УДК 681.3.06: 681.323

ПРЕОБРАЗОВАНИЕ СОРТИРОВКИ ХОАРА В ПАРАЛЛЕЛЬНУЮ ФОРМУ НА ОСНОВЕ МАТРИЦ СРАВНЕНИЙ

Я.Е. Ромм, В.В. Виноградский

Таганрогский государственный педагогический институт,
347926, Таганрог, Россия, ул. Инициативная, 48.
Тел.: 7(863)(4) 60 1753; факс 7(863)(4) 60 5397.
E-mail: Romm@List.ru

Изложена схема преобразования сортировки Хоара в параллельную форму с помощью матриц сравнения. В параллельной форме сортировка устойчива и устанавливает взаимно однозначное соответствие между входными и выходными индексами сортируемых элементов. Временная сложность максимально параллельного видоизменения имеет оценку $O(\log_2 n)$ на $O\left(n^2\right)$ процессорах.

It is shown the transformation of Hoar's sorting in the parallel form with the help of matrix-comparison. In the parallel form the sorting is stable and it sets up a one-for-one correspondence between inlet and outlet indexes of sorting elements. Time complexity of maximum parallel modifications estimates $O(\log_2 n)$ on $O\left(n^2\right)$ processors.

Постановка вопроса. Сортировка Ч. Хоара – одна из наиболее быстрых последовательных сортировок [1, 2], однако, ее видоизменения в параллельную форму не достигают минимальных значений временной сложности без ограничений, включающих, аналогично [3], деление на лучший и худший случаи. Далее рассматривается преобразование данной сортировки в параллельную форму со свойством устойчивости и временной сложностью $O(\log_2 n)$ в общем случае.

Сортировка понимается как внутренняя сортировка по ключу, описывается на примере одномерного массива, упорядочение выполняется по неубыванию "$\le$". Временная сложность сортировки $T(R)$ измеряется количеством ее последовательных шагов (бинарных сравнений), R – число процессоров, обмен не учитывается.

Под устойчивостью [1] сортировки понимается сохранение порядка равных элементов.

Для излагаемого далее преобразования существенны свойства прямой и обратной адресности. Под прямой адресностью понимается сопоставление входным индексам сортируемых элементов индексов этих же элементов на выходе сортировки; сопоставление индексам выходных элементов их индексов на входе сортировки понимается как обратная адресность. В случае устойчивости сортировки прямая и обратная адресность определяют взаимно однозначное соответствие входных и выходных индексов.

Преобразуется следующая разновидность [2] сортировки Хоара. Из массива (здесь и далее нумерация элементов массива от единицы)

$$A = (a[1],\ a[2],\ \dots,\ a[n]) \tag{1}$$

выбирается средний элемент

$$k := n\ div\ 2;\quad a_{cp} := a[k]. \tag{2}$$

Последовательно от начала массива отыскивается первый элемент $a[i] > a_{cp}$ при условии $i < k$, затем последовательно от конца массива – первый элемент $a[j] < a_{cp}$ при условии $j > k$. Найденные элементы меняются местами. Затем в том же порядке просмотра ищется следующая пара элементов из (1), аналогично сравнимых с a_{cp}, найденные элементы меняются местами. В продолжении процесса в левой части массива останутся элементы не большие a_{cp}, в правой – не меньшие a_{cp}. Если по ходу просмотра слева от a_{cp} не окажется большего элемента, то соответственный элемент из правой части массива дописывается в начало его левой части, если справа от a_{cp} не окажется меньшего элемента, то соответственный элемент из левой части массива дописывается в конец его правой части. В итоге достигается упорядочение элементов из (1) относительно среднего элемента (2). Аналогичное упорядочение выполняется для каждого в отдельности из обоих подмассивов слева и справа от a_{cp}, затем для каждого в отдельности из вновь полученных подмассивов и так до окончания сортировки.

Временная сложность упорядочения (1) в лучшем случае имеет оценку $O(n\log_2 n)$, в худшем – $O\left(n^2\right)$ [1].

Требуется преобразовать рассматриваемую сортировку в параллельную форму с временной сложностью $T(R)=O(\log_2 n)$ и оценить R; преобразованная сортировка должна обладать устойчивостью, прямой и обратной адресностью.

Преобразование сортировки Хоара в параллельную форму. Предлагаемое преобразование основано на том, что сравнение всех элементов подмассива со средним можно производить взаимно независимо и синхронно (далее – параллельно) по всем элементам подмассива и параллельно по всем подмассивам на каждом шаге. По результатам сравнений подсчетом значений индексов вычисляются адреса, по которым все элементы каждого подмассива, включая средний, параллельно перемещаются внутри подмассива на выходе его преобразования.

Рассматриваются видоизменения сортировки Хоара для лучшего и худшего случаев, затем параллельный вариант в общем случае с оценкой $T\left(n^2\right)=O(\log_2 n)$.

При оценке временной сложности не учитывается время арифметического сложения индексов для подсчета значений адресов.

Преобразование сортировки Хоара с временной сложностью $O(\log_2 n)$ в лучшем случае. Пусть на время рассуждения предполагается, что на каждом шаге сортировки в каждом подмассиве в качестве среднего элемента возможно выбрать медиану (элемент, разделяющий подмассив на два равных по размеру подмассива). Далее такой случай условно классифицируется как лучший, при изложении схемы будет подразумеваться, что шаги сортировки соответствуют этому абстрактному случаю в некотором приближении, не искажающем абстрактной оценки временной сложности. В данном случае количество подмассивов и, соответственно, с разницей на единицу, число взаимно упорядоченных средних элементов возрастает в геометрической прогрессии с суммой 2^i-1 на выходе i-го шага. При $i=\lceil \log_2(n+1) \rceil$ сортировка закончится.

Здесь и далее $[x]$ – целая часть числа x, $\lceil x \rceil$ – ближайшее целое, не меньшее x.

Каждый шаг можно максимально параллельно выполнить следующим образом. Пусть для определенности элементы (1) – числа, упорядочение выполняется по арифметическому неравенству $\le$, массив (1) идентифицируется как подмассив на первом шаге описываемого алгоритма. При данных допущениях параллельно по всем подмассивам и всем элементам каждого подмассива выполняется вычитание среднего элемента от каждого из элементов подмассива (сравнение всех элементов подмассива со средним элементом) табл. 1.

Таблица 1. Результаты сравнений элементов подмассива с a_{cp}

	a_1	a_2	…	a_{n-1}	a_n
a_{cp}	γ_1	γ_2	…	γ_{n-1}	γ_n

Из табл. 1 вытекает

$$\gamma_i=\begin{cases}"+",\ a_i>a_{cp},\\ "0",\ a_i=a_{cp},\\ "-",\ a_i<a_{cp},\end{cases}\quad i=1,\ 2,\ \dots,\ n. \tag{3}$$

Пример 1. Для $A=(5, 8, 5, 1, 5, 2, 5, 10, -3)$, $a_{cp}=a_5=5$, табл. 1 примет вид:

	5	8	5	1	5	2	5	10	-3
5	0	+	0	-	0	-	0	+	-

В правую часть подмассива перемещаются те элементы левой части, которые при сравнении образовали положительную разность, они располагаются непосредственно справа от среднего элемента с сохранением взаимного порядка. В левую часть перемещаются элементы правой части, образовавшие отрицательную разность, они располагаются непосредственно слева от среднего элемента также с сохранением взаимного порядка.

Адрес перемещения элемента внутри правой части подмассива определяет индекс, равный сумме количества предшествующих «нулей» правой части, «плюсов», образованных предшествующими элементами в левой и правой частях подмассива и выходного значения индекса среднего элемента; внутри левой части – индекс, равный сумме количества «нулей» и «минусов», образованных предшествующими элементами.

Адрес перемещения элемента из левой части подмассива в правую определяется как сумма числа предшествующих элементу «плюсов» и выходного значения индекса среднего элемента, из правой части в левую – разностью между выходным значением индекса среднего элемента и числом следующих за элементом «минусов» в правой части.

Особенности способа перемещения имеют цель сохранить порядок равных («потенциально» равных) элементов. Нетрудно видеть, что данный способ обеспечивает достижение данной цели.

Перемещение выполняется параллельно во всех подмассивах по всем элементам внутри каждого подмассива.

По построению алгоритма для его выполнения достаточно n процессоров, что в рассматриваемом случае влечет оценку $T(n)=O(\log_2 n)$.

Замечание 1. По построению схемы способ параллельного вычисления адресов и параллельного перемещения элементов на каждом шаге инвариантен относительно выбора в подмассиве положения a_{cp}. Это положение можно выбирать не в середине, а в любом месте подмассива, включая его границы.

Общее для произвольного смещения a_{cp} преобразование одного произвольно взятого подмассива по шагам параллельной сортировки описывает

Алгоритм 1.

Шаг 1. В подмассиве (1) выбирается средний элемент

$$j_{cp} := n \ div \ 2;\ a_{cp} := a[\,j_{cp}\,]. \qquad (4)$$

Выполняется параллельное сравнение всех элементов подмассива со средним, результаты определяются из (3) по табл. 1.

Значение адреса перемещения элемента определяются по следующей схеме.

Если j_{cp} – индекс среднего элемента в подмассиве до перемещения, а $j_{cp}^{(0)}$ – после него, то, вообще говоря, $a_{cp}[\,j_{cp}\,] \to a_{cp}[\,j_{cp}^{(0)}\,]$, где

$$j_{cp}^{(0)} = 1 + i + j,$$

i – количество «минусов» и «нулей» в левой части, j – количество «минусов» в правой части подмассива.

Индекс среднего элемента в примере 1 $a_{cp}[5] \to a_{cp}[1+3+2] = a_{cp}[6]$.

Адрес перемещения элемента из левой части подмассива в правую определяются соотношением

$$a_{л}[k] \to a_{пр}[\,j_{cp}^{(0)} + 1 + \ell\,],$$

где k, $j_{cp}^{(0)} + 1 + \ell$ – индексы в подмассиве до и после перемещения, ℓ – количество элементов больших среднего в левой части подмассива с индексами меньшими k (количество предшествующих «плюсов»).

В примере 1 $a[2] \to a[6+1+0] = a\,[7]$.

Адрес перемещения элемента из правой части подмассива в левую определяются соотношением

$$a_{пр}[k] \to a_{л}[\,j_{cp}^{(0)} - 1 - m\,],$$

где k, $j_{cp}^{(0)} - 1 - m$ – индексы в подмассиве до и после перемещения, m – количество элементов меньших среднего с порядковыми номерами большими k (количество последующих «минусов»).

В примере 1 $a[6] \to a[6-1-0] = a[5]$, $a[9] \to a[6-1-1] = a[4]$.

Адрес перемещения элемента внутри левой части подмассива определяется соотношением

$$a_{л}[k] \to a_{л}[1+p],$$

где k, $1+p$ – индексы в подмассиве до и после перемещения, p – количество элементов, до перемещения меньших среднего, либо равных ему, с порядковыми номерами меньшими k (количество предшествующих «нулей» и «минусов»).

В примере 1 $a[1] \to a[1+0] = a[1]$, $a[3] \to a[1+1] = a[2]$, $a[4] \to a[1+2] = a[3]$.

Адрес перемещения элемента внутри правой части подмассива определяется соотношением

$$a_{пр}[k] \to a_{пр}[\,j_{cp}^{(0)} + 1 + r + t\,],$$

где k, $1+ j_{cp}^{(0)} + r + t$ – индексы в помассиве до и после перемещения, r – количество элементов из левой части подмассива больших $a_{cp}[\,j_{cp}\,]$ (количество предшествующих «плюсов» слева от j_{cp}), t – количество элементов из правой части подмассива не меньших среднего с индексами меньшими k (количество предшествующих «плюсов» и «нулей» справа от j_{cp}).

В примере 1 $a[7] \to a[1+6+1+0] = a[8]$, $a[8] \to a[1+6+1+1] = a[9]$.

После параллельного перемещения элементов в конечное положение на выходе шага подмассив разделяется на два новых подмассива относительно перемещенного элемента $a_{cp}[j_{cp}^{(0)}]$.

В примере 1 на выходе шага 1 получится: (5, 5, 1, −3, 2), 5, (8, 5, 10).

Временная сложность шага $T_1(R) = \tau$, где $R = n-1$, τ – время одного бинарного сравнения.

Шаг i, $i \geq 2$. Процесс, описанный для шага 1, в лучшем случае параллельно повторяется для каждого из 2^{i-1} полученных на предыдущем шаге подмассивов.

Временная сложность шага составит $T_i(R) = \tau$, где $R \leq n$.

Шаг $i = \lceil \log_2(n+1) \rceil$. Этот шаг повторяет операции предыдущего шага в каждом из полученных на его выходе подмассивов, на выходе рассматриваемого шага достигается разделение всего преобразуемого массива на подмассивы, содержащие по одному элементу, это означает окончание сортировки.

По построению в лучшем случае для всех сравнений каждого шага достаточно не более n процессоров, в итоге $T(n) = O(\log_2 n)$. Схема адресации не меняет порядка равных элементов, обеспечивая устойчивость преобразованной сортировки.

Сортировка в худшем случае с временной сложностью $O(n)$. Следующий случай классифицируется как худший: на каждом шаге сортировки, при сохранении алгоритма 1, число средних элементов увеличивается на единицу (средний элемент постоянно перемещается на выходе шага в начало или конец преобразуемого подмассива). Число взаимно упорядоченных элементов на шаге увеличивается на единицу, для сортировки по алгоритму 1 потребуется не более n шагов, для выполнения каждого шага достаточно не более n процессоров. В этом случае $T(n) = O(n)$.

Улучшить последнюю оценку можно за счет увеличения на шаге числа параллельно работающих процессоров. Это предполагает соответственный рост числа параллельных сравнений на шаге, которые в контексте преобразуемой сортировки должны относиться к средним элементам. Отсюда следует рост числа средних элементов в каждом подмассиве в зависимости от номера шага. В описываемом далее алгоритме средние элементы, вообще говоря, уже не занимают серединного положения в подмассиве, они равно по длине всего массива отстоят друг от друга на входе шага и будут идентифицироваться с помощью кавычек как «H -отстоящие» элементы.

Параллельная сортировка с временной сложностью $O(\log_2 n)$ в общем случае. При произвольном смещении на шаге средних элементов сортировка заведомо выполнима за $\lceil \log_2(n+1) \rceil$ шагов, если на входе каждого шага количество «H -отстоящих» элементов увеличивается вдвое по сравнению с предыдущим. Необходимо, чтобы они затем взаимно упорядочивались с единичной временной сложностью на шаге и, с сохранением оценки, между ними на том же шаге оказывались расставленными все промежуточные по величине элементы подмассива. Расстановка должна соответствовать схеме Хоара и обеспечивать упорядоченность относительно всех «H -отстоящих» элементов подмассива.

С целью искомого преобразования на i-м шаге слева направо фиксируются «H -отстоящие» элементы вдвое ближе друг к другу, чем их аналоги на $(i-1)$-м шаге, номера фиксируемых элементов задаются равноотстоящими путем сквозной нумерации по всем элементам всего массива, полученного на выходе $(i-1)$-го шага.

Все подмассивы на i-м шаге обрабатываются параллельно.

В текущем подмассиве при этом выполняются следующие операции.

Чтобы взаимно упорядочить в подмассиве зафиксированные «H -отстоящие» элементы и с целью последующих преобразований, каждый из этих элементов параллельно сравнивается со всеми элементами подмассива, включая все другие «H -отстоящие» элементы этого подмассива. Аналогично схеме алгоритма 1, параллельно подсчитываются выходные адреса всех «H -отстоящих» элементов в подмассиве и осуществляется их параллельное перемещение по этим адресам. В результате все «H -отстоящие» элементы взаимно упорядочиваются.

Затем параллельно определяются выходные адреса всех остальных элементов подмассива, элементы параллельно перемещаются с сохранением взаимного порядка равных из них в интервалы между соответствующими парами «H -отстоящих» элементов.

Замечание 2. Если при сквозной нумерации элементов массива на входе шага какие-либо из «H -отстоящих» элементов совпадают с правыми границами подмассивов, сформированных на выходе предыдущего шага, и при этом левая граница таких подмассивов принадлежит замкнутому промежутку между соседними «H -отстоящими» элементами, то рассматриваемые подмассивы уже упорядочены относительно «H -отстоящих» элементов и их не требуется подвергать преобразованиям на этом шаге.

Замечание 3. Сортировка закончится, если все подмассивы упорядочатся так, что каждый из них состоит из одного элемента и границ, – взаимно упорядоченных «H -отстоящих» элементов.

Соответственно геометрической прогрессии количества взаимно упорядочиваемых элементов число последовательных шагов сортировки не превзойдет $\lceil \log_2(n+1) \rceil$.

При формальном описании данного алгоритма массив (1) по-прежнему идентифицируется как подмассив на первом шаге.

Алгоритм 2.

Шаг 2. Выполняется первый шаг алгоритма 1.

Шаг i, $i \ge 2$. Пусть в результате $(i-1)$-го шага сортируемый массив оказался разделенным на m подмассивов, состоящих, соответственно, из $\ell_1, \ell_2, ..., \ell_m$ элементов. Пусть при этом в результате $i-1$ предшествующих шагов свое конечное положение с взаимной упорядоченностью и упорядоченностью относительно них промежуточных элементов заняли $2^0 + 2^1 + \ldots + 2^{i-2} = 2^{i-1} - 1$ «H-отстоящих» элементов. Тогда на входе i-го шага фиксируются 2^{i-1} новых «H-отстоящих» элементов, которые подлежат на этом шаге взаимному упорядочению, кроме того, упорядочению относительно «H-отстоящих» подлежат остальные элементы подмассива. Фиксирование равноотстоящих («H-отстоящих») элементов выполняется сквозной нумерацией по всем элементам полного массива, полученного на выходе предыдущего шага.

Пусть для каждого j, $1 \le j \le m$, в j-м подмассиве (с числом элементов ℓ_j) на входе i-го шага зафиксировано k_j «H-отстоящих» элементов вида

$$a[\beta],\ a[\beta + \left[\frac{n}{2^i}\right]],\ a[\beta + 2\cdot\left[\frac{n}{2^i}\right]],\ ...,\ a[\beta + (k_j - 1)\cdot\left[\frac{n}{2^i}\right]],$$

при этом $\sum\limits_{1 \le j \le m} k_j = 2^{i-1}$.

Для искомого упорядочения параллельно по всем j, $1 \le j \le m$, выполняется параллельное сравнение всех зафиксированных на данном шаге k_j «H-отстоящих» элементов j-го подмассива со всеми элементами этого подмассива.

Сравнения можно представить в виде матрицы.

В табл. 2 символ ∞ обозначает число, заведомо большее модуля любого из сравниваемых чисел,

$$\gamma_{pr} = \begin{cases} "+",\ a_p > a_r, \\ "0",\ a_p = a_r, \\ "-",\ a_p < a_r \end{cases} \quad p = 0, 1, \ldots, k_j + 1; \quad r = 1, 2, \ldots, \ell_j.$$

Табл. 2 включает в себя, в частности, сравнения всех «H-отстоящих» элементов данного подмассива между собой. Выбирая соответственные им столбцы и строки, можно получить отдельную матрицу сравнений «H-отстоящих» элементов j-го подмассива между собой, она представляет собой квадратную матрицу с нулевой диагональю и антисимметричными относительно этой диагонали элементами [4].

Таблица 2. Матрица сравнений элементов j-го подмассива с «H-отстоящими» элементами

	a_1	a_2	…	a_{ℓ_j}
$-\infty$	γ_{01}	γ_{02}	…	$\gamma_{0\ell_j}$
$a[\beta]$	γ_{11}	γ_{12}	…	$\gamma_{1\ell_j}$
$a[\beta + \left[\frac{n}{2^i}\right]]$	γ_{21}	γ_{22}	…	$\gamma_{2\ell_j}$
…	…	…	…	…
$a[\beta + (k_j - 1)\cdot\left[\frac{n}{2^i}\right]]$	$\gamma_{k_j 1}$	$\gamma_{k_j 2}$	…	$\gamma_{k_j \ell_j}$
∞	$\gamma_{(k_j+1)1}$	$\gamma_{(k_j+1)2}$	…	$\gamma_{(k_j+1)\ell_j}$

Далее в табл. 3 значения $\tilde{\gamma}_{pr}$ выбраны на пересечениях соответственных строк и столбцов,

$$\tilde{\gamma}_{pr} = -\tilde{\gamma}_{rp},\ r = 1, 2, ..., k_j,\ p = 1, 2, ..., k_j.$$

По результатам сравнений производится подсчет адресов «H -отстоящих» элементов для их взаимного упорядочения. Адрес k -го слева направо «H -отстоящего» элемента получается как число, равное количеству «нулей» и «плюсов» k -го столбца матрицы, расположенных над диагональю и на самой диагонали, сложенному (для сохранения порядка равных элементов) с количеством только «плюсов» этого же столбца ниже диагонали [4]. Пусть это количество обозначено ℓ .

Замечание 4. Сортировка устойчива и является адресной: выходной индекс, равный ℓ , сопоставляется входному индексу k : $c[\ell] := a[k]$. При этом запоминается обратный адрес отсортированного элемента $c[\ell]$ (его входной индекс) и располагается в порядке отсортированных элементов: $e[\ell] := k$. С таким запоминанием выходной массив может получаться из элементов входного без перемещения [5]: $c[\ell] = a[e[\ell]]$.

Таблица 3. Матрица взаимных сравнений «H -отстоящих» элементов j -го подмассива

	$a[\beta]$	$a[\beta+\left[\frac{n}{2^i}\right]]$	…	$a[\beta+(k_j-1)\cdot\left[\frac{n}{2^i}\right]]$
$a[\beta]$	0	$\tilde{\gamma}_{12}$	…	$\tilde{\gamma}_{1k_j}$
$a[\beta+\left[\frac{n}{2^i}\right]]$	$\tilde{\gamma}_{21}$	0	….	$\tilde{\gamma}_{2k_j}$
…	…	…	…	…
$a[\beta+(k_j-1)\cdot\left[\frac{n}{2^i}\right]]$	$\tilde{\gamma}_{k_j1}$	$\tilde{\gamma}_{k_j2}$	…	0

Параллельная перестановка по найденным адресам влечет взаимное упорядочение всех «H -отстоящих» элементов j -го подмассива. Схема соответствует модифицированной с максимальным параллелизмом и устойчивостью [4] сортировке подсчетом, оценивается временной сложностью $T(R)=O(1)$, где R составляет число, равное половине элементов табл. 3 за вычетом диагональных элементов, $R=\frac{1}{2}(k_j^2-k_j)$. В данном случае это количество включается как часть в число процессоров, выполняющих сравнения по табл. 2.

Следующий пример рассматриваемые преобразования.

Пример 2. Пусть j -й подмассив в результате $i-1$ -го шага принял вид

$$(22, 99, 100, 50, 23, 19, 23, 1, 6, -3, 22, 1, 22, -10, -10). \quad (5)$$

Пусть на i -м шаге в нем оказалось пять «H -отстоящих» элементов, начиная со второго, –

$$a[2] := 99,\ a[5] := 23,\ a[8] := 1,\ a[11] := 22,\ a[14] := -10. \quad (6)$$

Табл.2 для (5), (6) примет вид

Из столбцов и строк табл. 4, отмеченных пунктиром, выделяется матрица сравнений «H -отстоящих» элементов между собой:

Таблица 4. Матрица сравнений элементов подмассива с «H -отстоящими» элементами

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	22	99	100	50	23	19	23	1	6	-3	22	1	22	-10	-10
$-\infty$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
99	-	0	+	-	-	-	-	-	-	-	-	-	-	-	-
23	-	+	+	+	0	-	0	-	-	-	-	-	-	-	-
1	+	+	+	+	+	+	+	0	+	-	+	0	+	-	-
22	0	+	+	+	+	-	+	-	-	-	0	-	0	-	-
-10	+	+	+	+	+	+	+	+	+	+	+	+	+	0	0
∞	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Таблица 5. Матрица сравнений
«H -отстоящих» элементов подмассива

	99	23	1	22	-10
99	0	-	-	-	-
23	+	0	-	-	-
1	+	+	0	+	-
22	+	+	-	0	-
-10	+	+	+	+	0

В обозначениях замечания 4 взаимный порядок «H -отстоящих» элементов после подсчетов по столбцам примет вид:

$$a[2] \to c[1+4] = c[5],\ a[5] \to c[1+3] = c[4],\ a[8] \to c[1+1] = c[2],$$
$$a[11] \to c[2+1] = c[3],\ a[14] \to c[1+0] = c[1],$$

где в квадратных скобках выходной адрес «H -отстоящего» элемента

аддитивно представлен количеством «нулей» и «плюсов» не ниже диагонали и количеством «плюсов» ниже диагонали.

При этом $e[5]=2$, $e[4]=5$, $e[2]=8$, $e[3]=11$, $e[1]=14$.

Переставив в соответствии с найденным взаимным порядком строки «H -отстоящих» элементов табл. 2, можно параллельно выполнить по этой же таблице сравнения всех оставшихся элементов j-го подмассива с взаимно упорядоченными «H -отстоящими» элементами. Заново производить сравнения «H -отстоящих» элементов между собой уже не требуется, а значения остальных сравнений позволяют в точности указать положение сравниваемого элемента между двумя конкретными соседними «H -отстоящими» элементами (с учетом границ подмассива). Это достигается следующим образом.

Пусть в рассматриваемой таблице уже выполнена перестановка строк «H -отстоящих» элементов в соответствии их взаимному порядку (неубыванию) сверху вниз, в таком виде она идентифицируется как модифицированная табл. 2. Тогда результаты сравнения в ней «H -отстоящих» элементов с остальными определяются нестрогой монотонностью расположения «H -отстоящих» элементов: вниз от каждого «минуса» в столбце располагаются только «минусы», вверх от каждого «нуля» – только «нули» или «плюсы», вверх от каждого «плюса» – только «плюсы». Смена знака в столбце определяется как пара соседних клеток, верхняя из которых неотрицательна, нижняя – отрицательна. Строка со сменой знака идентифицируется по неотрицательному элементу.

Перестановка строк табл. 4 согласно взаимному порядку «H -отстоящих» элементов влечет модифицированную табл. 6.

Таблица 6. Матрица сравнений элементов подмассива с «H -отстоящими» элементами после их взаимного упорядочения

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	22	99	100	50	23	19	23	1	6	-3	22	1	22	-10	-10
$-\infty$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
– 10	+	+	+	+	+	+	+	+	+	+	+	+	+	0	0
1	+	+	+	+	+	+	+	0	+	-	+	0	+	-	-
22	0	+	+	+	+	-	+	-	-	-	0	-	0	-	-
23	-	+	+	+	0	-	0	-	-	-	-	-	-	-	-
99	-	0	+	-	-	-	-	-	-	-	-	-	-	-	-
∞	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Очевидно, смена знака в столбце с единственностью определяет положение элемента с номером данного столбца между двумя взаимно упорядоченными «H -отстоящими» элементами. Именно, если при смене знака результат сравнения с «H -отстоящим» элементом есть «плюс», то на выходе шага элемент должен расположиться после данного «H -отстоящего» (ниже) и перед следующим (выше). Если при смене знака результат сравнения с «H -отстоящим» элементом есть «ноль», то элемент на выходе шага должен расположиться после данного «H -отстоящего» и перед следующим в том случае, когда в табл. 2 его номер больше номера столбца (входного номера на шаге) меньшего из рассматриваемой пары «H -отстоящих» элементов; если его номер был меньше входного номера на шаге меньшего элемента рассматриваемой пары, то сравниваемый элемент должен расположиться перед этим меньшим на выходе шага.

Взаимный порядок элементов, которые на выходе шага расположатся между элементами пары «H -отстоящих», определяется числом «нулей» и «плюсов» в той строке модифицированной табл. 2, где этот элемент образовал смену знака, с учетом их расположения относительно номеров данной пары «H -отстоящих» элементов.

Элементы, которые при смене знака образовали положительную разность, располагаются непосредственно справа от меньшего элемента пары «H -отстоящих» с сохранением взаимного порядка. В левую часть перемещаются элементы, при смене знака образовавшие отрицательную разность, они располагаются непосредственно слева от меньшего элемента пары также с сохранением взаимного порядка. При соответствующей формализации вычисления адресов это сохранит порядок равных элементов.

Пусть «H -отстоящий» элемент ℓ-й строки модифицированной табл. 2 обозначен $a_H[\ell]$, его порядковый номер после перемещения в подмассиве на выходе шага обозначен $j_{H\ell}$ и пусть $a_H[\ell]$ в этой таблице находится в столбце с номером q (по замечанию 4 этот входной номер $a_H[\ell]$ восстанавливается по номеру ℓ в процессе взаимного упорядочения «H -отстоящих» элементов).

В рассматриваемых обозначениях,

$$j_{H\ell}=1+r\ , \tag{7}$$

где r – сумма числа всех минусов ℓ-й строки таблицы и числа нулей данной строки, предшествующих слева направо столбцу с номером q. Очевидно, r равно числу предшествующих в смысле отношения порядка элементов, при перемещении согласно (7) сохраняется порядок равных элементов.

Для остальных элементов адреса перемещений внутри подмассива формализуются следующим образом.

Пусть в той же строке модифицированной табл. 2 элемент a_p образовал смену знака типа «плюс, минус». Тогда выходной индекс на шаге для a_p определяется соотношением

$$a[p] \to a[j_{H\ell} + 1 + r_0], \tag{8}$$

где p – порядковый номер элемента в подмассиве на выходе предыдущего шага, $j_{H\ell} + 1 + r_0$ – его порядковый номер в этом подмассиве на выходе рассматриваемого шага, $j_{H\ell}$ – выходной номер $a_H[\ell]$ из (7), r_0 – число предшествующих элементу a_p «плюсов», образовавших в ℓ-й строке смену знака.

Пусть a_p образовал в ℓ-й строке смену знака типа «ноль, минус» при условии $p > q$ (входной номер a_p больше входного номера $a_H[\ell]$). Тогда выходной индекс на шаге для a_p определяется из соотношения

$$a[p] \to a[j_{H\ell} + 1 + r_1 + t_1], \tag{9}$$

где p и $j_{H\ell} + 1 + r_1 + t_1$ имеют такой же смысл, как в (8), r_1 – число всех «плюсов», образовавших смену знака в ℓ-й строке, t_1 – число всех тех «нулей», образовавших в ℓ-й строке смену знака, которые находятся в столбцах с номерами, большими q и меньшими p.

Пусть a_p образовал смену знака типа «ноль, минус» при условии $p < q$ (входной номер a_p меньше входного номера $a_H[\ell]$). Тогда

$$a[p] \to a[j_{H\ell} - 1 - r_2] \tag{10}$$

где r_2 – количество последующих за a_p «нулей», образовавших в ℓ-й строке смену знака, которые расположены в столбцах с номерами меньшими q.

Замечание 6. Схема обеспечивает устойчивость сортировки благодаря тому, что при подсчете адресов учитывается и в процессе перемещения сохраняется взаимное расположение равных (и «потенциально» равных) элементов во всех рассмотренных случаях (7) – (10).

В примере 2 конечное положение «H-отстоящих» элементов на выходе шага определяется согласно (7):

$$a_H[2] \to a[1+13] = a[14],\ a_H[5] \to a[1+10] = a[11],$$

$$a_H[8] \to a[1+3] = a[4],\ a_H[11] \to a[1+8] = a[9],\ a_H[14] \to a[1+0] = a[1].$$

Конечные адреса остальных элементов подмассива на выходе шага определяется в соответствии с (8) – (10):

$$a[1] \to a[9-1-0] = a[8],\ a[3] \to a[14+1+0] = a[15],\ a[4] \to a[11+1+0] = a[12],\ a[6] \to a[4+1+0] = a[5],$$
$$a[7] \to a[11+1+1] = a[13],\ a[9] \to a[4+1+1] = a[6],\ a[10] \to a[1+1+0] = a[2],\ a[12] \to a[4+1+2] = a[7],$$
$$a[13] \to a[9+1+0] = a[10],\ a[15] \to a[1+1+1] = a[3].$$

На выходе данного шага, после перемещения всех элементов, подмассив (5) разделится на подмассивы относительно взаимно упорядоченных «H-отстоящих» элементов и примет вид

$$\big(-10, (-3, -10), 1, (19, 6, 1, 22), 22, (22), 23, (50, 23), 99, (100)\big),$$

где выходные подмассивы выделены круглыми скобками.

Шаг $i = \lceil \log_2(n+1) \rceil$. Этот шаг повторяет операции предыдущего шага в каждом из полученных на его выходе подмассивов, на выходе рассматриваемого шага достигается окончание сортировки.

С учетом взаимной независимости сравнений временная сложность i-го шага есть $T(R) = \tau$, где количество процессоров (с учетом сквозной нумерации «H-отстоящих» элементов) определяется числом одновременных сравнений 2^{i-1} «H-отстоящих» элементов с не более чем $n - 2^{i-1} + 1$ остальными элементами. Отсюда $R \le 2^{i-1} \cdot (n - 2^{i-1} + 1)$. При этом предполагается, что физическая перестановка «H-отстоящих» элементов не выполняется, а учитывается с помощью обратной адресации (иначе $T(R) = 2\tau$).

Временная сложность всей сортировки составит $T(R)=O(\log_2 n)$, где $R \le \max\limits_{1 \le i \le \log_2 n} 2^{i-1}\cdot(n-2^{i-1}+1)$ или $R=\frac{(n+1)^2}{4}$.

Таким образом, имеет место

Теорема 1. По изложенной схеме сортировка Хоара преобразуется в форму параллельного алгоритма, временная сложность которого $T(R)=O(\log_2 n)$ достигается с использованием $R=\frac{(n+1)^2}{4}$ процессоров. В преобразованной форме сортировка устойчива и устанавливает взаимно однозначное соответствие между индексами входных и выходных элементов.

Физически выделять из табл. 4 табл. 5 не необходимо, все требуемые для последней столбцы и строки сравнений содержит табл. 4. Не необходим также физический перевод табл. 4 в форму табл. 6, поскольку сортировка подсчетом на основе табл. 5 дает адреса перестановки строк с взаимным порядком «H -отстоящих» элементов и соответственные этим адресам обратные адреса строк табл. 4, с которыми эквивалентна табл. 6.

Предложенное преобразование сохраняет свойства прямой и обратной адресности, если соответственно перемещению элементов перемещать их входные индексы.

Заключение. Преобразование сортировки Хоара в форму параллельного алгоритма в лучшем случае достигает временной сложности $T(n)=O(\log_2 n)$, в худшем – $T(n)=O(n)$. Предложено преобразование этой сортировки в максимально параллельную форму в общем случае с оценкой временной сложности $T(R)=O(\log_2 n)$ при использовании $R=\frac{(n+1)^2}{4}$ процессоров. Такая оценка сопоставима [6] с оценкой параллельной сортировки слиянием по матрицам сравнений, но выше описанная сортировка отличается возможностью выполнения с числом процессоров $R=O(n)$ в лучшем случае. В предложенной форме сортировка устойчива, при этом соблюдается взаимно однозначное соответствие входных и выходных индексов элементов сортируемого массива. На основе данных свойств преобразованная сортировка применима по аналогии с методами из [5] – [7] для локализации нулей и экстремумов функций, для поиска и распознавания.

1. *Вирт Н.* Алгоритмы и структуры данных. – М.: Наука, 1989. – 384 с.
2. *Кнут Д.* Искусство программирования для ЭВМ. Т.3. Сортировка и поиск. – М.: Мир, 1978. – 844 с.
3. *Baudet G., Stevenson D.* Optimal sorting algorithms for parallel computers // IEEE Transactions of computers. – January 1978. – C. 27, N 1. – P. 84 – 87.
4. *Ромм Я.Е.* Параллельная сортировка слиянием по матрицам сравнений. II // Кибернетика и системный анализ. – 1995. – № 4. – С. 13 – 37.
5. *Ромм Я.Е., Гуревич М.Ю., Белоконова С.С., Соловьева И.А.* Вычисление нулей и полюсов функций на основе устойчивой адресной сортировки с приложением к поиску и распознаванию // Материалы 4 Междунар. научно-практ. конф. по программированию УкрПРОГ'2004.Киев, 2004. – № 2/3. – С. 462 – 472.
6. *Ромм Я.Е., Заика И.В., Тюшнякова И.А.* Идентификация экстремумов функций на основе сортировки с приложением к вычислительным схемам алгебры, анализа и распознаванию изображений // Материалы 5 Международной научно-практ. конф. по программированию УкрПРОГ'2006. – Киев, 2006. № 2-3. – С. 708 – 717.
7. *Ромм Я.Е.* Локализация и устойчивое вычисление нулей многочлена на основе сортировки. I // Кибернетика и системный анализ. – 2007. – № 1. – С. 165 – 182.