УДК 004.896

С.С. СИНЕЛЬНИКОВ

Донецкий государственный институт искусственного интеллекта, Украина

ПРИМЕНЕНИЕ ЧИСЛЕННЫХ МЕТОДОВ К ЗАДАЧЕ ПОИСКА ДАННЫХ В ОТСОРТИРОВАННОМ МАССИВЕ

Рассмотрены методы поиска данных в отсортированном массиве, показана связь численных методов нахождения нулей функции с бинарным и интерполяционным методом поиска. Применены методы касательных и итерационный метод к задаче поиска данных.

поиск данных, численные методы, массив, бинарный поиск, интерполяционный метод, метод хорд, метод касательных, итерационный метод

Введение

Компьютеры являются основными средствами хранения и быстрой обработки информации. Они внедрены во многие отрасли, и являются важной частью решения многих задач. Извлечение информации по запросу пользователя – одна из немногих задач, для решения которых целесообразно развивать существующие методы поиска данных. Методы поиска данных связаны со многими разделами современной науки: функциональный анализ, исследование операций, базы данных и прочие.

Задача поиска [1 – 3] заключается в нахождении индекса i элемента массива f по заранее известному значению элемента $=key$, т.е. поиск индекса i, при котором $f[i]=key$.

Анализ методов поиска данных

В настоящее время известными методами поиска данных в отсортированном массиве из N элементов являются бинарный поиск [1, 2], интерполяционный метод [1], и метод с применением бинарных деревьев [1]. Скорость поиска для каждого из методов соответственно порядка $\log_2 N$, $1\log_2\log_2 N$, $\log_2 N$.

Достоинство алгоритма бинарного поиска в том, что он не требует построения дополнительных структур и прост в программировании. Недостаток метода бинарного поиска в том, что он уступает по скорости интерполяционному методу. Достоинство интерполяционного метода заключается в его скорости, а недостаток в том, что он требует близких по значению элементов и не приемлет резких скачков между рядом стоящими элементами. Алгоритм с применением бинарных деревьев (как и бинарного поиска) не требователен к таким резким скачкам, но требует дополнительной организации древовидной структуры расположения элементов, что не всегда допустимо.

В данной работе будет показана связь численных методов нахождения нулей функции с бинарным и интерполяционным методом поиска, а также применены метод касательных и итерационный метод к задаче поиска данных.

Применение метода касательных к задаче поиска данных

Рассмотрим следующий отсортированный массив (таблично заданную функцию) элементов (индексация начинается с нуля):

$f[20]$ = {8, 67, 99, 106, 123, 140, 148, 153, 162, 169, 190, 191, 198, 263, 267, 322, 326, 346, 348, 441}.

Предположим, что необходимо найти индекс элемента со значением $key=267$.

Вычтем из всех элементов массива key и построим график зависимости значения $f[i]-key$ от индекса i (рис. 1).

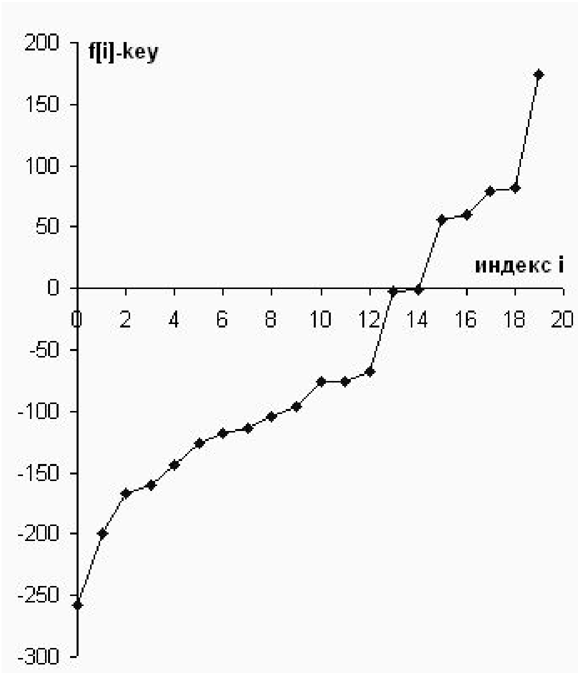

Рис. 1. График зависимости значения $f[i]-key$ от индекса i

Задача поиска элемента сводится к нахождению нуля построенной функции ($f[i]=0$). Полученная функция является возрастающей на отрезке [0, 19] и меняющей знак на концах отрезка, что, по аналогии с численными методами нахождения нулей функции, дает возможность применения методов дихотомии (деления пополам) и секущих (хорд) (если первая и вторая производная $f[i]$ знакопостоянны) [4]. Бинарный поиск и интерполяционный метод являются соответственно алгоритмом дихотомии и модифицированным методом хорд (рис. 2).

Возникает вопрос – можно ли применять другие численные методы для нахождения нулей функции $f[i]-key$ и при каких условиях? Применять другие численные методы можно к данной задаче. Например, метод касательных можно применять, если первая и вторая производная $f[i]$ знакопостоянны, т.е. для таблично заданной функции получим:

$$\begin{cases} f[i] \le f[i+1]; \\ f[i+1]-f[i] \le f[i+2]-f[i+1], \end{cases} \tag{1}$$

или

$$\begin{cases} f[i] \le f[i+1]; \\ f[i+1]-f[i] \ge f[i+2]-f[i+1]. \end{cases} \tag{2}$$

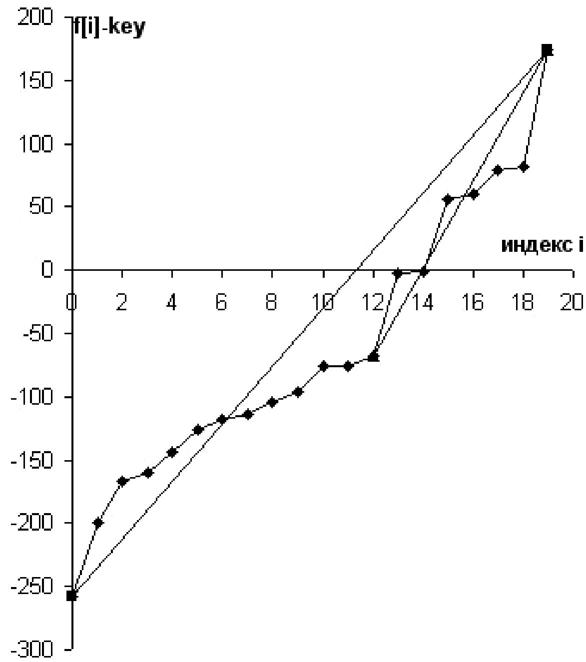

Рис. 2. Применение модифицированного метода хорд (интерполяционного метода)

Так как массив отсортирован по возрастанию, то условие (3)

$$f[i] \le f[i+1] \tag{3}$$

выполняется всегда. Остальные условия необходимо проверять, прежде чем использовать метод касательных.

Рассмотрим следующий массив из N=20 элементов, удовлетворяющий требованиям (1):

f [20]={4, 5, 11, 24, 39, 60, 90, 121, 162, 212, 267, 328, 395, 472, 548, 635, 773, 1121, 1671, 2531};

График таблично заданной функции f представлен на рис. 3.

Применим метод касательных для поиска элемента равного $key = 267$.

Проведем прямую через точки N–1 и N–2 (последнюю и предпоследнюю). Пересечение с осью OI дает значение 16. 16-й элемент не равен 267, поэтому его использовать в дальнейших расчетах нет смысла, и далее проводим прямую через точки 14 и 15. Пересечение с осью OI дает значение 10, которое равно 267.

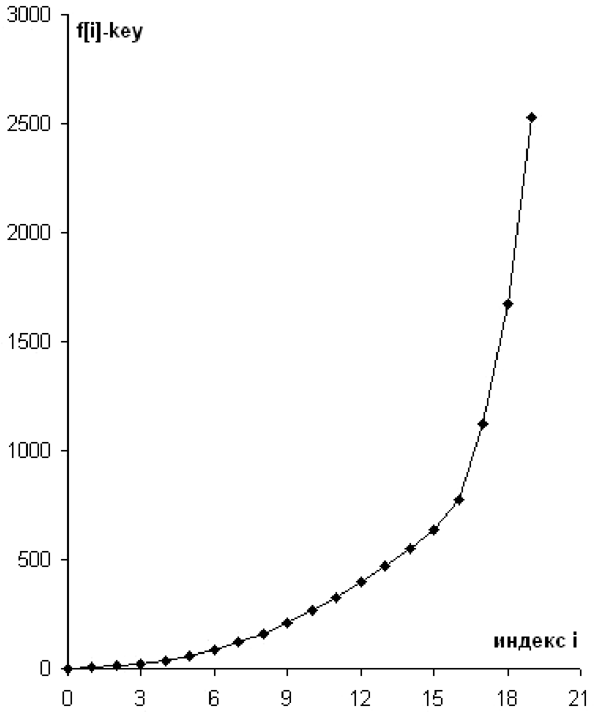

Рис. 3. График функции $f[i]$ для метода касательных

Искомый элемент найден и потребовалось две итерации.

В обоих случаях производилось округление ново полученного значения I в меньшую сторону (дробная часть отбрасывалась).

Очевидно, что данный алгоритм дольше всего будет искать первый элемент с индексом ноль (рис. 4) – 4 итерации, а быстрее всего *N*–1, *N*–2 и *N*–3.

Среднее число итераций для метода касательных составляет

$$\frac{(4\cdot 1+9\cdot 2+4\cdot 3+3\cdot 4)}{20}=\frac{46}{20}=2{,}3\,.$$

Среднее число итераций для метода хорд равно

$$\frac{(3\cdot 1+2\cdot 2+2\cdot 3+2\cdot 4+3\cdot 5+8\cdot 4)}{20} = \frac{68}{20} = 3{,}4\,.$$

В данном случае метод касательных лучше метода хорд на треть.

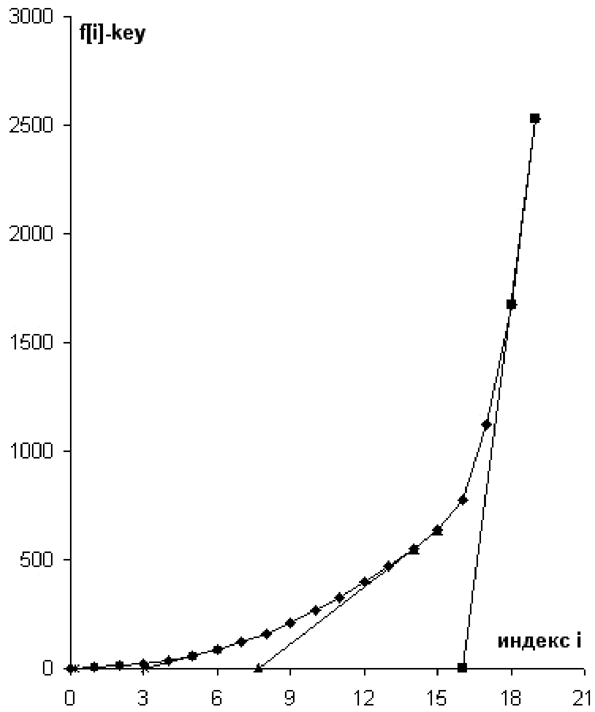

Рис. 4. Применение метода касательных

Формальное описание метода касательных

Формально метод касательных можно записать следующим образом.

Дан отсортированный по возрастанию массив f из *N* элементов. Необходимо найти индекс *k* в массиве f , где $f[k]=key$.

Если каждый элемент массива f удовлетворяет требованиям (1) или (2), то метод касательных применим.

Если каждый элемент массива f удовлетворяет требованиям (1), то алгоритм должен начать работу с *N*–1 элемента (последнего) и округление нового значения I_{j+2} должно производиться в меньшую сторону. Если каждый элемент массива f удовлетворяет требованиям (2), то алгоритм должен начать работу с 0 элемента (первого) и округление нового значения I_{j+2} должно производиться в большую сторону.

В случае (1) проведем прямую через точки ($I_j, f[I_j]$) и ($I_{j+1}, f[I_{j+1}]$), где $I_j = N-1$, а

$I_{j+1} = N-2$. Для случая (2) выбираем точки $(0, f[0])$ и $(1, f[1])$. Это имитация касательной.

$$\frac{I - I_{j+1}}{I_j - I_{j+1}} = \frac{f(I) - f[I_{j+1}]}{f[I_j] - f[I_{j+1}]}. \qquad (4)$$

Найдем пересечение прямой, заданной (4), с осью OI:

$$I = I_{j+1} - \frac{f[I_{j+1}]}{f[I_j] - f[I_{j+1}]},$$

и получим реккурентную формулу (5) для получения нового индекса

$$I_{j+2} = I_{j+1} - \frac{f[I_{j+1}]}{f[I_j] - f[I_{j+1}]}. \qquad (5)$$

Алгоритм завершает работу в том случае, когда $f[I_{j+2}] = key$, а искомый индекс $k = I_{j+2}$.

Для доказательства сходимости метода сравним тангенсы углов наклона прямых, проходящих через точки $(i, f[i])$ и $(i+1, f[i+1])$, $(i+1, f[i+1])$ и $(i+2, f[i+2])$:

$$\frac{f[i+1] - f[i]}{i+1-i} \le \frac{f[i+2] - f[i+1]}{i+2-i-1},$$

$$f[i+1] - f[i] \le f[i+2] - f[i+I]. \qquad (6)$$

Полученное условие (6) выполняется (требование (1)) и означает, что угол наклона первой прямой, проходящий через точки $(i, f[i])$ и $(i+1, f[i+1])$ меньше или равен, чем второй прямой, проходящий через точки $(i+1, f[i+1])$ и $(i+2, f[i+2])$. Пусть первая прямая пересекает ось I в точке (I1,0), а вторая прямая – в точке (I2,0). Тогда I1 ≤ I2. Применяя этот результат ко всем прямым, проведенным, начиная с точки с искомым индексом *k*, и следующей за ней *k*+1, получим:

$$I_k \le I_{k+I} \le \ \ldots \le I_{N-1}. \qquad (7)$$

Выражение (7) означает, что пересечение с осью OI любой прямой, проведенной через две любые точки $(i, f[i])$ и $(i+1, f[i+1])$, не могут быть меньше I_k – искомого индекса и, так как угол наклона прямых постоянно уменьшается, рано или поздно мы доберемся до I_k. Таким образом, сходимость метода достигнута. Аналогично можно доказать сходимость метода касательных для условий (2).

Применение метода итераций к задаче поиска

Рассмотрим возможность применения метода итераций [4 – 7] для поиска данных в отсортированном массиве из *N* элементов по ключу.

Необходимые требования для применения метода итераций: функция должна быть возрастающей, меняющей знак на концах отрезка и первая производная знакопостоянна. Каждое из условий для отсортированного массива выполняется.

Для поиска новой точки по методу итераций применяется следующая реккурентная формула:

$$I_{n+1} = \varphi(I_n),$$

где $\varphi(I) = I - f(I) \cdot C$, причем C выбирается таким образом, чтобы функция $\varphi(I)$ удовлетворяла условиям сходимости метода итераций.

Знак C определяется знаком производной. Так как массив отсортирован по возрастанию, то C>0. C выбирается из соображений (8):

$$C \le \frac{2}{|f'(x)|}, \qquad (8)$$

т.е. $C \le \dfrac{2}{\max(|f'(x)|)}$. Для таблично заданной функции получим:

$$C \le \frac{2}{\max\{|f(I+1) - f(I)|,\ I = \overline{0, N-2}\}}. \qquad (9)$$

Таким образом, реккурентная формула для получения нового значения примет вид (10):

$$I_{n+1} = I_n - f[I_n] \cdot C, \qquad (10)$$

где константа C определяется в соответствии с условием (9).

Начальный I_n берется равным *N*–1. Завершение алгоритма достигается при $f[I_{n+1}] = key$.

Заметим, что метод итераций сходится быстрее, если на каждом шаге пересчитывать константу C. Это выполнять уместно, так как диапазон поиска сужается. Недостаток данной модификации заключаются в накладных расходах (дополнительный массив для хранения максимальных значений констант C на каждом диапазоне поиска).

Применим метод итераций для поиска элемента равного $key = 267$ в массиве из N=20 элементов:

f[20]={4, 5, 11, 24, 39, 60, 90, 121, 162, 212, 267, 328, 395, 472, 548, 635, 773, 1121, 1671, 2531}.

Определим $C = \frac{2}{2531\text{-}1671} = \frac{2}{860}$. Выполнив 4 итерации алгоритма (для новых значений дробную часть отбрасываем), получим I_1=13, I_2=12, I_3=11, I_4=10.

В данном случае алгоритм итераций сработал медленнее алгоритма касательных, но быстрее алгоритма хорд. Достоинством алгоритма есть то, что он не требователен к условиям (1) и (2), в отличие от алгоритма касательных. Следует отметить, что алгоритм хорошо работает, если скачки между каждыми соседними элементами примерно одинаковы.

В алгоритме итераций также необходимо производить дополнительные расчеты по нахождению константы C, что также относится к его недостаткам.

Заключение

В данной работе была показана связь численных методов нахождения нулей функции с бинарным и интерполяционным методом поиска.

Были применены метод касательных и итерационный метод к задаче поиска данных, что является новизной в решении этой задачи. Были показаны условия, при которых следует применять метод касательных и итерационный метод для задачи поиска данных, выявлены достоинства и недостатки каждого из методов.

Применение метода касательных и итерационного методов позволит в ряде случаев увеличить скорость поиска данных в любом программном комплексе, использующем поиск данных (например, сервере баз данных, файловых системах), что говорит о высокой практической значимости полученных результатов.

Данная работа представляет научный интерес для специалистов, занимающихся разработкой программного обеспечения, связанного с поиском данных, для разработчиков серверов баз данных, прикладного программного обеспечения для роботов.

Литература

1. Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн Алгоритмы: построение и анализ. - 2-е издание. – Вильямс, 2005. – 1296 с.
2. Кнут Д. Искусство программирования. - 3-е издание. – Вильямс, 2005. – 720 с.
3. Седжвик Р. Фундаментальные алгоритмы на С++: Ч. 1-4: Анализ, структуры данных, сортировка, поиск: Пер. с англ. (третья ред.). – Diasoft, 2001. – 687 с.
4. Березин И.С., Жидков Н.П. Методы вычислений. – М.: Физматгиз, 1962. – Т. 1. – 632 с.
5. Вержбицкий В.М. Численные методы. – М: Высшая школа, 2001. – 382 с.
6. Волков Е.А. Численные методы. – М.: Наука, 1982. – 254 с.
7. Калиткин Н.Н. Численные методы. – М.: Наука, 1978. – 512 с.

Поступила в редакцию 20.11.2006

Рецензент: д-р физ.-мат. наук, проф. Б.В. Бондарев, Донецкий национальный университет.