УДК 004.67, 519.688

М.Ю. Ястребов

ПОИСК ФИБОНАЧЧИ – ОБОСНОВАНИЕ, СИСТЕМА СЧИСЛЕНИЯ И ХЕШИРОВАНИЕ

M.Yu. Yastrebov

THE FIBONACCI SEARCH — THE SUBSTANTIATION, THE NUMBER SYSTEM AND THE HASHING

Доказана полнота поиска Фибоначчи. Алгоритм поиска позволяет представлять искомые числа кортежами специального вида из трехсимвольного алфавита. Преобразование троичной записи числа к допустимому кортежу дает процедуру хеширования.

ПОИСК ФИБОНАЧЧИ. ПОЛНОТА ПОИСКА. ПРЕДСТАВЛЕНИЕ ЧИСЕЛ. ТРОИЧНАЯ СИСТЕМА. ХЕШИРОВАНИЕ.

The article is aimed to prove the completeness of the Fibonacci search. The algorithm of the search allows representing the searched-for numbers as finite sequences (n-tuples) of a special type combined of the alphabet consisting of three symbols. The conversion of the tripartite record of a number into the admissible tuple results in the hashing procedure.

FIBONACCI SEARCH. COMPLETENESS OF THE SEARCH. REPRESENTATION OF NUMBERS. TERNARY SYSTEM. HASHING.

Изложение алгоритма поиска в отсортированном по возрастанию массиве a_i с использованием последовательно убывающих чисел Фибоначчи в качестве шагов смещения (Ф-поиска), приведенное в основополагающих руководствах по программированию (прежде всего, у Д. Кнута в [1]), нуждается, на наш взгляд, в теоретическом обосновании. Как справедливо отмечает сам Д. Кнут, «Технология поиска Фибоначчи, на первый взгляд, представляется весьма загадочной и, если просто взять программу и постараться понять, как она работает, вам покажется, что это полное шаманство» [1, с. 450].

К сожалению, дальнейшее изложение у Д. Кнута не «превращает шаманство в обычный танец с бубном». Остаются открытыми следующие вопросы. Почему Ф-поиск, реализованный для индексов ($a_i = i$), позволяет получить все заявленные в качестве возможного ключа поиска значения i от 1 до $F_{k+1} - 1$? Как связаны алгоритм Ф-поиска и структура дерева Фибоначчи?

Поскольку для упорядоченного массива неравенство между элементами равносильно такому же неравенству между индексами $(a_i <,> a_j) \Leftrightarrow (i <,> j)$, можно для упрощения обозначений считать, что $a_i = i$, так что ключ поиска сравнивается непосредственно с индексами.

Напомним формулировки рекурсивного определения дерева Фибоначчи Δ_k порядка k с корнем F_k и алгоритма Ф-поиска.

Определение. Если $k = 0$ или $k = 1$, то Δ_0 – лист со значением 0 или 1 соответственно; если $k \geq 2$, то корнем является F_k, левым поддеревом – Δ_{k-1}, правым – Δ_{k-2} с увеличенными на F_k значениями.

Алгоритм

Шаг 1. Начальные присвоения: $i := F_k$; шаг смещения вправо $p := F_{k-1}$; шаг смещения влево $q := F_{k-2}$.

Шаг 2. Развилка: $q = 0 \Rightarrow$ поиск завершен неудачно; $K = i \Rightarrow$ элемент найден; $K < i \Rightarrow$ переход к шагу 3; $K > i \Rightarrow$ переход к шагу 4.

Шаг 3. $q = 0 \Rightarrow$ поиск завершен неудачно; сдвиг влево: $i := i - q$; уменьшение смещений: $(p, q) := (q, p - q)$; переход к шагу 2.

Шаг 4. $p = 1 \Rightarrow$ поиск завершен неудачно; сдвиг вправо: $i := i + q$; уменьшение смещений: $p := p - q$; $q := q - p$; переход к шагу 2.

Обоснование Ф-поиска

Утверждение 1. Наибольший индекс, который можно получить с помощью алгоритма Ф-поиска [1, с. 451], отправляясь от первого проверяемого индекса F_k при ключе поиска $K > F_k$, есть $F_{k+1} - 1$.

Доказательство. Если k четное, то есть $k = 2t$, постоянные сдвиги вправо для увеличения очередного i в процессе поиска (шаг 4) означают прибавление очередного по убыванию слагаемого F_{k-2j}. В результате получается сумма чисел $F_k + F_{k-2} + ... + F_2$, равная, как известно, $F_{k+1} - 1$ [2]. Если же k нечетное, то есть $k = 2t - 1$, постоянные сдвиги вправо для увеличения очередного i в процессе поиска (шаг 4) означают прибавление очередного по убыванию слагаемого F_{k-2j-1} до слагаемого F_3. В результате получается сумма чисел $F_{2t-1} + F_{2t-3} + ... + F_3 = F_{2t} - F_1 = F_{2t} - 1$.

Лемма 1. Пусть $k \ge 5$ *и* $\beta(k) = F_{k-4} + {} + F_{k-5} ... + F_1$. Тогда

$$F_k + F_{k-2} - \beta(k) = F_k + 1.$$

Доказательство. Индукция по k. При $k = 5$ (база индукции) имеем:

$$\beta(5) = F_1 = 1;$$

$$F_5 + F_3 - \beta(5) = 5 + 2 - 1 = 6 = F_5 + 1.$$

Индукционный переход:

$$F_{k+1} + F_{k-1} - \beta(k+1) =$$
$$= F_{k+1} + (F_k - F_{k-2}) - F_{k-3} - \beta(k) =$$

(применяем к $\beta(k)$ индукционное предположение)

$$= F_{k+1} + F_k - F_{k-2} - F_{k-3} - (F_{k-2} - 1) =$$
$$= F_{k+1} + 1 + ((F_k - F_{k-2}) - F_{k-3}) -$$
$$- F_{k-2} = F_{k+1} + 1. \blacksquare$$

Лемма 2. Наименьший индекс, который можно получить с помощью алгоритма Ф-поиска, отправляясь от первого проверяемого индекса F_k при ключе поиска $K > K_i$, есть $F_k + 1$.

Доказательство. Постоянные сдвиги влево (после перехода на первой итерации алгоритма Ф-поиска к $F_k + F_{k-2}$) с целью уменьшения очередного i (шаг 3) означают вычитание очередного по убыванию слагаемого F_{k-2j}, как это имеет место и в формуле для $\beta(k)$. $\blacksquare$

Очередная цель – убедиться, что таким путем можно получить и все промежуточные индексы между $F_k + 1$ и $F_{k+1} - 1$ правого поддерева. Поскольку индексы этого поддерева получаются увеличением на F_k левого поддерева уровня $k - 2$, исследуем предварительно Ф-поиск в левом поддереве исходного уровня k.

Теорема 1. Начинающийся с проверки F_k Ф-поиск с ключом $K \in \{1, 2, ..., F_{k+1} - 1\}$ позволяет получить все индексы $\{1, 2, ..., F_k - 1\} \cup \{F_k\} \cup \{F_k + 1, ..., F_{k+1} - 1\}$, причем ровно по одному разу.

Доказательство. Индукция по k. Базу индукции для $k = 3$ (а также для $k = 4,5$) проверяется непосредственно. Индукционный переход. Поиск, начинающийся с проверки элемента F_{k+1}, после перехода влево на первой итерации (шаг 3) приводит к проверке числа $F_{k+1} - F_{k-1} = F_k$ с исходными параметрами сдвига $p = F_{k-1}, q = F_{k-2}$, что соответствует Ф-поиску для предшествующего стартового значения F_k. Тогда дальнейший поиск, по индукционному предположению, позволяет получить все индексы $\{1, ..., F_{k+1} - 1\}$. Поиск, начинающийся с перехода вправо, приводит к индексу $F_{k+1} + F_{k-1}$. При этом уменьшаются параметры сдвига: $p = F_{k-2}$, $q = F_{k-3}$. Эти параметры являются исходными для Ф-поиска, начинающегося с проверки F_{k-1}, что приводит к

Допустимые кортежи при $k = 6$

i	1	2	3	4	5	6	7	8	9	10	11	12
c_4	−1	−1	−1	−1	−1	−1	−1	0	1	1	1	1
c_3	−1	−1	−1	−1	0	1	1	0	0	0	0	0
c_2	−1	−1	0	1	0	0	0	0	−1	−1	0	1
c_1	−1	0	0	0	0	−1	0	0	−1	0	0	0

отрезку индексов $\{1, ..., F_k - 1\}$. Но поскольку разность между стартовыми индексами $(F_{k+1} + F_{k-1}) - F_{k-1}$ есть F_{k+1}, Ф-поиск приводит к смещению этого отрезка: $\{1 + F_{k+1}, ..., F_k - 1 + F_{k+1}\} = \{F_{k+1}, ..., F_{k+2} - 1\}$. ■

Ф-поисковое представление

Ф-поиск индекса $i \in \{1, ..., F_{k+1} - 1\}$ со стартовым индексом F_k равносилен представлению $i = F_k + c_{k-2}F_{k-2} + c_{k-2}F_{k-2} + c_{k-2}F_{k-2} + ... + c_1F_1$, и задается кортежем $g_k(i) = (c_{k-1,i}^{(k)}, c_{k-2,i}^{(k)}, ..., c_{2,i}^{(k)}, c_{1,i}^{(k)})$, в котором компоненты $c_{j,i}^{(k)}$ могут принимать значения из множества $\{-1, 0, 1\}$. В то время как общее число таких кортежей равно 3^{k-2}, Ф-поиску соответствуют лишь $F_{k+1} - 1$ из них.

Назовем кортеж *допустимым*, если он соответствует Ф-поиску некоторого $i \in \{1, ..., F_{k+1} - 1\}$. В таблице приведены допустимые кортежи для $k = 6$.

По-видимому, до сих пор не было отмечено, что при фиксированном k (и, значит, фиксированном F_k) кортеж $g_k(i)$ можно рассматривать как представление числа i в своеобразной системе счисления с основанием F_k, которую назовем *Ф-поисковым представлением*. Оно отличается от *фибоначчиевой записи* в виде последовательности нулей и единиц [2]. Последняя, в отличие от $g_k(i)$, опирается на последовательное выделение из n максимально возможных слагаемых-чисел Фибоначчи.

Назовем при заданном k кортеж $(c_{k-2}, c_{k-3, ...,} c_1)$ со значениями компонент $c_j \in \{-1, 0, 1\}$ *правильным*, если он удовлетворяет следующим условиям:

1) если при некотором j выполняется $c_j \neq 0, c_{j-1} = 0, c_{j-2} = 0$, то $c_{j-3} = c_{j-4} = ... = c_1 = 0$, т. е. два или более нулей подряд могут только замыкать кортеж;

2) если $c_j = 1$, то $c_{j-1} = 0$;

3) $c_1 \neq 1$;

4) если $c_{k-2} = 0$, то $c_{k-3} = c_{k-4} = ... = c_1 = 0$, т. е. с нуля может начинаться только нулевой кортеж (что имеет место, когда ключ поиска равен F_k);

5) если $c_j = -1,\ c_{j-1} = 0$, то $c_{j-2} \neq 1$ (запрещена комбинация −1, 0, 1).

Лемма 3. При заданном k число $m(k)$ правильных кортежей равно $F_{k+2} - 1$.

Доказательство. Пусть $G(k)$ − множество правильных кортежей, $G_{-1}(k)$ − подмножество кортежей, начинающихся с −1, $G_{1,0}(k)$ − подмножество кортежей, начинающихся с 1,0. Тогда

$$G(k) = \{(0, 0, ...,)\} \cup G_{-1}(k) \cup G_{1,0}(k). \quad (*)$$

База индукции при $k = 4$ и $k = 5$ проверяется непосредственным перебором возможных кортежей.

Индукционный переход. Если к любому кортежу из $G(k - 1)$ приписать слева минус единицу, получится кортеж из $G_{-1}(k)$, так что $|G_{-1}(k)| \leq |G(k-1)|$. Обратно, если из кортежа, принадлежащего $G_{-1}(k)$, убрать начальную минус единицу, получится кортеж из $G(k - 1)$, так что $|G_{-1}(k)| \geq |G(k-1)|$. Следовательно, $|G_{-1}(k)| = |G(k-1)|$. Аналогично устанавливается, что $|G_{1,0}(k)| = |G(k-2)|$. Применяя индукционное предположение, получаем в силу (*): $(F_{k+2} - 1) + 1 + F_{k+1} - 1 = F_{k+3} - 1$. ■

Кортежи, отвечающие значению $k + 1$, получаются из кортежей, отвечающих значению k следующим образом.

Теорема 2. Для того чтобы кортеж

$(c_{k-1}, c_{k-2,\dots,} c_1)$ был допустимым, необходимо и достаточно, чтобы он был правильным

Доказательство. Необходимость условия следует из данных таблицы. Убедимся в его достаточности. Допустимые кортежи представляют числа $\{1, 2, \dots, F_{k+2} - 1\}$, так что их уже $F_{k+2} - 1$. В силу леммы они покрывают все возможные варианты правильных кортежей. ■

Построение дерева поиска

Алгоритм Ф-поиска позволяет задать дерево, узлами которого являются индексы $i \in \{1, \dots, F_{k+1} - 1\}$. Именно, при $k \ge 3$ корнем дерева объявляется $u_1 = F_k$, его левым потомком – $u_2 = F_k - F_{k-2} = F_{k-1}$, его правым потомком – $u_3 = F_k + F_{k-2}$. Это соответствует тому, что при $i = u_1$ в кортеже $g_k(u_1)$ имеем $c_{k-1,i}^{(k)} = c_{k-2,i}^{(k)} = \dots = c_{1,i}^{(k)} = 0$; при $i = u_2$ в кортеже $g_k(i)$ имеем $c_{k-1,i}^{(k)} = -1,\ c_{k-2,i}^{(k)} = \dots = c_{1,i}^{(k)} = 0$; при $i = u_3$ в кортеже $g_k(i)$ имеем $c_{k-1,i}^{(k)} = 1,\ c_{k-2,i}^{(k)} = \dots = c_{1,i}^{(k)} = 0$. Далее, при $j \ge 2$ задаем рекурсивно: левый потомок для u_j есть индекс, который получается при одношаговом сдвиге из u_{j-1} на $-q$ (шаг 3); правый потомок для u_j есть индекс, который получается при одношаговом сдвиге из u_{j-1} на $+q$ (шаг 4). В силу утверждения 4, каждый из индексов окажется включенным в дерево, причем ровно один раз.

Утверждение 4. Дерево, структура которого задается алгоритмом Ф-поиска, является деревом Фибоначчи.

Доказательство. Индукция по k. База индукции при $k = 3 \Leftrightarrow F_k = 2$ проверяется непосредственно. Индукционный переход. Рассмотрим дерево, которое строится по алгоритму Ф-поиска для $k + 1$. Согласно алгоритму, левым потомком узла F_{k+1} (шаг 3 с $q = F_{(k+1)-2} = F_{k-1}$) является $F_{k+1} - F_{k-1} = F_k$, потомки которого, в силу индукционного предположения, образуют дерево Фибоначчи порядка k. Вместе с F_{k+1} в качестве родителя они оказываются левым поддеревом дерева Фибоначчи порядка $k + 1$ в соответствии с определением последнего. Правым потомком является $F_{k+1} + F_{k-1}$; дальнейший поиск от него происходит с исходными параметрами $p = F_{k-2}, q = F_{k-3}$. Это соответствует, согласно индукционному предположению, дереву Фибоначчи с корнем F_{k-1}. Сдвиг значений узлов последнего на F_{k-1} совпадает с условием, входящим в определение дерева Фибоначчи, поскольку разница индексов $(k + 1) - (k - 1) = 2$. ■

Процедура хеширования, основанная на Ф-поисковом представлении

Записанное в троичной позиционной системе счисления число $n \in \{1, 2, \dots, 3^k - 1\} = A(k)$ задается кортежем g из k нулей, единиц и двоек. Если заменить все двойки на минус единицы, получим кортеж g', который может оказаться Ф-поисковым представлением некоторого другого числа $n' \in \{1, 2, \dots, F_{k+3} - 1\}$: $g' = g_{k+2}(n')$. В этом случае хеширование разреженного массива $a_n, n \in A(k)$ означает соответствие $a_n \to b_{n'}$, и мы задаем хеш-функцию в виде $h(n) = n' = F_{k+1} + \sum_{i=1}^{k-1} c_i F_i$.

Наряду с преобразованием кортежа троичной записи в потенциально Ф-поисковое представление по схеме $(0 \to 0, 1 \to 1, 2 \to -1)$, возможны еще пять других вариантов преобразования (т. е. всего 3!).

Если же кортеж g' не является допустимым, то для хеширования необходим механизм его преобразования в допустимый. Любой предложенный механизм должен пройти экспериментальную проверку на предмет равномерности в среднем распределения числа коллизий на один допустимый кортеж.

Априори представляется разумным следующий механизм преобразования. Запрещенными в кортеже являются:

1) тройки (0, 0, –1), (0, 0, 1), (0, 0, 0) – последняя, если не является концевой;
2) пары (1, –1), (1, 1);
3) конечная пара $(\forall, 1)$, где $\forall = 0, 1, -1$;
4) начальная пара (0, 1), (0, –1);
5) тройка (–1, 0, 1).

Поэтому сначала в конечной паре $(\forall, 1)$, если она присутствует, $\forall$ заменяется на 0 при четном количестве минус-единиц в кортеже и на –1 при нечетном. Затем начальная пара (0, 1), если она присутствует, заменяется на (−1, 1), а начальная пара (0, −1) − на (−1, −1). Далее случаи 3 и 4 уже встретиться не могут. Затем при движении слева направо с последовательным сдвигом на одну позицию тройка (0, 0, 1) заменяется на (0, 1, 0), тройка (0, 0, −1) − на (0, −1, 0); неконцевая тройка (0, 0, 0) заменяется на (0, 1, 0) при четном количестве минус единиц в исходном кортеже и на (0, −1, 0) при нечетном. После прохода по всему кортежу случай 1 больше встретиться не может. Для устранения случая 2 при движении слева направо пары (1, −1), (1, 1) заменяются на пару (−1, −1). Новых нулей при этом не появляется. На последнем этапе тройки (−1, 0, 1) заменяются на (−1, 0, −1).

Заметим, принудительное введение в кортеж минус единиц в большем числе случаев, чем введение единиц, хорошо согласуется с тем, что правое поддерево имеет порядок на два меньший, чем левое (в правом поддереве первая компонента кортежа является единицей, а в правом − минус единицей).

Поскольку $F_i = \left\langle \alpha^i / \sqrt{5} \right\rangle$, где $\langle ... \rangle$ обозначает ближайшее целое, а $\alpha = (1+\sqrt{5})/2$, то коэффициент экономии памяти можно оценить как $(\sqrt{5}/\alpha^2)\beta^k$, где $\beta = 3/\alpha$.

Резюмируя, следует сказать, что процедура поиска Фибоначчи может использоваться для формирования как новой трехсимвольной системы счисления, так и новой процедуры хеширования, не предполагающей выполнение «медленных» операций умножения и деления. Рассмотренный подход дает методологическую основу для использования в указанных целях и других поисковых процедур. Наконец, дополнительную вариативность Ф-поисковому подходу придает возможность изменения корневого узла F_k.

СПИСОК ЛИТЕРАТУРЫ

1. **Кнут, В.** Искусство программирования [Текст] / В. Кнут // Сортировка и поиск. −2-е изд. − М.: ИД «Вильямс», 2003. − Т. 3. −820 с.

2. **Воробьев, Н.Н.** Числа Фибоначчи [Текст] / Н.Н. Воробьев. − М.: Наука, 1978. −144 с.

REFERENCES

1. **Knut V.** Iskusstvo programmirovaniia; T. 3. Sortirovka i poisk; 2-e izd. — Moscow: Izdatel'skii dom «Vil'iams», 2003. — 820 s. (rus)

2. **Vorob'ev N.N.** Chisla Fibonachchi. — Moscow: Nauka, 1978. — 144 s. (rus)

ЯСТРЕБОВ Михаил Юрьевич − *заведующий кафедрой математики Государственной морской академии имени адмирала С.О. Макарова, профессор.*

198035, Россия, Санкт-Петербург, ул. Двинская, д. 5/7.

E-mail: mikyast@gmail.com

YASTREBOV, Mikhail Yu. *Admiral Makarov State University of Maritime and Inland Shipping.*

198035, Dvinskaya str., 5/7, St. Petersburg, Russia.

E-mail: mikyast@gmail.com