УДК 373
DOI 10.25688/2072-9014.2022.59.1.03

О. Д. Любутов

ПРИМЕНЕНИЕ ОФИСНЫХ ПРИЛОЖЕНИЙ ДЛЯ ИЗУЧЕНИЯ ДЕРЕВА ФЕНВИКА ПРИ ПОДГОТОВКЕ ШКОЛЬНИКОВ 8–9-Х КЛАССОВ К ОЛИМПИАДАМ ПО ИНФОРМАТИКЕ

В статье рассматривается использование популярного электронного образовательного ресурса MS Excel для изучения дерева Фенвика при подготовке школьников к олимпиадам по информатике.

Ключевые слова: электронный образовательный ресурс; методика обучения информатике; олимпиада по информатике; префиксная сумма; дерево Фенвика; дерево отрезков.

O. D. Lyubutov

THE USE OF OFFICE APPLICATIONS FOR THE STUDY OF THE SEGMENT TREE IN PREPARING SCHOOLCHILDREN OF THE 8–9 FORMS FOR COMPUTER SCIENCE OLYMPIADS

The article considers the use of a popular ELR of MSExcel to explain a Fenwick tree when training students for IT Olympiads.

Keywords: electronic learning resources; IT training methodology; IT Olympiads; prefix sum; Fenwick tree; segment tree.

Современные школьные олимпиады [3; 5–7] требуют от участников, работающих в обстановке ограниченного времени, полной концентрации внимания на решаемой задаче. Но решение задач на олимпиадах по информатике, в отличие от олимпиад по другим предметам, требует знания специфических алгоритмов и в особенности знания специфических структур данных, изучение которых не предусмотрено даже в рамках углубленной школьной программы по информатике.

Именно такой структурой данных является дерево Фенвика. К сожалению, ученики 8–9-х классов чаще всего не имеют представления о подобной структуре данных, методах ее реализации на языках программирования, способах ее реализации в программном коде для решения олимпиадных задач по информатике. Между тем знание свойств подобных структур и умелое использование в своих программах дерева Фенвика является необходимым условием для решения многих олимпиадных задач.

Познакомить учеников с деревом Фенвика, наглядно показать организацию этой структуры данных, способы ее применения при решении олимпиадных задач может модель, реализованная на основе офисного приложения MS Excel [1; 2; 4; 8].

Известно, что большое количество олимпиадных задач по информатике требуют для своего решения расчета префиксных сумм. Префиксной суммой массива $a_0, a_1, a_2, ..., a_n$ называется массив $b_0, b_1, b_2, ..., b_n$, определяющийся следующим образом:

$$
\begin{aligned}
&b_{-1} = 0;\\
&b_0 = a_0;\\
&b_1 = a_0 + a_1;\\
&b_2 = a_0 + a_1 + a_2;\\
&...\\
&b_n = a_0 + a_1 + ... + a_n .
\end{aligned}
$$

Очевидно, что для вычисления значения b_k нет необходимости каждый раз складывать значения всех элементов от a_0 до a_k. Достаточно вычислить значение суммы $b_{k-1} + a_k$. Имея заполненный массив b, можно за одну операцию вычислить сумму на отрезке. Например, если требуется вычислить сумму всех элементов от a_x до a_y включительно, достаточно вычислить разность $b_y - b_{x-1}$.

Приведем пример классической задачи с использованием префиксных сумм.

Задача «Рулетка». В игре «Рулетка» выигрыш регулярно сменяется проигрышем, и наоборот. Как выигрыш, так и проигрыш — это целые числа (положительные или отрицательные). Пускай имеется список из N партий, сыгранных в течение дня ($1 \le N \le 1000$), и для каждой партии задано значение выигрыша (или проигрыша) X_i ($-10000 \le X_i \le 10000$; $1 \le i \le N$). Известно, что один из игроков сыграл несколько партий подряд. Требуется определить сумму максимального выигрыша игрока, а также номер партии k, с которой он начал игру, и номер партии m, после которой он закончил играть ($1 \le k, m \le N$).

Для решения данной задачи можно использовать префиксную сумму. Пройдя по массиву выигрышей и создав массив с префиксной суммой, можно за одно арифметическое действие (вычитание) находить выигрыш между k-й и m-й партиями:

$$b_m - b_{k-1}.$$

Таким образом, сложность алгоритма, решающего данную задачу, сводится к асимптотике O (n^2), что вполне допустимо для заданной размерности N. Если же не использовать префиксную сумму, то сложность возрастает до O (n^3), что не позволит программе уложиться в отведенное время.

Обычно школьники довольно легко справляются с решением подобных задач. Сложности вызывают задачи, в которых вычисления префиксных сумм чередуются с изменениями значений в исходном массиве. После каждого такого изменения необходимо заново пересчитывать префиксную сумму. Рассмотрим следующую задачу.

Задача «Мороженое». В одном городе живет дегустатор мороженого. В каждый из d дней ($1 \leq d \leq 10\,000$) он выходит из дома и идет по улице, на которой расположены N ($1 \leq N \leq 10\,000$) киосков с мороженым, пронумерованных от 0 до $N - 1$. В каждом киоске продается один тип мороженого стоимостью S_k ($1 \leq S_k \leq 100$, $0 \leq k \leq N - 1$). Дегустатор идет вдоль улицы, удерживаясь от покупки мороженого. У киоска с номером a у него заканчивается сила воли, и он начинает покупать по одному мороженому в каждом киоске, начиная с киоска a.

Последнее мороженое он покупает в киоске с номером b ($0 \leq a \leq b < N$). Известно, что ежедневно (в i-й день) только один из киосков (с номером k_i) может изменить цену мороженого на следующий ($i + 1$) день. Требуется написать программу, которая по входным значениям d, N, ($S_0 \div S_{N-1}$), а также по ежедневным значениям a_i, b_i, k_i, S_{ki} определит сумму, которую дегустатор затратил на покупку мороженого за d дней.

Если применять для решения данной задачи префиксную сумму (как в предыдущей задаче), то асимптотическая сложность алгоритма будет O (n^2), потому что при изменении цены на мороженое придется выполнять массовую операцию по пересчету префиксной суммы массива. Такая же сложность алгоритма будет и в случае, если решать данную задачу методом «грубой силы» и ежедневно вычислять сумму цен на мороженое во всех киосках, начиная с киоска с номером a_i и заканчивая киоском с номером b_i. Для более эффективного решения данной задачи можно использовать дерево Фенвика.

Деревом Фенвика называется структура данных, позволяющая изменять значения массива и находить значение некоторой ассоциативной функции f на произвольном отрезке $a_0, a_1, a_2, \ldots, a_n$ массива за асимптотику O ($\log_2 n$). В отличие от дерева отрезков функция должна быть обратимой. Чаще всего в качестве f берутся функции суммы, произведения, сложения по модулю два. Дерево Фенвика реализуется с помощью дополнительного массива размерности n ($f_0, f_1, f_2, \ldots, f_n$).

Рассмотрим операции, которые можно реализовывать с помощью дерева Фенвика:

1. Операция инициализации (или изменения) значения элемента входного массива. При изменении значения a_i изменяется не только значение f_i, но и все элементы массива f, индексы которых определяются следующей формулой:

$$i_{next} = i_{prev} \mid (i_{prev} + 1).$$

Согласно этой формуле в массиве f будет изменено не более чем $\log_2 n$ элементов.

2. Операция вычисления функции на префиксе. Для вычисления функции, например суммы элементов $a_0, a_1, a_2, \ldots, a_k$, необходимо вычислить сумму элементов массива f, индексы которых определяются следующей формулой:

$$i_{next} = i_{prev} \,\&\, (i_{prev} + 1) - 1 \text{ (при этом } i_0 = k).$$

Согласно этой формуле в итоговую сумму будет добавлено не более $\log_2 k$ элементов массива f.

Знаки в формулах вычисления очередного значения индекса массива («|» и «&») означают поразрядную дизъюнкцию и поразрядную конъюнкцию. Вычисление значения суммы на отрезке осуществляется вычитанием значений сумм на префиксах (sum (a, b) = sum (b) – sum (a – 1)).

Следует заметить, что подобное математическое описание дерева Фенвика вызывает определенные сложности при усвоении учениками 8–9-х классов. Для повышения наглядности имеет смысл представить формулы изменения индексов в виде числовых рядов следующего вида:

Для изменения элемента:	Требуется изменить значения элементов:
a_0	$f_0, f_1, f_3, f_7, f_{15}, f_{31}, f_{63}, \ldots$
a_1	$f_1, f_3, f_7, f_{15}, f_{31}, f_{63}, \ldots$
a_2	$f_2, f_3, f_7, f_{15}, f_{31}, f_{63}, \ldots$
a_3	$f_3, f_7, f_{15}, f_{31}, f_{63}, \ldots$

Для вычисления суммы:	Требуется сложить значения элементов:
$a_0 \div a_1$	f_1
$a_0 \div a_2$	f_1, f_2
$a_0 \div a_5$	f_3, f_5
$a_0 \div a_7$	f_7
$a_0 \div a_{14}$	$f_7, f_{11}, f_{13}, f_{14}$

Еще большей наглядности изучаемого материала можно достигнуть, если создать функционирующую модель дерева Фенвика, чтобы учащиеся могли самостоятельно вводить значения во входной массив (a), при этом наблюдая изменения значений в массиве дерева (f) и изменение значений в выходном массиве префиксных сумм (*sum*).

Реализовать подобную модель можно с помощью программы MS Excel, входящей в пакет MS Office (см. рис. 1). В данной модели изображены три одномерных массива: a – массив исходных (входных) значений, f — массив, реализующий дерево Фенвика, и массив *sum* – выходной массив префиксных сумм. При вводе чисел в столбец *B* осуществляется автоматический пересчет значений в строке 3 и столбце *V*. Стрелками указаны взаимосвязи при вычислении элементов в массиве f и массиве *sum*. При работе с данной моделью ученики изменяют значения в массиве a и в динамике наблюдают соответствующие изменения в массивах *sum* и f. Стрелки помогают понять, почему произошли изменения в тех или иных элементах массивов.

Если присвоить элементу входного массива (например, a_2) значение 5, то значения массивов f и *sum* изменятся (см. рис. 2).

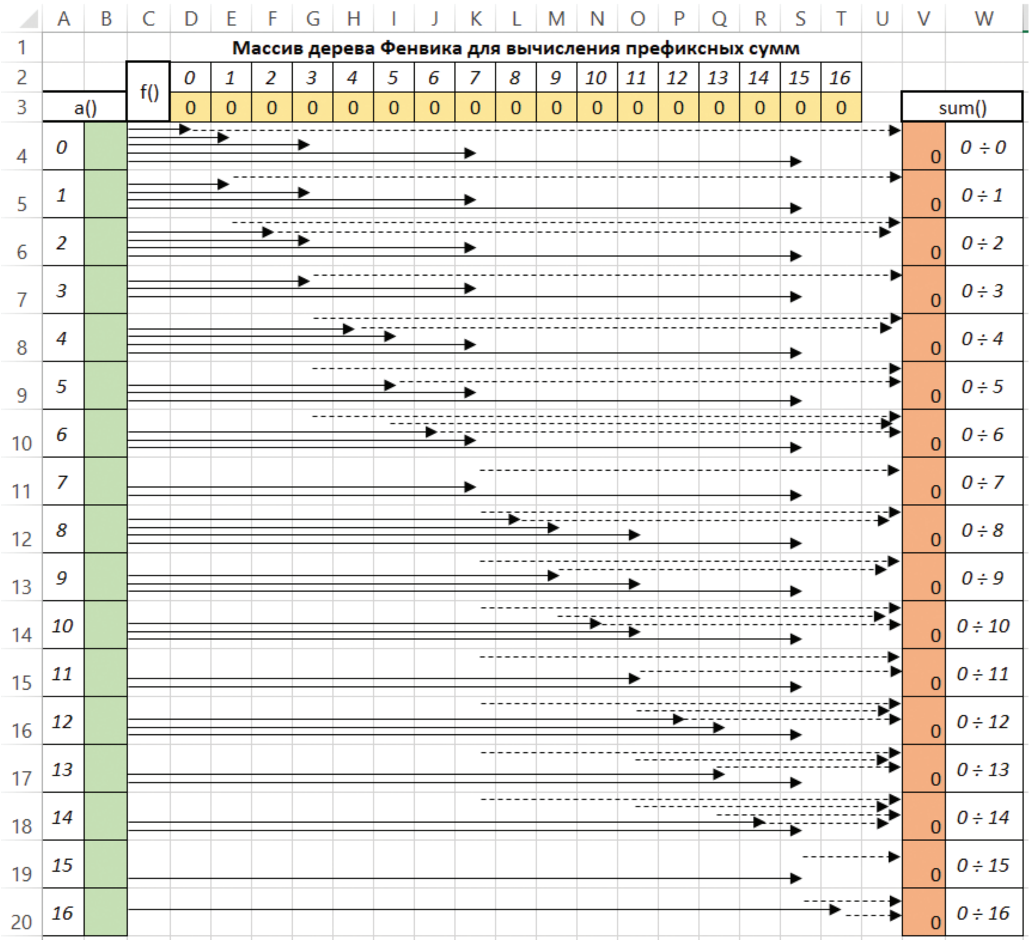

Рис. 1. Функционирующая модель дерева Фенвика

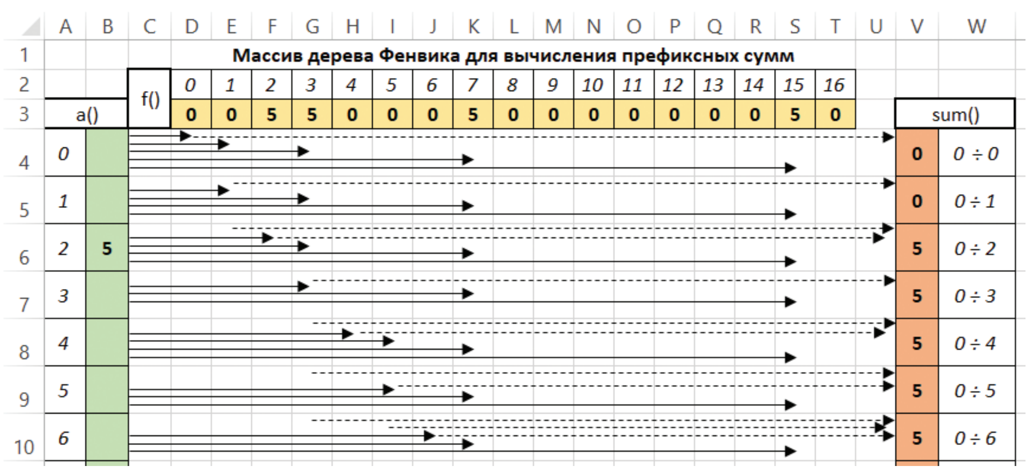

Рис. 2. Массив дерева Фенвика для вычисления префиксных сумм

Из рисунка 2 видно, что все суммы, начиная с 0 ÷ 2, стали равны 5. Если присвоить элементу входного массива (например, a_4) значение –3, то значения массивов *f* и *sum* опять изменятся (рис. 3).

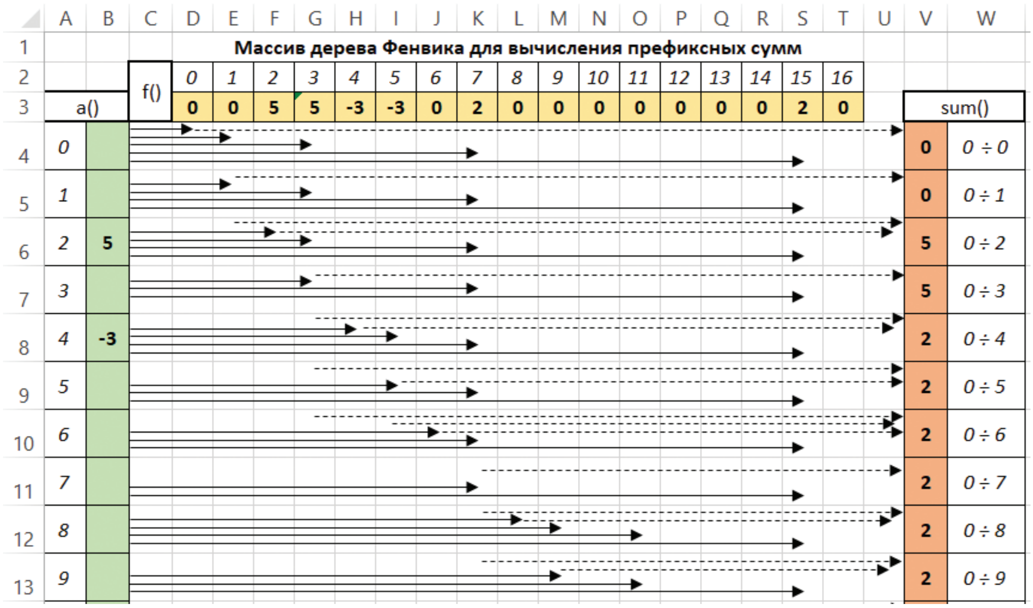

Рис. 3. Массив дерева Фенвика для вычисления префиксных сумм

С помощью дерева Фенвика можно реализовать решение задачи «Мороженое» за асимптотику O ($n \log_2 n$), так как n — количество дней, а изменение цены на мороженое и вычисление стоимости съеденного мороженого на отрезке улицы имеют сложность O ($\log_2 n$).

С помощью дерева Фенвика можно решать и обратную задачу, когда входными значениями являются изменения данных на отрезке, а выходными — значения элементов массива. В качестве обратной рассмотрим следующую задачу.

Задача «Белки». В городском парке живут M ($1 \le M \le 10\,000$) белок. На зиму белки запасают орехи. Для этого они облюбовали аллею из N ($1 \le N \le 10\,000$) деревьев, пронумерованных от 0 до $N - 1$. В каждом дереве есть дупло. Каждая белка, собрав S_k орехов ($1 \le S_k \le 100$, $0 \le k \le N - 1$) раскладывает по одному ореху в каждое дупло подряд, начиная с дерева с номером a и заканчивая деревом с номером b ($b = a + S_k - 1$, $0 \le a \le b < N$). Требуется написать программу, которая по входным значениям M, N, ($S_0 \div S_{N-1}$) определит номер дерева, в дупле которого будет максимальное количество орехов.

Для решения этой задачи построим обратную модель дерева Фенвика (см. рис. 4). В этой модели входными являются значения массива *sum*, а выходными — значения в массиве *a*. Направление стрелок указывает порядок расчета. Для задания входного значения k на отрезке от a до b требуется ввести два значения: значение k в префикс b (sum[b] := k) и значение $-k$

в префикс *a* – 1 (sum[a–1] := –k). Например, если требуется присвоить всем элементам выходного массива от a[3] до a[6] значение 5, то нужно присвоить sum[6] число 5, а sum[2] — число –5 (рис. 4).

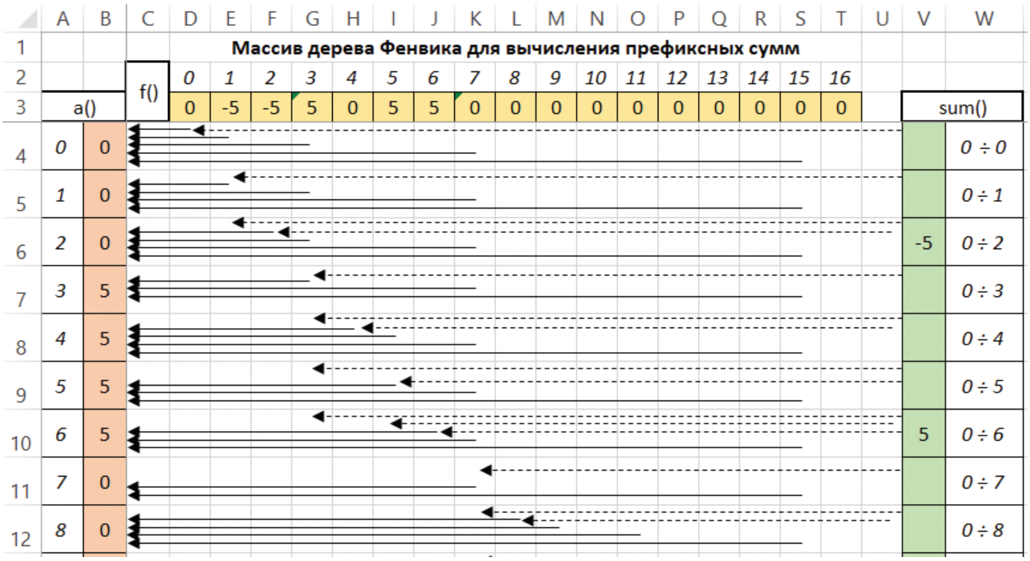

Рис. 4. Массив дерева Фенвика для вычисления префиксных сумм

Проиллюстрируем задачу «Белки» на небольшом наборе данных (рис. 5). Пусть у нас есть 5 белок и 16 деревьев (от 0 до 15):

– первая белка размещает 11 орехов с дерева № 0 по дерево № 10;
– вторая белка размещает 11 орехов с дерева № 5 по дерево № 15;
– третья белка размещает 5 орехов с дерева № 2 по дерево № 6;
– четвертая белка размещает 5 орехов с дерева № 6 по дерево № 10;
– пятая белка размещает 10 орехов с дерева № 4 по дерево № 13.

В столбце *V* таблицы записаны единицы в элементах, соответствующих концам отрезков, и минус единицы в элементах, предшествующих началам отрезков. В столбце *B* таблицы мы видим результат — количество орехов в каждом дереве. Легко заметить, что максимальное количество орехов было положено в дупло дерева № 6. Сложность алгоритма решения данной задачи имеет асимптотику $O(n\log_2 n)$, потому что и операция по раскладке орехов одной белкой, и проверка количества орехов в одном дупле имеют сложность $O(\log_2 n)$.

Если сравнивать дерево Фенвика с аналогичными структурами данных, например с деревом отрезков, то можно отметить следующие преимущества и недостатки (см. табл. 1).

Таким образом, можно сделать вывод о том, что повышение наглядности при изучении дерева Фенвика с помощью электронной таблицы, позволяющей визуализировать процессы, происходящие в дереве, поможет школьникам лучше подготовиться к решению олимпиадных задач по информатике.

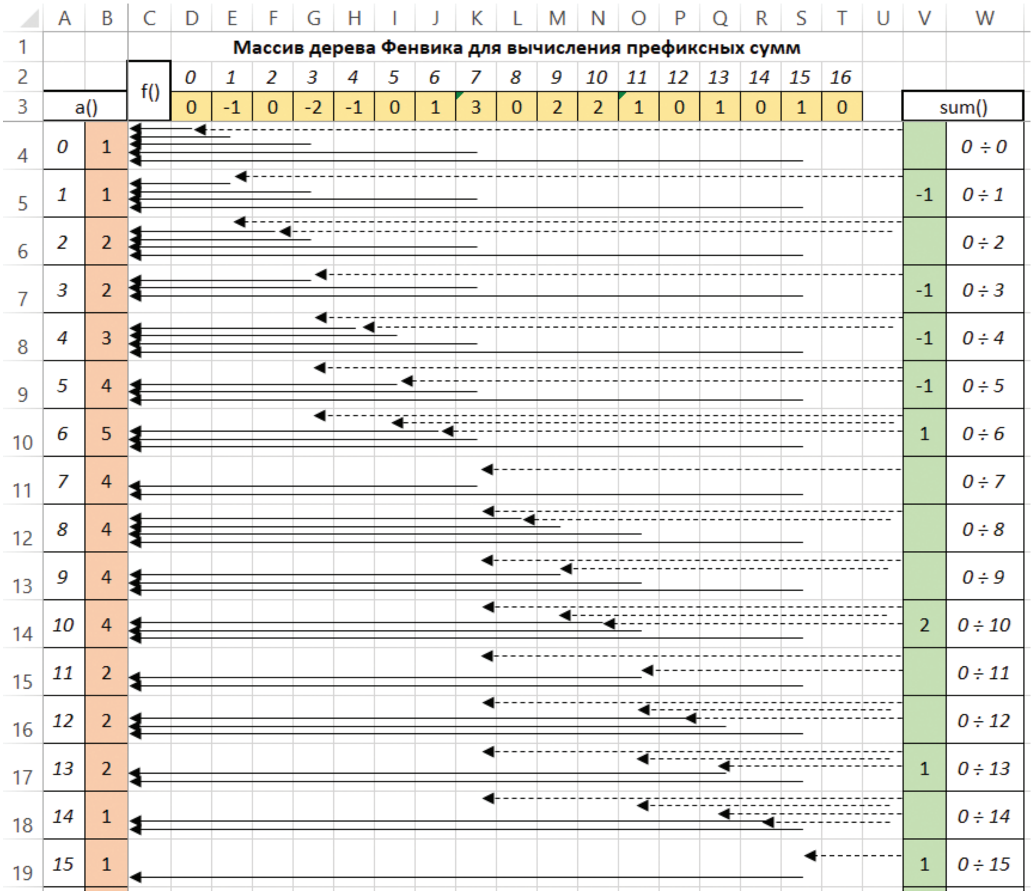

Рис. 5. Массив дерева Фенвика для вычисления префиксных сумм

Таблица 1

Структура данных	Преимущества	Недостатки
Дерево Фенвика	Простота программной реализации. Реализуется итерационно. Занимаемый объем памяти n	Позволяет реализовывать только обратимые функции
Дерево отрезков	Позволяет реализовывать не только обратимые функции, но и необратимые (например, максимум)	Требует рекурсивной реализации и объема памяти $2n$

Литература

1. Бондаренко С., Бондаренко М. Excel 2003. Популярный самоучитель. СПб.: Питер, 2005. 320 с.

2. Киммел П. Excel 2003 и VBA. Справочник программиста / П. Киммел и др. М.: Вильямс, 2005. 725 с.

3. Кирюхин В. М. Информатика: всероссийские олимпиады. Вып. 1. М.: Просвещение, 2008. 220 с.

4. Левитин А. В. Алгоритмы. Введение в разработку и анализ. М.: Вильямс, 2006. 576 с.

5. Меньшиков Ф. В. Олимпиадные задачи по программированию (+CD). СПб.: Питер, 2006. 315 с.

6. Мозговой М. В. Занимательные алгоритмы. Самоучитель. СПб.: Питер, 2004. 208 с.

7. Скиена С. С., Ревилла М. А. Олимпиадные задачи по программированию. Руководство по подготовке к соревнованиям. М.: КУДИЦ-Образ, 2005. 416 с.

8. Уокенбах Дж. Библия пользователя Excel 2003 (+CD). М.: Диалектика, 2007. 768 с.

Literatura

1. Bondarenko S., Bondarenko M. Excel 2003. Populyarny`j samouchitel`. SPb.: Piter, 2005. 320 s.

2. Kimmel P. Excel 2003 i VBA. Spravochnik programmista / P. Kimmel i dr. M.: Vil`yams, 2005. 725 s.

3. Kiryuxin V. M. Informatika: vserossijskie olimpiady`. Vy`p. 1. M.: Prosveshhenie, 2008. 220 s.

4. Levitin A. V. Algoritmy`. Vvedenie v razrabotku i analiz. M.: Vil`yams, 2006. 576 s.

5. Men`shikov F. V. Olimpiadny`e zadachi po programmirovaniyu (+CD). SPb.: Piter, 2006. 315 s.

6. Mozgovoj M. V. Zanimatel`ny`e algoritmy`. Samouchitel`. SPb.: Piter, 2004. 208 s.

7. Skiena S. S., Revilla M. A. Olimpiadny`e zadachi po programmirovaniyu. Rukovodstvo po podgotovke k sorevnovaniyam. M.: KUDICz-Obraz, 2005. 416 s.

8. Uokenbax Dzh. Bibliya pol`zovatelya Excel 2003 (+CD). M.: Dialektika, 2007. 768 s.