



Data Structures and Algorithms

Алгоритмы. Сортировка.



Сортировка

Сортировка - упорядочение элементов последовательности в возрастающем порядке.

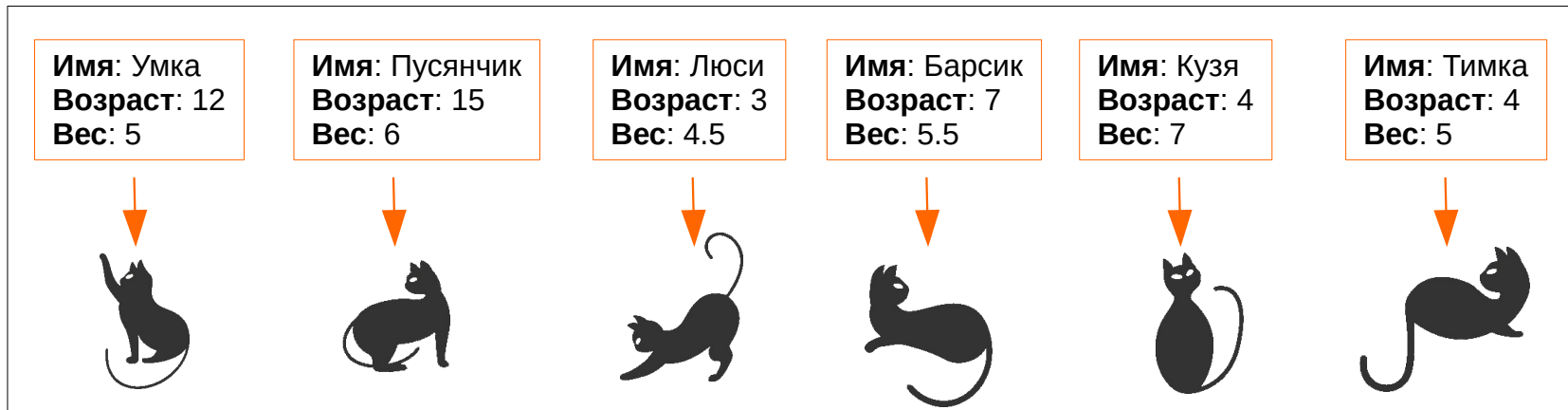
Дано: последовательность из N элементов $p_1, p_2, p_3 \dots$ в которой каждый элемент p представляет собой запись, которая содержит данные и некоторый ключ k (по нему и проводится сортировка).

Задача сортировки - такая перестановка элементов, что бы их ключи образовали возрастающую последовательность.

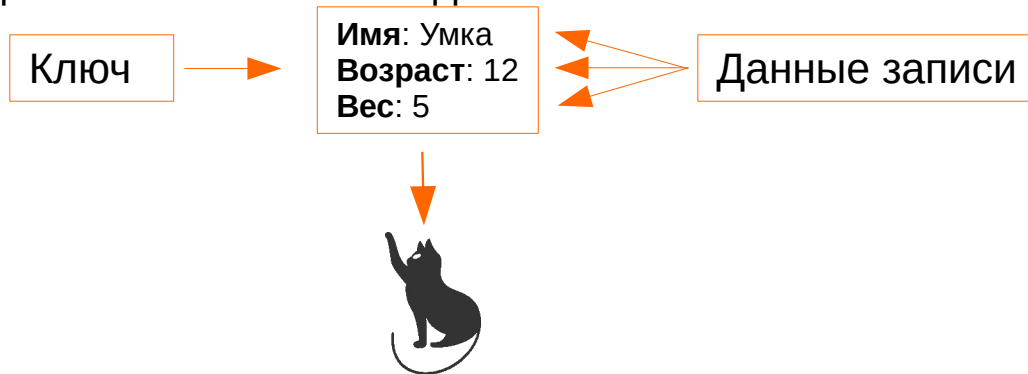
$$k_1 \leq k_2 \leq k_3 \leq \dots$$



Сортировка — объяснение на кошках

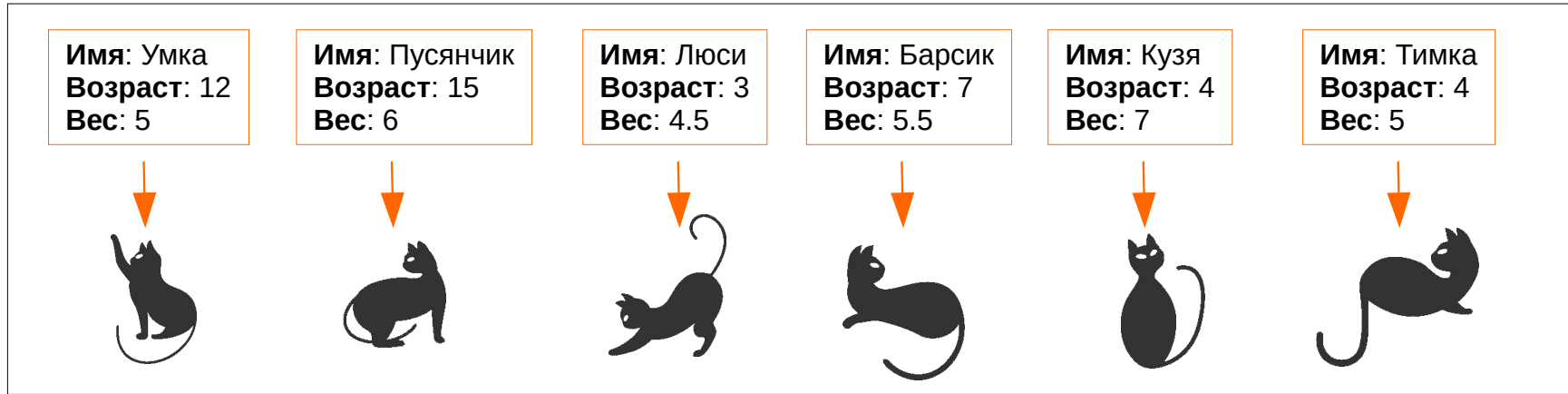


Есть набор элементов последовательности (как вы догадались это кошки). Выберем в качестве ключа возраст. Тогда элемент этой последовательности можно описать следующим образом:





Сортировка — объяснение на кошках



Последовательность элементов отсортирована по возрастанию





Требования к ключам элементов

Для множества ключей должно быть определено отношение порядка « $<$ ». Причем для любых 3-х ключей (например a , b , c) должны выполняться такие условия:

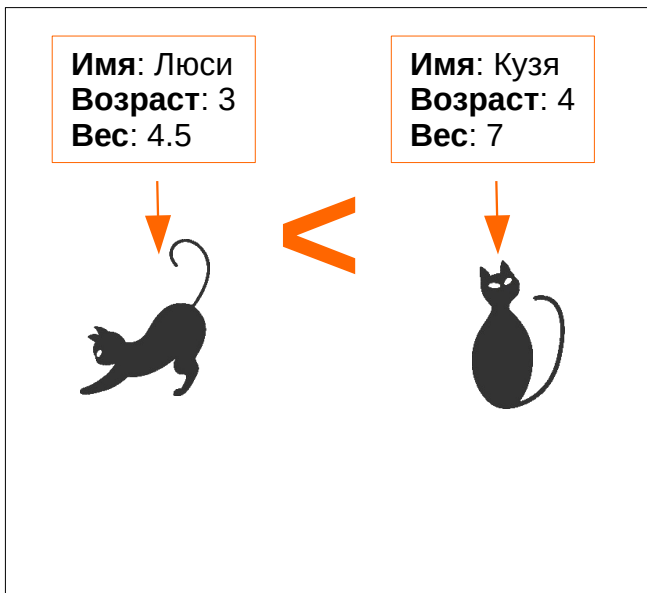
- **Закон трихотомии.** Справедливым является одно и только одно из соотношений: $a < b$, $a > b$, $a = b$.
- **Закон транзитивности.** Если $a < b$ и $b < c$, то $a < c$.

Данные условия определяют математическое понятие линейного или совершенного упорядочения, а удовлетворяющие им множества поддаются сортировке большинством методов.



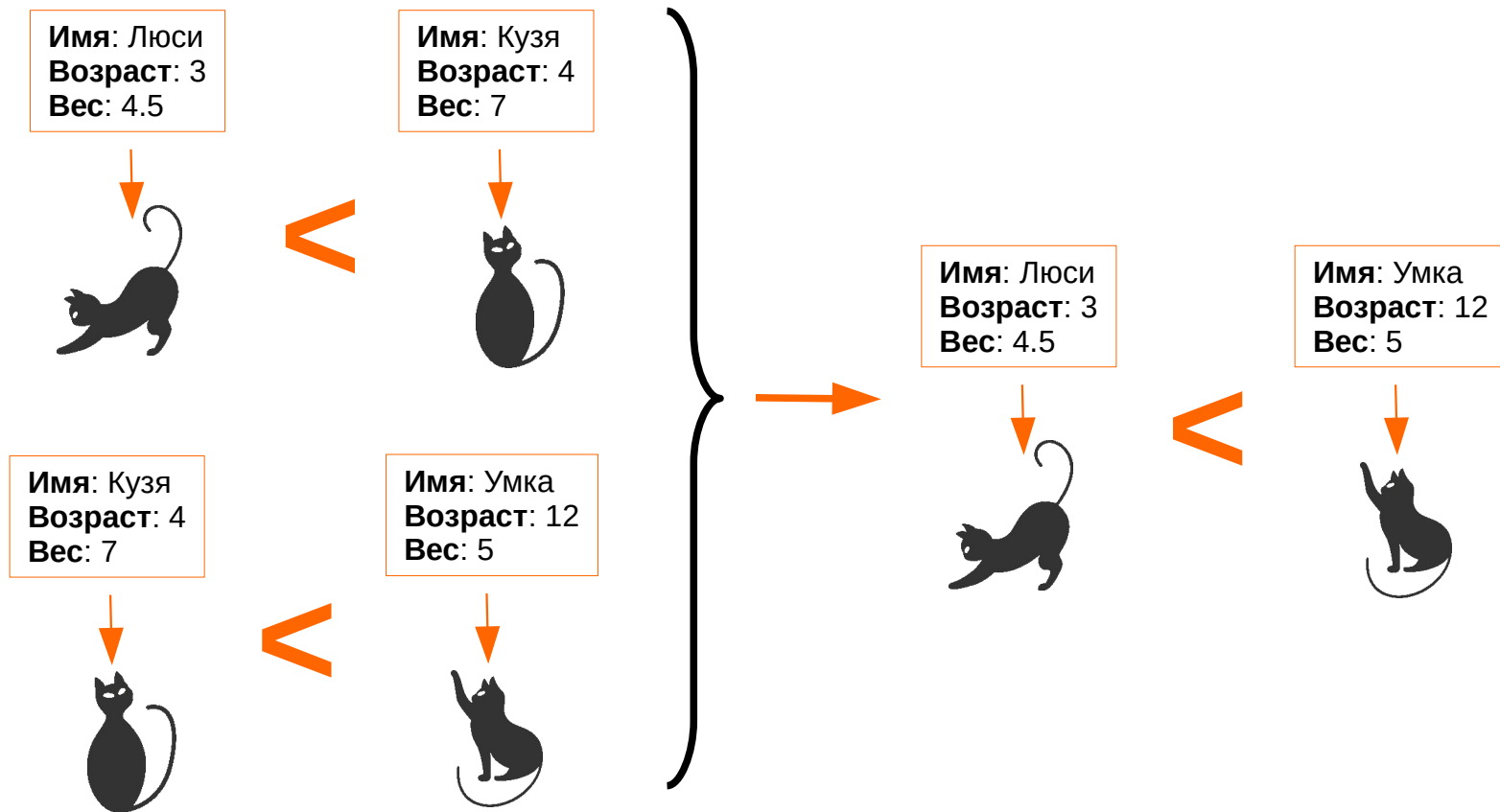
Закон трихотомии

В качестве ключей сортировки будем как и прежде использовать возраст кошки.





Закон транзитивности





Виды сортировки

Устойчивая (стабильная) сортировка — сортировка, которая не меняет относительный порядок сортируемых элементов, имеющих одинаковые ключи, по которым происходит сортировка.

Математическая формулировка:

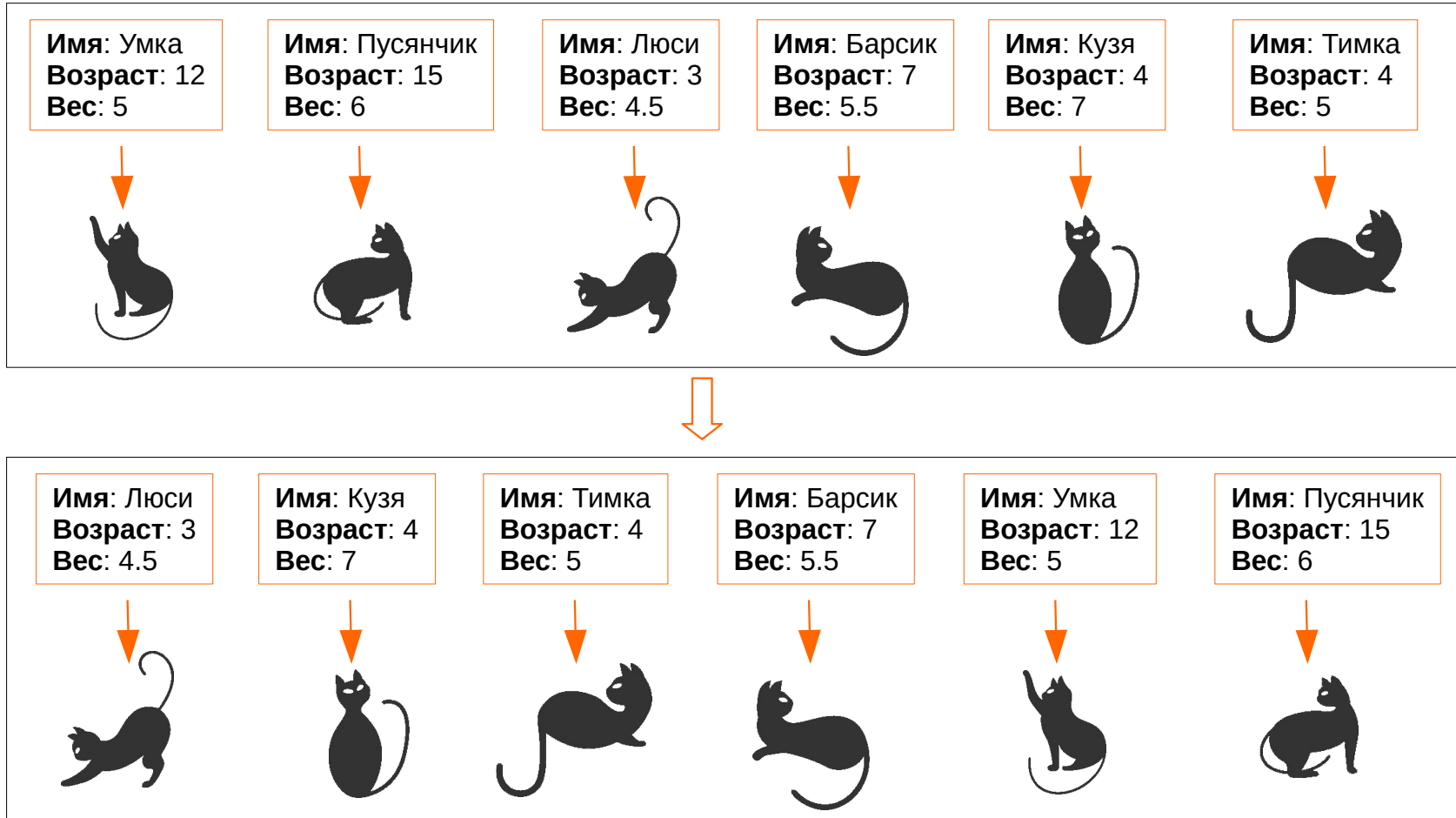
$$\forall (i < j \in \mathbb{N}, k_i = k_j) n_i < n_j$$

Для любых индексов в последовательности $i < j$, в случае если ключи одинаковы, то и $n_i < n_j$.

Неустойчивая — не сохраняет относительный порядок элементов имеющих одинаковые ключи.



Устойчивая сортировка





Устойчивая сортировка

На предыдущем слайде изображены две последовательности элементов. Верхняя не отсортированная последовательность. Нижняя отсортированная последовательность. Отметим два элемента не отсортированной последовательности (это коты с именами **Кузя** и **Тимка**) с одинаковыми значениями ключей (у них одинаковый возраст). В не отсортированной последовательности **Тимка** стоял правее **Кузи**. В отсортированной последовательности эти элементы сменили свое местоположение, но друг относительно друга они сохранили порядок следования. Т.е. **Тимка** стоит правее **Кузи**.

Алгоритмы сортировки которые сохраняют порядок при сортировке называют **устойчивыми**.



Алгоритмы устойчивой и неустойчивой сортировки

Алгоритмы устойчивой сортировки	Алгоритмы неустойчивой сортировки
Сортировка пузырьком	Сортировка выбором
Сортировка перемешиванием	Сортировка расчёской
Сортировка вставками	Сортировка Шелла
Гномья сортировка	Пирамидальная сортировка
Сортировка слиянием	Плавная сортировка
Сортировка с помощью двоичного дерева	Быстрая сортировка
Сортировка Timsort	

В таблице приведены примеры алгоритмов как устойчивой так и неустойчивой сортировки.



Внешняя и внутренняя сортировки

Внутренняя сортировка — данные последовательности целиком вмещаются в оперативную память.

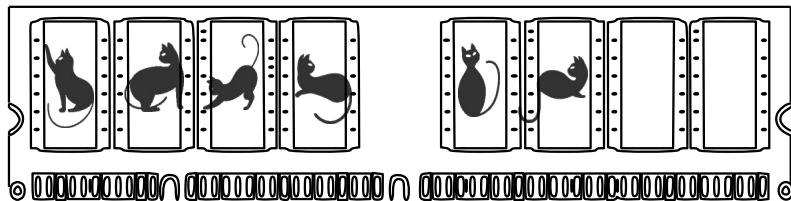
- Доступ возможен к произвольному элементу последовательности .
- Элементы последовательности упорядочиваются в памяти без дополнительных затрат.

Внешняя сортировка — данные не помещаются в оперативную память.

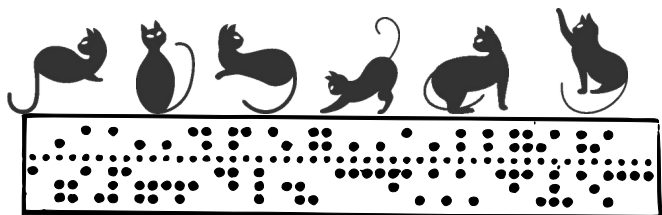
- Последовательный доступ к элементу последовательности. Т.е. можно прочесть только текущий элемент, за ним следующий и т. д.
- Затраты по свободную навигацию по элементам неоправданно высоки.



Внешняя и внутренняя сортировки



Внутренняя



Внешняя



Некоторые критерии алгоритмов сортировки

Естественность поведения — эффективность метода при обработке уже упорядоченных или частично упорядоченных данных. Алгоритм ведёт себя естественно, если учитывает эту характеристику входной последовательности и работает лучше.

Использование операции сравнения - алгоритмы, использующие для сортировки сравнение элементов между собой, называются основанными на сравнениях. Минимальная трудоемкость худшего случая для этих алгоритмов составляет: $O(n \cdot \ln(n))$



Список литературы

- 1) Ананий Левитин. Алгоритмы: введение в разработку и анализ. : Пер. с англ. — М. : Издательский дом "Вильямс", 2006. — 576 с. — ISBN 5-8459-0987-2. Стр. [45-47]
- 2) Роберт Седжвик, Кевин Уэйн. Алгоритмы на Java. 4-е издание. : М.: Вильямс, 2013. — 843 с. — ISBN: 978-5-8459-1781-2 Стр.[228-233]