

СБОРНИК ЗАДАЧ ПО ТЕОРИИ АЛГОРИТМОВ.

Структуры данных.
Часть. Бинарный поиск элемента в массиве.

УДК 510.51(075.8)

ББК 22.12я73-1

С23

А в т о р ы :

**С. А. Соболев, К. Ю. Вильчевский, В. М. Котов,
Е. П. Соболевская**

Р е ц е н з е н т ы :

кафедра информатики и методики преподавания информатики
физико-математического факультета Белорусского государственного
педагогического университета им. М. Танка

(заведующий кафедрой, кандидат педагогических наук,
доцент *С. В. Вабищевич*);

профессор кафедры информационных технологий в культуре
Белорусского государственного университета культуры и искусства,
кандидат физико-математических наук, доцент *П. В. Гляков*

Часть 1

БИНАРНЫЙ ПОИСК

Операция поиска некоторого элемента x в произвольном массиве выполняется за время $\Theta(n)$ (в худшем случае нужно просмотреть последовательно все элементы). Однако если массив уже отсортирован, поиск элемента можно выполнить за время $O(\log n)$. Этот метод известен как бинарный поиск (аналогичные названия: поиск делением пополам, дихотомия) и часто используется на практике.

1.1. ОПИСАНИЕ МЕТОДА

Предположим, что есть некоторая упорядоченная последовательность элементов A , где $a_0 \leq a_1 \leq \dots \leq a_{n-1}$. Необходимо проверить, есть ли среди них заданный элемент x . Первоначально определяем границы $[l, r)$ области поиска как $l = 0$, $r = n$. Определяем индекс центрального элемента области поиска. Предположим, что это число $k = \lfloor \frac{l+r}{2} \rfloor$. Сравниваем a_k — элемент последовательности — и число x . Если элементы совпадают, то поиск завершён. Если $x < a_k$, то продолжаем аналогичные действия, изменяя правую границу области поиска на k . Если $x > a_k$, то продолжаем аналогичные действия, изменяя левую границу области поиска на $k + 1$. Алгоритм прекращает работу, как только будет найден требуемый элемент либо станет верным равенство $l = r$ (эта ситуация говорит о том, что элемента в последовательности нет).

```
def BinarySearch(a, x):  
    l = 0, r = len(a)  
    while l < r:  
        k = (l + r) // 2  
        if x == a[k]:  
            return True  
        else if x < a[k]:
```

```
        r = k
    else: # x > a[k]
        l = k + 1
    return False
```

В псевдокоде предполагается, что оператор `//` выполняет целочисленное деление с округлением вниз. Вместо выражения $\lfloor \frac{l+r}{2} \rfloor$ можно использовать эквивалентное $l + \lfloor \frac{r-l}{2} \rfloor$, чтобы избежать целочисленного переполнения в момент вычисления суммы.

Можно показать, что каждая новая итерация этого алгоритма приводит к уменьшению оставшейся части последовательности поиска не менее чем в два раза, поэтому алгоритм гарантированно завершается за $O(\log n)$ шагов. Частой проблемой в реализации является следующая: вместо строки кода

```
l = k + 1
```

ошибочно пишут строку

```
l = k
```

При этом наблюдается такой эффект: как только полуинтервал $[l, r)$ сходится к единичной длине, т. е. становится верно $r = l + 1$, мы всякий раз получаем $k = \lfloor \frac{l+r}{2} \rfloor = l$ и выполняем тождественное присваивание $l = l$ — алгоритм за циклируется.

1.2. БИНАРНЫЙ ПОИСК САМОГО ЛЕВОГО ЭЛЕМЕНТА

Нетрудно модифицировать описанный выше алгоритм бинарного поиска, чтобы он выдавал индекс элемента x в случае, если такой элемент есть. Однако при наличии в массиве равных элементов непонятно, индекс какого из них будет выдан. Иногда требуется получить самое левое или самое правое вхождение. Обобщением этой задачи является задача поиска индекса первого элемента, большего, чем x , либо равного ему. Реализуем операцию `LOWERBOUND`, которая решает эту задачу. В случае отсутствия в массиве подходящих элементов договоримся, что возвращаемое значение будет равно n .

```
def LowerBound(a, x):
    l = 0, r = len(a)
    while l < r:
        k = (l + r) // 2
        if x <= a[k]:
```

```

        r = k
    else: # x > a[k]
        l = k + 1
    return l

```

Кроме этого, может потребоваться найти индекс первого элемента, строго большего, чем x . Такая операция называется `UPPERBOUND`, её реализация отличается от предыдущей только одним знаком неравенства.

```

def UpperBound(a, x):
    l = 0, r = len(a)
    while l < r:
        k = (l + r) // 2
        if x < a[k]:
            r = k
        else: # x >= a[k]
            l = k + 1
    return l

```

Заметим, что для любого x полуинтервал от `LOWERBOUND`(A, x) (включительно) до `UPPERBOUND`(A, x) (не включительно) определяет участок массива, содержащий элементы x . Разность между этими двумя величинами равна количеству элементов массива A , которые равны x .

Пример 1.1. Рассмотрим следующий массив длины 9:

0	1	2	3	4	5	6	7	8
2	4	5	5	5	27	67	91	97

Предположим, что искомый элемент равен 5. В этом случае функция `LOWERBOUND` вернёт значение индекса 2, функция `UPPERBOUND` вернёт 5.

В стандартной библиотеке языка C++ есть функция `std::binary_search`, которая выполняет бинарный поиск и возвращает логическое значение (есть элемент или нет). Кроме того, определены функции `std::lower_bound` и `std::upper_bound`, которые действуют аналогично рассмотренным и возвращают итераторы. В языке Java для классов `Arrays` и `Collections` определён статический метод `binarySearch`, который совмещает в себе описанные выше функции `BinarySearch` и `LowerBound`, однако является менее гибким (при наличии в массиве нескольких элементов, равных искомому, метод может вернуть индекс любого). В языке Python бинарный поиск реализован в стандартном модуле `bisect`.

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

1. Алгоритмы: построение и анализ / Т. Кормен [и др.]. — М. : Вильямс, 2005. — 1296 с.
2. Котов В. М., Мельников О. И. Информатика. Методы алгоритмизации : учеб. пособие для 10–11 кл. общеобразоват. шк. с углубл. изучением информатики. — Минск : Нар. асвета, 2000. — 221 с.
3. Котов В. М., Соболевская Е. П., Толстиков А. А. Алгоритмы и структуры данных : учеб. пособие. — Минск : БГУ, 2011. — 267 с. — (Классическое университетское издание).
4. Сборник задач по теории алгоритмов : учеб.-метод. пособие / В. М. Котов [и др.]. — Минск : БГУ, 2017. — 183 с.
5. Теория алгоритмов : учеб. пособие / П. А. Иржавский [и др.]. — Минск : БГУ, 2013. — 159 с.
6. Соболев С. А., Котов В. М., Соболевская Е. П. Опыт использования образовательной платформы Insight Runner на факультете прикладной математики и информатики Белорусского государственного университета // Роль университетского образования и науки в современном обществе : материалы междунар. науч. конф., Минск, 26–27 февр. 2019 г. / Белорус. гос. ун-т ; редкол.: А. Д. Король (пред.) [и др.]. — Минск : БГУ, 2019. — С. 263–267.
7. Соболев С. А., Котов В. М., Соболевская Е. П. Методика преподавания дисциплин по теории алгоритмов с использованием образовательной платформы iRunner // Судьбы классического университета: национальный контекст и мировые тренды [Электронный ресурс] : материалы XIII Респ. междисциплинар. науч.-теорет. семинара «Инновационные стратегии в современной социальной философии» и междисциплинар. летней школы молодых ученых «Экология культуры», Минск, 9 апр. 2019 г. / Белорус. гос. ун-т ; сост.: В. В. Анохина, В. С. Сайганова ; редкол.: А. И. Зеленков (отв. ред.) [и др.] — С. 346–355.

СОДЕРЖАНИЕ

Часть 1. БИНАРНЫЙ ПОИСК

1.1. Описание метода	4
1.2. Бинарный поиск самого левого элемента	5

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ	7
--------------------------------	---

Учебное издание

Соболь Сергей Александрович
Вильчевский Константин Юрьевич
Котов Владимир Михайлович и др.

СБОРНИК ЗАДАЧ ПО ТЕОРИИ АЛГОРИТМОВ. СТРУКТУРЫ ДАННЫХ

Учебно-методическое пособие

Редактор *Х. Х. XXXXXXXX*
Художник обложки *С. А. Соболь*
Технический редактор *Х. Х. XXXXXXXX*
Компьютерная вёрстка *С. А. Соболя*
Корректор *Х. Х. XXXXXXXX*

Подписано в печать 29.02.2020. Формат 60×84/16. Бумага офсетная.
Печать офсетная. Усл. печ. л. 10,69. Уч.-изд. л. 9,6.
Тираж 150 экз. Заказ

Белорусский государственный университет.
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий № 1/270 от 03.04.2014.
Пр. Независимости, 4, 220030, Минск.

Республиканское унитарное предприятие
«Издательский центр Белорусского государственного университета».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий № 2/63 от 19.03.2014.
Ул. Красноармейская, 6, 220030, Минск.