

## **ПРИМЕНЕНИЕ АЛГОРИТМОВ СОРТИРОВКИ И ПОИСКА В ЭКОНОМИЧЕСКОМ АНАЛИЗЕ ДАННЫХ**

**Зайнуллина А.И.**, студент,

**Куценко С.М.**, к.п.н., доцент

Казанский государственный энергетический университет, г. Казань, Россия

**Аннотация.** В статье рассматриваются теоретические основы и особенности применения алгоритмов сортировки и поиска при обработке экономических данных. Показана роль алгоритмов в повышении эффективности вычислений и организации структурированной аналитической информации.

**Ключевые слова:** алгоритм, сортировка, поиск, сложность, оптимизация, данные, программирование, анализ.

Современная экономика основана на анализе большого объема информации, который

требует высокой скорости обработки и точности. Основой эффективной обработки данных является использование алгоритмов сортировки и поиска для обеспечения структуры и разумной организации информации. Эти алгоритмы составляют основу вычислительной логики и используются не только для программирования, но и для задач прикладного анализа. Целью данной статьи является исследование принципа, классификации и применения алгоритмов сортировки и поиска для решения задач, связанных с обработкой экономических данных.

Эффективность алгоритма оценивается на основе времени выполнения и использования памяти, которые определяются временной и пространственной сложностью [4]. Для выражения этих характеристик используется асимптотическая оценка, позволяющая сравнивать алгоритмы независимо от конкретной реализации.

Сортировка данных – это процесс размещения элементов в массиве или списке в соответствии с заданными критериями (например, по возрастанию или алфавитному порядку номеров). Алгоритмы сортировки можно разделить на простые алгоритмы и эффективные алгоритмы. Простые алгоритмы включают пузырьковую сортировку, сортировку вставкой и сортировку выбором. Эти алгоритмы просты в реализации, но плохо работают при обработке больших массивов данных [5]. Эффективные методы, такие как быстрая сортировка, сортировка слиянием и сортировка в куче, используют принцип рекурсивного разделения данных, что позволяет значительно сократить временные затраты на обработку большого объема входных данных.

Алгоритм поиска предназначен для поиска элементов в структуре данных на основе заданных критериев. Самым простым алгоритмом поиска является линейный поиск,

который сканирует все элементы по порядку, пока не будет найден нужный элемент. Бинарный поиск – более эффективный алгоритм [1]. Он использует принцип, согласно которому массив должен быть предварительно отсортирован и разделен на две половины, что значительно сокращает количество операций. В соответствии с конкретными характеристиками структуры данных могут использоваться более сложные методы поиска, такие как интерполяционный поиск, учитывающий распределение значений элементов.

На практике для реализации алгоритмов сортировки и поиска используются различные языки программирования, включая Python, C++, Java и др [2]. Независимо от языка, основные этапы реализации включают определение структуры данных, организацию цикла обработки и реализацию логических условий для обеспечения корректной работы алгоритма.

Рассмотрим реализацию алгоритма сортировки. Алгоритм пузырьковой сортировки попарно сравнивает последовательные элементы массива и меняет их местами при нарушении порядка. Этот процесс повторяется несколько раз для полной сортировки массива. Этот метод очень прост, но он неэффективен для больших наборов данных. Более эффективным методом является сортировка слиянием, при которой исходный массив разбивается на подмассивы, каждый подмассив сортируется, а результат соединяется с одним отсортированным массивом. Этот метод обеспечивает высокую скорость и стабильность работы с различными структурами входных данных.

Быстрая сортировка основана на выборе опорного элемента и разделении массива на две части – с элементами, меньшими и большими опорного значения. Далее для каждой

из частей рекурсивно выполняется тот же алгоритм. Этот метод отличается высокой эффективностью на больших объемах данных, хотя его производительность может снижаться при неудачном выборе опорного элемента. Пирамидальная сортировка использует структуру данных, называемую бинарной кучей, и позволяет достичь высокой стабильности и предсказуемости времени выполнения.

При реализации алгоритма поиска важно выбрать структуру данных и способ доступа к элементу. Линейный поиск хорош для небольших массивов, поскольку все элементы просматриваются по порядку. В отличие от этого, бинарный поиск требует предварительной сортировки данных, но временные затраты на поиск определенного значения значительно ниже. Более сложные структуры, такие как деревья поиска, хэш-таблицы и индексы баз данных, могут использоваться в поисковых приложениях для выполнения операций поиска с постоянным или логарифмическим временем.

Практическая работа с алгоритмами сортировки и поиска предполагает проведение сравнительного анализа их производительности на различных наборах данных. Для этого формируются выборки различного объема и структуры, после чего измеряется время выполнения операций [3]. В результате можно определить зависимость между сложностью алгоритма и эффективностью его работы. Такие эксперименты показывают, что простые алгоритмы эффективны лишь при небольших объемах данных, в то время как для больших массивов предпочтительно использовать алгоритмы с логарифмической или квазилинейной сложностью.

При анализе экономических показателей обычно необходимо систематизировать информацию на основе определенных стандартов, таких как показатели по уровню

дохода, торговому дню и динамике изменений. Алгоритмы сортировки позволяют систематизировать данные, выявлять закономерности и создавать аналитические отчеты.

Алгоритм поиска позволяет пользователю быстро получить доступ к определённой информации, такой как сведения о компании с самым высоким доходом или самым низким обменным курсом за определённый период времени. Использование эффективных алгоритмов сортировки и поиска значительно ускорило обработку таких запросов и повысило производительность информационных систем.

Базы данных, электронные таблицы, библиотеки программного обеспечения для анализа (pandas, NumPy и др.) [6]. Новейшая аналитическая платформа сочетает в себе классические и оптимизированные методы, однако необходимо понимать эти алгоритмы, чтобы адаптировать их к конкретным задачам анализа, программным решениям и повысить общую эффективность.

Важным аспектом использования алгоритмов является их оптимизация под конкретные условия. В зависимости от количества и структуры данных выберите алгоритм, который обеспечивает наилучший баланс между скоростью и потреблением ресурсов. Для небольших наборов данных желательно использовать простые алгоритмы, поскольку это сводит к минимуму вычислительные затраты. Однако при обработке больших объемов информации следует использовать более сложные алгоритмы, чтобы минимизировать время выполнения операции.

Современные вычислительные системы предоставляют возможности для параллельной обработки данных с использованием многопоточных алгоритмов сортировки и поиска. Распределяя вычислительные задачи между несколькими процессорами, вы можете значительно ускорить обработку больших объемов информации. Некоторые языки программирования и библиотеки предоставляют встроенные функции, оптимизированные для конкретных аппаратных архитектур, что позволяет разработчикам добиться оптимальной производительности без ручной настройки алгоритма.

Алгоритмы сортировки и поиска являются наиболее важными инструментами обработки данных в программировании и играют важную роль в организации и анализе экономической информации. Их использование обеспечивает структурирование, оптимизацию и совершенствование процесса анализа. Понимание принципов работы алгоритмов, их характеристик и областей применения способствует развитию алгоритмического мышления и необходимо для проектирования современного программного обеспечения и аналитических систем.

В условиях растущего объема данных и возрастающей сложности вычислительных задач грамотный выбор и оптимизация алгоритмов являются основой надежности и производительности систем экономического анализа.

## Литература

Автор: Зайнуллина А.И. Куценко С.М.  
17.11.2025 11:25 -

---

1. 6 алгоритмов поиска на Java: от простого к сложному // Proplib : сайт. 19 мар. 2019.

URL

: <https://proplib.io/p/6-search-algorithms-java>. (дата обращения: 08.11.2025).

2. Это база. Алгоритмы сортировки для начинающих // Habr / Selectel. 17 окт. 2024. URL

: <https://habr.com/ru/companies/selectel/articles/851206/>.

(дата обращения: 08.11.2025).

3. Исследование эффективности алгоритмов сортировки: сравнительный анализ производительности на различных структурах данных // Nagoroh.ru : проект.

URL:

<https://nagoroh.ru/p/project/issledovanie-effektivnosti-algoritmov-sortirovki-sravnitelnyi-analiz-pr-oizvoditelnosti-na-razlicnyh-strukturax-dannyh>.

(дата обращения: 08.11.2025).

4. Кайда Н. Что такое «O» большое: объясняем на простых примерах // Proplib : сайт.

27 апр. 2024.

URL

:

[https](https://proplib.io/p/6-search-algorithms-java)

[://](https://proplib.io/p/6-search-algorithms-java)

[proplib](https://proplib.io/p/6-search-algorithms-java)

[.](https://proplib.io/p/6-search-algorithms-java)

[io](https://proplib.io/p/6-search-algorithms-java)

[/](https://proplib.io/p/6-search-algorithms-java)

[p](https://proplib.io/p/6-search-algorithms-java)

[/](https://proplib.io/p/6-search-algorithms-java)

[chto](https://proplib.io/p/6-search-algorithms-java)

[-](https://proplib.io/p/6-search-algorithms-java)

[takoe](https://proplib.io/p/6-search-algorithms-java)

[-](https://proplib.io/p/6-search-algorithms-java)

[o](https://proplib.io/p/6-search-algorithms-java)

Автор: Зайнуллина А.И. Куценко С.М.  
17.11.2025 11:25 -

---

-  
bolshoe  
-  
obyasnyаем  
-  
na  
-  
prostyh  
-  
primerah  
-2024-04-27.

(дата обращения: 08.11.2025).

5. Методы сортировки: от пузырьковой до быстрой – сравнение алгоритмов // SkyPro : Wiki-раздел.

URL:  
<https://sky.pro/wiki/analytics/metody-sortirovki-ot-puzyrkovoј-do-bystroj-sravnenie-algoritmov/>.  
(  
дата обращения: 08.11.2025).

6. Типы и архитектуры параллельных вычислений: какие они бывают? // Habr / Fplus Tech

.  
URL  
: [https://habr.com/ru/companies/fplus\\_tech/articles/881488/](https://habr.com/ru/companies/fplus_tech/articles/881488/).

(дата обращения: 08.11.2025).