

*Глухов А.М. Сравнительный анализ трудоемкости алгоритмов поиска в программировании // Академия педагогических идей «Новация». Серия: Студенческий научный вестник. – 2018. – №5 (май). – АРТ 225-эл. – 0,3 п.л. - URL: <http://akademnova.ru/page/875550>*

**РУБРИКА: ТЕХНИЧЕСКИЕ НАУКИ**

**УДК-004**

**Глухов Александр Михайлович**

Студент 5 курса, вечернего факультета

*Научный руководитель:* Абрамова О. Ф.

доцент кафедры Информатики и технологии программирования

Волжский политехнический институт (филиал) ФГБОУ ВО

«Волгоградский государственный технический университет»

(ВПИ (филиал) ВолгГТУ)

г. Волжский, Российская Федерация

e-mail: [glukhov\\_alexander@outlook.com](mailto:glukhov_alexander@outlook.com)

**СРАВНИТЕЛЬНЫЙ АНАЛИЗ ТРУДОЕМКОСТИ  
АЛГОРИТМОВ ПОИСКА В ПРОГРАММИРОВАНИИ**

*Аннотация:* В статье рассмотрены особенности и различные виды алгоритмов поиска целевого значения, а так же рассмотрены основные виды оценки эффективности алгоритма поиска.

*Ключевые слова:* алгоритм поиска, линейный алгоритм поиска, двоичный (бинарный) алгоритм поиска, интерполирующий алгоритм поиска, временная эффективность, пространственная эффективность.

**Glukhov Alexander Mikhailovich**

5th year student, evening faculty

Supervisor: **O. F. Abramova**

associate Professor of Informatics and programming technology  
Volzhsky Polytechnic Institute (branch) of the Federal state budgetary  
educational institution of higher education " Volgograd state technical  
University»

(VPI (branch) VSTU)

Volzhsky, Russian Federation

## **COMPARATIVE ANALYSIS OF THE COMPLEXITY OF THE SEARCH ALGORITHMS IN PROGRAMMING**

*Abstract:* the article describes the features and different types of algorithms for finding the target value, as well as the main types of evaluating the effectiveness of the search algorithm.

*Keywords:* search algorithm, linear search algorithm, binary search algorithm, interpolating search algorithm, time efficiency, spatial efficiency.

### **1. Алгоритмы поиска**

Поиск необходимой информации – одна из фундаментальных задач в множестве различных сфер, и в первую очередь в программировании.

Алгоритм поиска – это последовательный набор команд и инструкций, предназначенный для поиска целевого значения (ключа поиска) среди заданного множества данных, например, списка или массива.

В случае разговора об алгоритмах поиска мы предполагаем, что любая информация содержится в записях, из которых состоит некий список, который в свою очередь представляет собой массив данных в программе. Записи (элементы списка) в массиве данных идут последовательно, без каких-либо «промежутков» и(или) пропусков между ними. Номера записей в списке начинаются с индекса 1 (стоит отметить, что в самой программе первым индексом элемента списка является 0, вторым – 1 и т.д.) до  $N$  – конечного и полного числа записей.

Порядок записей случаен для неотсортированного списка или массива данных, и упорядочен по возрастанию или убыванию в отсортированном. Для массива данных, в котором значения записей идут в случайном, неотсортированном порядке, возможен только метод последовательного перебора каждого значения до момента нахождения искомого значения. Для массивов данных с упорядоченной последовательностью элементов списка возможно более сложные и более эффективные алгоритмы поиска.

## 2. Разновидности алгоритмов поиска

3. Для поиска необходимой информации возможно использование как собственноручного написанного алгоритма поиска, так и использование уже готовых стандартных алгоритмов. Все известные алгоритмы поиска и сортировки прошли множество стадий улучшения и оптимизации, что значительно повышает их эффективность по сравнению с собственноручно написанными алгоритмами.

4. Существует огромное множество разнообразных алгоритмов поиска для чисел, строк, символов, кратчайшего пути, адреса данных и т.д. Их сложность варьируется от самых простых алгоритмов поиска,

работающих по принципу последовательного сравнения данных списка с искомым значением, до чрезвычайно эффективных, но имеющих ограничения алгоритмов бинарного поиска, а также алгоритмов, основанных на представлении базового множества данных в иной, более подходящей для выполнения задачи поиска форме.

5. Для решения задачи поиска необходимых данных не существует единого универсального алгоритма, который одинаково эффективно подходил бы для всех случаев и задач. Некоторые из алгоритмов выполняются быстрее остальных, но для их работы требуется большой объем памяти. Другие выполняются очень быстро, но их можно применять только для предварительно отсортированных массивов и так далее. Как правило, каждый алгоритм поиска предназначен для решения только своей определенной задачи, и плохо справляются с поиском в других, не подходящих для него условиях.

6. Рассмотрим следующие виды алгоритмов поиска целевого значения:

1. Линейный алгоритм поиска
2. Двоичный (бинарный) алгоритм поиска
3. Интерполирующий алгоритм поиска

### 2.1. Линейный алгоритм поиска

Линейный алгоритм поиска является самым простым из алгоритмов поиска, который работает как с сортированными, так и не с сортированными массивами данных.

Главным плюсом данного алгоритма является возможность его простой реализации и сама возможность его применения к неупорядоченным массивам данных.

Принцип работы данного алгоритма заключается в последовательном сравнении каждого элемента массива данных с искомым значением, начиная с первого элемента и до момента совпадения значений (или до конца массива, в случае если искомое значение отсутствует).

У алгоритма последовательного поиска два наихудших случая. В первом случае искомый элемент находится в самом конце списка или вообще стоит последним. Во втором же случае искомого элемента вообще нет в списке. В обоих случаях придется проверить каждый элемент массива данных на совпадение, прежде чем найти искомый элемент или выяснить что его вообще нет в массиве данных. В результате будет проделано  $N$  сравнений, где  $N$  – число элементов в списке.

## 2.2. Двоичный (бинарный) алгоритм поиска

Если задачу поиска искомого значения необходимо осуществить в отсортированном/упорядоченном по возрастанию или убыванию массиве данных, то в таком случае возможно применение алгоритма двоичного (бинарного) поиска.

Данный алгоритм поиска работает по принципу деления массива данных на две равные части, причем поиск осуществляется только в одной из этих частей, которая затем также делится на две равные части, и так до тех пор, пока не будет обнаружено искомое значение, или же не будет определено его отсутствие. Использование принципа деления зоны поиска вдвое возможно только по причине того, что массив данных изначально упорядочен. После поиска среднего элемента и сравнения его с искомым значением, алгоритм определяет в какой из двух получившихся зон находится искомый элемент, после чего начинает дальнейший поиск.

### 2.3. Интерполирующий алгоритм поиска

Скорость работы данного алгоритма поиска превосходит скорость работы двоичного-бинарного (особенно это заметно в случае работы с большими объемами массива данных). Также одним из весомых преимуществ данного алгоритма является возможность искать не только числовые значения, но и текстовую информацию.

Вместо сравнения каждого элемента с искомым значением, как при линейном поиске, данный алгоритм производит «предсказание» местонахождения элемента: производится оценка области поиска по «расстоянию» между ключом и текущим значением элемента. Сначала вычисляется длина области поиска, затем длина от начала области до числа  $X$  (например, середины массива данных). Данные вычисления производятся как с адресом элемента, так и с его значением. В конце длина области умножается на длину между значениями, и результат прибавляется к значению из начала области.

Таким образом, алгоритм просто вычисляет по формуле область массива данных, где может находиться искомое значение с помощью принципа интерполяции, пытаюсь примерно определить закон возрастания элементов в массиве данных. Если вычисленное значение не равно искомому, тогда происходит сдвиг границы поиска и вычисления повторяются. Если результат вычислений больше, то сдвигается правая граница области поиска, если меньше – левая. Подобно бинарному поиску, отсекая область за областью, алгоритм постепенно находит искомое значение (если конечно оно вообще есть).

## 3. Сравнительный анализ алгоритмов поиска

Существует два основных вида оценки эффективности алгоритма:

1. Временная – отвечает за измерение скорости работы алгоритма поиска;
2. Пространственная – отвечает за измерение необходимой для работы алгоритма поиска объема памяти.

В обоих случаях оценка сложности зависит от множества параметров: от размера массива входных данных (ведь массив данных из 100 элементов будет обработан быстрее, чем аналогичный массив из 1000 элементов), от мощности процессора, типа данных, языка программирования и т.д.

Многие алгоритмы также предлагают выбор между объёмом памяти и скоростью. Как правило, многие алгоритмы поиска предлагают выбор между занимаемым объемом памяти и скоростью работы. Например, требуемую задачу поиска можно решить быстро, используя большой объём памяти, или медленно, но потребуется совсем немного памяти. Особенно актуально это для алгоритмов поиска кратчайшего пути.

Для анализа проведем теоретический поиск числа в одномерном массиве данных на 256 элементов. Данные в массиве предзаполнены и сортированы по возрастанию. Мы не будем проводить анализ поиска случайного числа в массиве данных, который заполнен случайными числами, ввиду неоднородности выходных данных. Будем учитывать только оценку алгоритма по времени скорости работы.

Предположим, некоторому теоретическому алгоритму поиска нужно выполнить  $5n^3 + 8n$  условных операций, чтобы обработать  $n$  элементов массива входных данных. При возрастании переменной  $n$ , на итоговую оценку времени работы будет в большей степени влиять возведение  $n$  в куб, чем умножение его на 5 или же прибавление  $8n$ . В

таком случае говорят, что временная оценка эффективности данного алгоритма поиска равна  $O(n^3)$ , т. е. зависит от размера входных данных кубически.

Использование заглавной буквы  $O$  (или так называемая  $O$ -нотация) пришло к нам из математики, где её применяют для сравнения асимптотического поведения функций.

### 3.1. Линейный алгоритм поиска

Данный алгоритм поиска работает по принципу последовательного сравнения каждого элемента массива данных с искомым значением до момента совпадения. Соответственно, в нашем примере  $O = n$ , где  $n$  – адрес ячейки, содержащей искомое значение. Эффективность данного алгоритма меняется в зависимости от «удаленности» искомой ячейки данных от «начала» массива данных. В лучшем случае  $O = 1$ , т.е. искомое значение будет находиться в первой ячейке. В худшем случае  $O =$  длине массива данных, если искомое значение находится в самом конце или вообще отсутствует. В нашем примере эффективность времени работы данного алгоритма поиска составит  $O = 200$ .

### 3.2. Двоичный (бинарный) алгоритм поиска

Данный алгоритм поиска работает по принципу последовательного деления массива данных на равные области поиска. Эффективность времени работы данного алгоритма поиска вычисляется по формуле  $O = \log_2(n)$ , и в нашем примере соответствует  $O = \log_2 256 = 8$ . Преимуществом данного алгоритма поиска является точно предсказываемое время работы.

### 3.3. Интерполирующий алгоритм поиска

Данный алгоритм поиска подобен двоичному, но работает на основе математической формулы – принципа интерполяции. Для данного

алгоритма поиска не существует формулы, способной точно определить его время работы, так как данный алгоритм работает на математической формуле, значения которой изменяются в зависимости от наполнения массива данных. Согласно Таблице 3.1 мы можем видеть, что время поиска в алгоритме постоянно меняется в зависимости от расположения ячейки искомого значения в массиве данных.

Таблица 3.1 – Время работы (циклов поиска) алгоритмов поиска

	Линейный	Двоичный	Интерполирующий
n = 5	5	8	1
n = 42	42	8	4
n = 50	50	8	3
n = 84	84	8	3
n = 100	100	8	2
n = 126	126	8	2
n = 168	168	8	2
n = 200	200	8	3
n = 210	210	8	1
n = 250	250	8	2
n = 255	255	8	1

n – адрес искомой ячейки в массиве данных.

## 4. Заключение

В результате проделанной работы мы наблюдаем, что применение линейного алгоритма допустимо только в случае невозможности сортировки массива данных, в противном случае следует использовать интерполирующий алгоритм поиска, который обладает гораздо большей скоростью работы.

### Список использованной литературы:

1. Абрамова О.Ф., Александрина А. Ю. Анализ методов организации и проведения внеучебных конкурсных мероприятий в дистанционном формате / О.Ф. Абрамова, А.Ю. Александрина // Открытое и дистанционное образование. - 2017. - № 2 (66). - С. 14-25
2. Круподеров Д.Д., Абрамова О.Ф. Анализ проблематики проведения web-олимпиад // Материалы VIII Международной студенческой электронной научной конференции «Студенческий научный форум» URL: [www.scienceforum.ru/2016/1378/18758](http://www.scienceforum.ru/2016/1378/18758) (дата обращения: 17.12.2016).
3. Абрамова О.Ф. Программная реализация веб-системы для проведения дистанционных конкурсов с мультимедийным контентом / О.Ф. Абрамова // Актуальные вопросы профессионального образования (АВПО). - 2017. - № 4 (9) Декабрь. - С. 25-39.
4. Александрина А.Ю., Александрина А. Ю., Каблов В. Ф. Разработка специализированных программных продуктов как форма научно-исследовательской работы студентов направления «Химическая технология» / А.Ю. Александрина, В.Ф. Каблов, О.Ф. Абрамова // Вестник Российского ун-та дружбы народов. Серия «Информатизация образования». - 2015. - № 4. - С. 59-66.
5. Буньковский Д.В. Формирование системы менеджмента качества в предпринимательском проекте/ Д.В. Буньковский // Вестник НГИЭИ. - 2016. - № 7 (62). - С. 22-31.
6. Гончар А.А., Абрамова О. Ф. Исследование предметной области и анализ осуществимости разработки коммуникационной программной системы для вуза [Электронный ресурс] / А.А. Гончар, О.Ф. Абрамова // Форум молодых учёных : электрон. науч. журнал. - 2017. - № 6 (10). – 16 с. – Режим доступа : [http://forum-nauka.ru/domains\\_data/files/10/Gonchar%20A.%20A.pdf](http://forum-nauka.ru/domains_data/files/10/Gonchar%20A.%20A.pdf).
7. Борзин Р.Ю., Абрамова О. Ф., Лясин Д. Н. Исследование предметной области и анализ осуществимости разработки программной системы для автоматизации проведения видеотрансляций [Электронный ресурс] / Р.Ю. Борзин, О.Ф. Абрамова, Д. Н. Лясин // Форум молодых учёных : электрон. науч. журнал. - 2017. - № 6 (10). – 26 с. – Режим доступа : [http://forum-nauka.ru/domains\\_data/files/10/BORZIN%20R.Yu..pdf](http://forum-nauka.ru/domains_data/files/10/BORZIN%20R.Yu..pdf).

**Всероссийское СМИ**

**«Академия педагогических идей «НОВАЦИЯ»**

Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.

(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

**Сайт:** akademnova.ru

**e-mail:** akademnova@mail.ru

8. Кащенко Я.В., Абрамова О. Ф., Рыбанов А. А. Исследование предметной области и анализ осуществимости разработки программной системы для учёта учебных и научных достижений студента вуза [Электронный ресурс] / Я.В. Кащенко, О.Ф. Абрамова, А.А. Рыбанов // Форум молодых учёных : электрон. науч. журнал. - 2017. - № 6 (10). - 17 с. - Режим доступа [http://forum-nauka.ru/domains\\_data/files/10/Kashenko%20Ya.V..pdf](http://forum-nauka.ru/domains_data/files/10/Kashenko%20Ya.V..pdf).

9. Мельниченко Д.В., Абрамова О. Ф. Исследование логических проблем юзабилити сайтов и анализ существующих решений [Электронный ресурс] / Д.В. Мельниченко, О.Ф. Абрамова // Современная техника и технологии. - 2015. - № 1. - С. Режим доступа : <http://technology.snauka.ru/2015/01/5360>.

10. Сулейманов А.Ю., Абрамова О. Ф. Анализ проблем автоматизации бизнес-процессов многопрофильных образовательных учреждений [Электронный ресурс] / А.Ю. Сулейманов, О.Ф. Абрамова // Современная техника и технологии. - 2015. - № 6. - Режим доступа : <http://technology.snauka.ru/2015/06/6792>.

***Дата поступления в редакцию: 17.05.2018 г.***

***Опубликовано: 21.05.2018 г.***

***© Академия педагогических идей «Новация». Серия «Студенческий научный вестник», электронный журнал, 2018***

***© Глухов А.М., 2018***