

Выравнивания. Продолжение.

Алгоритмы в биоинформатике

Мелешко Дмитрий
meleshko.dmitrii@gmail.com

Что было на прошлой лекции?

- Дали определение выравнивания и веса выравнивания.
- Узнали что замены неравноценны.
- Обсудили как устроены матрицы замен BLOSUM и PAM.
- Обсудили что один длинный гэп более вероятен чем много коротких.
- Использовали субаддитивные функции штрафов за гэпы, в частности линейную.

Что будет на этой лекции?

- Обсудим затраты памяти на выравнивание, научимся экономить.
- Поговорим про идеи альтернативных алгоритмов выравнивания.

Алгоритмы выравнивания и память

- Needleman–Wunsch
 $O(n^2)$
- Gotoh's
 $O(n^2)$
- Smith/Waterman
 $O(n^2)$

Алгоритмы выравнивания и память

Найдем выравнивание X и Y хромосом человека.

$$\text{size}(X) = 156040895 \sim 10^8$$

$$\text{size}(Y) = 57227415 \sim 10^7$$

Нам понадобится больше 10^{15} бит

Алгоритм Хиршберга!

Леммы:

- Пусть $D(a, b)$ расстояние выравнивания, тогда $D(a^{-1}, b^{-1}) = D(a, b)$, где x^{-1} последовательность в обратном порядке.

x

Алгоритм Хиршберга!

Леммы:

- Пусть $D(a, b)$ расстояние выравнивания, тогда $D(a^{-1}, b^{-1}) = D(a, b)$, где x^{-1} последовательность в обратном порядке.
- Пусть $D(a, b)$ — расстояние редактирования и $a = a_l + a_r$, где $a_l = a[1..k]$, $a_r = a[k+1..n]$. Тогда такое i , что для $b_l = b[1..i]$, $b_r = b[i+1..m]$ выполняется $D(a, b) = D(a_l, b_l) + D(a_r, b_r)$

Алгоритм Хиршберга!

Пруф:

Дано некоторое разбиение $a = a_l + a_r$.

Допустим что $\nexists i$ такого, для которого выполняется

$$D(a, b) = D(a_l, b_l) + D(a_r, b_r)$$

Алгоритм Хиршберга!

Пруф:

∄ некоторое разбиение $a = a_l + a_r$.

Допустим что $\nexists i$ такого, для которого выполняется

$$D(a, b) = D(a_l, b_l) + D(a_r, b_r)$$

⇒ если мы разобьем a на a_l и a_r , и выровняем a_l с любым из префиксов b , то a_r нельзя будет выровнять ни с одним из суффиксов b , так, чтобы получилось как в выравнивании a, b .

		G	A	T						
	0	1	2	3	6	6	7	7	8	A
A	1	1	1	2	5	5	6	6	7	A
A	2	2	1	2	4	4	5	5	6	G
G	3	2	2	2	3	4	4	4	5	A
A	4	3	2	3	2	3	3	3	4	G
G	5	4	3	3	1	2	2	2	3	T
T	6	5	4	3	2	1	2	1	2	A
A	7	6	5	4	3	2	1	1	1	C
C	8	7	6	5	4	3	2	1	0	
					T	A	C	A		

$$a_l = GAT, \quad a_r = TACA$$

Алгоритм Хиршберга!

Замечание

Допустим что $\exists i$ такое, что выполняется

$$D(a, b) > D(a_l, b_l) + D(a_r, b_r)$$

Но тогда мы бы объединили выравнивание a_l, b_l и a_r, b_r и получили бы более оптимальное выравнивание!

		G	A	T						
	0	1	2	3	6	6	7	7	8	A
A	1	1	1	2	5	5	6	6	7	A
A	2	2	1	2	4	4	5	5	6	G
G	3	2	2	2	3	4	4	4	5	A
A	4	3	2	3	2	3	3	3	4	G
G	5	4	3	3	1	2	2	2	3	T
T	6	5	4	3	2	1	2	1	2	A
A	7	6	5	4	3	2	1	1	1	C
C	8	7	6	5	4	3	2	1	0	
					T	A	C	A		

$$a_l = GAT, \quad a_r = TACA$$

Алгоритм Хиршберга!

Следствие

Получается что если $a = a_l + a_r$, то найдется такой индекс i , что для $b_l = b[..i]$, $b_r = b[i..]$ выполняется $D(a, b) = D(a_l, b_l) + D(a_r, b_r)$ но такого индекса где $D(a, b) > D(a_l, b_l) + D(a_r, b_r)$ точно не найдется.

Значит можно выравнивать суффикс и префикс и искать минимум!

		G	A	T						
	0	1	2	3	6	6	7	7	8	A
A	1	1	1	2	5	5	6	6	7	A
A	2	2	1	2	4	4	5	5	6	G
G	3	2	2	2	3	4	4	4	5	A
A	4	3	2	3	2	3	3	3	4	G
G	5	4	3	3	1	2	2	2	3	T
T	6	5	4	3	2	1	2	1	2	A
A	7	6	5	4	3	2	1	1	1	C
C	8	7	6	5	4	3	2	1	0	
					T	A	C	A		

$$a_l = GAT, \quad a_r = TACA$$

Алгоритм Хиршберга.

1. Разбиваем a пополам, $a = a_{\dots\frac{1}{2}} + a_{\frac{1}{2}\dots}$
 2. Находим расстояние выравнивание от $a_{\dots\frac{1}{2}}$ до всех префиксов b и то же самое для $a_{\frac{1}{2}\dots}^{-1}$ и b^{-1}
 3. Находим разбиение b такое, при котором минимально $D(a_{\dots\frac{1}{2}}, b_l) + D(a_{\frac{1}{2}\dots}, b_r)$, записываем, где это произошло.
 4. Запускаемся рекурсивно на $a_{\dots\frac{1}{2}}, b_l$ и на $a_{\frac{1}{2}\dots}, b_r$
-

Алгоритм Хиршберга. Пример.

$a = AGTACGCA$

$b = TATGC$

удаления и вставки -2

совпадения 2

мутации -1

$H(AGTACGCA, TATGC)$

$NW(AGTA, b)$

		T	A	T	G	C
	0	-2	-4	-6	-8	-10
A	-2	-1	0	-2	-4	-6
G	-4	-3	-2	-1	0	-2
T	-6	-2	-4	0	-2	-1
A	-8	-4	0	-2	-1	-3

$NW(\text{rev}(CGCA), \text{rev}(b))$

		C	G	T	A	T
	0	-2	-4	-6	-8	-10
A	-2	-1	-3	-5	-4	-6
C	-4	0	-2	-4	-6	-5
G	-6	-2	2	0	-2	-4
C	-8	-4	0	1	-1	-3

Алгоритм Хиршберга. Пример.

$H(AGTACGCA, TATGC)$

$a = AGTACGCA$

$b = TATGC$

удаления и вставки -2

совпадения 2

мутации -1

$NW(AGTA, b)$

		T	A	T	G	C
	0	-2	-4	-6	-8	-10
A	-2	-1	0	-2	-4	-6
G	-4	-3	-2	-1	0	-2
T	-6	-2	-4	0	-2	-1
A	-8	-4	0	-2	-1	-3

$NW(\text{rev}(CGCA), \text{rev}(b))$

		C	G	T	A	T
	0	-2	-4	-6	-8	-10
A	-2	-1	-3	-5	-4	-6
C	-4	0	-2	-4	-6	-5
G	-6	-2	2	0	-2	-4
C	-8	-4	0	1	-1	-3

Алгоритм Хиршберга. Пример.

$a = AGTACGCA$

$b = TATGC$

удаления и вставки -2

совпадения 2

мутации -1

$H(AGTACGCA, TATGC)$

$NW(AGTA, b)$

		T	A	T	G	C
	0	-2	-4	-6	-8	-10
A	-2	-1	0	-2	-4	-6
G	-4	-3	-2	-1	0	-2
T	-6	-2	-4	0	-2	-1
A	-8	-4	0	-2	-1	-3

$NW(\text{rev}(CGCA), \text{rev}(b))$

		C	G	T	A	T
	0	-2	-4	-6	-8	-10
A	-2	-1	-3	-5	-4	-6
C	-4	0	-2	-4	-6	-5
G	-6	-2	2	0	-2	-4
C	-8	-4	0	1	-1	-3

ScoreL = $[-8, -4, 0, -2, -1, -3]$
 rev(ScoreR) = $[-3, -1, 1, 0, -4, -8]$
 Sum = $[-11, -5, 1, -2, -5, -11]$

Алгоритм Хиршберга. Пример.

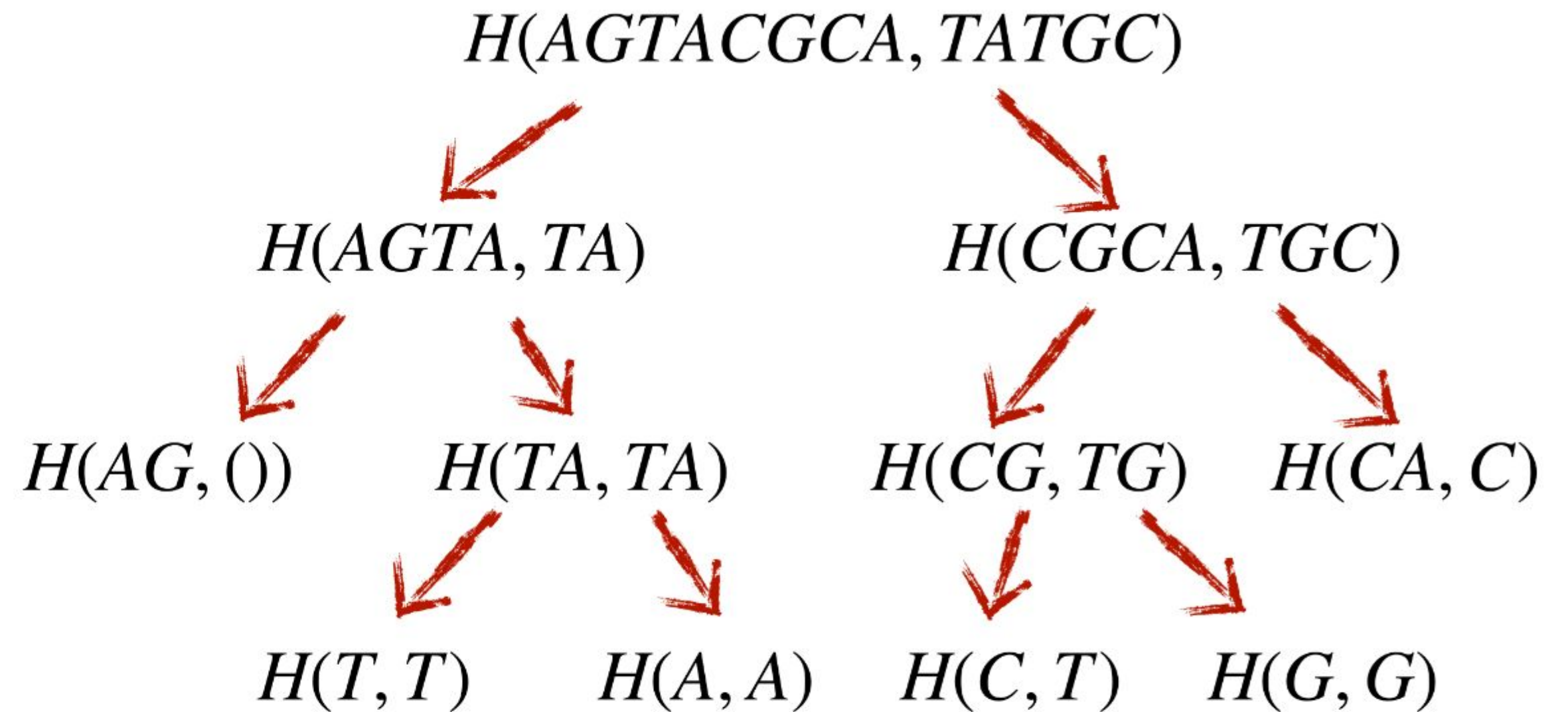
$a = AGTACGCA$

$b = TATGC$

удаления и вставки -2

совпадения 2

мутации -1



Алгоритм Хиршберга.

Сложность.

- Сумма количества операций на каждом шаге

Алгоритм Хиршберга. Сложность.

- Сумма количества операций на каждом шаге

$nm +$

$\frac{n}{2}(m - i) + \frac{n}{2}(i) +$

$\frac{n}{4}(m - i - j) + \frac{n}{4}(j) + \frac{n}{4}(i - k) + \frac{n}{4}(k) +$

$\dots =$

Алгоритм Хиршберга. Сложность.

- Сумма количества операций на каждом шаге

$$nm +$$

$$\frac{n}{2}(m - i) + \frac{n}{2}(i) +$$

$$\frac{n}{4}(m - i - j) + \frac{n}{4}(j) + \frac{n}{4}(i - k) + \frac{n}{4}(k) +$$

$$\dots =$$

$$= nm + \frac{n}{2}m + \frac{n}{4}m + \dots = nm \sum_{t=0}^{\log_2(n)} \frac{1}{2^t}$$

Алгоритм Хиршберга. Сложность.

- Сумма количества операций на каждом шаге

$$nm +$$

$$\frac{n}{2}(m - i) + \frac{n}{2}(i) +$$

$$\frac{n}{4}(m - i - j) + \frac{n}{4}(j) + \frac{n}{4}(i - k) + \frac{n}{4}(k) +$$

$$\dots =$$

$$= nm + \frac{n}{2}m + \frac{n}{4}m + \dots = nm \sum_{t=0}^{\log_2(n)} \frac{1}{2^t}$$

Алгоритм Хиршберга. Сложность.

- Сумма количества операций на каждом шаге

$$nm +$$

$$\frac{n}{2}(m - i) + \frac{n}{2}(i) +$$

$$\frac{n}{4}(m - i - j) + \frac{n}{4}(j) + \frac{n}{4}(i - k) + \frac{n}{4}(k) +$$

$$\dots =$$

$$= nm + \frac{n}{2}m + \frac{n}{4}m + \dots = nm \sum_{t=0}^{\log_2(n)} \frac{1}{2^t} \leq 2nm \Rightarrow O(nm)$$

Алгоритм Хиршберга. Оценка.

1. По памяти

$$O(n)$$

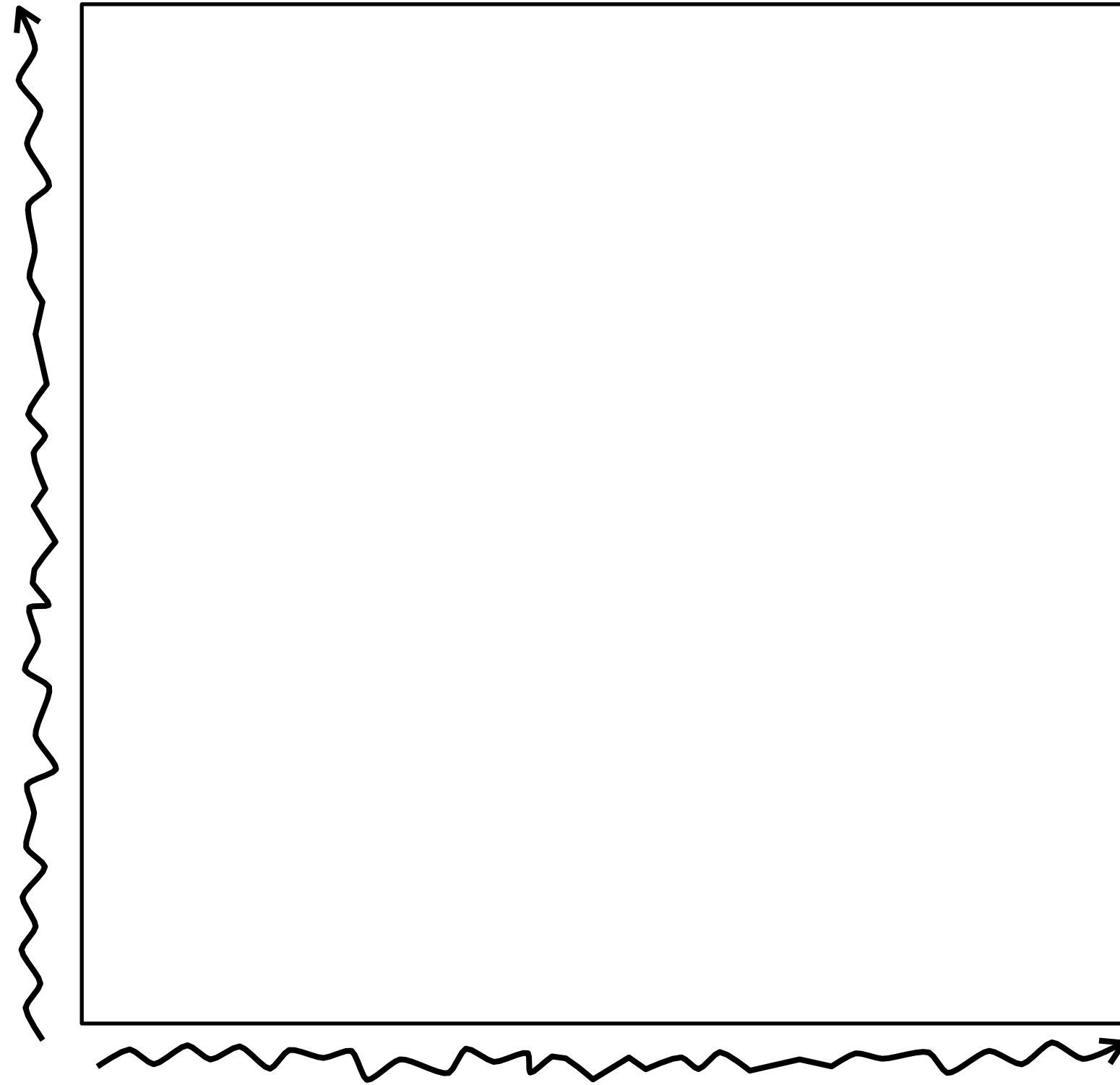
2. По времени

$$O(n^2)$$

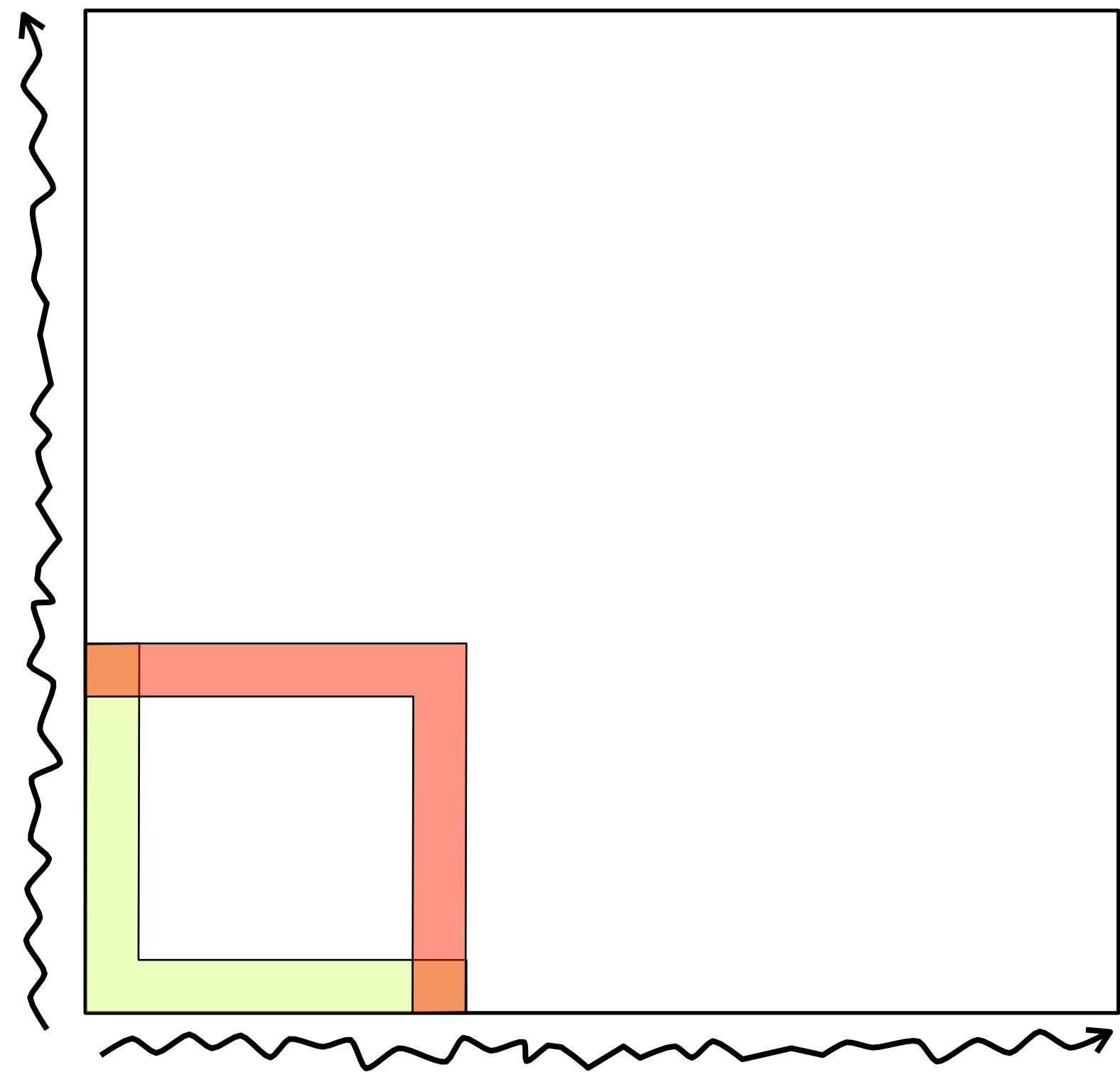
3. Используется асимптотически меньше памяти, а скорость хуже только на константу!

Метод четырех русских

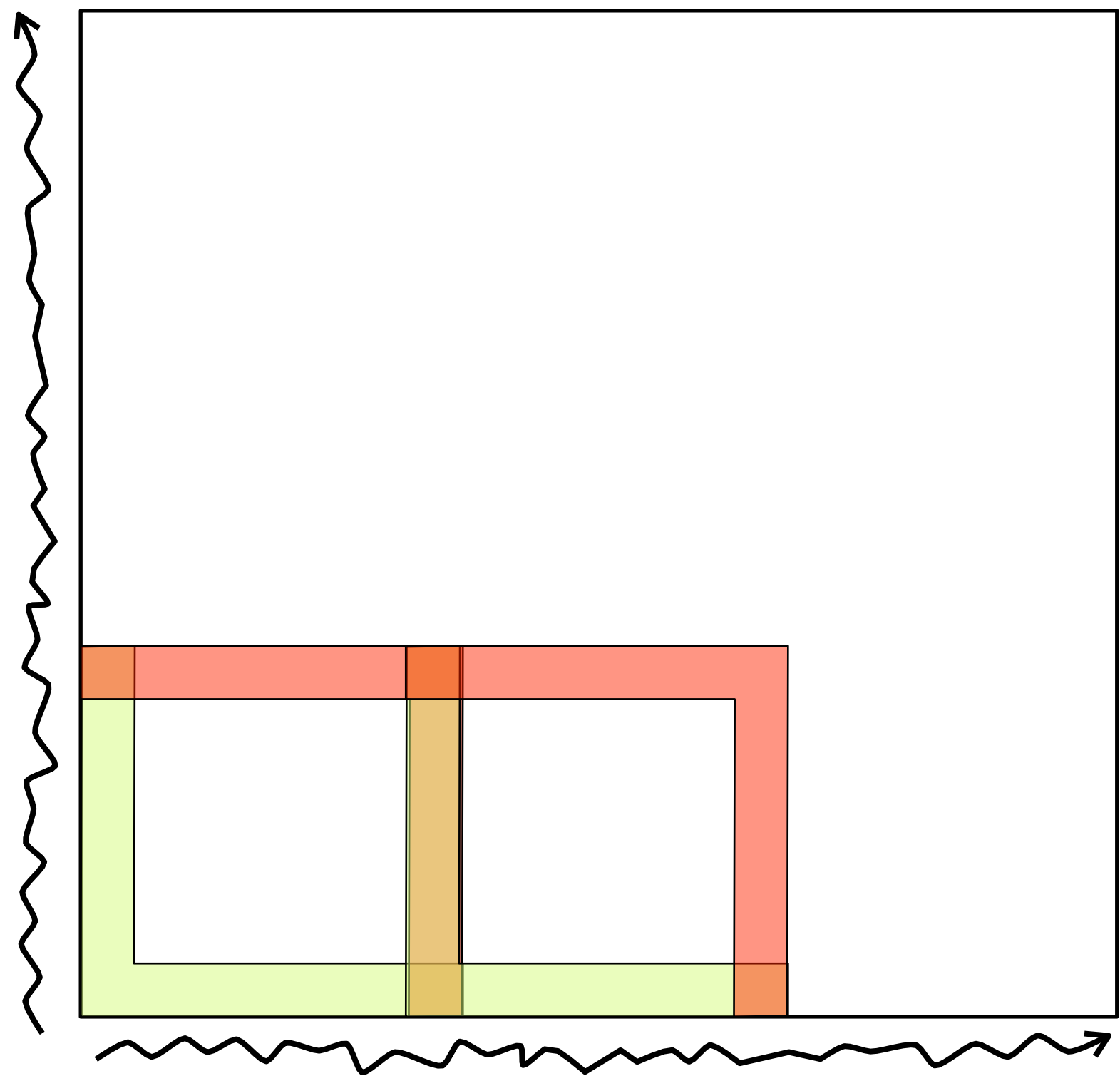
Метод четырех русских



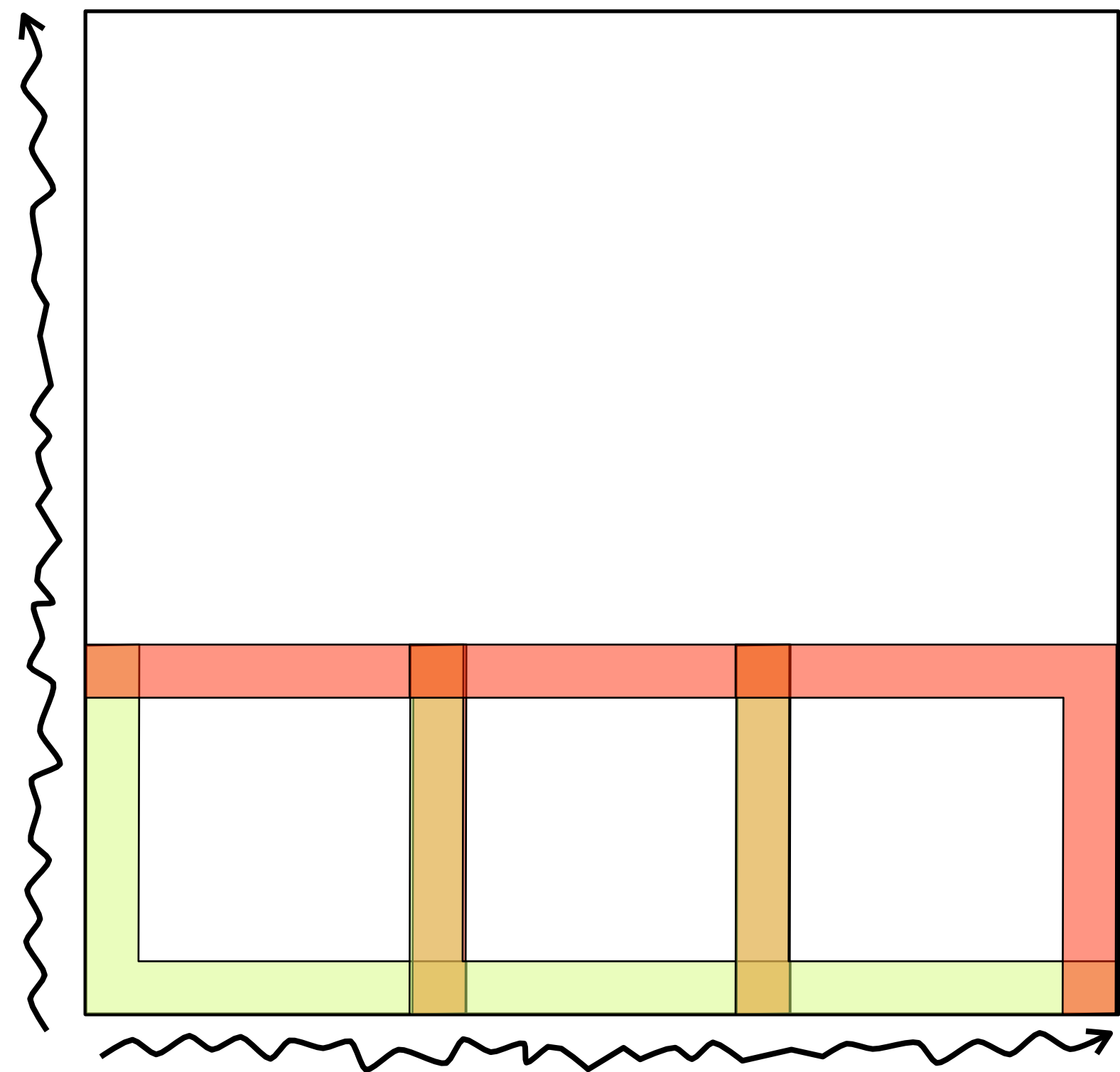
Метод четырех русских



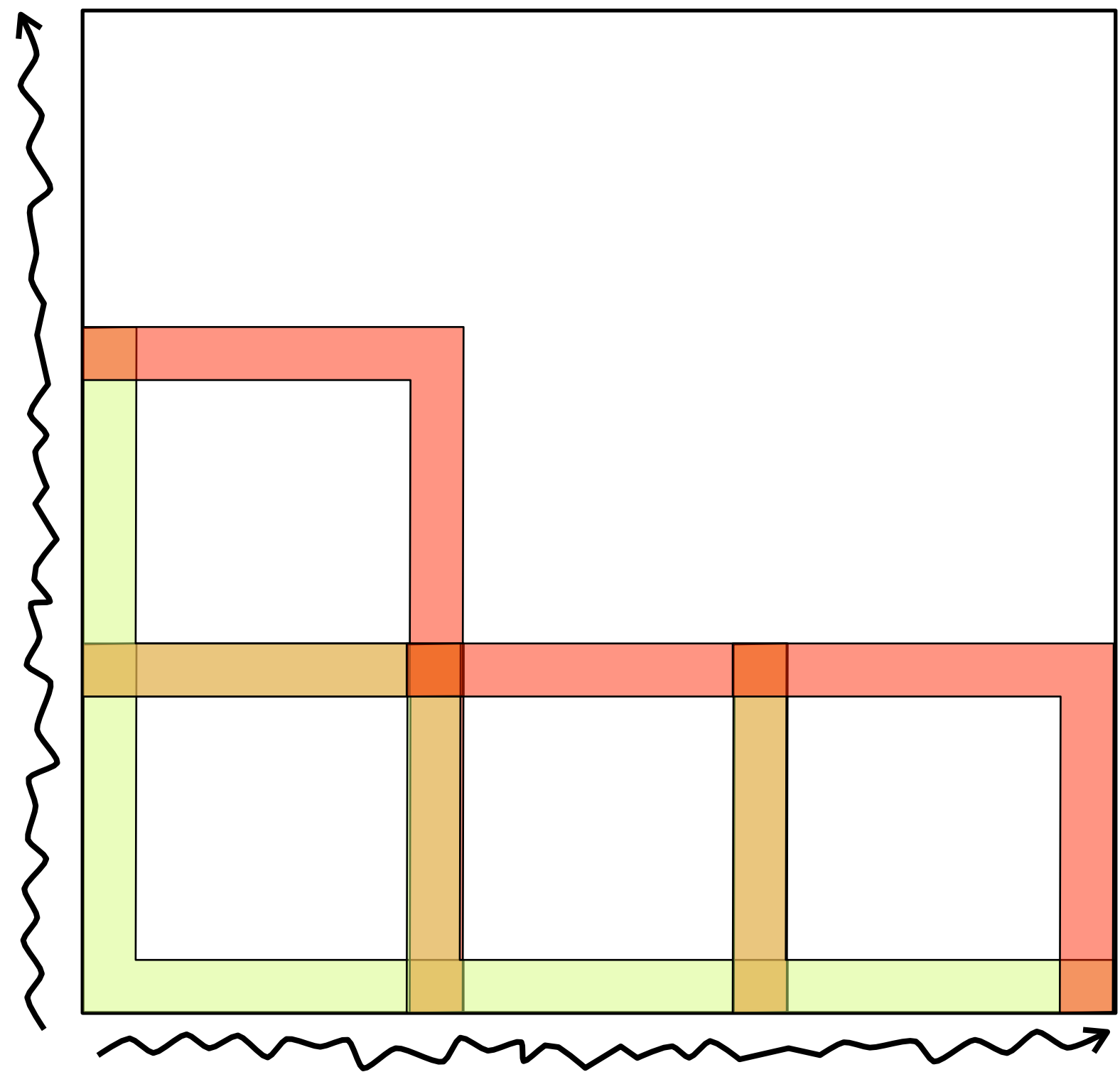
Метод четырех русских



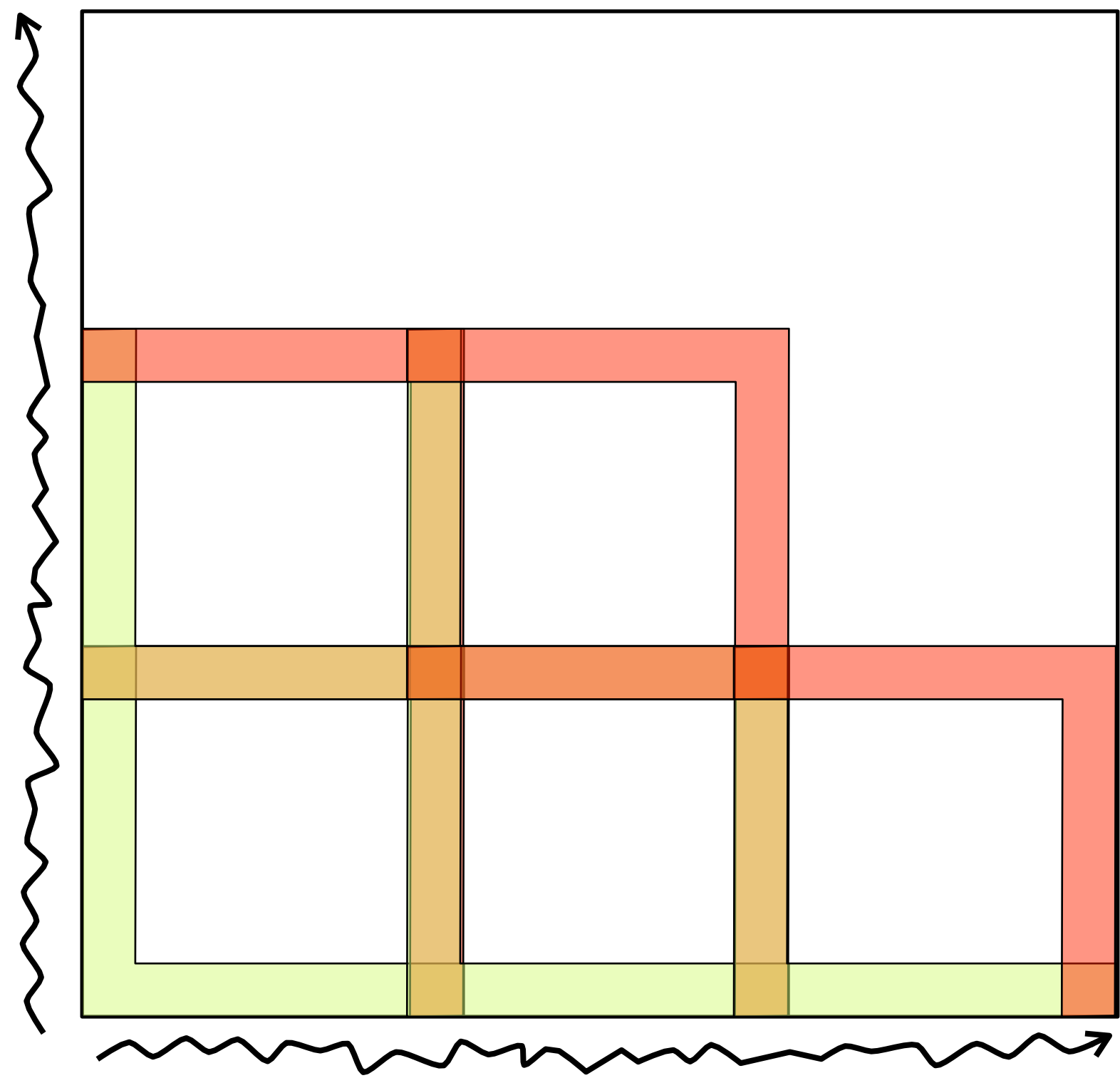
Метод четырех русских



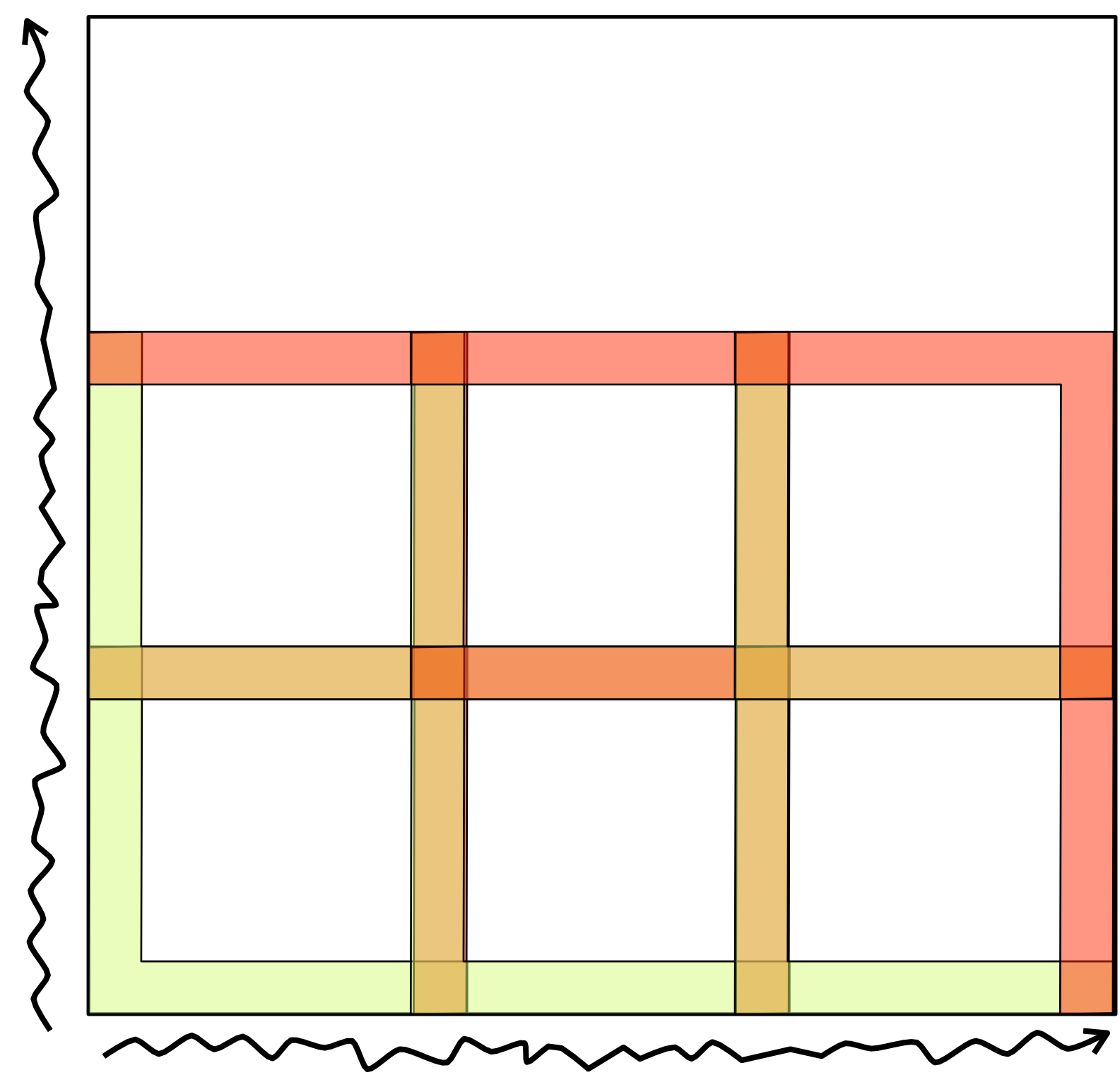
Метод четырех русских



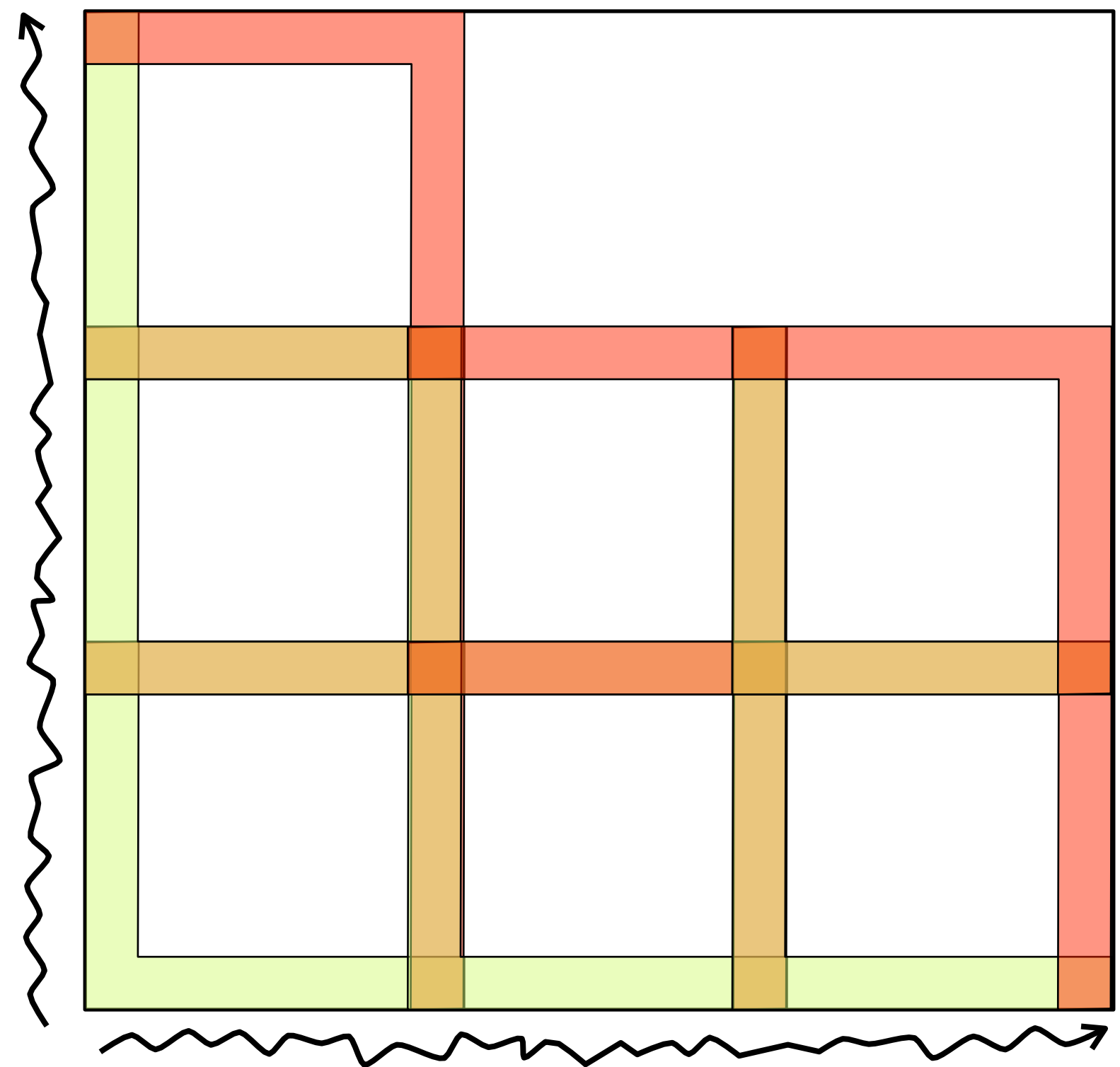
Метод четырех русских



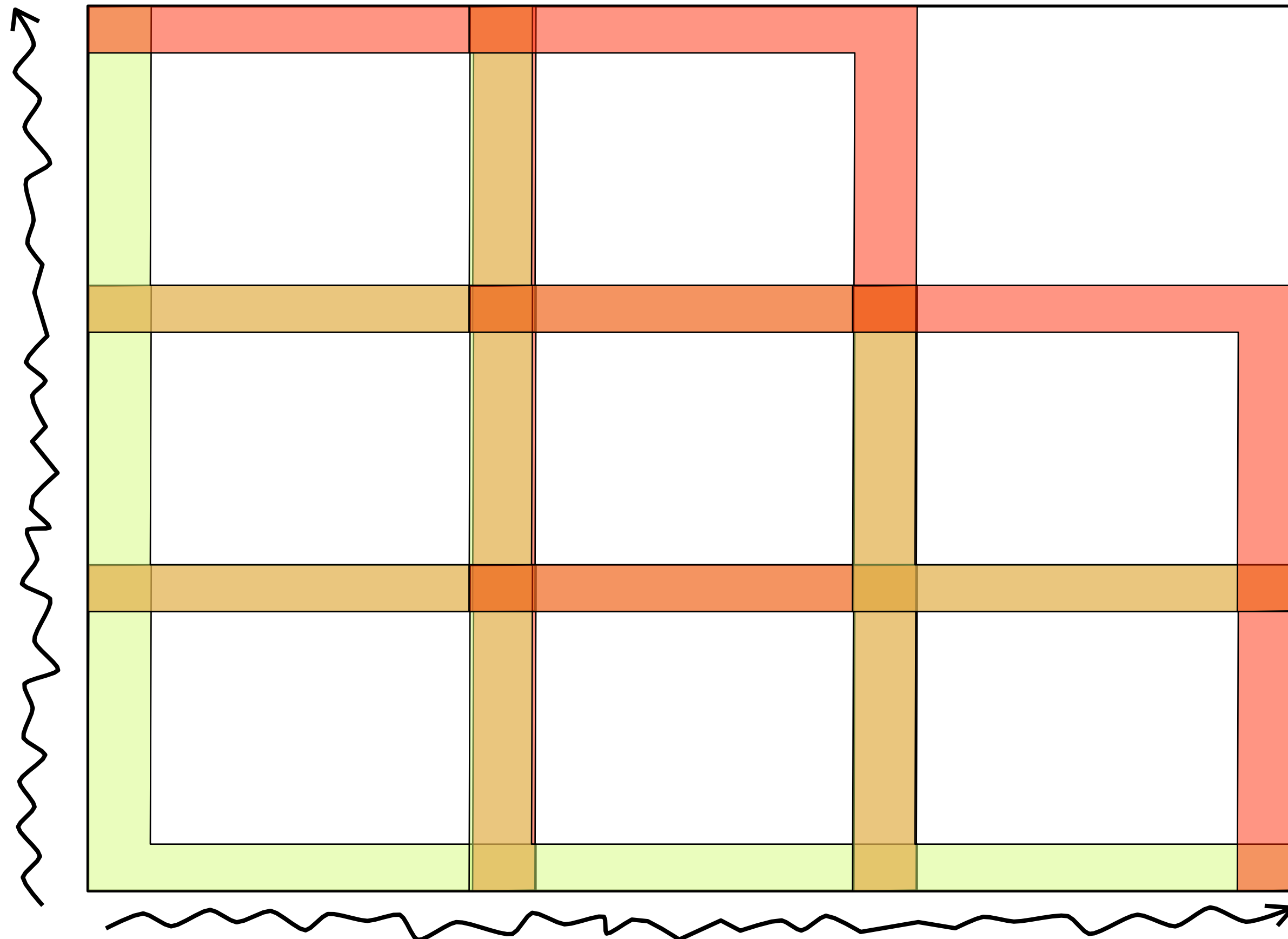
Метод четырех русских



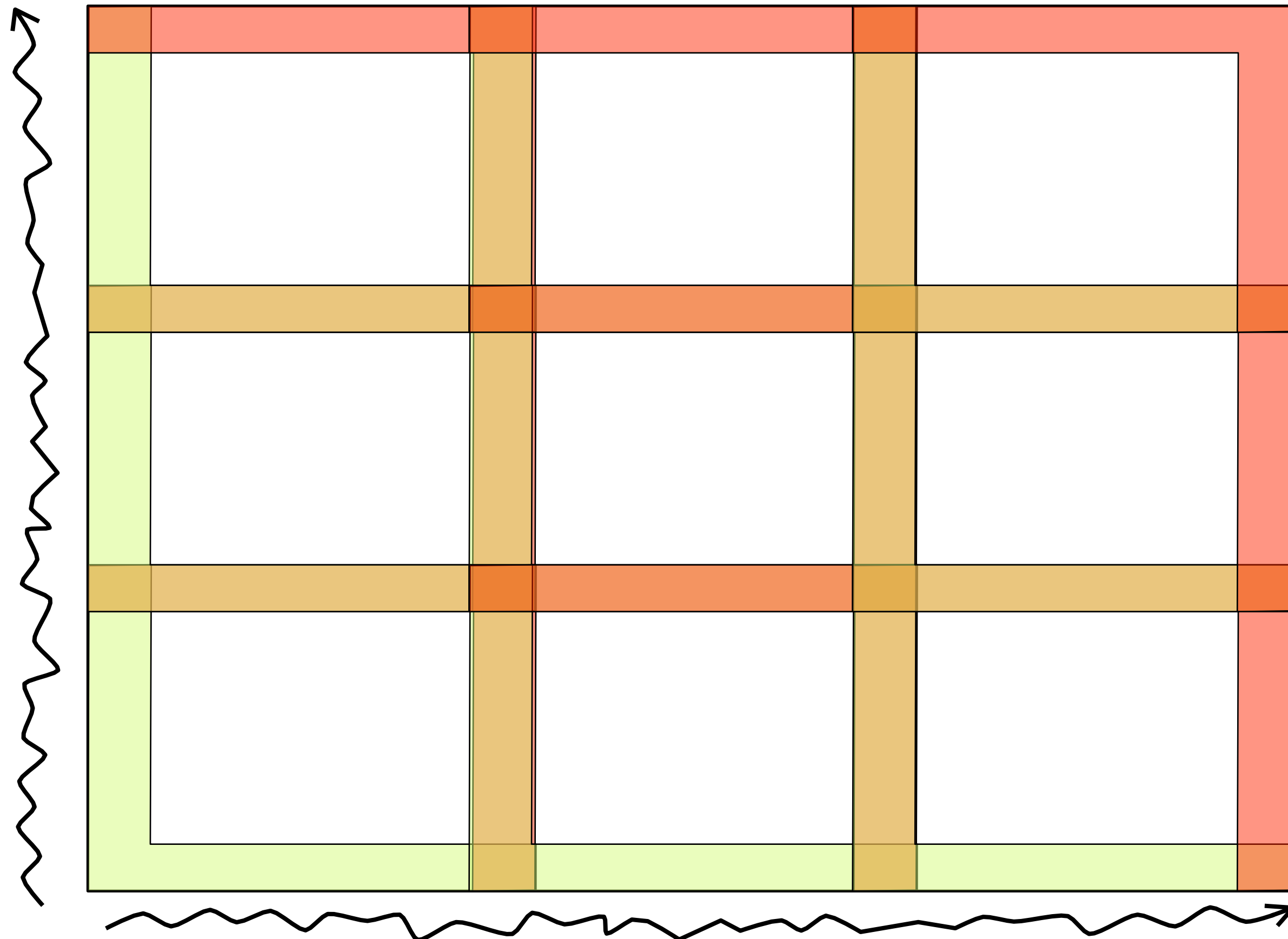
Метод четырех русских



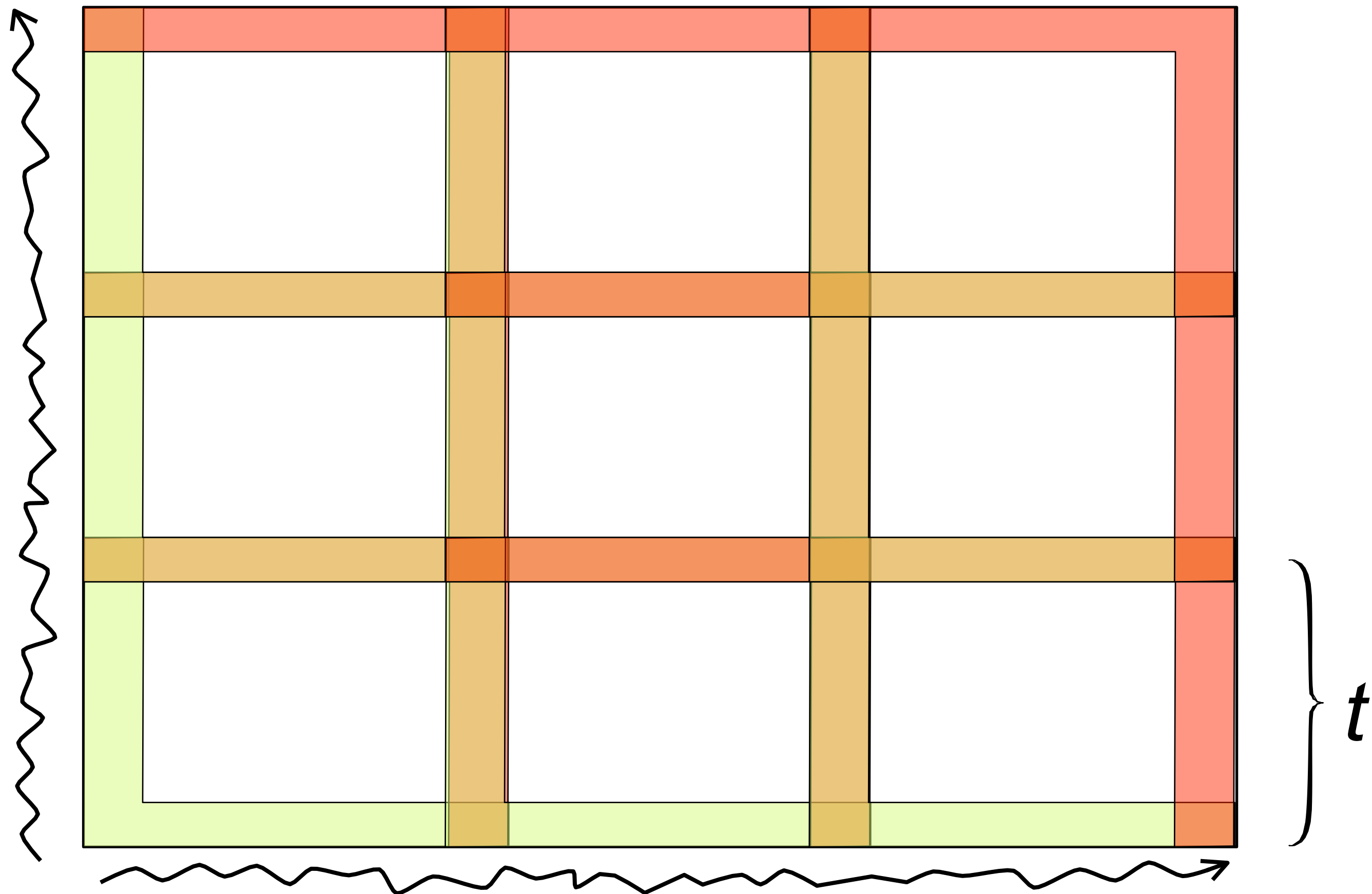
Метод четырех русских



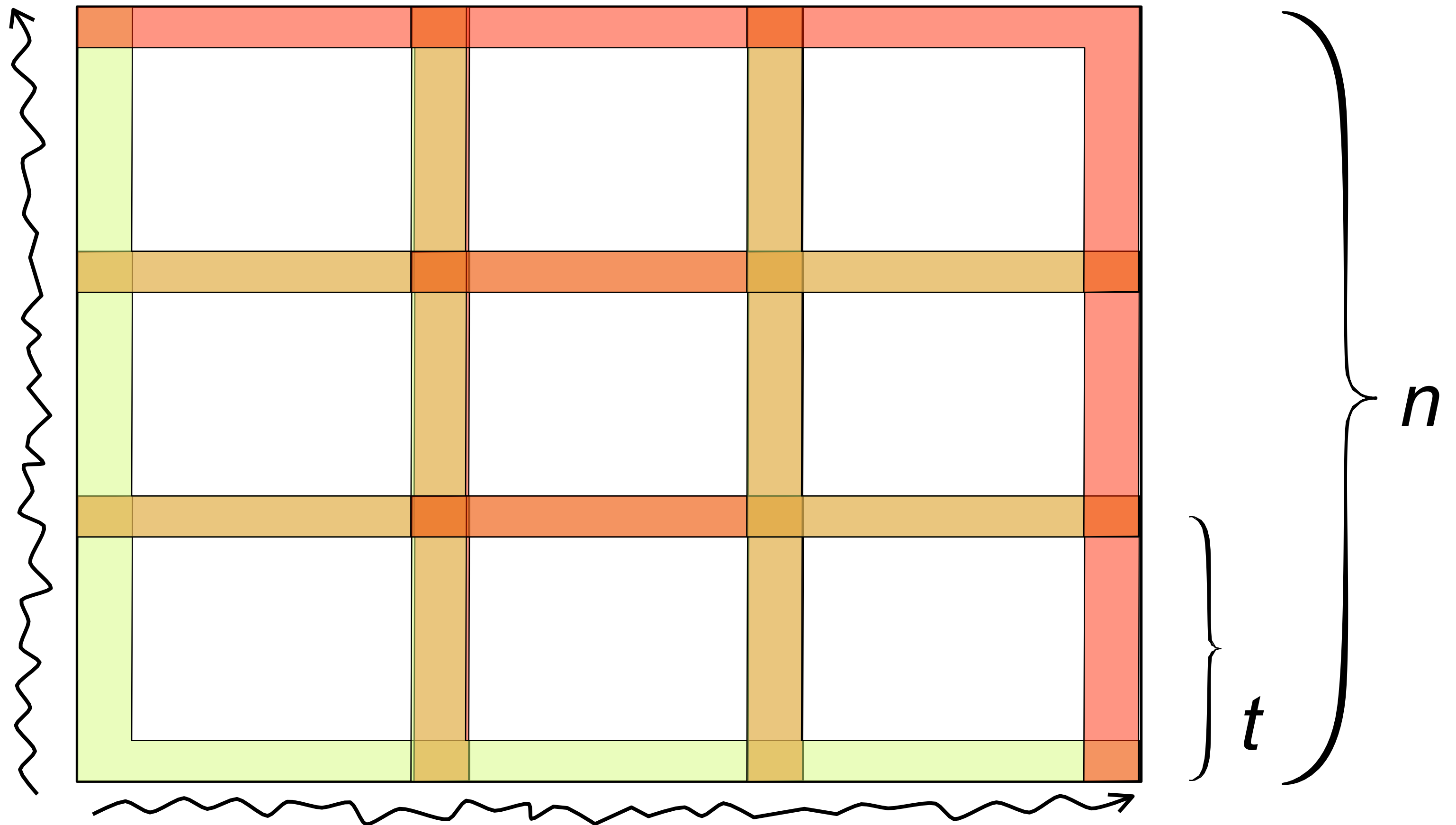
Метод четырех русских



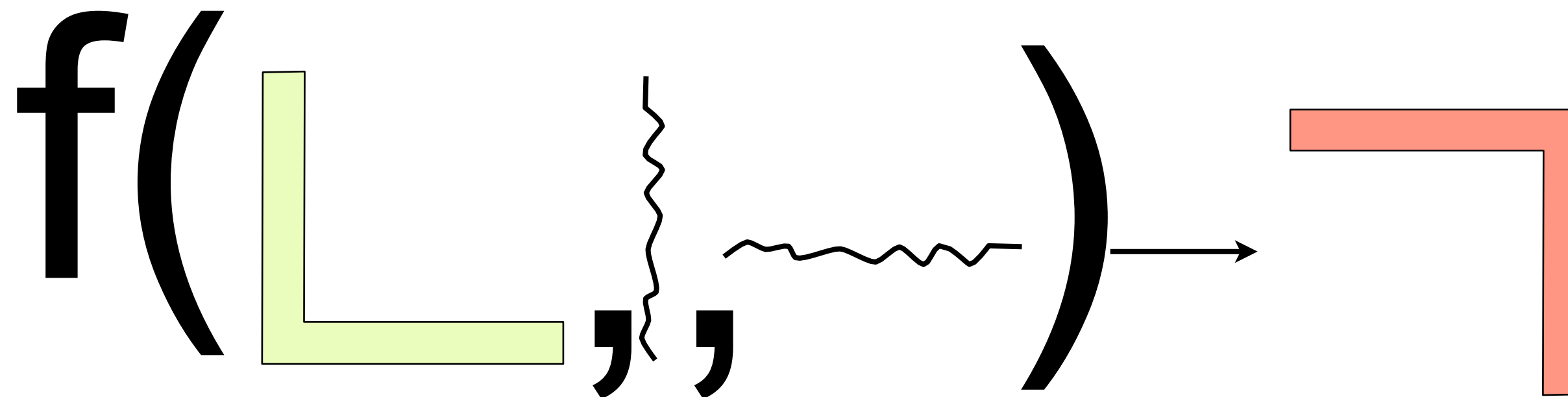
Метод четырех русских



Метод четырех русских

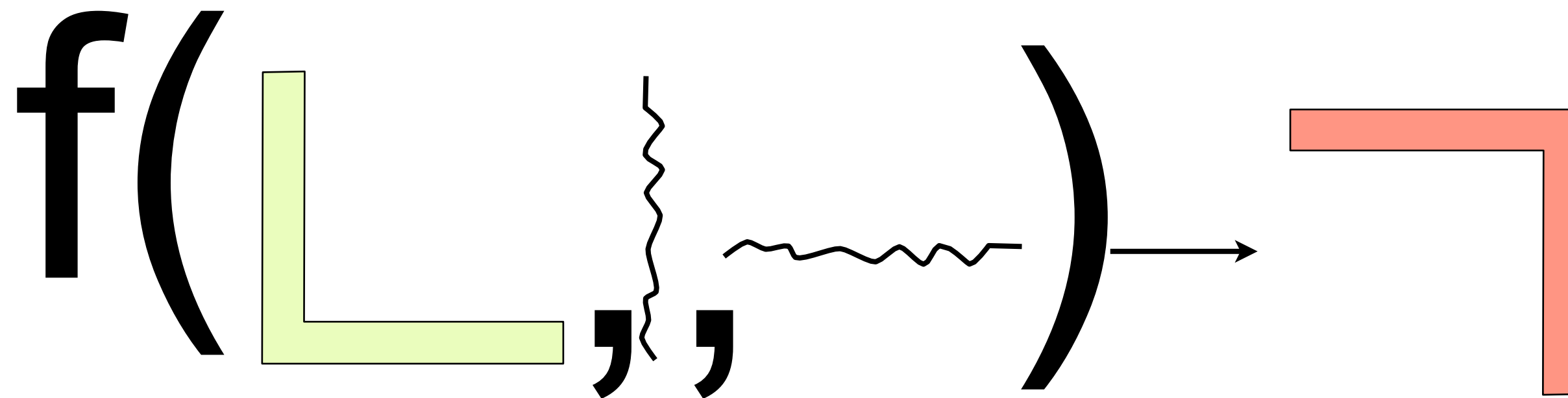


Блок-функция



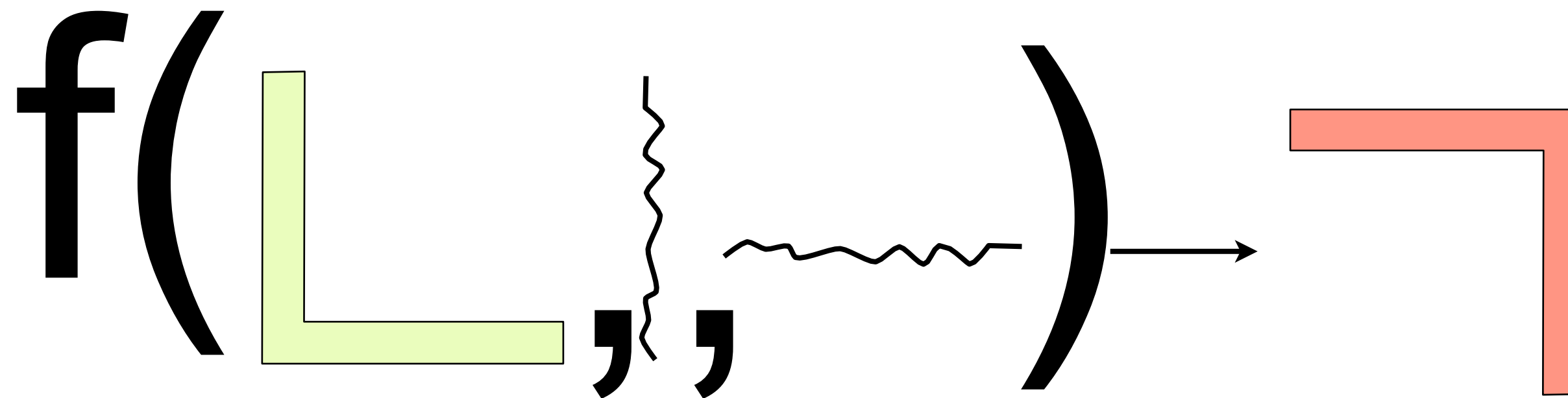
Если мы считаем данную функцию за $O(t^2)$, то расстояние выравнивания можно посчитать за ?

Блок-функция



Если мы считаем данную функцию за $O(t^2)$, то расстояние выравнивания можно посчитать за $O(n^2)$

Блок-функция



Если мы считаем данную функцию за $O(t^2)$, то расстояние выравнивания можно посчитать за $O(n^2)$

Однако эту функцию можно посчитать за $O(t)$

Упрощения и предположения

Рассчитываем обычный edit distance: гэпы и мисматчи стоят 1, а совпадения 0.

Размер алфавита Σ - константа.

$n = k(t-1)$ для некоторого целого k (блоки идеально покрывают матрицу, пересекаясь ровно по одной строке/столбцу)

Предпосчитаем функцию f

Предпосчитаем $f(x)$ для всех возможных $x = (\sqsubset, \sim, \sqsupset)$.

Сколько существует различных x ?

Предпосчитаем функцию f

Предпосчитаем $f(x)$ для всех возможных $x = (\sqcup, \sim, \{ \}$).

Сколько существует различных x ?

$$\underbrace{(n+1)^{2t}}_{\substack{\text{Каждая ячейка} \\ \text{содержит число} \\ \text{от 0 до } n}} \underbrace{|\Sigma|^{2t}}_{\substack{\text{Столько бывает} \\ \text{входных строк} \\ \text{длины } t.}}$$

Расчитать каждую стоит $O(t^2)$, и всего получается $O((n+1)^{2t} |\Sigma|^{2t} t^2) = O(n^2)$. Плохо!

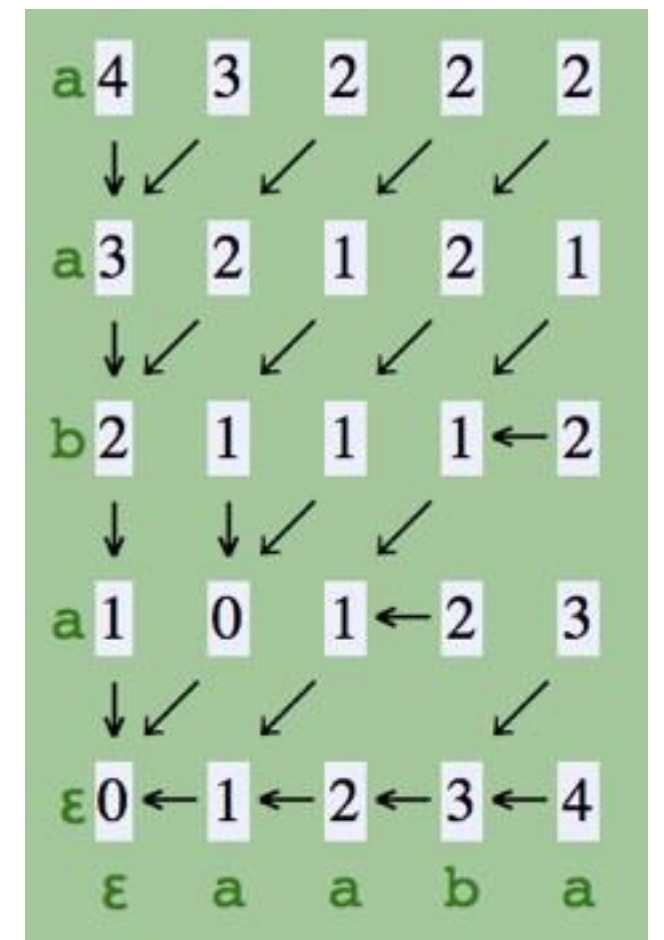
Кодирование отступов

Хитрость заключается в понимании того, что на самом деле существует меньше принципиально возможных различных входных данных для x .

Элементы строк и столбцов во входных данных не являются независимыми.

Пусть, D — матрица, а $D(i,j)$ — значение в позиции i,j .

Лемма: Соседние значения D в строке, столбце или диагонали отличаются не более чем на 1.



Кодирование отступов

Строка матрицы кодируется как изначальное значение плюс вектор над $\{-1,0,1\}$:

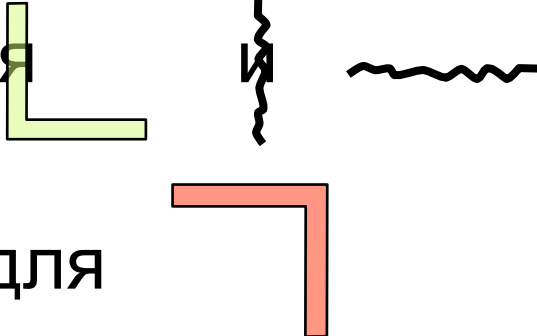
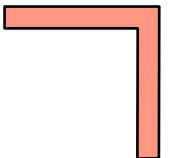
Пример. 567767 \rightarrow 5 1 1 0 -1 1

Определение. Вектор отступов — это кодировка строки или столбца, как указано выше, за исключением того, что первая запись имеет значение 0.

Пример. 567767 \rightarrow 0 1 1 0 0 -1 1

Итак, по первому значению и вектору смещения вы можете восстановить строку или столбец.

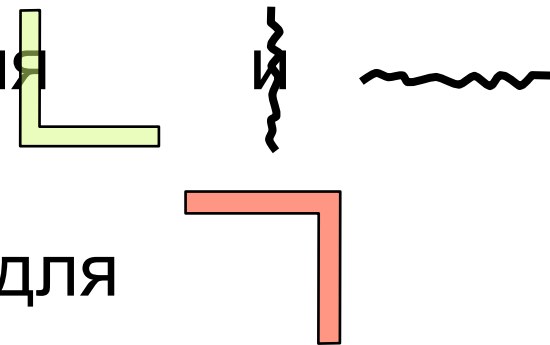
Кодирование отступов

Теорема. из векторов отступов для  можно посчитать вектор отступов для 

1	C+3	C+2	C+1	C+2
1	C+2	C+1	C+1	C+1
1	C+1	C+1	C	C+1
0	C	C	C+1	C+2
	0	0	1	1

Кодирование отступов

Теорема. из векторов отступов для



можно посчитать вектор отступов для

1	C+3	C+2	C+1	C+2
1	C+2	C+1	C+1	C+1
1	C+1	C+1	C	C+1
0	C	C	C+1	C+2
	0	0	1	1

	0	-1	-1	1	
1	C+3	C+2	C+1	C+2	1
1	C+2	C+1	C+1	C+1	0
1	C+1	C+1	C	C+1	-1
0	C	C	C+1	C+2	0
	0	0	1	1	

Анализ быстродействия

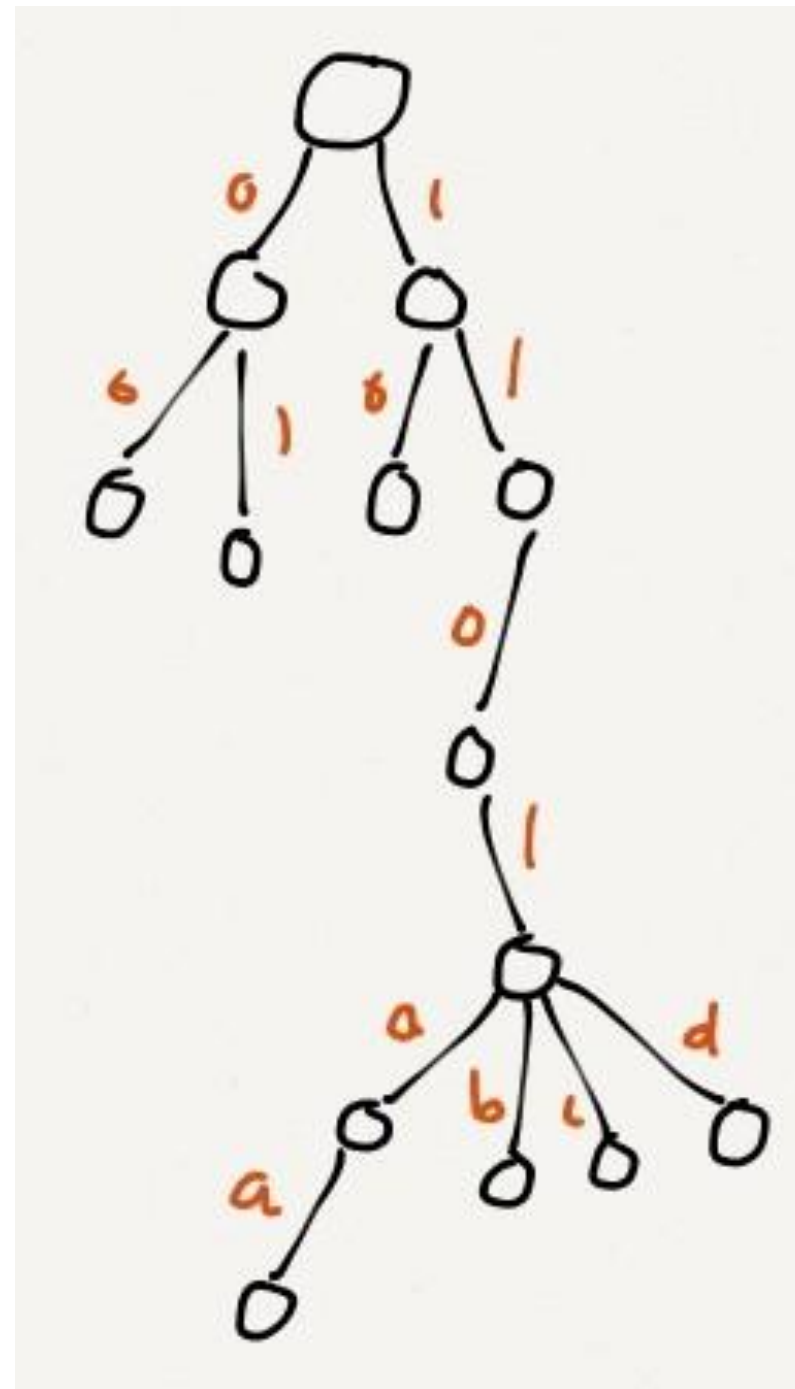
Всего $3^{2(t-1)}$ векторов.

Всего $3^{2(t-1)} |\Sigma|^{2t}$ ВОЗМОЖНЫХ ВХОДОВ в f .

Вычисление всех $f(x)$ занимает $O((3|\Sigma|)^{2t} t^2)$.

Предположим $t = \log_{2|\Sigma|} n$, мы получим $O(n(\log n)^2)$

Как быстро находить значения f



Depth $\approx 4t = O(t)$

Анализ быстройдействия

Всего блоков $O(n^2 / t^2)$.

Если поиск $f(x)$ занимает $O(t)$, то заполнить матрицу
 $O(tn^2/t^2) = O(n^2/t)$

Если $t = O(\log n)$ то:

$$O(n^2 / \log n + n(\log n)^2) = O(n^2 / \log n)$$

На практике

Удобно брать $t = \text{const}$.

Не дает асимптотического ускорения, но работает быстрее в const раз.

Burrows-Wheeler transform

a c a a c g

Burrows-Wheeler transform

a c a a c g \$

Burrows-Wheeler transform

a c a a c g \$
\$ a c a a c g

Burrows-Wheeler transform

a c a a c g \$
\$ a c a a c g
g \$ a c a a c

Burrows-Wheeler transform

a	c	a	a	c	g	\$
\$	a	c	a	a	c	g
g	\$	a	c	a	a	c
c	g	\$	a	c	a	a
a	c	g	\$	a	c	a
a	a	c	g	\$	a	c
c	a	a	c	g	\$	a

Burrows-Wheeler transform

\$ a c a a c g
a a c g \$ a c
a c a a c g \$
a c g \$ a c a
c a a c g \$ a
c g \$ a c a a
g \$ a c a a c

Burrows-Wheeler transform

\$	a	c	a	a	c	g
a	a	c	g	\$	a	c
a	c	a	a	c	g	\$
a	c	g	\$	a	c	a
c	a	a	c	g	\$	a
c	g	\$	a	c	a	a
g	\$	a	c	a	a	c

Burrows-Wheeler transform

a c a a c g \$



g c \$ a a a c

Burrows-Wheeler transform

a c a a c g \$
↑ ?
g c \$ a a a c

Burrows-Wheeler transform

\$	a	c	a	a	c	g
a	a	c	g	\$	a	c
a	c	a	a	c	g	\$
a	c	g	\$	a	c	a
c	a	a	c	g	\$	a
c	g	\$	a	c	a	a
g	\$	a	c	a	a	c

Burrows-Wheeler transform

\$	a	c	a	a	c	g
a	a	c	g	\$	a	c
a	c	a	a	c	g	\$
a	c	g	\$	a	c	a
c	a	a	c	g	\$	a
c	g	\$	a	c	a	a
g	\$	a	c	a	a	c

Burrows-Wheeler transform

g	\$	a	c	a	a	c
c	a	a	c	g	\$	a
\$	a	c	a	a	c	g
a	a	c	g	\$	a	c
a	c	a	a	c	g	\$
a	c	g	\$	a	c	a
c	g	\$	a	c	a	a

Burrows-Wheeler transform

\$ a c a a c g
a a c g \$ a c
a c a a c g \$
a c g \$ a c a
c a a c g \$ a
c g \$ a c a a
g \$ a c a a c

Burrows-Wheeler transform

g

c

\$

a

a

a

c

\$ a c a a c g

a a c g \$ a c

a c a a c g \$

a c g \$ a c a

c a a c g \$ a

c g \$ a c a a

g \$ a c a a c

Burrows-Wheeler transform

\$

a

a

a

c

c

g

\$ a c a a c g

a a c g \$ a c

a c a a c g \$

a c g \$ a c a

c a a c g \$ a

c g \$ a c a a

g \$ a c a a c

Burrows-Wheeler transform

\$
a
a
a
c
c
g

\$	a	c	a	a	c	g
a	a	c	g	\$	a	c
a	c	a	a	c	g	\$
a	c	g	\$	a	c	a
c	a	a	c	g	\$	a
c	g	\$	a	c	a	a
g	\$	a	c	a	a	c

Burrows-Wheeler transform

g \$
c a
\$ a
a a
a c
a c
c g

\$ a c a a c g
a a c g \$ a c
a c a a c g \$
a c g \$ a c a
c a a c g \$ a
c g \$ a c a a
g \$ a c a a c

Burrows-Wheeler transform

\$ a

a a

a c

a c

c a

c g

g \$

\$ a c a a c **g**

a a c g **\$** a **c**

a c a a c g **\$**

a c g **\$** a c **a**

c a a c g **\$** **a**

c g **\$** a c a **a**

g **\$** a c a a **c**

Burrows-Wheeler transform

\$ a

a a

a c

a c

c a

c g

g \$

\$ a c a a c g

a a c g \$ a c

a c a a c g \$

a c g \$ a c a

c a a c g \$ a

c g \$ a c a a

g \$ a c a a c

Burrows-Wheeler transform

g \$ a

c a a

\$ a c

a a c

a c a

a c g

c g \$

\$ a c a a c g

a a c g \$ a c

a c a a c g \$

a c g \$ a c a

c a a c g \$ a

c g \$ a c a a

g \$ a c a a c

Burrows-Wheeler transform

\$ a c

a a c

a c a

a c g

c a a

c g \$

g \$ a

\$ a c a a c **g**

a a c g \$ a **c**

a c a a c g **\$**

a c g \$ a c **a**

c a a c g \$ **a**

c g \$ a c a **a**

g \$ a c a a **c**

Burrows-Wheeler transform

\$ a c
a a c
a c a
a c g
c a a
c g \$
g \$ a

\$	a	c	a	a	c	g
a	a	c	g	\$	a	c
a	c	a	a	c	g	\$
a	c	g	\$	a	c	a
c	a	a	c	g	\$	a
c	g	\$	a	c	a	a
g	\$	a	c	a	a	c

Burrows-Wheeler transform

g \$ a c

c a a c

\$ a c a

a a c g

a c a a

a c g \$

c g \$ a

\$ a c a a c g

a a c g \$ a c

a c a a c g \$

a c g \$ a c a

c a a c g \$ a

c g \$ a c a a

g \$ a c a a c

Burrows-Wheeler transform

\$ a c a
a a c g
a c a a
a c g \$
c a a c
c g \$ a
g \$ a c

\$ a c a a c g
a a c g \$ a c
a c a a c g \$
a c g \$ a c a
c a a c g \$ a
c g \$ a c a a
g \$ a c a a c

Burrows-Wheeler transform

g \$ a c a

c a a c g

\$ a c a a

a a c g \$

a c a a c

a c g \$ a

c g \$ a c

\$ a c a a c **g**

a a c g \$ a **c**

a c a a c g **\$**

a c g \$ a c **a**

c a a c g \$ **a**

c g \$ a c a **a**

g \$ a c a a **c**

Burrows-Wheeler transform

\$ a c a a
a a c g \$
a c a a c
a c g \$ a
c a a c g
c g \$ a c
g \$ a c a

\$ a c a a c g
a a c g \$ a c
a c a a c g \$
a c g \$ a c a
c a a c g \$ a
c g \$ a c a a
g \$ a c a a c

Burrows-Wheeler transform

g \$ a c a a
c a a c g \$
\$ a c a a c
a a c g \$ a
a c a a c g
a c g \$ a c
c g \$ a c a

\$ a c a a c g
a a c g \$ a c
a c a a c g \$
a c g \$ a c a
c a a c g \$ a
c g \$ a c a a
g \$ a c a a c

Burrows-Wheeler transform

\$ a c a a c
a a c g \$ a
a c a a c g
a c g \$ a c
c a a c g \$
c g \$ a c a
g \$ a c a a

\$ a c a a c g
a a c g \$ a c
a c a a c g \$
a c g \$ a c a
c a a c g \$ a
c g \$ a c a a
g \$ a c a a c

Burrows-Wheeler transform

g \$ a c a a c
c a a c g \$ a
\$ a c a a c g
a a c g \$ a c
a c a a c g \$
a c g \$ a c a
c g \$ a c a a

\$ a c a a c g
a a c g \$ a c
a c a a c g \$
a c g \$ a c a
c a a c g \$ a
c g \$ a c a a
g \$ a c a a c

Burrows-Wheeler transform

\$ a c a a c g

a a c g \$ a c

a c a a c g \$

a c g \$ a c a

c a a c g \$ a

c g \$ a c a a

g \$ a c a a c

\$ a c a a c g

a a c g \$ a c

a c a a c g \$

a c g \$ a c a

c a a c g \$ a

c g \$ a c a a

g \$ a c a a c

First-last property

$a_k c_i a_m a_n c_j g_x \$$

First-last property

$\$$ ₁	a	c	a	a	c	g
a ₁	a	c	g	$\$$	a	c
a ₂	c	a	a	c	g	$\$$
a ₃	c	g	$\$$	a	c	a
c ₁	a	a	c	g	$\$$	a
c ₂	g	$\$$	a	c	a	a
g ₁	$\$$	a	c	a	a	c

First-last property

$\$$ ₁	a	c	a	a	c	g ₁
a ₁	a	c	g	$\$$	a	c
a ₂	c	a	a	c	g	$\$$ ₁
a ₃	c	g	$\$$	a	c	a
c ₁	a	a	c	g	$\$$	a
c ₂	g	$\$$	a	c	a	a
g ₁	$\$$	a	c	a	a	c

First-last property

$\$1$ a c a a c g_1

a_1 a c g \$ a c

a_2 c a a c g $\$1$

a_3 c g \$ a c a

c_1 a a c g \$ a

c_2 g \$ a c a a

g_1 \$ a c a a c

First-last property

a_3 c g \$ a c **a**
 c_1 a a c g \$ **a**
 c_2 g \$ a c a **a**

$\$1$ a c a a c **g_1**
 a_1 a c g \$ a **c**
 a_2 c a a c g **$\$1$**

a_3 c g \$ a c **a**
 c_1 a a c g \$ **a**
 c_2 g \$ a c a **a**

 g_1 \$ a c a a **c**

First-last property

a **a**₃ c g \$ a c
a **c**₁ a a c g \$
a **c**₂ g \$ a c a

\$₁ a c a a c **g**₁
a₁ a c g \$ a **c**
a₂ c a a c g **\$**₁
a₃ c g \$ a c **a**
c₁ a a c g \$ **a**
c₂ g \$ a c a **a**
g₁ \$ a c a a **c**

First-last property

a **a**₃ c g \$ a c
a **c**₁ a a c g \$
a **c**₂ g \$ a c a

\$₁ a c a a c **g**₁
a₁ a c g \$ a **c**
a₂ c a a c g **\$**₁
a₃ c g \$ a c **a**
c₁ a a c g \$ **a**
c₂ g \$ a c a **a**
g₁ \$ a c a a **c**

First-last property

a_1 a_3 c g \$ a c
 a_2 c_1 a a c g \$
 a_3 c_2 g \$ a c a

$\$1$ a c a a c g_1
 a_1 a c g \$ a c
 a_2 c a a c g $\$1$
 a_3 c g \$ a c a
 c_1 a a c g \$ a
 c_2 g \$ a c a a
 g_1 \$ a c a a c

First-last property

a_3 c g \$ a c a_1
 c_1 a a c g \$ a_2
 c_2 g \$ a c a a_3

$\$1$ a c a a c g_1
 a_1 a c g \$ a c
 a_2 c a a c g $\$1$
 a_3 c g \$ a c a
 c_1 a a c g \$ a
 c_2 g \$ a c a a
 g_1 \$ a c a a c

First-last property

a_3 c g \$ a c a_1
 c_1 a a c g \$ a_2
 c_2 g \$ a c a a_3

$\$1$ a c a a c g_1
 a_1 a c g \$ a c
 a_2 c a a c g $\$1$
 a_3 c g \$ a c a
 c_1 a a c g \$ a
 c_2 g \$ a c a a
 g_1 \$ a c a a c

First-last property

a_3 c g \$ a c a_1
 c_1 a a c g \$ a_2
 c_2 g \$ a c a a_3

$\$1$ a c a a c g_1
 a_1 a c g \$ a c
 a_2 c a a c g $\$1$
 a_3 c g \$ a c a_1
 c_1 a a c g \$ a_2
 c_2 g \$ a c a a_3
 g_1 \$ a c a a c

First-last property

$\$1$	a	c	a	a	c	g_1
a_1	a	c	g	$\$$	a	c_1
a_2	c	a	a	c	g	$\$1$
a_3	c	g	$\$$	a	c	a_1
c_1	a	a	c	g	$\$$	a_2
c_2	g	$\$$	a	c	a	a_3
g_1	$\$$	a	c	a	a	c_2

First-last property

$a_2 \ c_1 \ a_1 \ a_3 \ c_2 \ g_1 \ \$_1$

First-last property

$\$1$	a_2	c_1	a_1	a_3	c_2	g_1
a_1	a_3	c_2	g_1	$\$1$	a_2	c_1
a_2	c_1	a_1	a_3	c_2	g_1	$\$1$
a_3	c_2	g_1	$\$1$	a_2	c_1	a_1
c_1	a_1	a_3	c_2	g_1	$\$1$	a_2
c_2	g_1	$\$1$	a_2	c_1	a_1	a_3
g_1	$\$1$	a_2	c_1	a_1	a_3	c_2

First-last property

$\$1$	a_2	c_1	a_1	a_3	c_2	g_1
a_1	a_3	c_2	g_1	$\$1$	a_2	c_1
a_2	c_1	a_1	a_3	c_2	g_1	$\$1$
a_3	c_2	g_1	$\$1$	a_2	c_1	a_1
c_1	a_1	a_3	c_2	g_1	$\$1$	a_2
c_2	g_1	$\$1$	a_2	c_1	a_1	a_3
g_1	$\$1$	a_2	c_1	a_1	a_3	c_2

Suffix array

a c a a c g \$

Suffix array

0. a c a a c g \$
1. c a a c g \$
2. a a c g \$
3. a c g \$
4. c g \$
5. g \$
6. \$

Suffix array

6. \$
2. a a c g \$
0. a c a a c g \$
3. a c g \$
1. c a a c g \$
4. c g \$
5. g \$

Suffix array

6.	\$	a	c	a	a	c	g
2.	a	a	c	g	\$	a	c
0.	a	c	a	a	c	g	\$
3.	a	c	g	\$	a	c	a
1.	c	a	a	c	g	\$	a
4.	c	g	\$	a	c	a	a
5.	g	\$	a	c	a	a	c

Suffix array and BWT

6. \$ a c a a c g

2. a a c g \$ a c

0. a c a a c g \$

3. a c g \$ a c a

1. c a a c g \$ a

4. c g \$ a c a a

5. g \$ a c a a c

\$ a c a a c g

a a c g \$ a c

a c a a c g \$

a c g \$ a c a

c a a c g \$ a

c g \$ a c a a

g \$ a c a a c

Suffix array and BWT

$B[i] = \$$ if $S[i] = 0$

$B[i] = X[S[i] - 1]$ otherwise

Suffix array interval

6. \$
2. a a c g \$
0. a c a a c g \$
3. a c g \$
1. c a a c g \$
4. c g \$
5. g \$

$R(" ") = (0, 6)$

Suffix array interval

6. \$

$R(\text{"a"}) = (1, 3)$

2. a a c g \$

0. a c a a c g \$

3. a c g \$

1. c a a c g \$

4. c g \$

5. g \$

Suffix array interval

6. \$ $R(\text{"a c"}) = (2, 3)$

2. a a c g \$

0. a c a a c g \$

3. a c g \$

1. c a a c g \$

4. c g \$

5. g \$

Suffix array interval

$$R_L(W) = \min \{k: W \text{ is prefix of } X_{S[k]} \}$$

$$R_H(W) = \max \{k: W \text{ is prefix of } X_{S[k]} \}$$

Suffix array interval

$$C(x) = | \{ 0 \leq j \leq n-2 : X[j] < x \} |$$

a c a a c g \$

$$C(a) = 0, C(c) = 3, C(g) = 5, \dots$$

Suffix array interval

→ 6. \$
→ 2. a a c g \$
0. a c a a c g \$
→ 3. a c g \$
1. c a a c g \$
→ 4. c g \$
→ 5. g \$

Suffix array interval

$$O(x, i) = | \{ 0 \leq j \leq i : B[j] = x \} |$$

g c \$ a a a c

$$O(a, 0) = 0, O(a, 1) = 0, O(a, 2) = 0, O(a, 3) = 1, \\ O(a, 4) = 2, \dots$$

Suffix array interval

							a
$\$_1$	a	c	a	a	c	g_1	0
a_1	a	c	g	$\$$	a	c_1	0
a_2	c	a	a	c	g	$\$_1$	0
a_3	c	g	$\$$	a	c	a_1	
c_1	a	a	c	g	$\$$	a_2	
c_2	g	$\$$	a	c	a	a_3	
g_1	$\$$	a	c	a	a	c_1	

Suffix array interval

							a
$\$_1$	a	c	a	a	c	g_1	0
a_1	a	c	g	$\$$	a	c_1	0
a_2	c	a	a	c	g	$\$_1$	0
a_3	c	g	$\$$	a	c	a_1	1
c_1	a	a	c	g	$\$$	a_2	
c_2	g	$\$$	a	c	a	a_3	
g_1	$\$$	a	c	a	a	c_1	

Suffix array interval

							a
$\$_1$	a	c	a	a	c	g_1	0
a_1	a	c	g	$\$$	a	c_1	0
a_2	c	a	a	c	g	$\$_1$	0
a_3	c	g	$\$$	a	c	a_1	1
c_1	a	a	c	g	$\$$	a_2	2
c_2	g	$\$$	a	c	a	a_3	
g_1	$\$$	a	c	a	a	c_1	

Suffix array interval

							a
$\$_1$	a	c	a	a	c	g_1	0
a_1	a	c	g	$\$$	a	c_1	0
a_2	c	a	a	c	g	$\$_1$	0
a_3	c	g	$\$$	a	c	a_1	1
c_1	a	a	c	g	$\$$	a_2	2
c_2	g	$\$$	a	c	a	a_3	3
g_1	$\$$	a	c	a	a	c_1	

Suffix array interval

	a							
$\$ _1$	a	c	a	a	c	$g _1$	0	
$a _1$	a	c	g	$\$$	a	$c _1$	0	
$a _2$	c	a	a	c	g	$\$ _1$	0	
$a _3$	c	g	$\$$	a	c	$a _1$	1	
$c _1$	a	a	c	g	$\$$	$a _2$	2	
$c _2$	g	$\$$	a	c	a	$a _3$	3	
$g _1$	$\$$	a	c	a	a	$c _2$	3	

Suffix array interval

	a							
$\$_1$	a	c	a	a	c	g_1	0	
a_1	a	c	g	$\$$	a	c_1	0	
a_2	c	a	a	c	g	$\$_1$	0	
a_3	c	g	$\$$	a	c	a_1	1	
c_1	a	a	c	g	$\$$	a_2	2	
c_2	g	$\$$	a	c	a	a_3	3	
g	$\$$	a	c	a	a	c	3	

$$O(a, 1) = 0$$

$$O(a, 4) = 2$$

Suffix array interval

								a
\$ ₁	a	c	a	a	c	g ₁	0	0
a ₁	a	c	g	\$	a	c ₁	0	0
a ₂	c	a	a	c	g	\$ ₁	0	0
a ₃	c	g	\$	a	c	a ₁	1	1
c ₁	a	a	c	g	\$	a ₂	2	2
c ₂	g	\$	a	c	a	a ₃	3	3
g	\$	a	c	a	a	c	3	3

$$O(a, 1) = 0$$

$$[a _1, a _2]$$

$$O(a, 4) = 2$$

Suffix array interval

		a	c	g	t
$\$_1$	a c a a c g₁	0	0	1	0
a₁	a c g $\$$ a c₁	0	1	1	0
a₂	c a a c g $\\$₁	0	1	1	0
a₃	c g $\$$ a c a₁	1	1	1	0
c₁	a a c g $\$$ a₂	2	1	1	0
c₂	g $\$$ a c a a₃	3	1	1	0
g₁	$\$$ a c a a c₁	3	2	1	0

Suffix array interval

							a	c	g	t
$\$$ ₁	a	c	a	a	c	g ₁	0	0	1	0
a ₁	a	c	g	\$	a	c ₁	0	1	1	0
a ₂	c	a	a	c	g	\$ ₁	0	1	1	0
a ₃	c	g	\$	a	c	a ₁	1	1	1	0
c ₁	a	a	c	g	\$	a ₂	2	1	1	0
c ₂	g	\$	a	c	a	a ₃	3	1	1	0
g ₁	\$	a	c	a	a	c ₂	3	2	1	0

$$O(c, -1) = 0$$

[c₁]

$$O(c, 4) = 1$$

First-last property

$\$1$	a	c	a	a	c	g_1
a_1	a	c	g	$\$$	a	c_1
a_2	c	a	a	c	g	$\$1$
a_3	c	g	$\$$	a	c	a_1
c_1	a	a	c	g	$\$$	a_2
c_2	g	$\$$	a	c	a	a_3
g_1	$\$$	a	c	a	a	c_2

Suffix array interval

$$R_L(xW) = C(x) + O(x, R_L(W)) - 1$$

$$R_H(xW) = C(x) + O(x, R_H(W)) - 1$$

$$R_L("") = 0$$

$$R_H("") = \text{len}(X) - 1$$

Intuition behind pattern search

a

a c a a c g \$

Intuition behind pattern search

a

a c **a** **a** c g \$

Intuition behind pattern search

c a

a c a a c g \$

Intuition behind pattern search

c a

a c a a c g \$

$\$_1$	a	c	a	a	c	g₁
a₁	a	c	g	\$	a	c₁
a₂	c	a	a	c	g	\$₁
a₃	c	g	\$	a	c	a₁
c₁	a	a	c	g	\$	a₂
c₂	g	\$	a	c	a	a₃
g	\$	a	c	a	a	c

Intuition behind pattern search

c a

a c a a c g \$

$\$1$ a c a a c **g₁**

a₁	a	c	g	\$	a	c₁
a₂	c	a	a	c	g	\$₁
a₃	c	g	\$	a	c	a₁

c₁ a a c g \$ **a₂**

c₂ g \$ a c a **a₃**

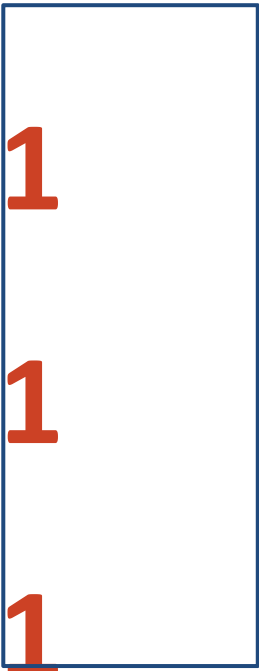
g \$ a c a a **c**

Intuition behind pattern search

c a

a c a a c g \$

$\$1$	a	c	a	a	c	g₁
a₁	a	c	g	\$	a	c₁
a₂	c	a	a	c	g	\$₁
a₃	c	g	\$	a	c	a₁
c₁	a	a	c	g	\$	a₂
c₂	g	\$	a	c	a	a₃
g	\$	a	c	a	a	c



Intuition behind pattern search

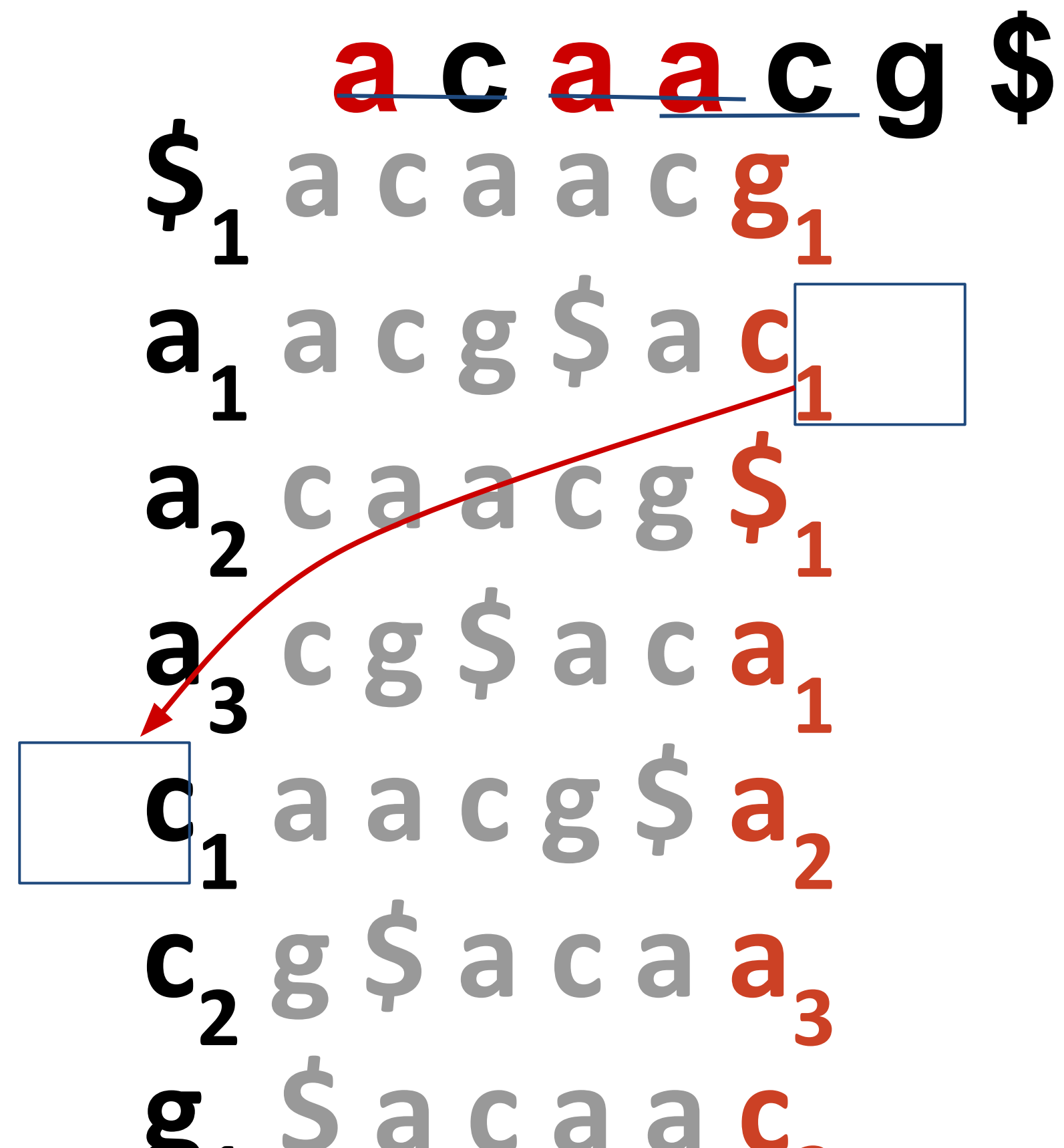
c a

a c a a c g \$

$\$1$	a	c	a	a	c	g₁
a₁	a	c	g	\$	a	c₁ <input type="text"/>
a₂	c	a	a	c	g	\$₁
a₃	c	g	\$	a	c	a₁
c₁	a	a	c	g	\$	a₂
c₂	g	\$	a	c	a	a₃
g	\$	a	c	a	a	c

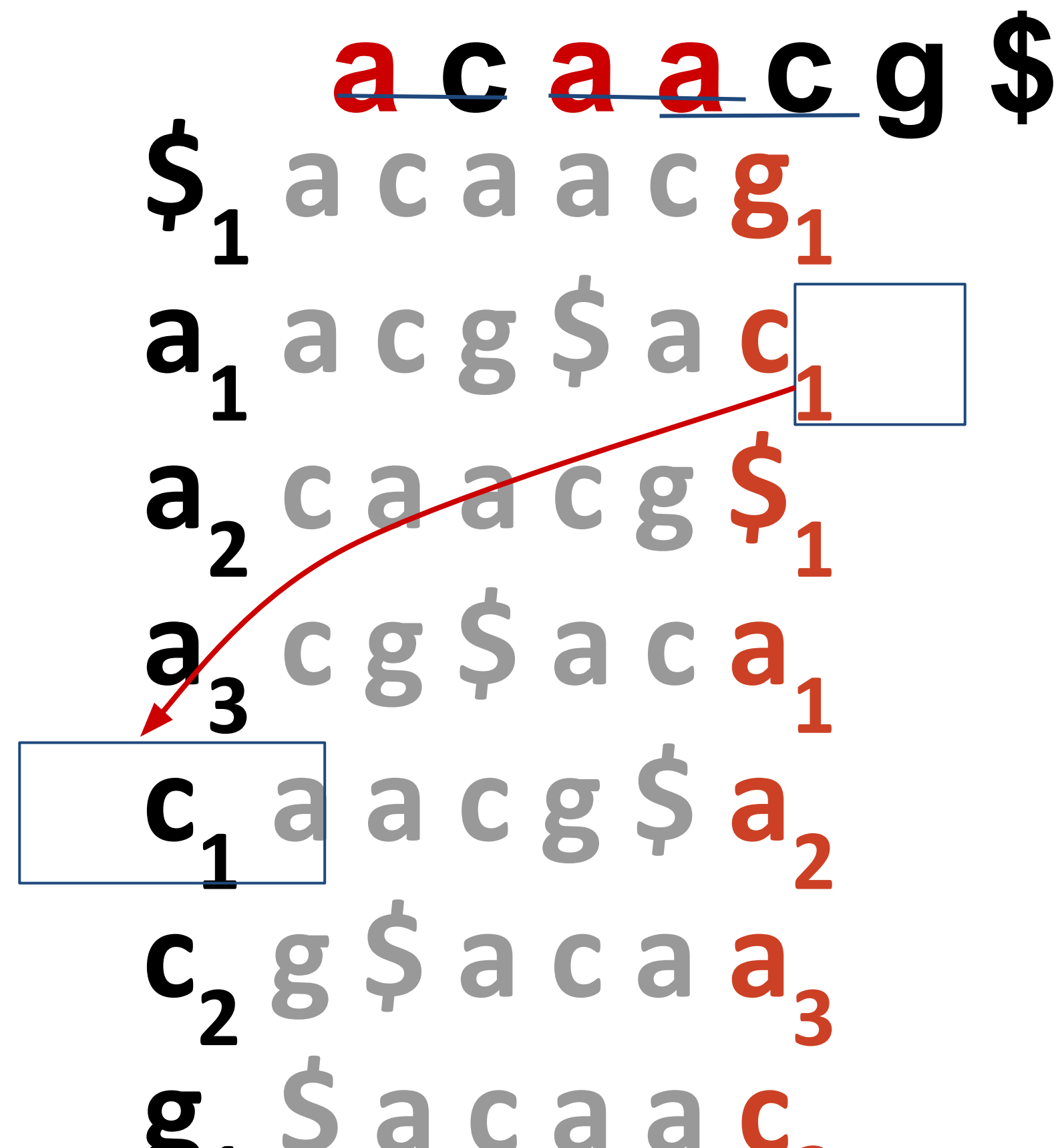
Intuition behind pattern search

c a



Intuition behind pattern search

c a



Suffix array interval

$$R_L(xW) = C(x) + O(x, R_L(W)) - 1$$

$$R_H(xW) = C(x) + O(x, R_H(W)) - 1$$

$$R_L("") = 0$$

$$R_H("") = \text{len}(X) - 1$$

Pattern search

$C(\$) = 0$
→

$C(a) = 1$
→

$C(c) = 4$
→

$C(g) = 6$
→

$C(t) = 7$
→

								a	c	g	t	
$\$$ ₁	a	c	a	a	c	g	₁	0	0	1	0	
a	₁	a	c	g	\$	a	c	₁	0	1	1	0
a	₂	c	a	a	c	g	\$	₁	0	1	1	0
a	₃	c	g	\$	a	c	a	₁	1	1	1	0
c	₁	a	a	c	g	\$	a	₂	2	1	1	0
c	₂	g	\$	a	c	a	a	₃	3	1	1	0
g	₁	\$	a	c	a	a	c	₁	3	2	1	0

Pattern search

a a c

$C(\$) = 0$
→

$C(a) = 1$
→

$C(c) = 4$
→

$C(g) = 6$
→

$C(t) = 7$
→

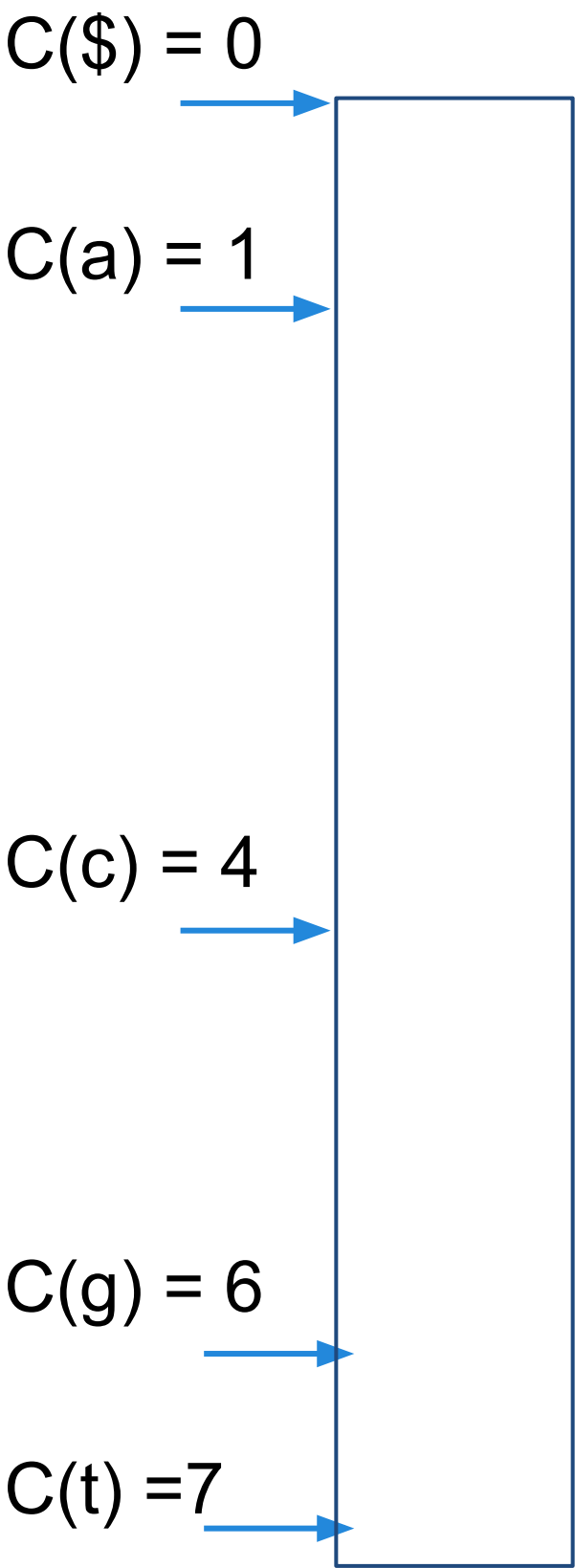
		a	c	g	t					
$\$$ ₁	a	c	a	a	c	g ₁	0	0	1	0
a ₁	a	c	g	$\$$	a	c ₁	0	1	1	0
a ₂	c	a	a	c	g	$\\$ ₁	0	1	1	0
a ₃	c	g	$\$$	a	c	a ₁	1	1	1	0
c ₁	a	a	c	g	$\$$	a ₂	2	1	1	0
c ₂	g	$\$$	a	c	a	a ₃	3	1	1	0
g ₁	$\$$	a	c	a	a	c ₂	3	2	1	0

Pattern search

a a c

$R_L('') = 0$

$R_H('') = 6$



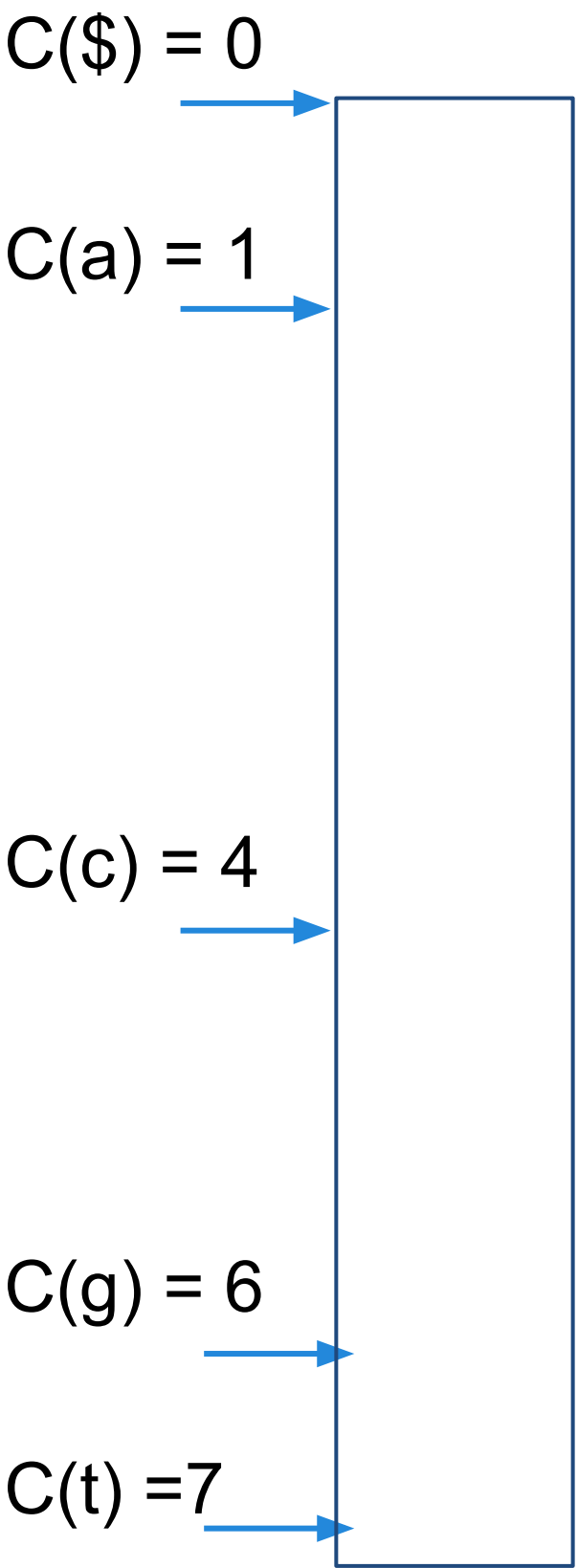
								a	c	g	t
$\$$ ₁	a	c	a	a	c	g	₁	0	0	1	0
a	₁	a	c	g	\$	a	c	0	1	1	0
a	₂	c	a	a	c	g	\$	0	1	1	0
a	₃	c	g	\$	a	c	a	1	1	1	0
c	₁	a	a	c	g	\$	a	2	1	1	0
c	₂	g	\$	a	c	a	a	3	1	1	0
g	₁	\$	a	c	a	a	c	3	2	1	0

Pattern search

a a **c**

$$R_L(xW) = C(x) + O(x, R_L(W)) - 1$$

$$R_H(xW) = C(x) + O(x, R_H(W)) - 1$$



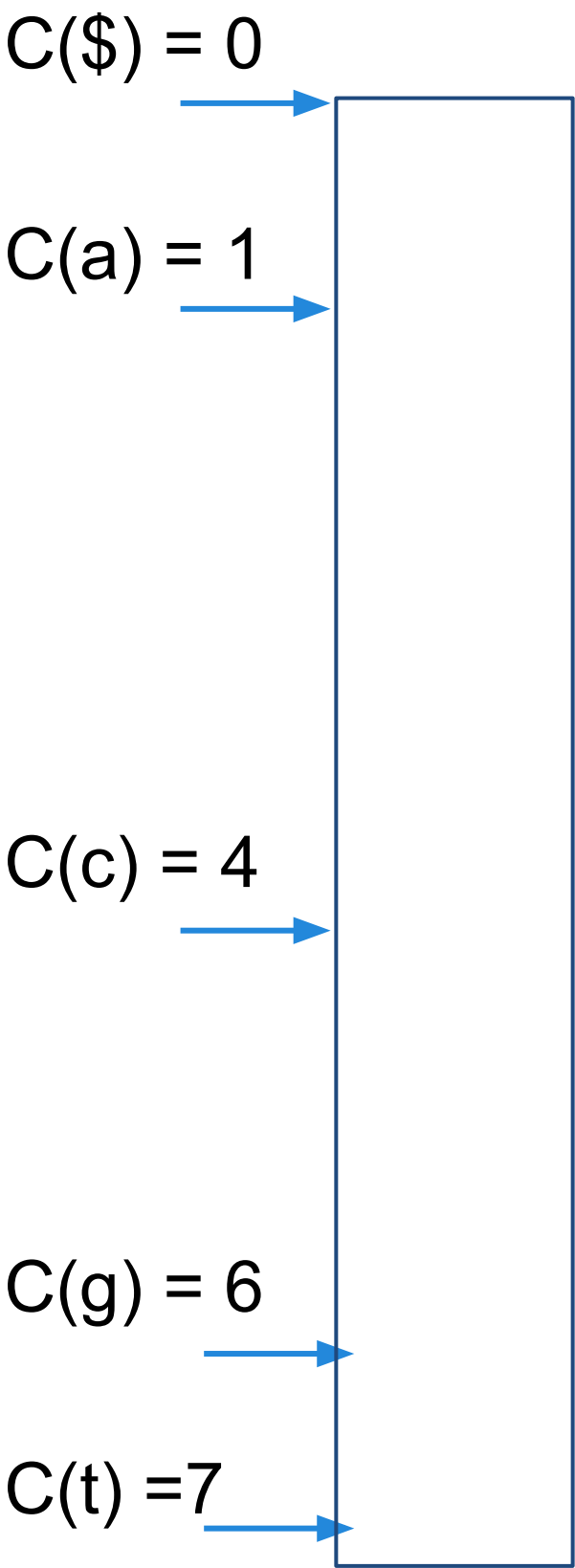
		a	c	g	t					
$\$$ ₁	a	c	a	a	c	g ₁	0	0	1	0
a ₁	a	c	g	\$	a	c ₁	0	1	1	0
a ₂	c	a	a	c	g	\$ ₁	0	1	1	0
a ₃	c	g	\$	a	c	a ₁	1	1	1	0
c ₁	a	a	c	g	\$	a ₂	2	1	1	0
c ₂	g	\$	a	c	a	a ₃	3	1	1	0
g ₁	\$	a	c	a	a	c ₂	3	2	1	0

Pattern search

a a c

$$R_L(c) = C(c) + O(c, R_L("")) - 1$$

$$R_H(c) = C(c) + O(c, R_H("")) - 1$$



		a	c	g	t					
$\$$ ₁	a	c	a	a	c	g ₁	0	0	1	0
a ₁	a	c	g	$\$$	a	c ₁	0	1	1	0
a ₂	c	a	a	c	g	$\\$ ₁	0	1	1	0
a ₃	c	g	$\$$	a	c	a ₁	1	1	1	0
c ₁	a	a	c	g	$\$$	a ₂	2	1	1	0
c ₂	g	$\$$	a	c	a	a ₃	3	1	1	0
g ₁	$\$$	a	c	a	a	c ₂	3	2	1	0

Pattern search

a a c

$$R_L(c) = C(c) + O(c, -1)$$

$$R_H(c) = C(c) + O(c, 6) - 1$$

$C(\$) = 0$
→

$C(a) = 1$
→

$C(c) = 4$
→

$C(g) = 6$
→

$C(t) = 7$
→

								a	c	g	t
\$ ₁	a	c	a	a	c	g ₁		0	0	1	0
a ₁	a	c	g	\$	a	c ₁		0	1	1	0
a ₂	c	a	a	c	g	\$ ₁		0	1	1	0
a ₃	c	g	\$	a	c	a ₁		1	1	1	0
c ₁	a	a	c	g	\$	a ₂		2	1	1	0
c ₂	g	\$	a	c	a	a ₃		3	1	1	0
g ₁	\$	a	c	a	a	c ₁		3	2	1	0

Pattern search

a a **c**

$$R_L(c) = C(c) + 0$$

$$R_H(c) = C(c) + 2 - 1$$

$$C(\$) = 0 \rightarrow$$

$$C(a) = 1 \rightarrow$$

$$C(c) = 4 \rightarrow$$

$$C(g) = 6 \rightarrow$$

$$C(t) = 7 \rightarrow$$

								a	c	g	t
\$ ₁	a	c	a	a	c	g₁		0	0	1	0
a ₁	a	c	g	\$	a	c₁		0	1	1	0
a ₂	c	a	a	c	g	\$₁		0	1	1	0
a ₃	c	g	\$	a	c	a₁		1	1	1	0
c ₁	a	a	c	g	\$	a₂		2	1	1	0
c ₂	g	\$	a	c	a	a₃		3	1	1	0
g ₁	\$	a	c	a	a	c₁		3	2	1	0

Pattern search

a a c

$$R_L(c) = 4 + 0 = 4$$

$$R_H(c) = 4 + 2 - 1 = 5$$

$C(\$) = 0$ →

$C(a) = 1$ →

$C(c) = 4$ →

$C(g) = 6$ →

$C(t) = 7$ →

								a	c	g	t	
$\$$ ₁	a	c	a	a	c	g	₁	0	0	1	0	
a	₁	a	c	g	\$	a	c	₁	0	1	1	0
a	₂	c	a	a	c	g	\$	₁	0	1	1	0
a	₃	c	g	\$	a	c	a	₁	1	1	1	0
c	₁	a	a	c	g	\$	a	₂	2	1	1	0
c	₂	g	\$	a	c	a	a	₃	3	1	1	0
g	₁	\$	a	c	a	a	c	₁	3	2	1	0

Pattern search

a a c

$$R_L(ac) = C(a) + O(a, R_L(c)) - 1$$

$$R_H(ac) = C(a) + O(a, R_H(c)) - 1$$

$C(\$) = 0$
→

$C(a) = 1$
→

$C(c) = 4$
→

$C(g) = 6$
→

$C(t) = 7$
→

		a	c	g	t					
$\$$ ₁	a	c	a	a	c	g ₁	0	0	1	0
a ₁	a	c	g	\$	a	c ₁	0	1	1	0
a ₂	c	a	a	c	g	\$ ₁	0	1	1	0
a ₃	c	g	\$	a	c	a ₁	1	1	1	0
c ₁	a	a	c	g	\$	a ₂	2	1	1	0
c ₂	g	\$	a	c	a	a ₃	3	1	1	0
g ₁	\$	a	c	a	a	c ₂	3	2	1	0

Pattern search

a a c

$$R_L(ac) = C(a) + O(a, 3)$$

$$R_H(ac) = C(a) + O(a, 5) - 1$$

$C(\$) = 0$
→

$C(a) = 1$
→

$C(c) = 4$
→

$C(g) = 6$
→

$C(t) = 7$
→

								a	c	g	t
\$ ₁	a	c	a	a	c	g ₁		0	0	1	0
a ₁	a	c	g	\$	a	c ₁		0	1	1	0
a ₂	c	a	a	c	g	\$ ₁		0	1	1	0
a ₃	c	g	\$	a	c	a ₁		1	1	1	0
c ₁	a	a	c	g	\$	a ₂		2	1	1	0
c ₂	g	\$	a	c	a	a ₃		3	1	1	0
g ₁	\$	a	c	a	a	c ₁		3	2	1	0

Pattern search

a a c

$$R_L(ac) = C(a) + O(a, 3)$$

$$R_H(ac) = C(a) + O(a, 5) - 1$$

$$C(\$) = 0 \rightarrow$$

$$C(a) = 1 \rightarrow$$

$$C(c) = 4 \rightarrow$$

$$C(g) = 6 \rightarrow$$

$$C(t) = 7 \rightarrow$$

								a	c	g	t
\$ ₁	a	c	a	a	c	g ₁		0	0	1	0
a ₁	a	c	g	\$	a	c ₁		0	1	1	0
a ₂	c	a	a	c	g	\$ ₁		0	1	1	0
a ₃	c	g	\$	a	c	a ₁		1	1	1	0
c ₁	a	a	c	g	\$	a ₂		2	1	1	0
c ₂	g	\$	a	c	a	a ₃		3	1	1	0
g ₁	\$	a	c	a	a	c ₁		3	2	1	0

Pattern search

a a c

$$R_L(ac) = C(a) + 1$$

$$R_H(ac) = C(a) + 3 - 1$$

$$C(\$) = 0$$

$$C(a) = 1$$

$$C(c) = 4$$

$$C(g) = 6$$

$$C(t) = 7$$

								a	c	g	t
\$ ₁	a	c	a	a	c	g ₁		0	0	1	0
a ₁	a	c	g	\$	a	c ₁		0	1	1	0
a ₂	c	a	a	c	g	\$ ₁		0	1	1	0
a ₃	c	g	\$	a	c	a ₁		1	1	1	0
c ₁	a	a	c	g	\$	a ₂		2	1	1	0
c ₂	g	\$	a	c	a	a ₃		3	1	1	0
g ₁	\$	a	c	a	a	c ₁		3	2	1	0

Pattern search

a a c

$$R_L(ac) = C(a) + 1$$

$$R_H(ac) = C(a) + 3 - 1$$

$C(\$) = 0$
→

$C(a) = 1$
→

$C(c) = 4$
→

$C(g) = 6$
→

$C(t) = 7$
→

								a	c	g	t
\$ ₁	a	c	a	a	c	g ₁		0	0	1	0
a ₁	a	c	g	\$	a	c ₁		0	1	1	0
a ₂	c	a	a	c	g	\$ ₁		0	1	1	0
a ₃	c	g	\$	a	c	a ₁		1	1	1	0
c ₁	a	a	c	g	\$	a ₂		2	1	1	0
c ₂	g	\$	a	c	a	a ₃		3	1	1	0
g ₁	\$	a	c	a	a	c ₁		3	2	1	0

Pattern search

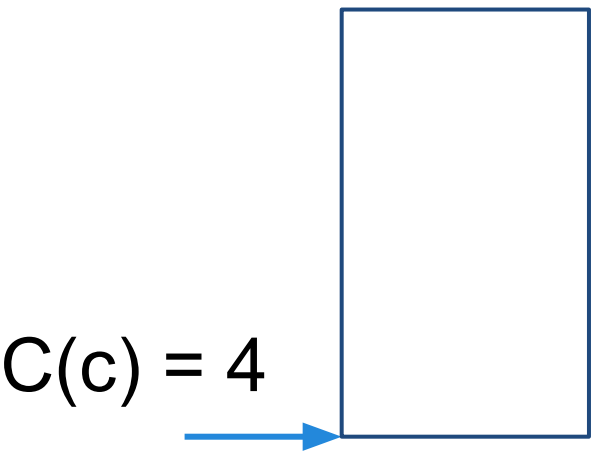
a a c

$R_L(ac) = 1 + 1 = 2$

$R_H(ac) = 1 + 3 - 1 = 3$

$C(\$) = 0$

$C(a) = 1$



$C(g) = 6$

$C(t) = 7$

								a	c	g	t
\$ ₁	a	c	a	a	c	g ₁		0	0	1	0
a ₁	a	c	g	\$	a	c ₁		0	1	1	0
a ₂	c	a	a	c	g	\$ ₁		0	1	1	0
a ₃	c	g	\$	a	c	a ₁		1	1	1	0
c ₁	a	a	c	g	\$	a ₂		2	1	1	0
c ₂	g	\$	a	c	a	a ₃		3	1	1	0
g ₁	\$	a	c	a	a	c ₁		3	2	1	0

Pattern search

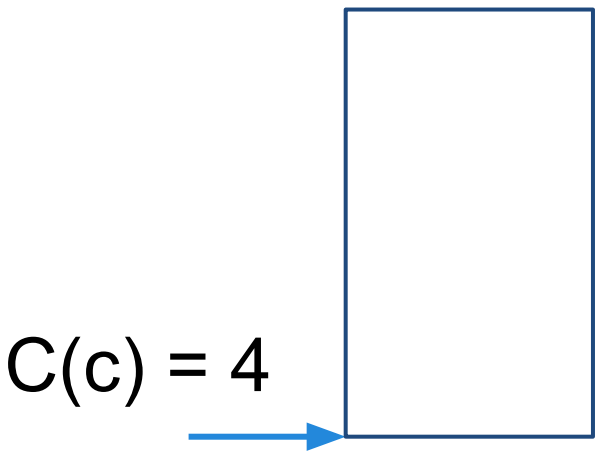
a a c

$$R_L(aac) = C(a) + O(a, R_L(ac)) - 1$$

$$R_H(aac) = C(a) + O(a, R_H(ac)) - 1$$

$C(\$) = 0$
→

$C(a) = 1$
→



$C(g) = 6$
→

$C(t) = 7$
→

								a	c	g	t
$\$$ ₁	a	c	a	a	c	g ₁		0	0	1	0
a ₁	a	c	g	$\$$	a	c ₁		0	1	1	0
a ₂	c	a	a	c	g	$\\$ ₁		0	1	1	0
a ₃	c	g	$\$$	a	c	a ₁		1	1	1	0
c ₁	a	a	c	g	$\$$	a ₂		2	1	1	0
c ₂	g	$\$$	a	c	a	a ₃		3	1	1	0
g ₁	$\$$	a	c	a	a	c ₁		3	2	1	0

Pattern search

a a c

$$R_L(aac) = C(a) + O(a, 2 - 1)$$

$$R_H(aac) = C(a) + O(a, 3) - 1$$

$$C(\$) = 0 \rightarrow$$

$$C(a) = 1 \rightarrow$$

$$C(c) = 4 \rightarrow$$

$$C(g) = 6 \rightarrow$$

$$C(t) = 7 \rightarrow$$

								a	c	g	t
\$ ₁	a	c	a	a	c	g ₁		0	0	1	0
a ₁	a	c	g	\$	a	c ₁		0	1	1	0
a ₂	c	a	a	c	g	\$ ₁		0	1	1	0
a ₃	c	g	\$	a	c	a ₁		1	1	1	0
c ₁	a	a	c	g	\$	a ₂		2	1	1	0
c ₂	g	\$	a	c	a	a ₃		3	1	1	0
g ₁	\$	a	c	a	a	c ₂		3	2	1	0

Pattern search

a a c

$$R_L(aac) = C(a) + 0$$

$$R_H(aac) = C(a) + 1 - 1$$

$$C(\$) = 0$$

$$C(a) = 1$$

$$C(c) = 4$$

$$C(g) = 6$$

$$C(t) = 7$$

								a	c	g	t
\$ ₁	a	c	a	a	c	g ₁		0	0	1	0
a ₁	a	c	g	\$	a	c ₁		0	1	1	0
a ₂	c	a	a	c	g	\$ ₁		0	1	1	0
a ₃	c	g	\$	a	c	a ₁		1	1	1	0
c ₁	a	a	c	g	\$	a ₂		2	1	1	0
c ₂	g	\$	a	c	a	a ₃		3	1	1	0
g ₁	\$	a	c	a	a	c ₂		3	2	1	0


Pattern search

a a c

$$R_L(aac) = 1 + 0 = 1$$

$$R_H(aac) = 1 + 1 - 1 = 1$$

$C(\$) = 0$
→

$C(a) = 1$
→ 

$C(c) = 4$
→

$C(g) = 6$
→

$C(t) = 7$
→

								a	c	g	t
$\$$ ₁	a	c	a	a	c	g ₁		0	0	1	0
a ₁	a	c	g	$\$$	a	c ₁		0	1	1	0
a ₂	c	a	a	c	g	$\\$ ₁		0	1	1	0
a ₃	c	g	$\$$	a	c	a ₁		1	1	1	0
c ₁	a	a	c	g	$\$$	a ₂		2	1	1	0
c ₂	g	$\$$	a	c	a	a ₃		3	1	1	0
g ₁	$\$$	a	c	a	a	c ₁		3	2	1	0

Pattern search

a a c

$R_L(aac) = 1$

$R_H(aac) = 1$



$C(\$) = 0$

$C(a) = 1$



$C(c) = 4$

$C(g) = 6$

$C(t) = 7$

		a	c	g	t					
$\$$ ₁	a	c	a	a	c	g ₁	0	0	1	0
a ₁	a	c	g	\$	a	c ₁	0	1	1	0
a ₂	c	a	a	c	g	\$ ₁	0	1	1	0
a ₃	c	g	\$	a	c	a ₁	1	1	1	0
c ₁	a	a	c	g	\$	a ₂	2	1	1	0
c ₂	g	\$	a	c	a	a ₃	3	1	1	0
g ₁	\$	a	c	a	a	c ₁	3	2	1	0

Pattern search

For how long does it work?

Pattern search

For how long does it work?

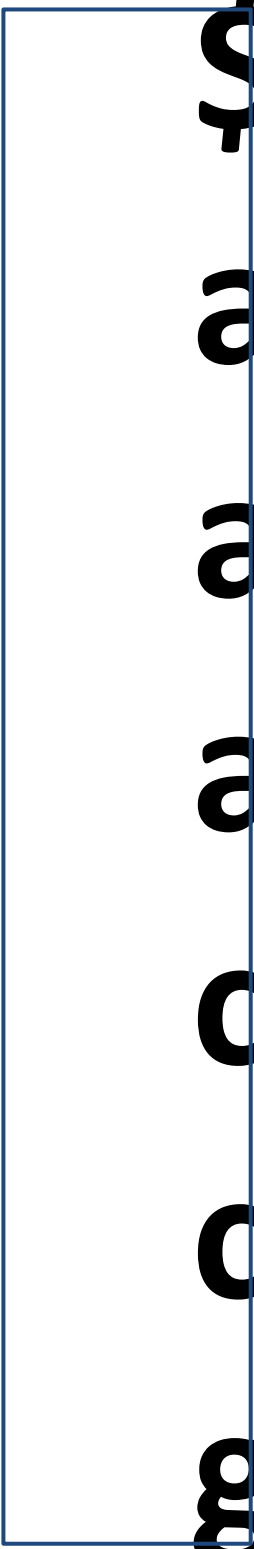
$O(m)$

Pattern search with errors

$\$_1$	a	c	a	a	c	g_1
a_1	a	c	g	$\$$	a	c_1
a_2	c	a	a	c	g	$\$_1$
a_3	c	g	$\$$	a	c	a_1
c_1	a	a	c	g	$\$$	a_2
c_2	g	$\$$	a	c	a	a_3
g_1	$\$$	a	c	a	a	c_2

g c a

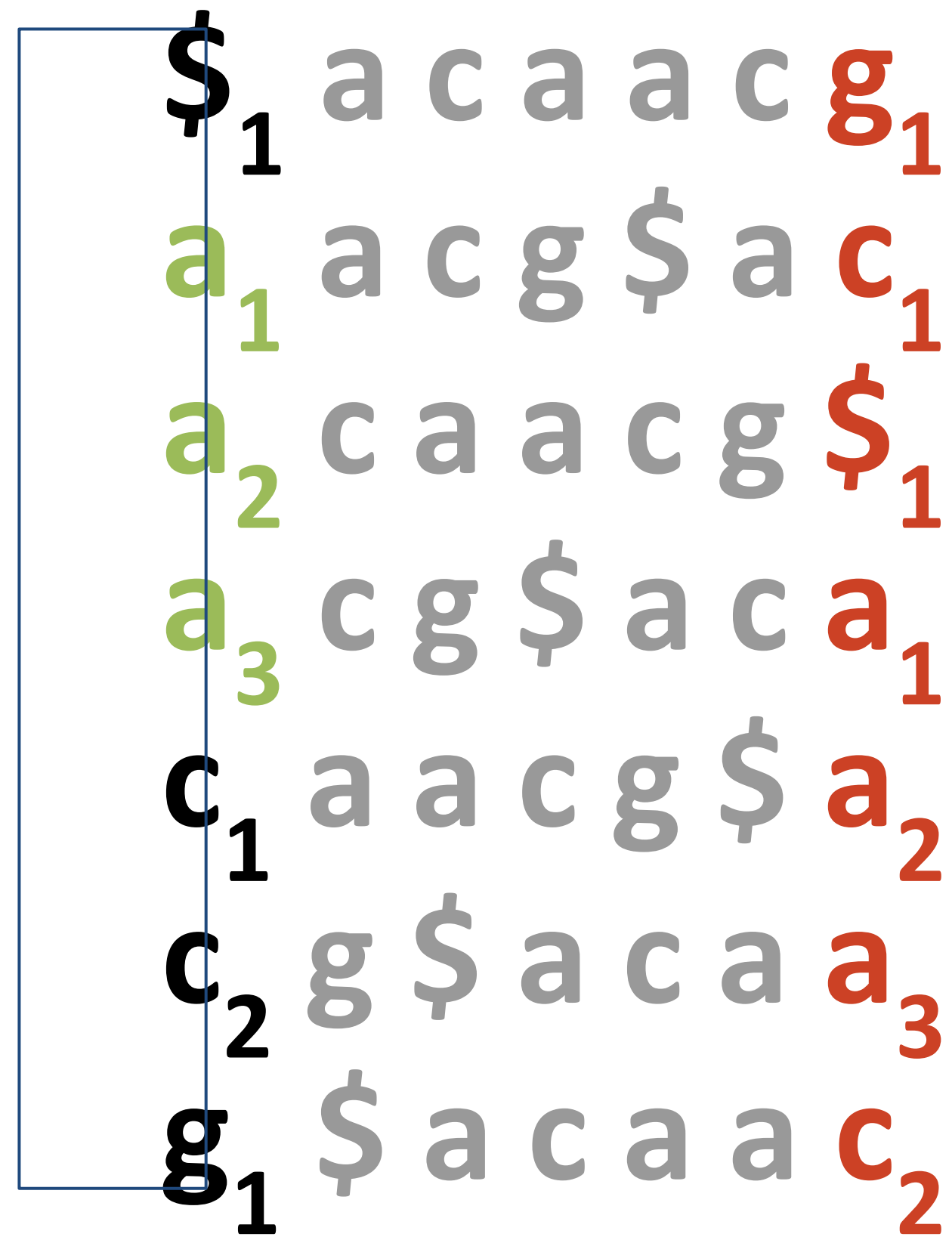
Pattern search with errors



$\$$ ₁	a	c	a	a	c	g ₁
a ₁	a	c	g	$\$$	a	c ₁
a ₂	c	a	a	c	g	$\$$ ₁
a ₃	c	g	$\$$	a	c	a ₁
c ₁	a	a	c	g	$\$$	a ₂
c ₂	g	$\$$	a	c	a	a ₃
g ₁	$\$$	a	c	a	a	c ₂

g c a


Pattern search with errors



$\$_1$	a	c	a	a	c	g_1
a_1	a	c	g	$\$$	a	c_1
a_2	c	a	a	c	g	$\$_1$
a_3	c	g	$\$$	a	c	a_1
c_1	a	a	c	g	$\$$	a_2
c_2	g	$\$$	a	c	a	a_3
g_1	$\$$	a	c	a	a	c_2

g c a

Pattern search with errors

	$\$$ ₁	a	c	a	a	c	g	₁
	a	a	c	g	\$	a	c	₁
	a	c	a	a	c	g	\$	₁
	a	c	g	\$	a	c	a	₁
	c	a	a	c	g	\$	a	₂
	c	g	\$	a	c	a	a	₃
	g	\$	a	c	a	a	c	₂

g c a

Pattern search with errors

	$\$$ ₁	a	c	a	a	c	g	₁
<div></div>	a	a	c	g	$\$$	a	c	₁
	a	c	a	a	c	g	$\$$	₁
	a	c	g	$\$$	a	c	a	₁
	c	a	a	c	g	$\$$	a	₂
	c	g	$\$$	a	c	a	a	₃
	g	$\$$	a	c	a	a	c	₂

g c a

Pattern search with errors

$\$$ ₁	a	c	a	a	c	g ₁
a ₁	a	c	g	$\$$	a	c ₁
a ₂	c	a	a	c	g	$\$$ ₁
a ₃	c	g	$\$$	a	c	a ₁
c ₁	a	a	c	g	$\$$	a ₂
c ₂	g	$\$$	a	c	a	a ₃
g ₁	$\$$	a	c	a	a	c ₂

g c a

Pattern search with errors

$\$$ ₁	a	c	a	a	c	g ₁
a ₁	a	c	g	\$	a	c ₁
a ₂	c	a	a	c	g	\$ ₁
a ₃	c	g	\$	a	c	a ₁
c ₁	a	a	c	g	\$	a ₂
c ₂	g	\$	a	c	a	a ₃
g ₁	\$	a	c	a	a	c ₂

g c a

Pattern search with errors

$\$$ ₁	a	c	a	a	c	g	₁
a	a	c	g	\$	a	c	₁
a	c	a	a	c	g	\$	₁
a	c	g	\$	a	c	a	₁
c	a	a	c	g	\$	a	₂
c	g	\$	a	c	a	a	₃
g	\$	a	c	a	a	c	₂

g c a

Pattern search with errors

$\$$ ₁	a	c	a	a	c	g	₁
a	a	c	g	\$	a	c	₁
a	c	a	a	c	g	\$	₁
a	c	g	\$	a	c	a	₁
<div></div>	c	a	a	c	g	\$	a
c	g	\$	a	c	a	a	₃
g	\$	a	c	a	a	c	₂

g c a

Pattern search with errors

$\$_1$	a	c	a	a	c	g_1
a_1	a	c	g	$\$$	a	c_1
a_2	c	a	a	c	g	$\$_1$
a_3	c	g	$\$$	a	c	a_1
c_1	a	a	c	g	$\$$	a_2
c_2	g	$\$$	a	c	a	a_3
g_1	$\$$	a	c	a	a	c_2

g c a

Pattern search with errors

$\$$ ₁	a	c	a	a	c	g ₁
a ₁	a	c	g	\$	a	c ₁
a ₂	c	a	a	c	g	\$ ₁
a ₃	c	g	\$	a	c	a ₁
c ₁	a	a	c	g	\$	a ₂
c ₂	g	\$	a	c	a	a ₃
g ₁	\$	a	c	a	a	c ₂

g c a

Pattern search with errors

$\$$ ₁	a	c	a	a	c	g ₁
a ₁	a	c	g	\$	a	c ₁
a ₂	c	a	a	c	g	\$ ₁
a ₃	c	g	\$	a	c	a ₁
c ₁	a	a	c	g	\$	a ₂
c ₂	g	\$	a	c	a	a ₃
g ₁	\$	a	c	a	a	c ₂



Pattern search with errors

$\$$ ₁	a	c	a	a	c	g ₁
a ₁	a	c	g	\$	a	c ₁
a ₂	c	a	a	c	g	\$ ₁
a ₃	c	g	\$	a	c	a ₁
c ₁	a	a	c	g	\$	a ₂
c ₂	g	\$	a	c	a	a ₃
g ₁	\$	a	c	a	a	c ₂

g c a

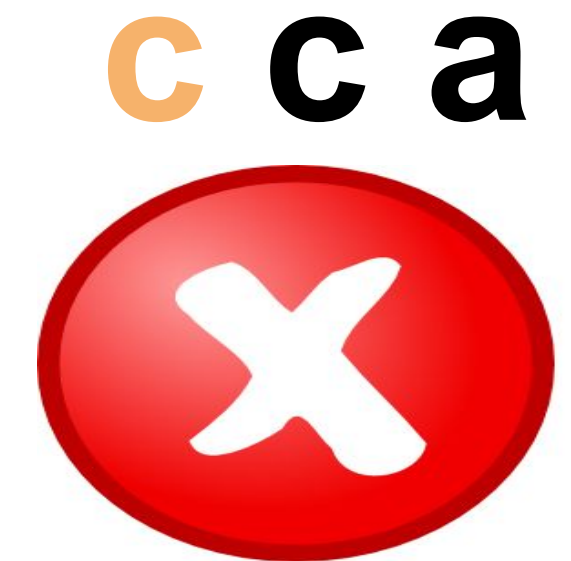
Pattern search with errors

$\$$ ₁	a	c	a	a	c	g ₁
a ₁	a	c	g	$\$$	a	c ₁
a ₂	c	a	a	c	g	$\$$ ₁
a ₃	c	g	$\$$	a	c	a ₁
c ₁	a	a	c	g	$\$$	a ₂
c ₂	g	$\$$	a	c	a	a ₃
g ₁	$\$$	a	c	a	a	c ₂

c c a

Pattern search with errors

$\$$ ₁	a	c	a	a	c	g ₁
a ₁	a	c	g	$\$$	a	c ₁
a ₂	c	a	a	c	g	$\$$ ₁
a ₃	c	g	$\$$	a	c	a ₁
c ₁	a	a	c	g	$\$$	a ₂
c ₂	g	$\$$	a	c	a	a ₃
g ₁	$\$$	a	c	a	a	c ₂



Pattern search with errors

$\$$ ₁	a	c	a	a	c	g ₁
a ₁	a	c	g	$\$$	a	c ₁
a ₂	c	a	a	c	g	$\$$ ₁
a ₃	c	g	$\$$	a	c	a ₁
c ₁	a	a	c	g	$\$$	a ₂
c ₂	g	$\$$	a	c	a	a ₃
g ₁	$\$$	a	c	a	a	c ₂

a c a

Pattern search with errors

$\$$ ₁	a	c	a	a	c	g	₁
a	a	c	g	\$	a	c	₁
a	c	a	a	c	g	\$	₁
a	c	g	\$	a	c	a	₁
c	a	a	c	g	\$	a	₂
c	g	\$	a	c	a	a	₃
g	\$	a	c	a	a	c	₂

a c a

Pattern search with errors

$\$$ ₁	a	c	a	a	c	g	₁
a	a	c	g	$\$$	a	c	₁
<div></div>	a	c	a	a	c	g	$\$$ ₁
a	c	g	$\$$	a	c	a	₁
c	a	a	c	g	$\$$	a	<div></div>
c	g	$\$$	a	c	a	a	₃
g	$\$$	a	c	a	a	c	₂

a c a

Pattern search with errors

$\$$ ₁	a	c	a	a	c	g ₁	
a ₁	a	c	g	\$	a	c ₁	
<div></div>	a ₂	c	a	a	c	g	\$ ₁
a ₃	c	g	\$	a	c	a ₁	
c ₁	a	a	c	g	\$	a ₂	
c ₂	g	\$	a	c	a	a ₃	
g ₁	\$	a	c	a	a	c ₂	

