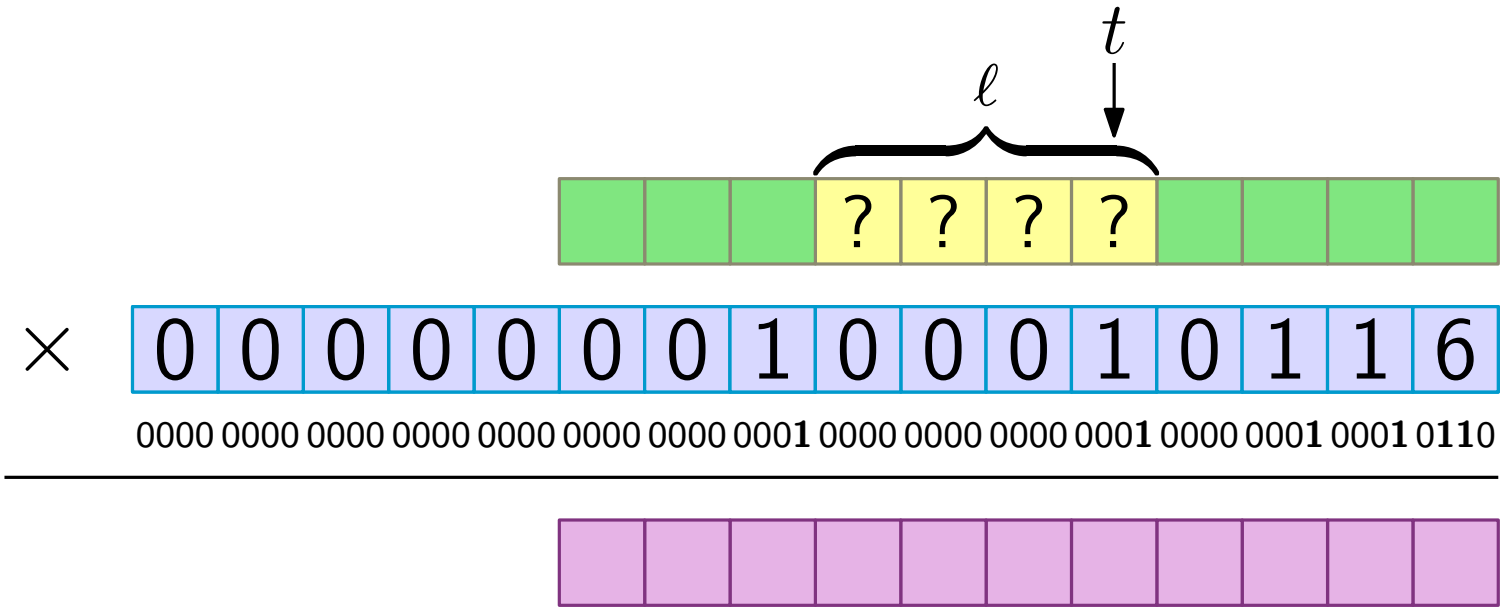


Arriving digit (fixed value)

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

Output digit

Multiplication

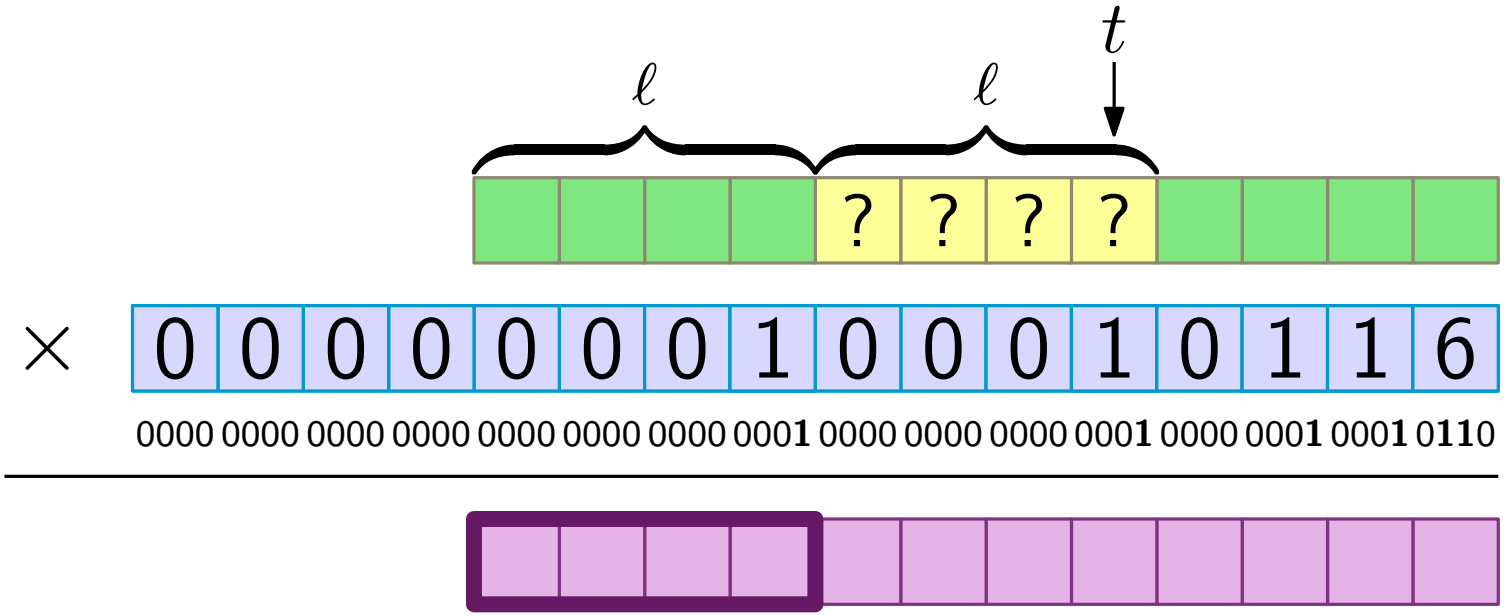


Arriving digit (fixed value)

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

Output digit

Multiplication

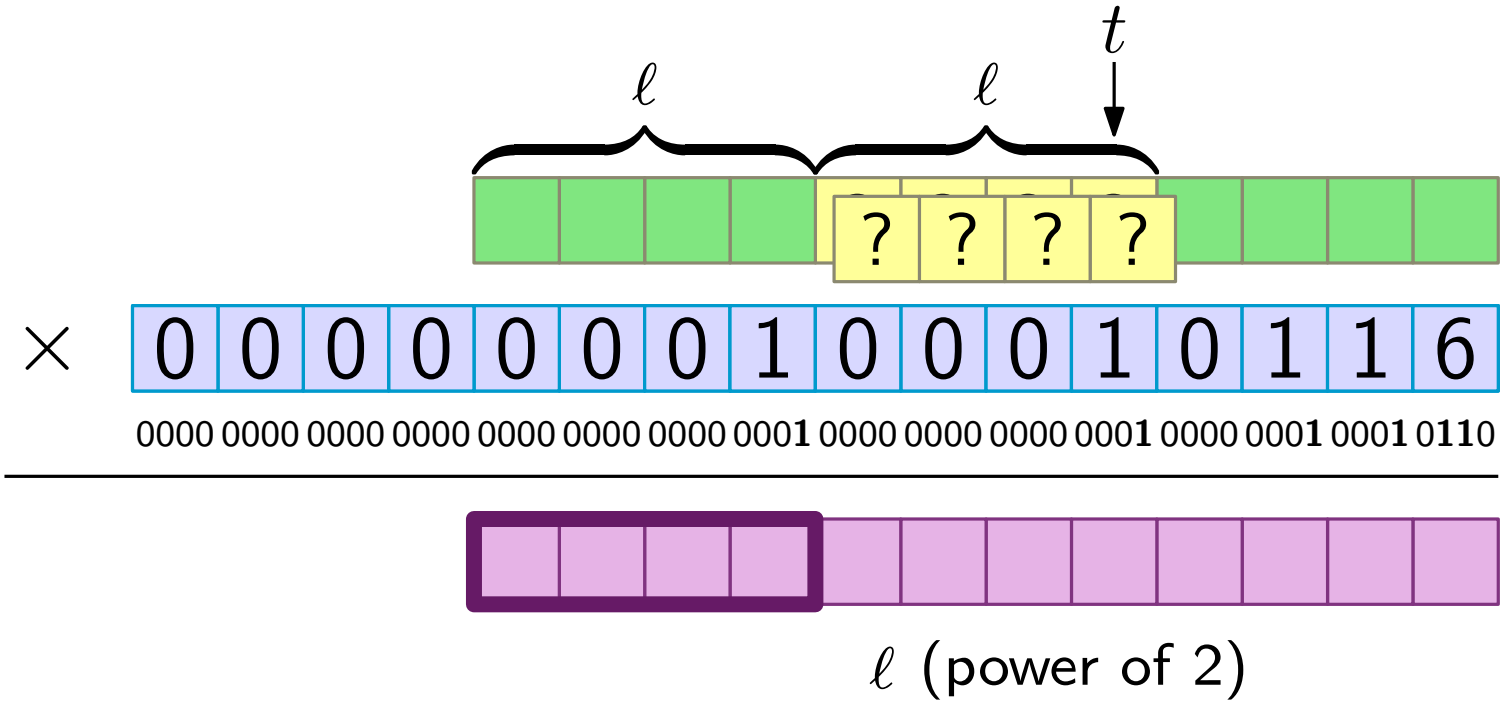


Arriving digit (fixed value)

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

Output digit

Multiplication

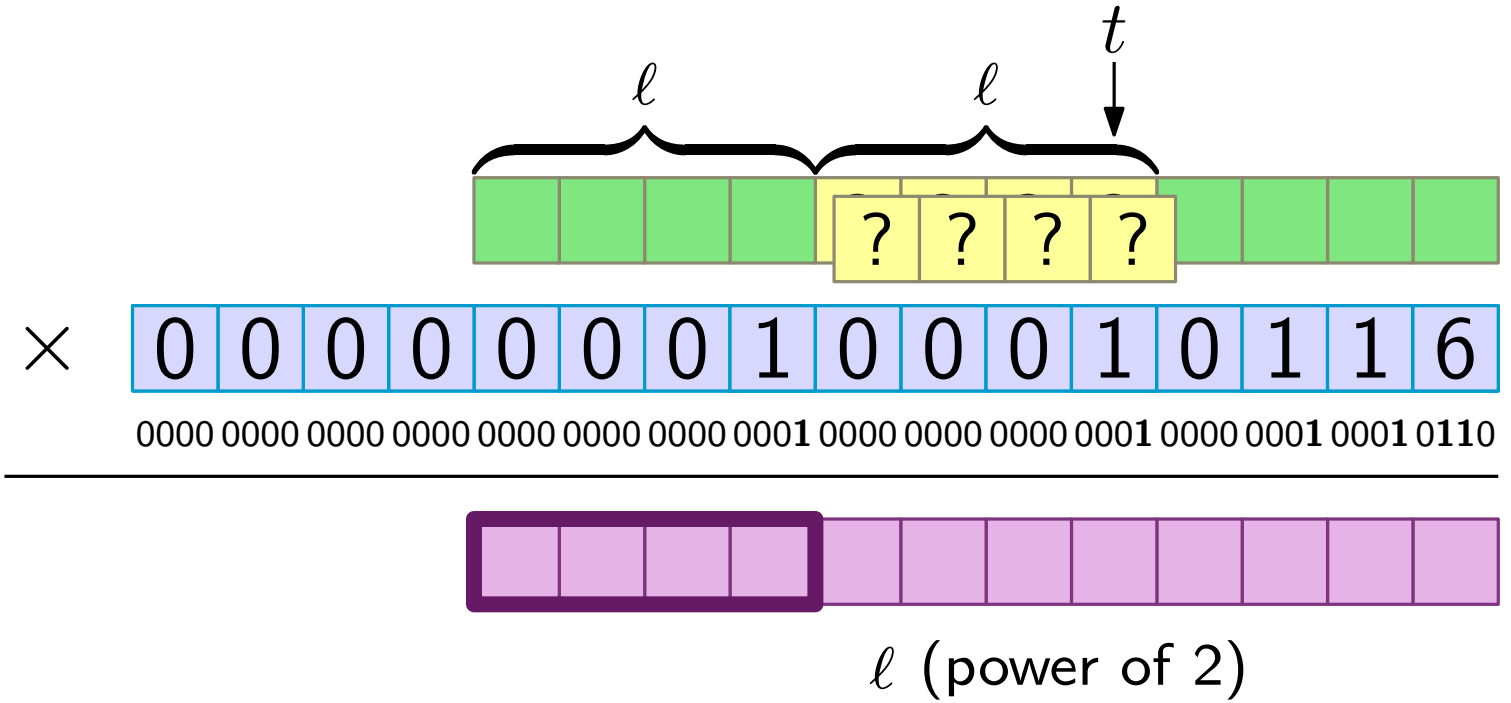


ℓ (power of 2)

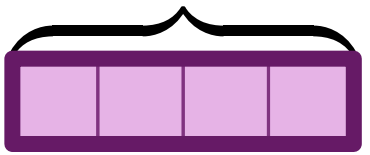
Under any fixed digits and any ,

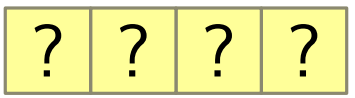
there can be at most **two** possible values of ? ? ? ?

Multiplication

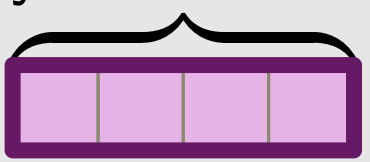


ℓ (power of 2)

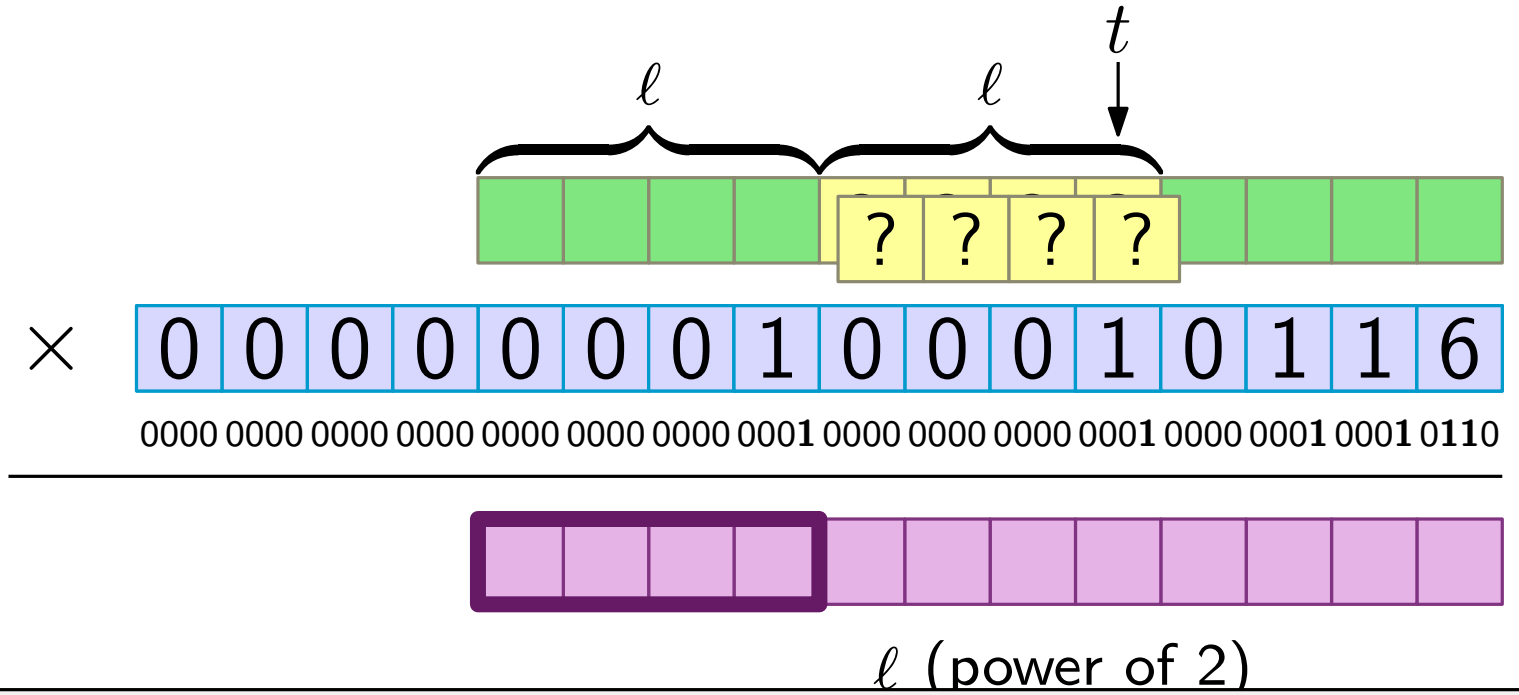
Under any fixed digits  and any  ,

there can be at most **two** possible values of 

The conditional entropy

$H($the outputs during  | all  fixed$) \geqslant \frac{\ell}{2}\delta - \frac{1}{2}$

Multiplication

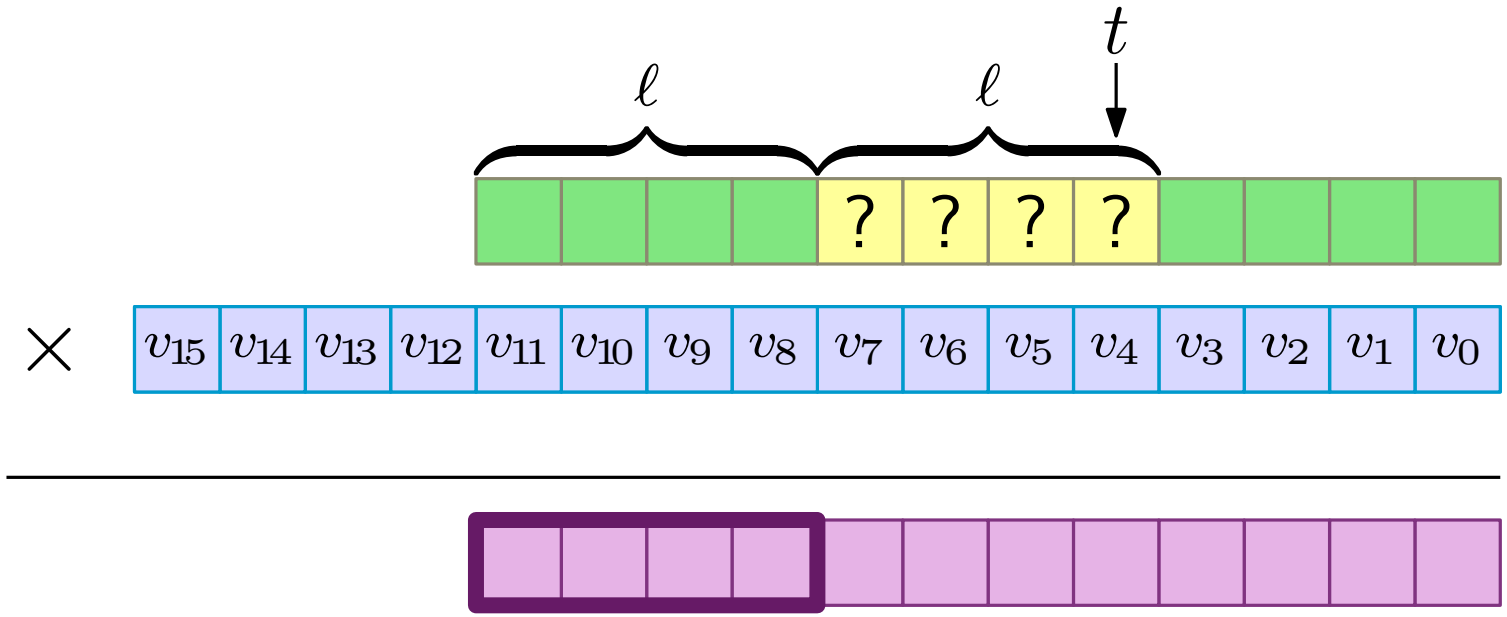


Using the information transfer method, we get the time lower bound: $\Omega\left(\frac{\delta}{w} \cdot n \log n\right)$

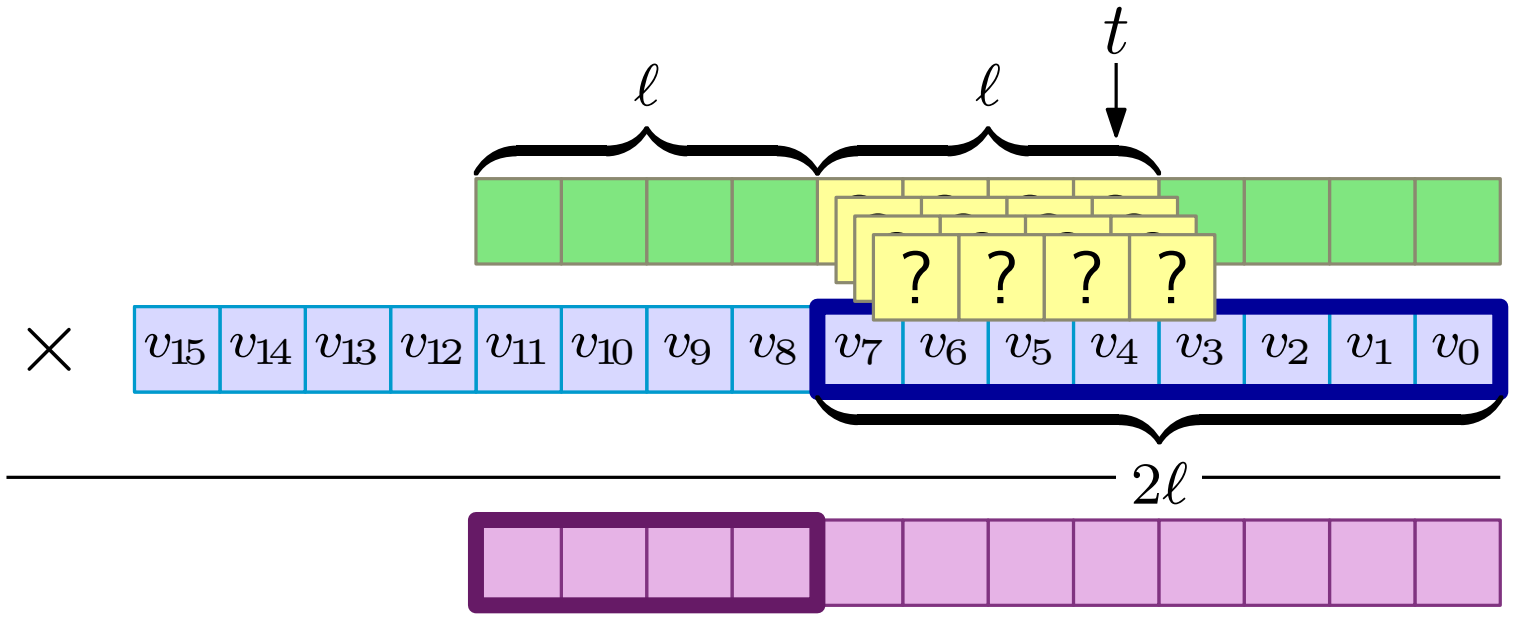
The conditional entropy

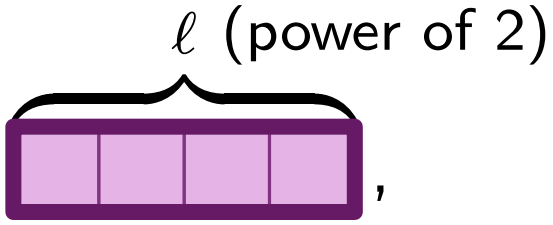
$H($the outputs during $\overbrace{\quad}^{\ell}$ | all ■ fixed$) \geqslant \frac{\ell}{2}\delta - \frac{1}{2}$

Multiplication

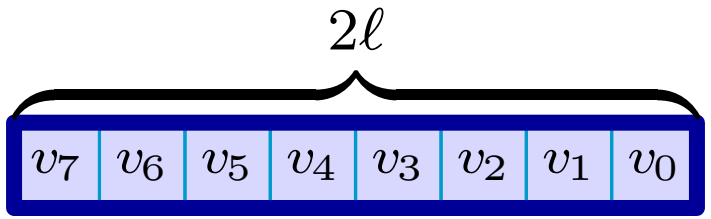


Multiplication

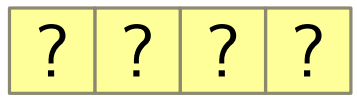




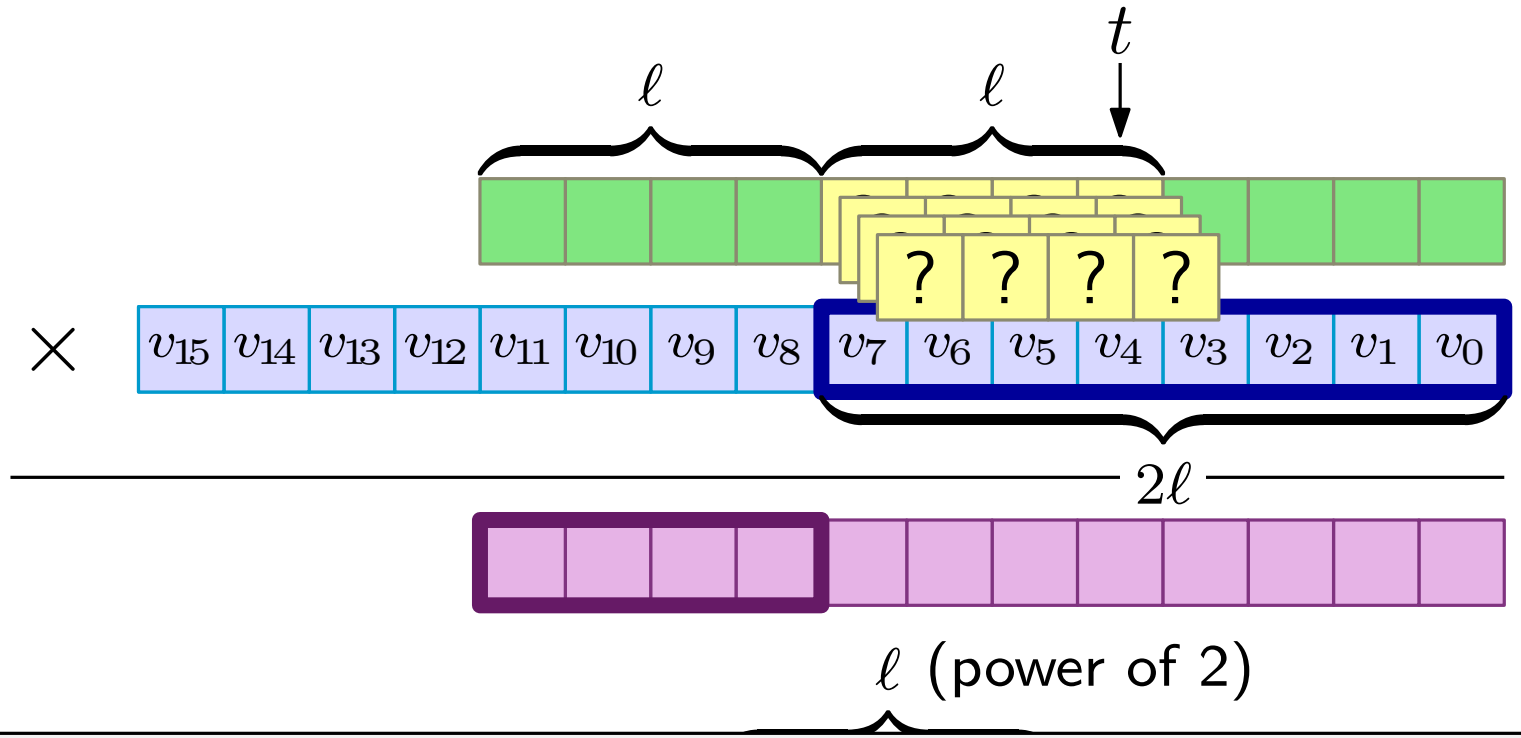
Under any fixed digits  and any ,



for at least half of all possible values

there can be at most **four** possible values of 

Multiplication



For a random fixed number, the time lower bound is also $\Omega\left(\frac{\delta}{w} \cdot n \log n\right)$

there can be at most **four** possible values of | ? | ? | ? | ? |

Thank you!