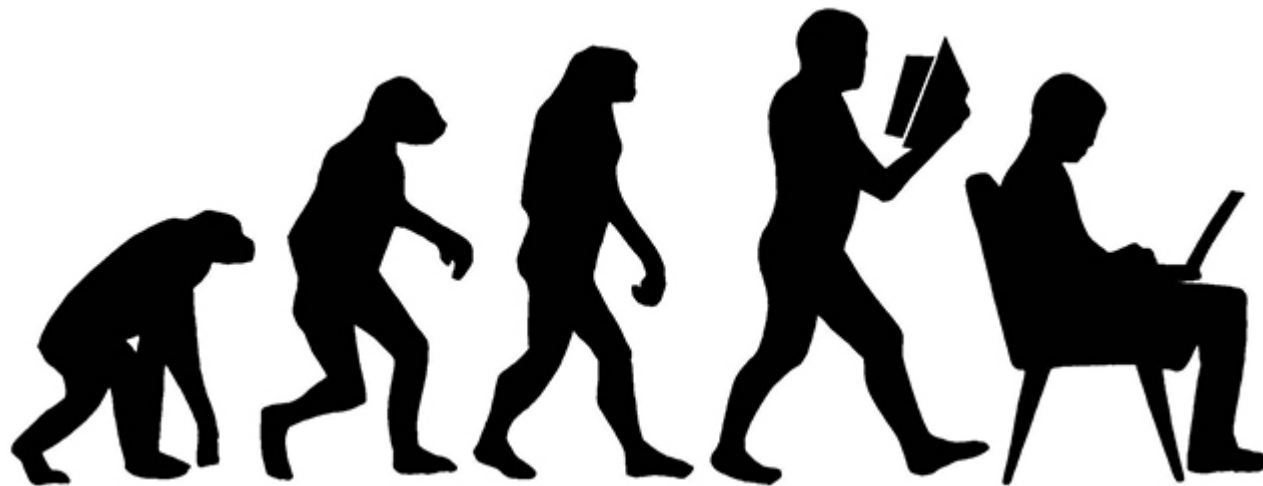


Поиск эволюционирующих подпоследовательностей (цепочек) временного ряда



*Эволюция человека — это цепь
непрерывных сомнений.*

С. Михалюк

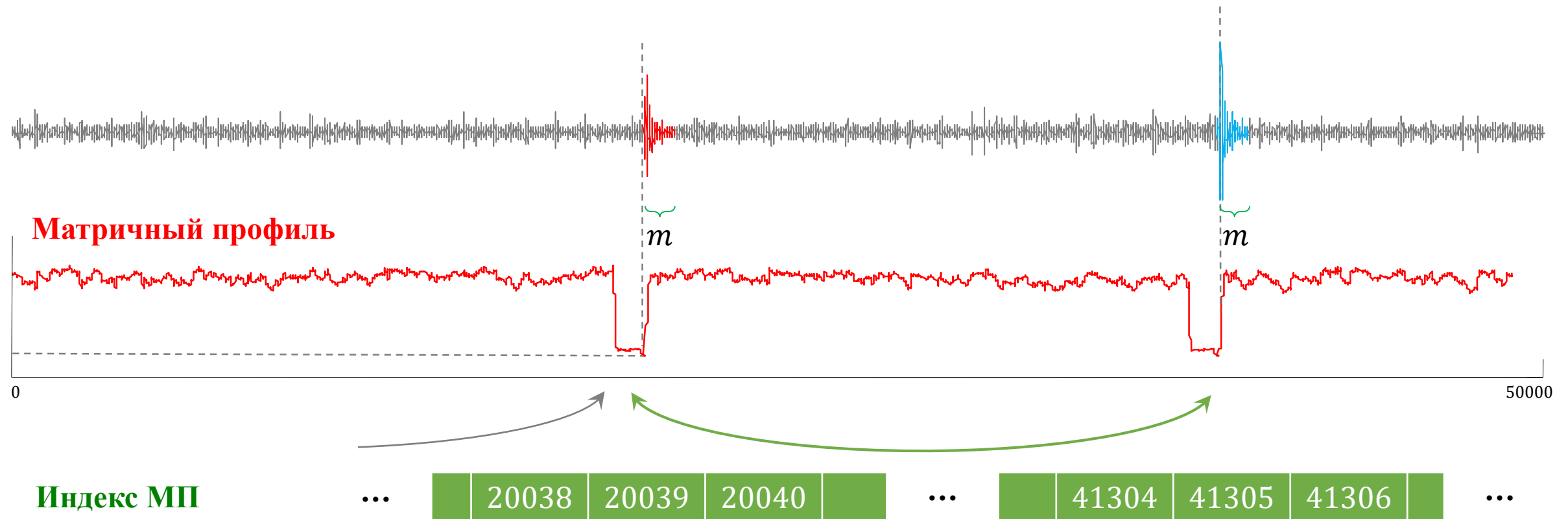
Содержание

- Понятие цепочки
- Алгоритмы поиска цепочек
- Примеры применения

Матричный профиль ряда (напоминание)

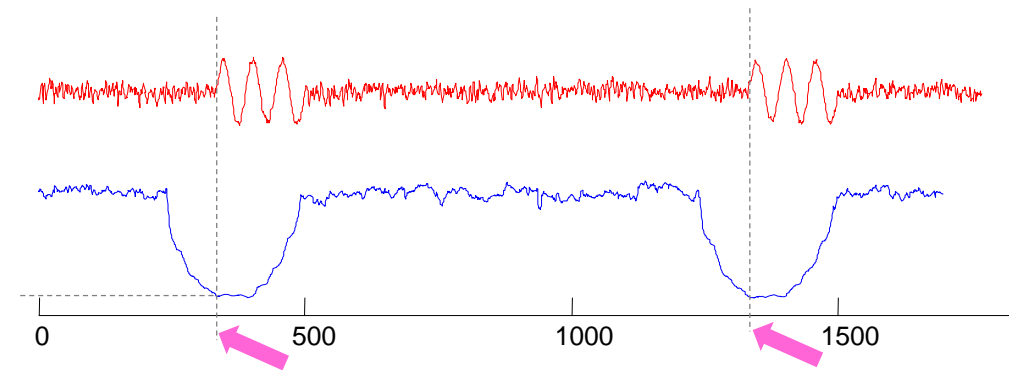
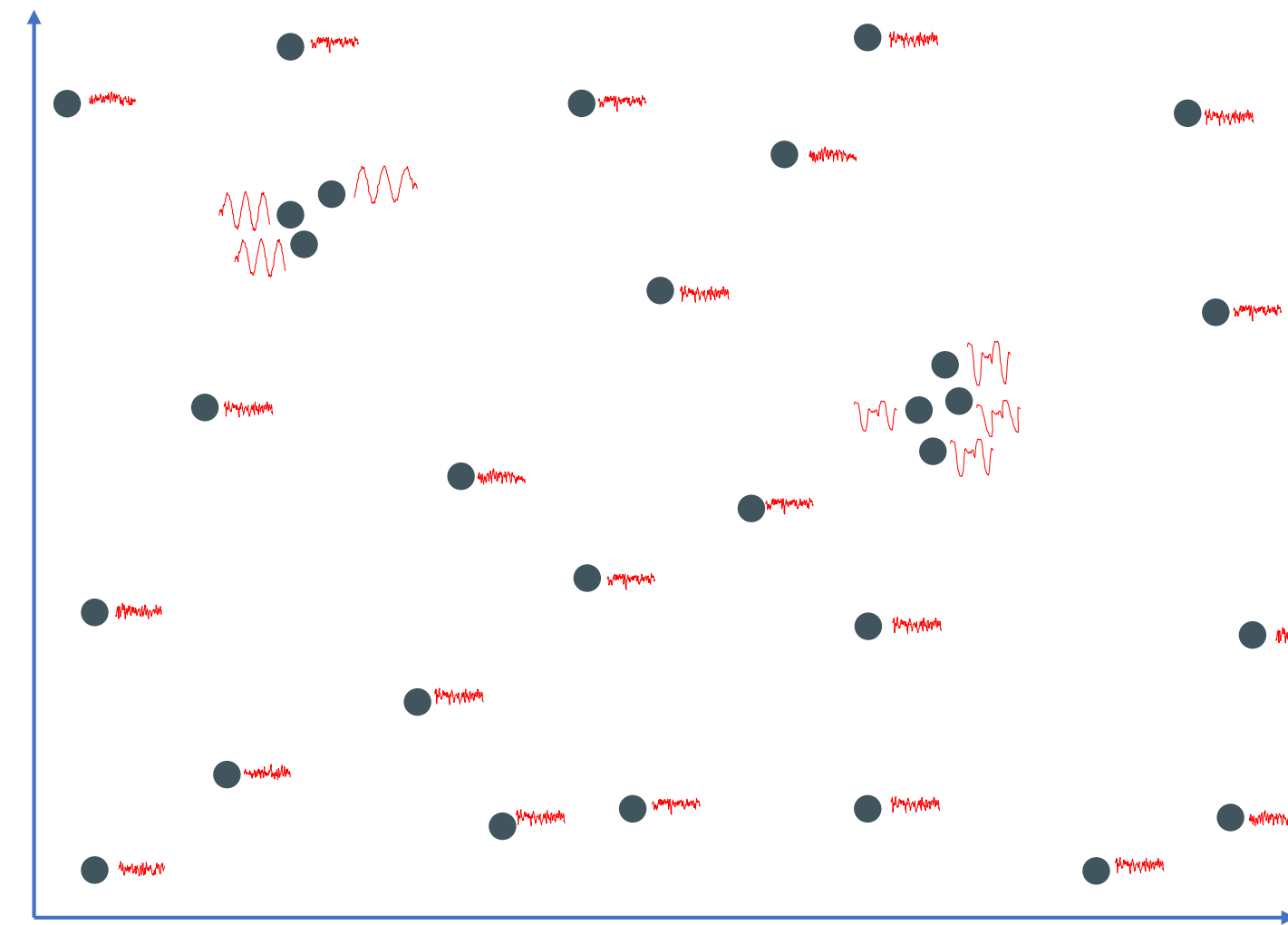
Матричный профиль показывает **расстояние** от каждой подпоследовательности до ее **ближайшего соседа**

Индекс матричного профиля показывает **место ближайшего соседа** каждой подпоследовательности



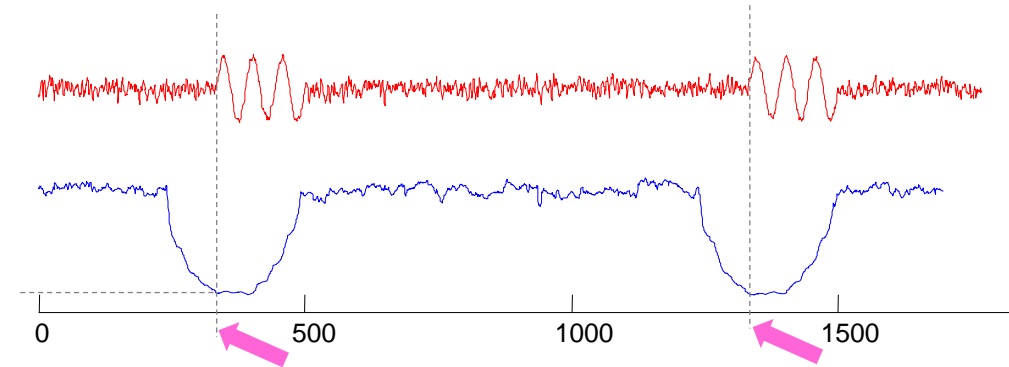
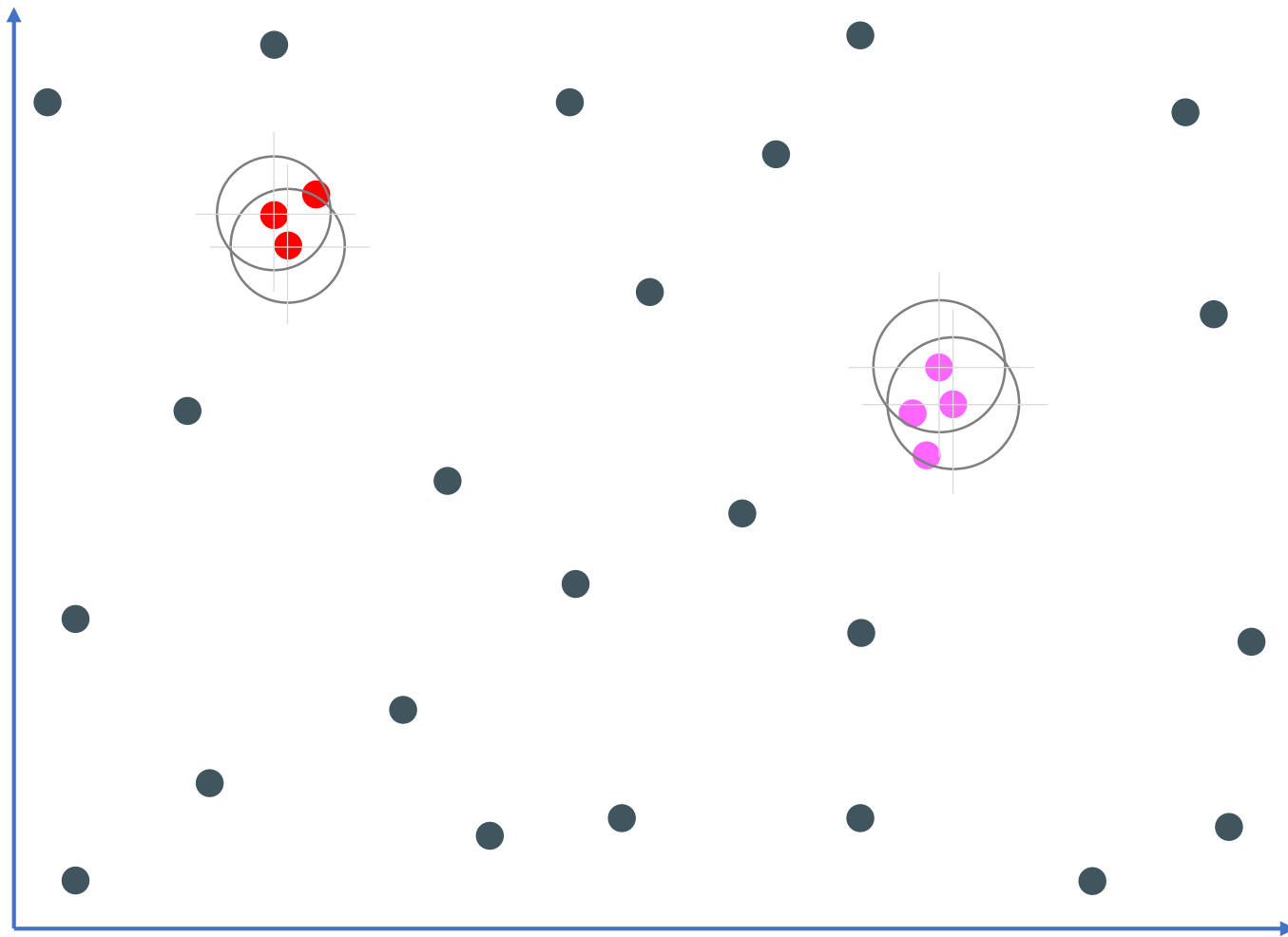
Минимумы МП соответствуют мотивам ряда

Упрощенное отображение мотивов ряда



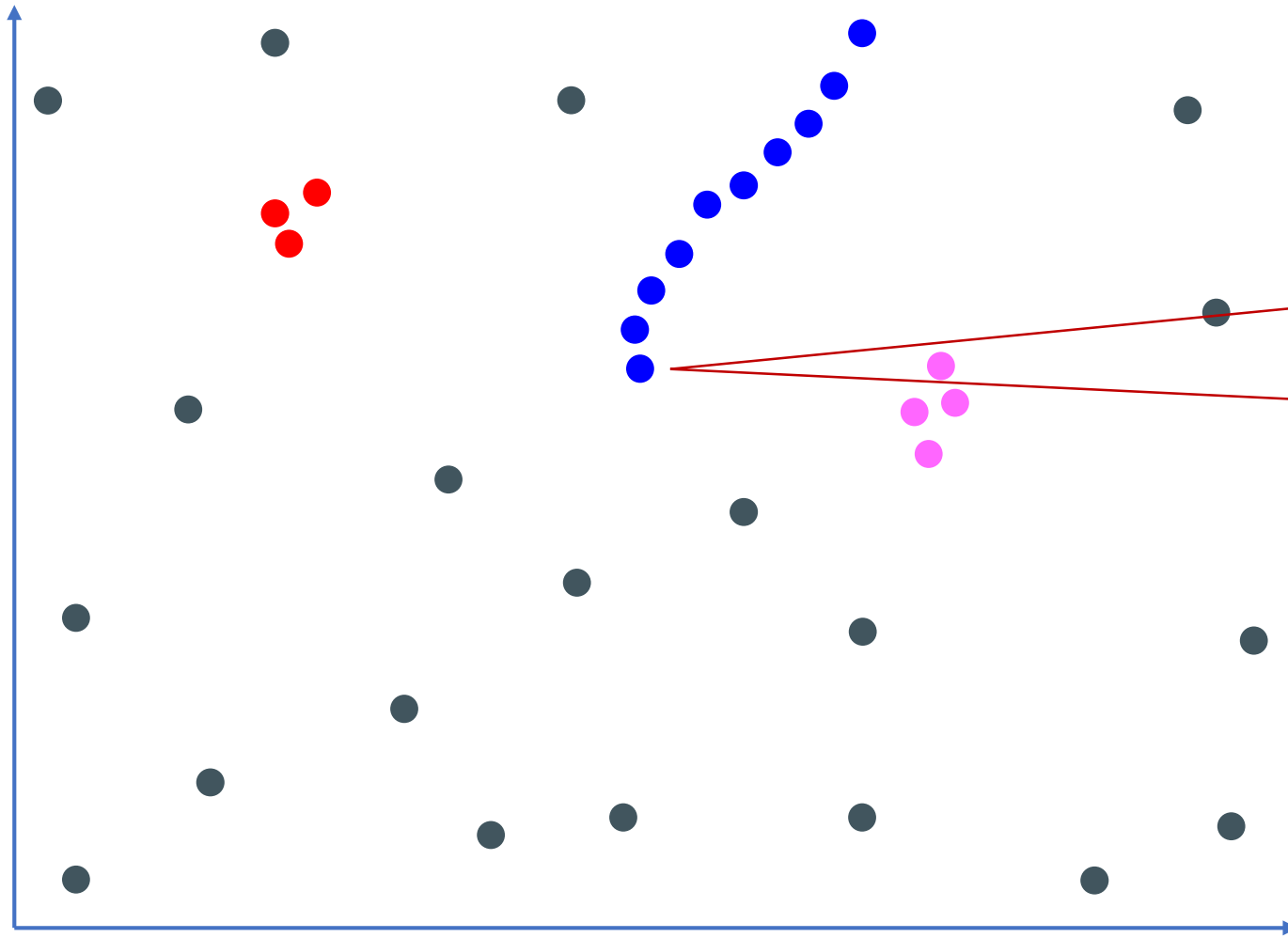
- Подпоследовательности – точки пространства \mathbb{R}^m
- Скопления соответствуют участкам ряда с локальными минимумами матричного профиля

Упрощенное отображение top-2 мотивов ряда



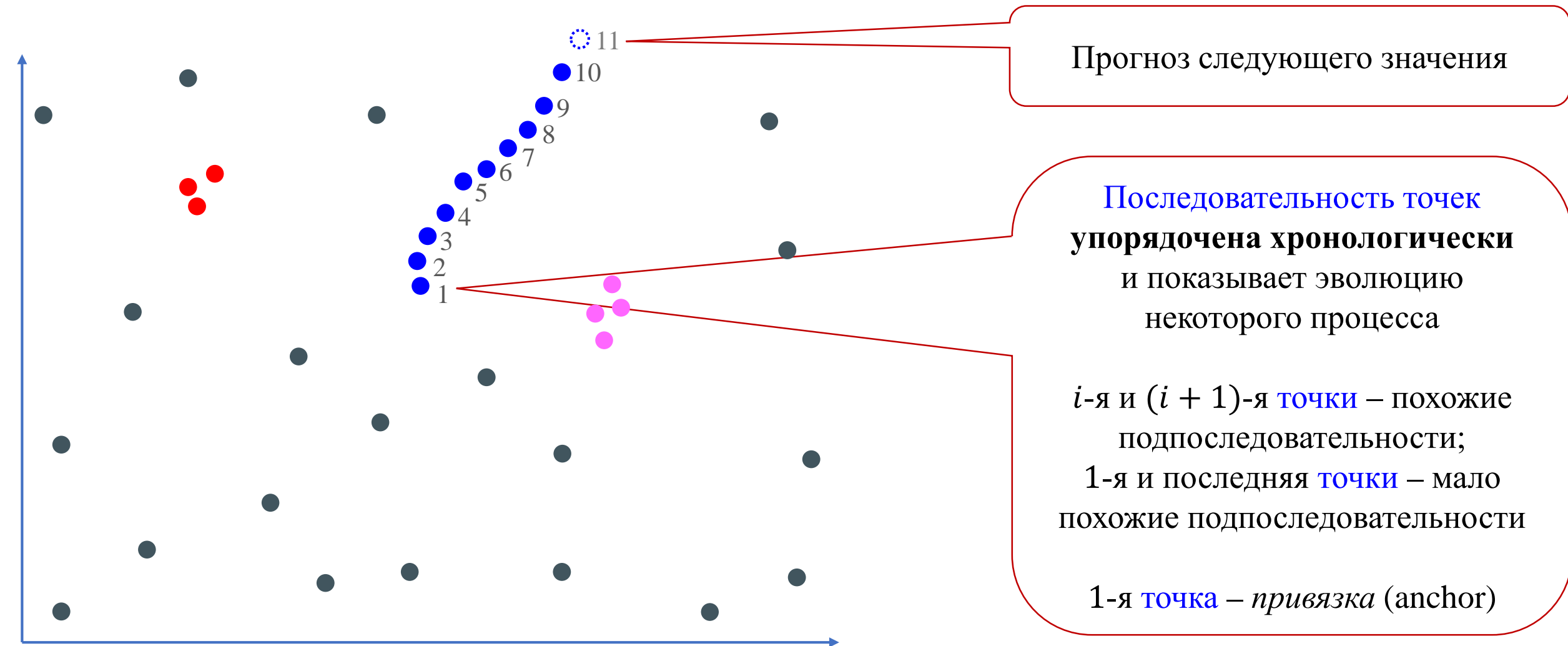
- Подпоследовательности – точки пространства \mathbb{R}^m
- Скопления соответствуют участкам ряда с локальными минимумами матричного профиля

От мотивов ряда к цепочкам



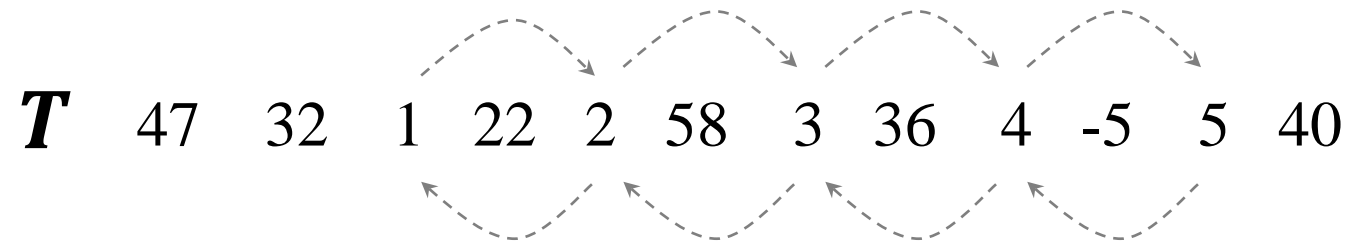
Последовательность точек
(не обязательно из одного мотива),
вероятно, формирующих
множество мотивов

От мотивов ряда к цепочкам



Определение цепочки

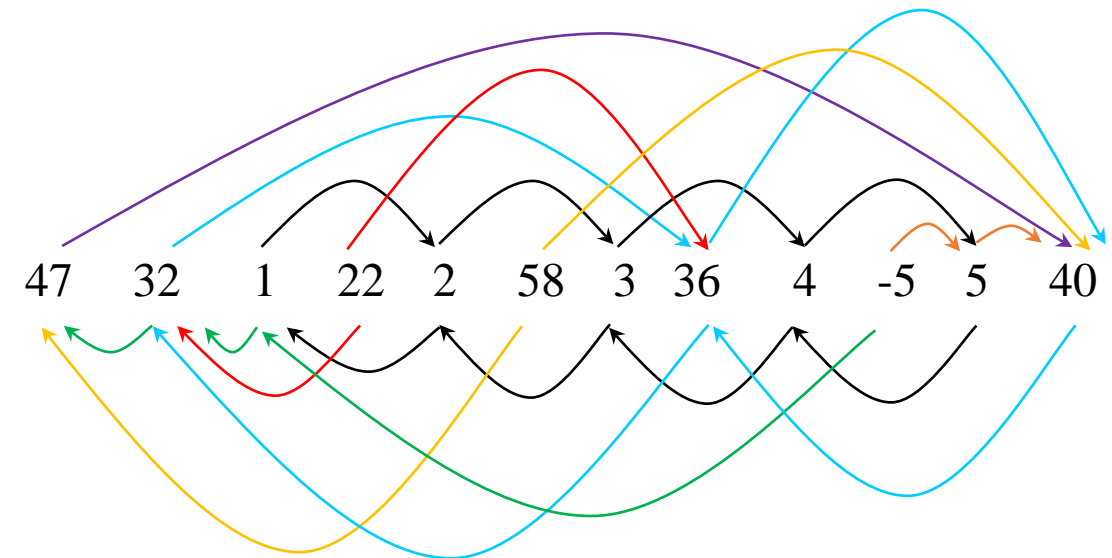
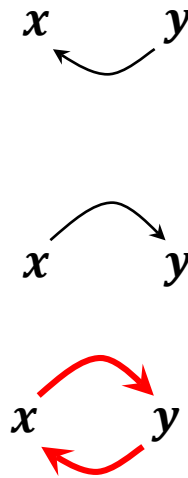
Упрощение: $m = 1$, $dist(T_{i,m}, T_{j,m}) = |t_i - t_j|$



Определение цепочки

- $LNN(\cdot)/RNN(\cdot)$ – ближайший сосед слева/справа от данной подпоследовательности
- Если x и y – два последовательных звена цепочки, то $y = RNN(x)$, $x = LNN(y)$

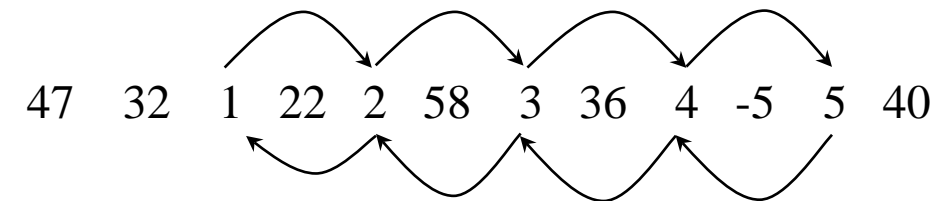
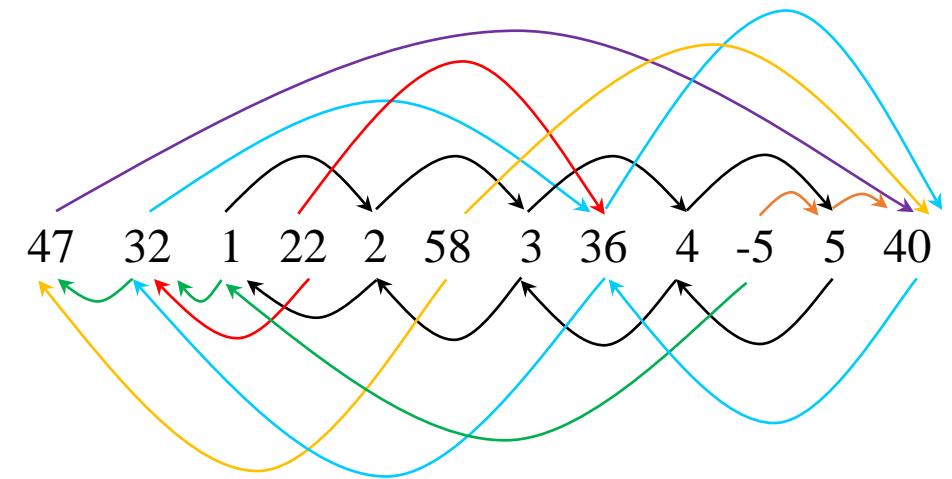
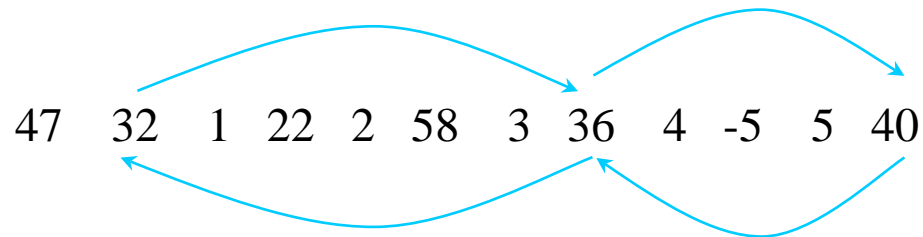
- Ближайший сосед слева LNN
- Ближайший сосед справа RNN
- Последовательные звенья соединены петлей



Определение цепочки

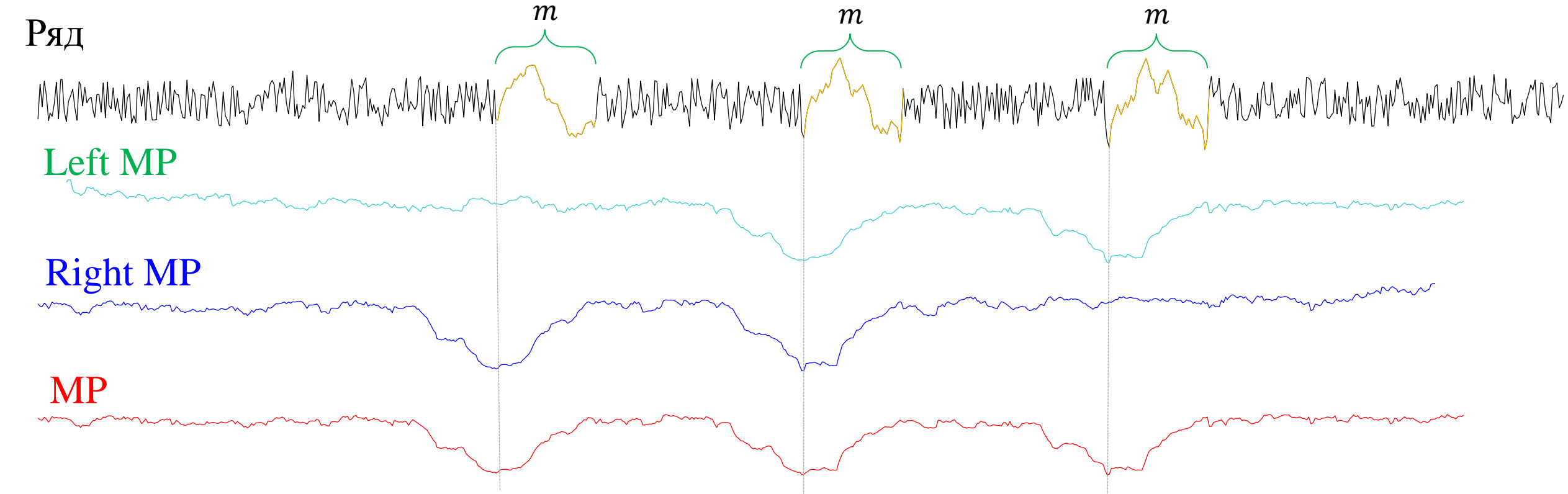
- $LNN(\cdot)/RNN(\cdot)$ – ближайший сосед слева/справа от данной подпоследовательности
- Если x и y – два последовательных звена цепочки, то $y = RNN(x)$, $x = LNN(y)$

- Ближайший сосед слева LNN
- Ближайший сосед справа RNN
- Последовательные звенья соединены петлей



Левый и правый матричные профили

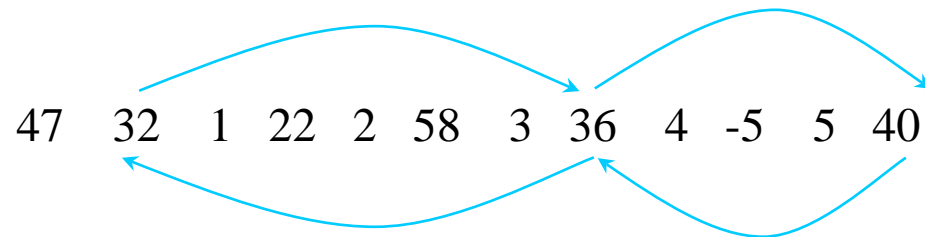
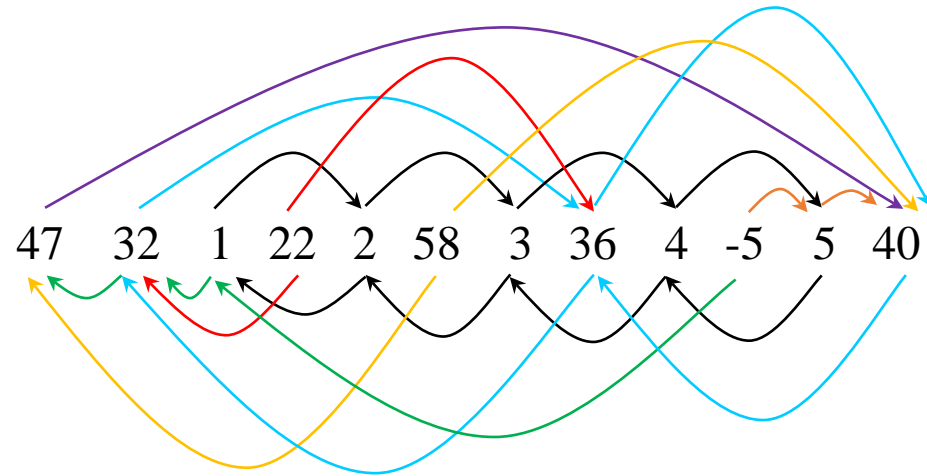
Левый (правый) матричный профиль и его индекс хранит информацию о ближайшем соседе слева (справа)



Левый и правый матричные профили

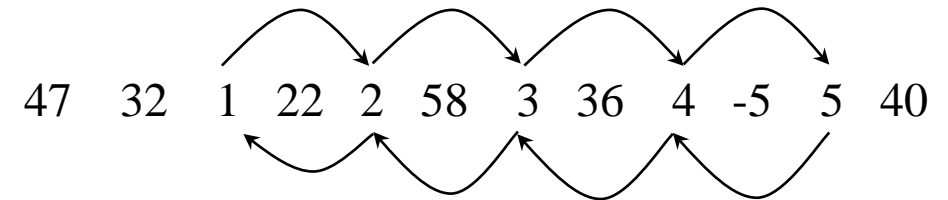
- Левый профиль расстояния для $T_{i,m}$: $DL_i = \{d_{i,l}\}_{l=1}^{i-m/4}$ $d_{i,j} = \text{ED}_{\text{norm}}(T_{i,m}, T_{j,m})$
- Правый профиль расстояния для $T_{i,m}$: $DR_i = \{d_{i,r}\}_{r=i+m/4}^{n-m+1}$
- Ближайший сосед слева для $T_{i,m}$: $LNN(T_{i,m}) = T_{j,m}$, $d_{i,j} = \min(DL_i)$
- Ближайший сосед справа для $T_{i,m}$: $RNN(T_{i,m}) = T_{j,m}$, $d_{i,j} = \min(DR_i)$
- Левый матричный профиль T : $PL = \{\min(DL_i)\}_{i=1}^{n-m+1}$
- Индекс левого матричного профиля T : $IL = \{IL_i\}_{i=1}^{n-m+1}$, $IL_i = j$ при $LNN(T_{i,m}) = T_{j,m}$
- Правый матричный профиль T : $PR = \{\min(DR_i)\}_{i=1}^{n-m+1}$
- Индекс левого матричного профиля T : $IR = \{IR_i\}_{i=1}^{n-m+1}$, $IR_i = j$ при $RNN(T_{i,m}) = T_{j,m}$
- Цепочка T : $TSC = \{T_{C_i,m}\}_{i=1}^k$, $\forall i \in 1..k-1$
 $C_i \leq C_{i+1}$, $RNN(T_{C_i,m}) = T_{C_{i+1},m}$, $LNN(T_{C_{i+1},m}) = T_{C_i,m}$

Закрепленные (anchored) и незакрепленные (unanchored) цепочки



Закрепленная цепочка:

$$32 \rightleftharpoons 36 \rightleftharpoons 40$$



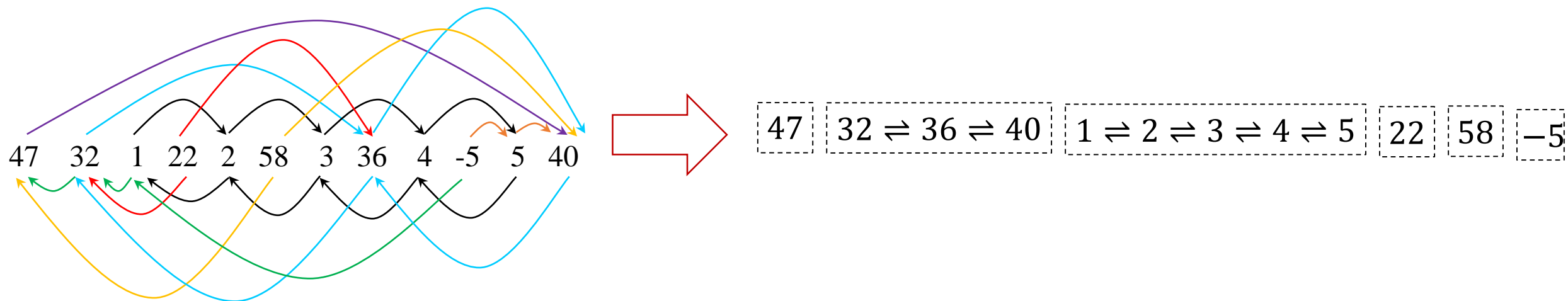
Незакрепленная цепочка

имеет max длину среди всех цепочек ряда:

$$1 \rightleftharpoons 2 \rightleftharpoons 3 \rightleftharpoons 4 \rightleftharpoons 5$$

Множество всех цепочек (All-Chain Set)

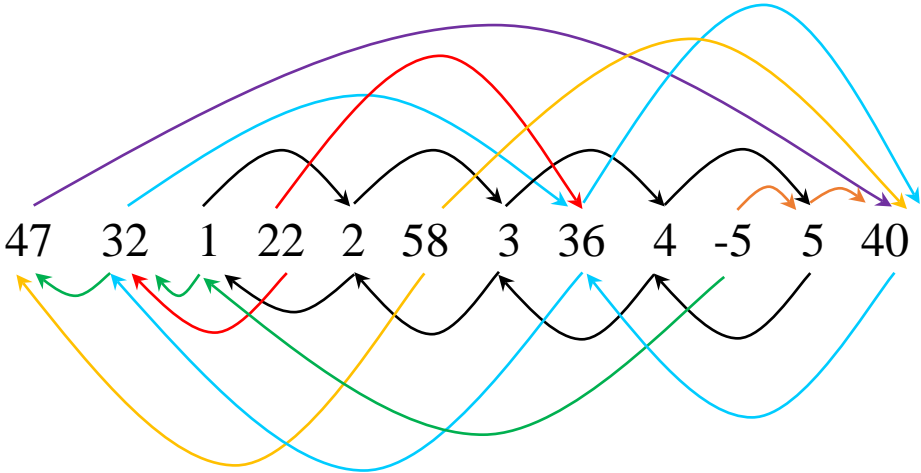
- Множество всех закрепленных цепочек, каждая из которых не включена в какую-либо другую цепочку
- Свойства МВЦ
 1. Включает в себя все элементы (подпоследовательности) ряда
 2. Каждый элемент (подпоследовательность) включен единожды



Содержание

- Понятие цепочки
- **Алгоритмы поиска цепочек**
- Примеры применения

Вычисление множества всех цепочек



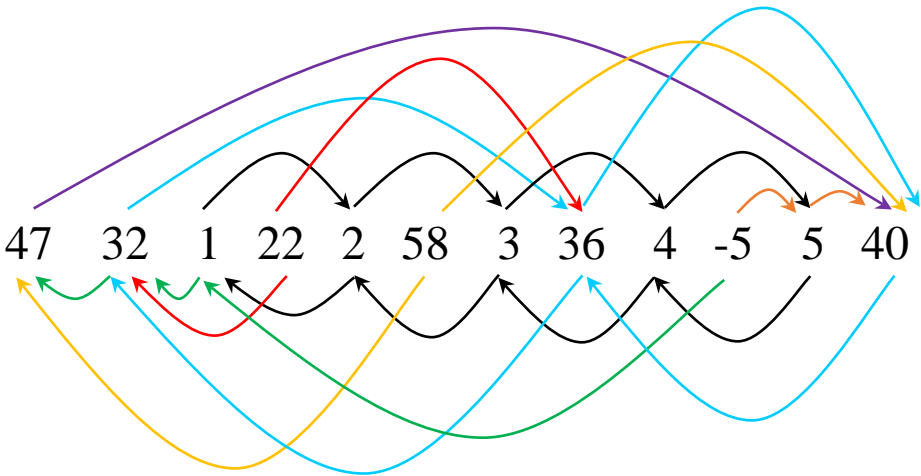
	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	47	32	1	22	2	58	3	36	4	-5	5	40
<i>IR</i>	12	8	5	8	7	12	9	12	11	11	12	0
<i>IL</i>	0	1	2	2	3	1	5	2	7	3	9	8

Текущая цепочка

Элемент	47	32	1	22	2	58	3	36	4	-5	5	40
Обработан?	0	0	0	0	0	0	0	0	0	0	0	0

Множество всех
цепочек

Вычисление множества всех цепочек



	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	47	32	1	22	2	58	3	36	4	−5	5	40
<i>IR</i>	12	8	5	8	7	12	9	12	11	11	12	0
<i>IL</i>	0	1	2	2	3	1	5	2	7	3	9	8

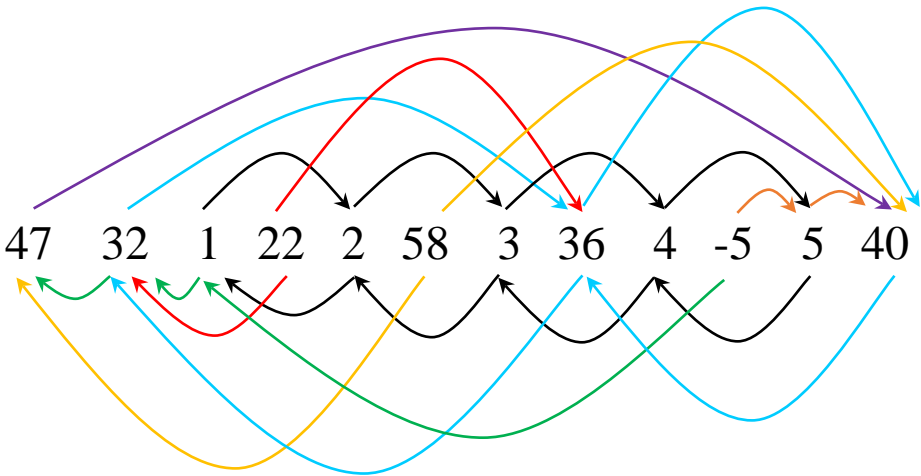
Текущая цепочка

Элемент	47	32	1	22	2	58	3	36	4	-5	5	40
Обработан?	0	0	0	0	0	0	0	0	0	0	0	0

Множество всех цепочек

--

Вычисление множества всех цепочек



	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	47	32	1	22	2	58	3	36	4	−5	5	40
<i>IR</i>	12	8	5	8	7	12	9	12	11	11	12	0
<i>IL</i>	0	1	2	2	3	1	5	2	7	3	9	8

Текущая цепочка

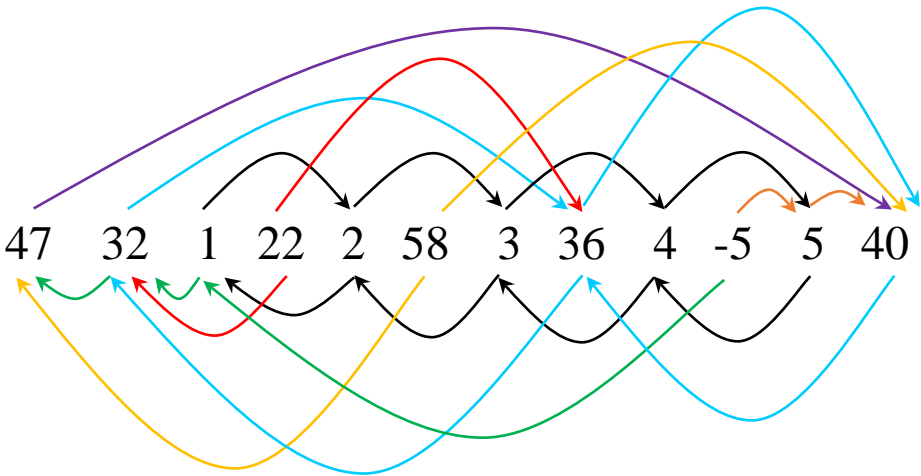
47

Элемент	47	32	1	22	2	58	3	36	4	-5	5	40
Обработан?	1	0	0	0	0	0	0	0	0	0	0	0

Множество всех цепочек

47

Вычисление множества всех цепочек



	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	47	32	1	22	2	58	3	36	4	−5	5	40
<i>IR</i>	12	8	5	8	7	12	9	12	11	11	12	0
<i>IL</i>	0	1	2	2	3	1	5	2	7	3	9	8

Текущая цепочка

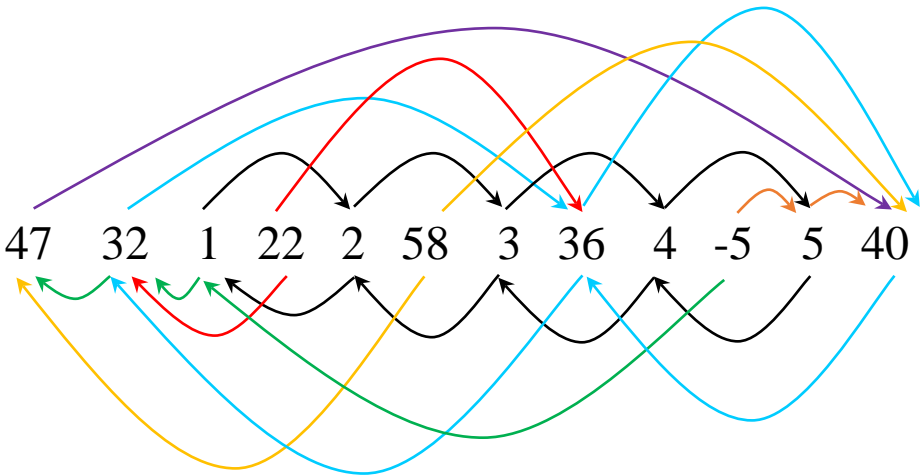
47

Элемент	47	32	1	22	2	58	3	36	4	−5	5	40
Обработан?	1	0	0	0	0	0	0	0	0	0	0	0

Множество всех цепочек

47

Вычисление множества всех цепочек



	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	47	32	1	22	2	58	3	36	4	−5	5	40
<i>IR</i>	12	8	5	8	7	12	9	12	11	11	12	0
<i>IL</i>	0	1	2	2	3	1	5	2	7	3	9	8

Текущая цепочка

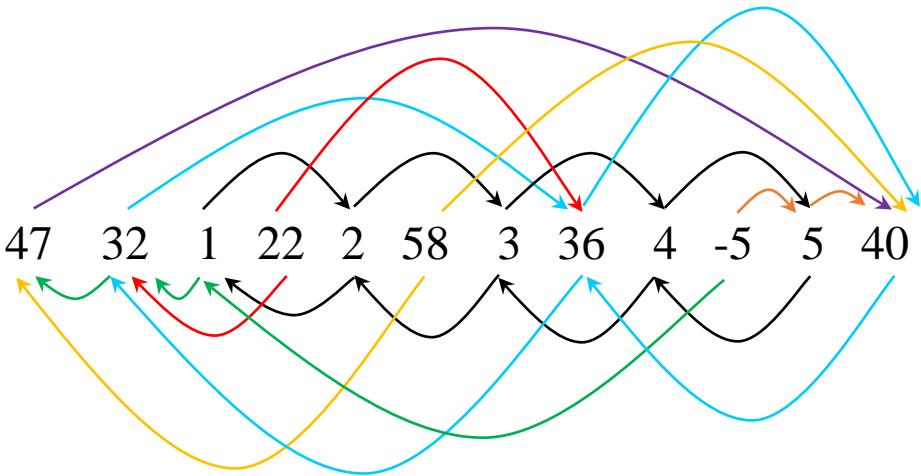
32 ⇒ 36 ⇒ 40

Элемент	47	32	1	22	2	58	3	36	4	-5	5	40
Обработан?	1	1	0	0	0	0	0	1	0	0	0	1

Множество всех цепочек

47 32 ⇒ 36 ⇒ 40

Вычисление множества всех цепочек



	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	47	32	1	22	2	58	3	36	4	−5	5	40
<i>IR</i>	12	8	5	8	7	12	9	12	11	11	12	0
<i>IL</i>	0	1	2	2	3	1	5	2	7	3	9	8

Текущая цепочка

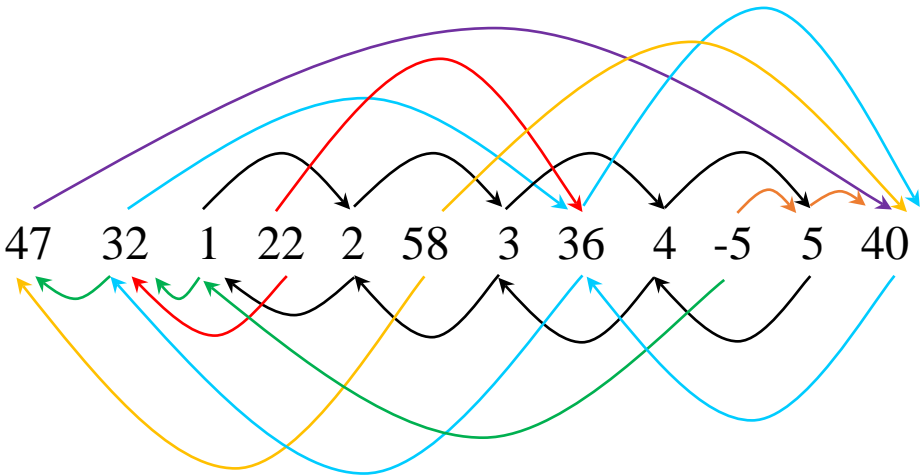
32 ⇒ 36 ⇒ 40

Элемент	47	32	1	22	2	58	3	36	4	−5	5	40
Обработан?	1	1	0	0	0	0	0	1	0	0	0	1

Множество всех цепочек

47 32 ⇒ 36 ⇒ 40

Вычисление множества всех цепочек



	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	47	32	1	22	2	58	3	36	4	−5	5	40
<i>IR</i>	12	8	5	8	7	12	9	12	11	11	12	0
<i>IL</i>	0	1	2	2	3	1	5	2	7	3	9	8

Текущая цепочка

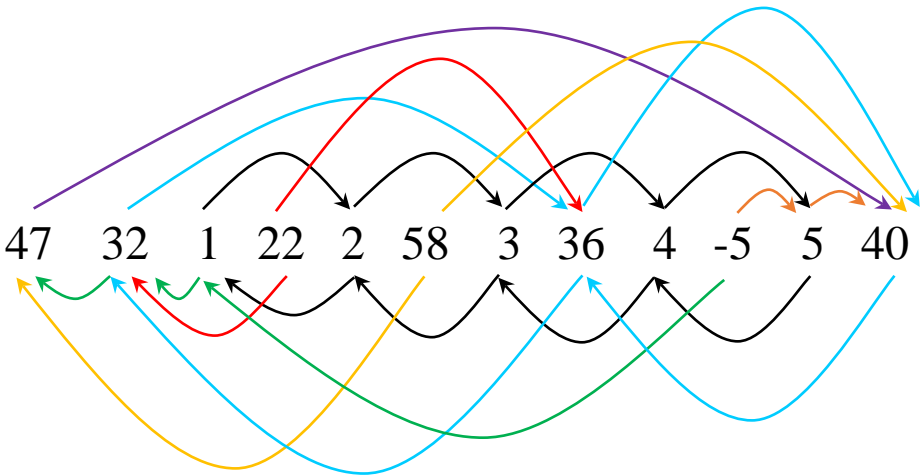
1 ⇒ 2 ⇒ 3 ⇒ 4 ⇒ 5

Элемент	47	32	1	22	2	58	3	36	4	−5	5	40
Обработан?	1	1	1	0	1	0	1	1	1	0	1	1

Множество всех цепочек

47 32 ⇒ 36 ⇒ 40 1 ⇒ 2 ⇒ 3 ⇒ 4 ⇒ 5

Вычисление множества всех цепочек



	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	47	32	1	22	2	58	3	36	4	−5	5	40
<i>IR</i>	12	8	5	8	7	12	9	12	11	11	12	0
<i>IL</i>	0	1	2	2	3	1	5	2	7	3	9	8

Текущая цепочка

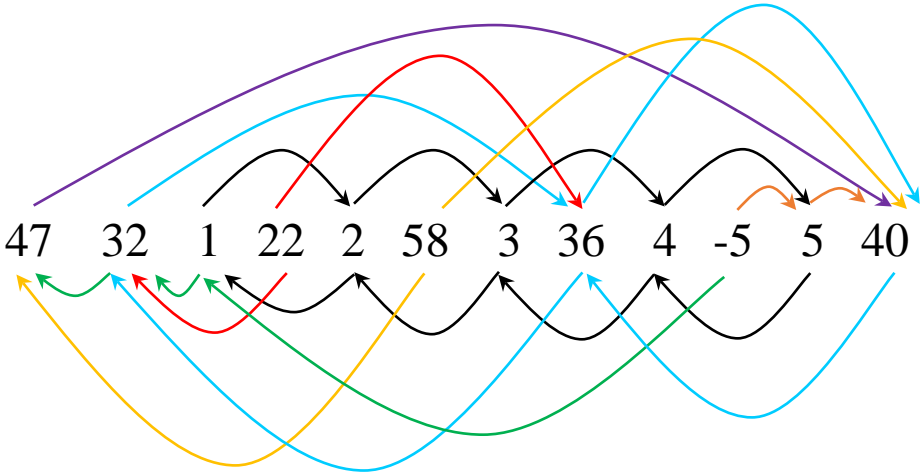
22

Элемент	47	32	1	22	2	58	3	36	4	−5	5	40
Обработан?	1	1	1	1	1	0	1	1	1	0	1	1

Множество всех цепочек

47 32 \Rightarrow 36 \Rightarrow 40 1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 5 22

Вычисление множества всех цепочек



	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	47	32	1	22	2	58	3	36	4	-5	5	40
<i>IR</i>	12	8	5	8	7	12	9	12	11	11	12	0
<i>IL</i>	0	1	2	2	3	1	5	2	7	3	9	8

Текущая цепочка

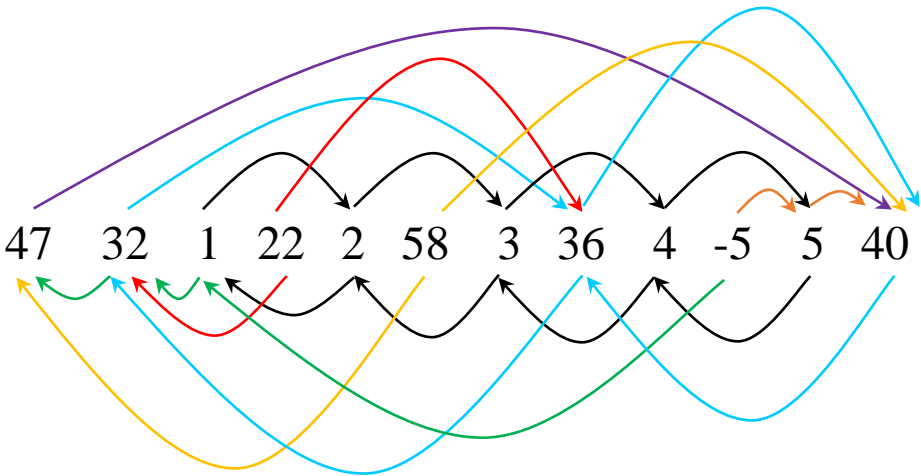
22

Элемент	47	32	1	22	2	58	3	36	4	-5	5	40
Обработан?	1	1	1	1	1	0	1	1	1	0	1	1

Множество всех цепочек

47 32 \Rightarrow 36 \Rightarrow 40 1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 5 22

Вычисление множества всех цепочек



	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	47	32	1	22	2	58	3	36	4	−5	5	40
<i>IR</i>	12	8	5	8	7	12	9	12	11	11	12	0
<i>IL</i>	0	1	2	2	3	1	5	2	7	3	9	8

Текущая цепочка

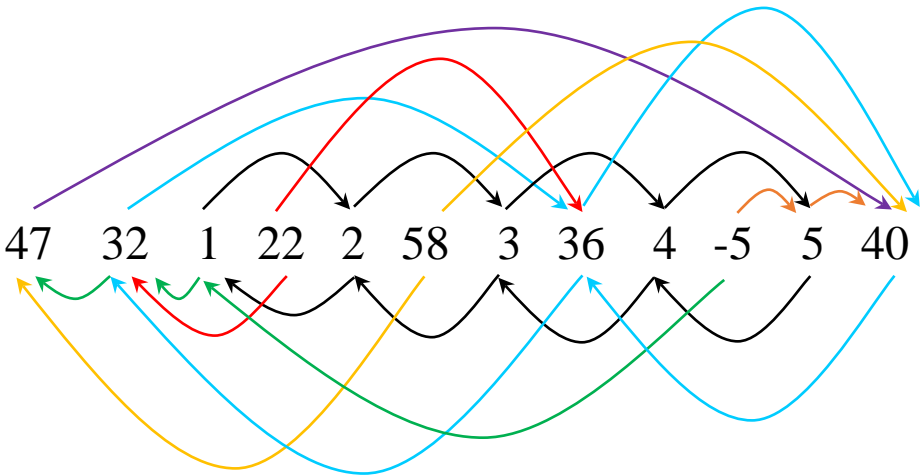
22

Элемент	47	32	1	22	2	58	3	36	4	−5	5	40
Обработан?	1	1	1	1	1	0	1	1	1	0	1	1

Множество всех цепочек

47 32 \Rightarrow 36 \Rightarrow 40 1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 5 22

Вычисление множества всех цепочек



	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	47	32	1	22	2	58	3	36	4	−5	5	40
<i>IR</i>	12	8	5	8	7	12	9	12	11	11	12	0
<i>IL</i>	0	1	2	2	3	1	5	2	7	3	9	8

Текущая цепочка

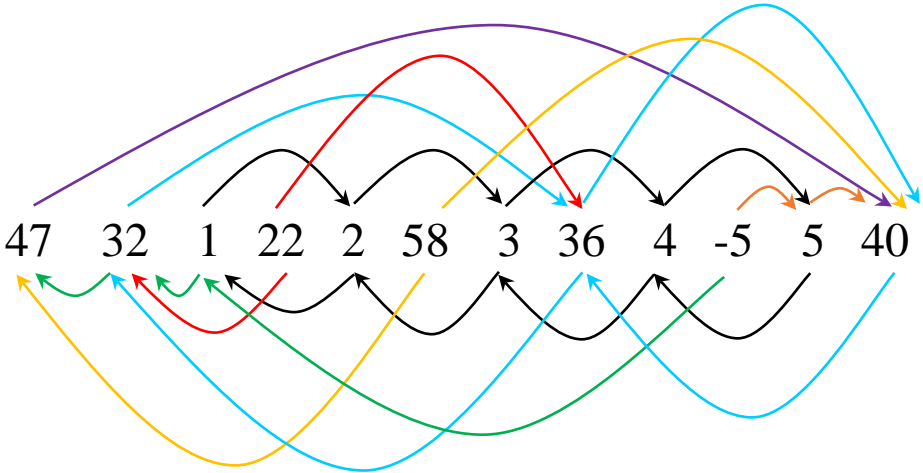
58

Элемент	47	32	1	22	2	58	3	36	4	−5	5	40
Обработан?	1	1	1	1	1	1	1	1	1	0	1	1

Множество всех цепочек

47 32 \Rightarrow 36 \Rightarrow 40 1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 5 22 58

Вычисление множества всех цепочек



	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	47	32	1	22	2	58	3	36	4	-5	5	40
<i>IR</i>	12	8	5	8	7	12	9	12	11	11	12	0
<i>IL</i>	0	1	2	2	3	1	5	2	7	3	9	8

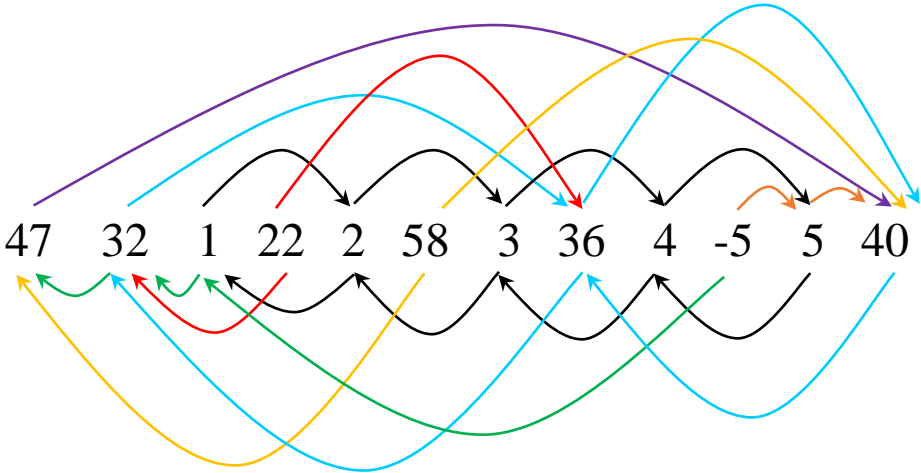
Текущая цепочка

Элемент	47	32	1	22	2	58	3	36	4	-5	5	40
Обработан?	1	1	1	1	1	1	1	1	1	1	1	1

Множество всех цепочек

47 32 \Rightarrow 36 \Rightarrow 40 1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 5 22 58 -5

Вычисление множества всех цепочек



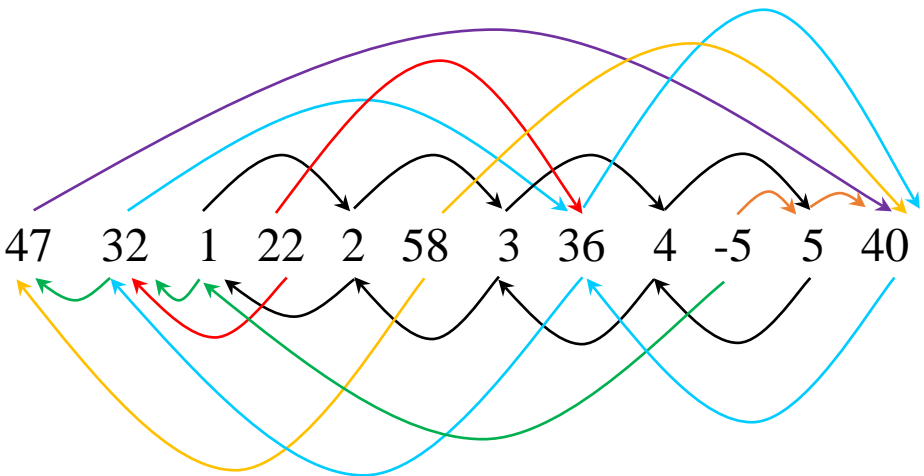
	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	47	32	1	22	2	58	3	36	4	-5	5	40
<i>IR</i>	12	8	5	8	7	12	9	12	11	11	12	0
<i>IL</i>	0	1	2	2	3	1	5	2	7	3	9	8

Элемент	47	32	1	22	2	58	3	36	4	-5	5	40
Обработан?	1	1	1	1	1	1	1	1	1	1	1	1

Множество всех цепочек 47 32 \Rightarrow 36 \Rightarrow 40 1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 5 22 58 -5

Левый и Правый индексы МП позволяют вычислить МВЦ за $O(n)$

Вычисление незакрепленной цепочки



	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	47	32	1	22	2	58	3	36	4	-5	5	40
<i>IR</i>	12	8	5	8	7	12	9	12	11	11	12	0
<i>IL</i>	0	1	2	2	3	1	5	2	7	3	9	8

Элемент	47	32	1	22	2	58	3	36	4	-5	5	40
Обработан?	1	1	1	1	1	1	1	1	1	1	1	1

Множество всех цепочек

47 32 \Rightarrow 36 \Rightarrow 40 1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 5 22 58 -5



Самая длинная цепочка

1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 5

Алгоритм поиска закрепленной цепочки (Anchored Time Series Chain)

Algorithm ATSC(IN: T , IR , IL , j ; OUT: C)

$C := \{j\}$

while $IR(j) \neq 0$ **and** $IL(IR(j)) = j$ **do**

$j := IR(j)$

$C := C \cup j$

return C

Алгоритм поиска незакрепленной цепочки

Algorithm ALLC(IN: T, IR, IL ; OUT: S, UC)

$Len := \{1\}_1^{n-m+1}; S := \emptyset$

for $i := 1$ **to** $|IR|$ **do**

if $Len(i) = 1$ **then**

$j := i; C := \{j\}$

while $IR(j) \neq 0$ **and** $IL(IR(j)) = j$ **do**

$j := IR(j); Len(j) := -1; Len(i) := Len(i) + 1; C := C \cup j$

$S := S \cup C$

$UC := ATSC(T, IL, IR, \arg \max Len(i))$

return S, UC

Исключает повторный просмотр цепочек, входящих в уже выявленные цепочки

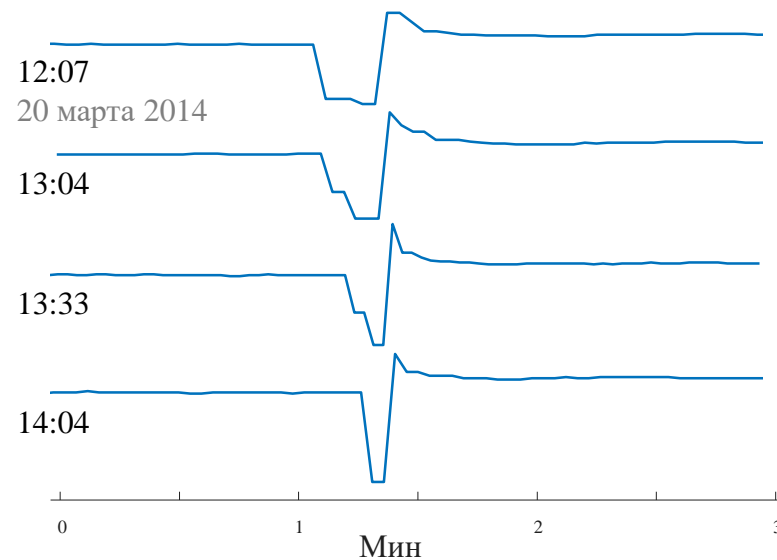
Содержание

- Понятие цепочки
- Алгоритмы поиска цепочек
- **Примеры применения**

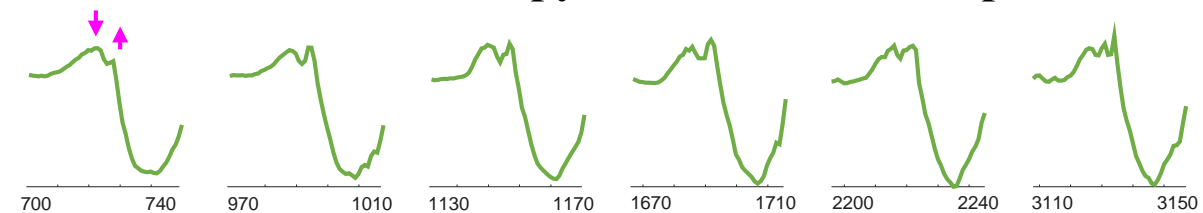
Поиск цепочек: энергопотребление, тренировки



Энергопотребление холодильника

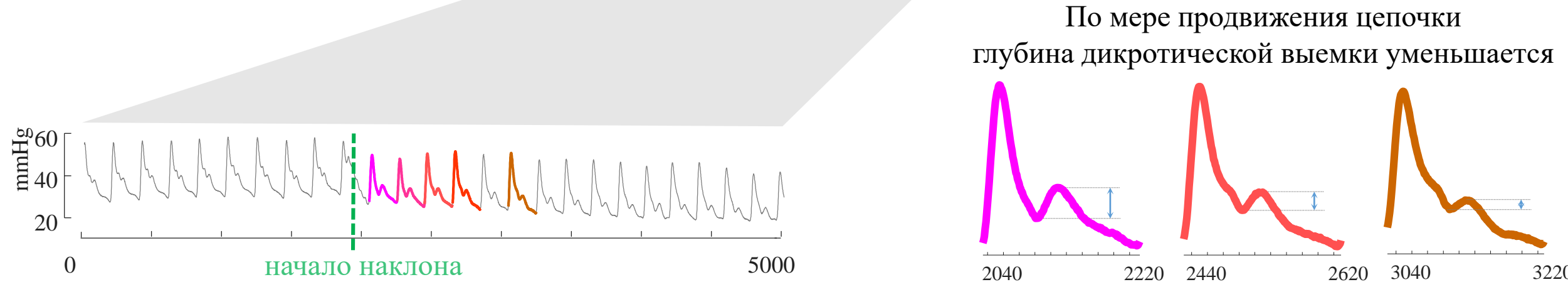
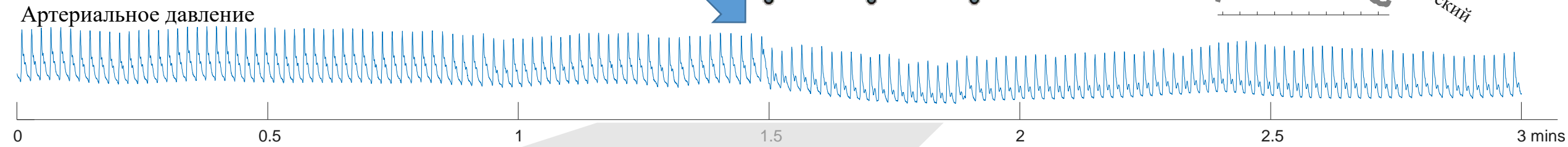
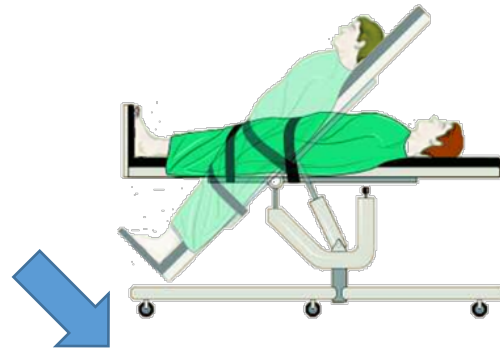


Запись датчика с левой икры спортсмена, когда он начал бег трусцой на беговой дорожке



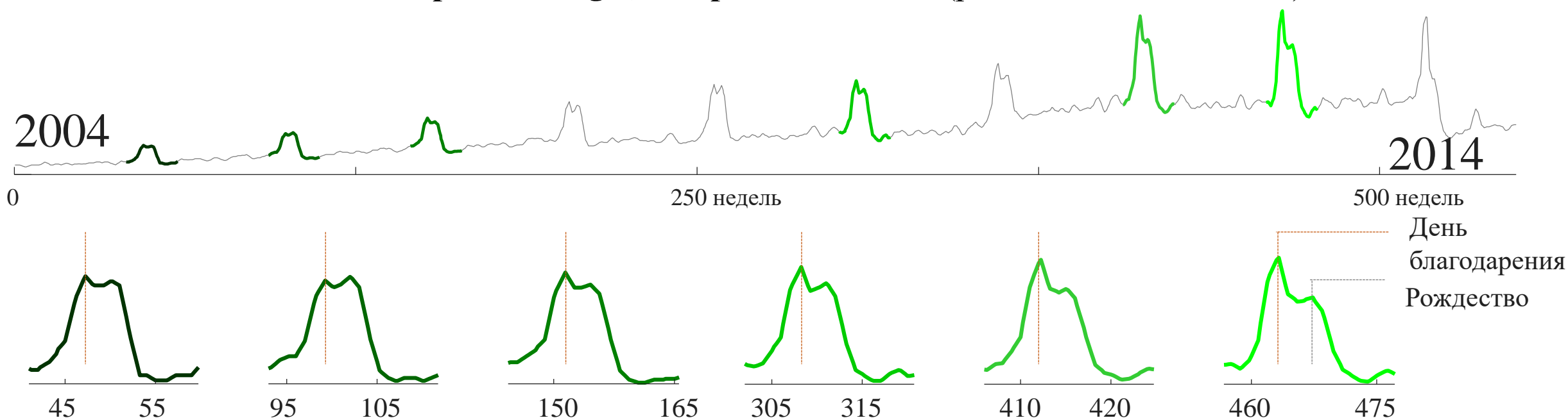
Применение цепочек: Эксперимент с наклоняемой мед. каталкой

Обнаружение цепочек
в ряде артериального давления,
 m – один удар сердца



Применение цепочек: Запросы Google

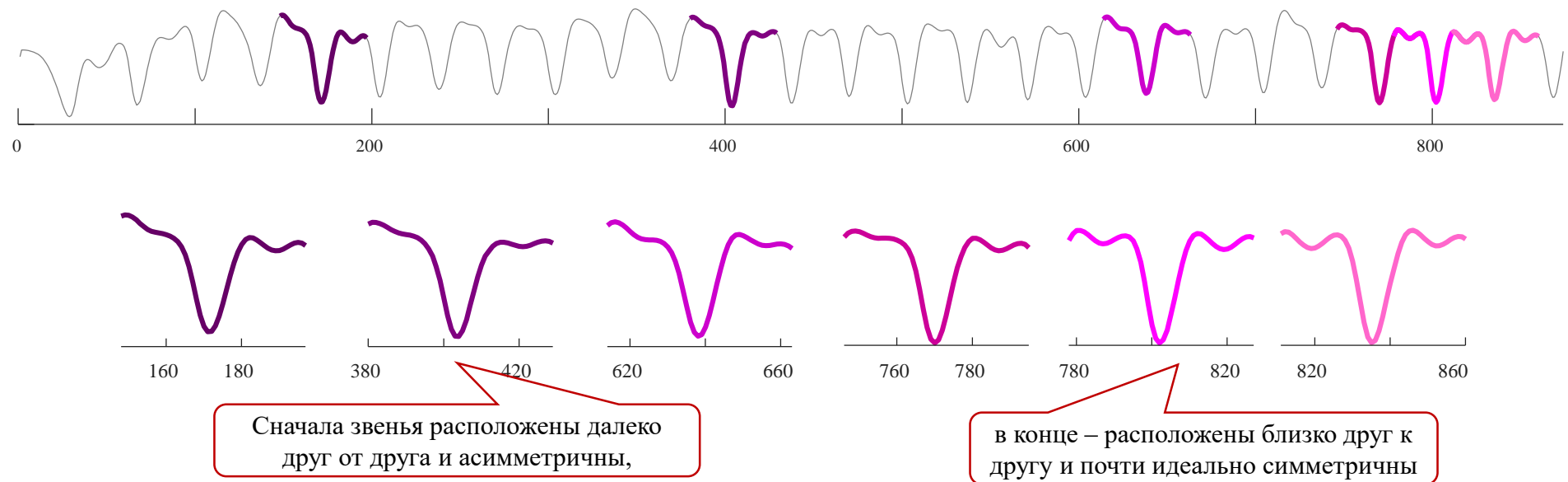
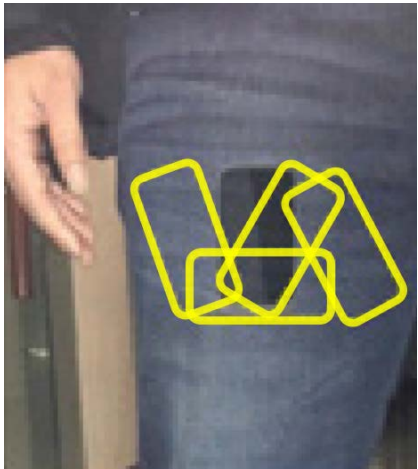
Число запросов Google, содержащих *Kohl's* (розничная сеть в США)



- **Рост важности Киберпонеделника:** за 10 лет выпуклость переходит от плавной и занимающей больший период между Днём благодарения и Рождеством к резкой и сосредоточенной на Дне благодарения
- Киберпонеделник (понеделник после Дня благодарения) – маркетинговый термин для побуждения людей делать покупки онлайн; пресс-релиз “Киберпонеделник становится одним из крупнейших дней онлайн-покупок в году” от 28 ноября 2005 г., дата совпадает с первым проблеском острого пика в цепочке

Применение цепочек: Биометрическая идентификация по походке

- Набор данных о походке записан для проверки гипотезы о биометрической идентификации. Рассматривается нестабильность мобильного телефона с точки зрения его ориентации и положения, когда он свободно помещается в карман
- Характер походки начинается как непредсказуемый (телефон болтается в кармане пользователя) и в конечном итоге успокаивается (телефон становится на место)



Литература

1. Zhu Y., Imamura M., Nikovski D., Keogh E.J. Matrix Profile VII: Time series chains: A new primitive for time series data mining. 2017 IEEE International Conference on Data Mining, ICDM 2017, New Orleans, LA, USA, November 18-21, 2017. pp. 695-704.
<https://doi.org/10.1109/ICDM.2017.79>
2. Zhu Y., Imamura M., Nikovski D., Keogh E.J. Introducing time series chains: A new primitive for time series data mining. Knowl. Inf. Syst. 60(2), 1135-1161 (2019). <https://doi.org/10.1007/s10115-018-1224-8>
3. Imamura M., Nakamura T., Keogh E.J. Matrix Profile XXI: A geometric approach to time series chains improves robustness. KDD'20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, CA, USA, August 23-27, 2020. 1114-1122.
<https://doi.org/10.1145/3394486.3403164>