

- СН. 2. Penza, 20 marta 2018 goda [Innovative scientific research: theory, methodology, practice: Collection of articles of the XIII International Scientific and Practical Conference: in 2 parts. Volume Part 2. Penza, March 20, 2018]. Penza: Nauka i Prosveshchenie (IP Gulyaev G.YU.), 2018, pp. 165-170. EDN YTKTDQ.
17. Song Y. [et al.]. An Ensemble of Retrieval-Based and Generation-Based Human-Computer Conversation Systems, *IJCAI*, 2018.
18. Yang L. [et al.]. A Hybrid Retrieval-Generation Neural Conversation Model, *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1341-1350.
19. Venkatesh A. [et al.]. On Evaluating and Comparing Conversational Agents, *arXiv preprint arXiv:1801.03625*, 2018.
20. Serban I. [et al.]. Multiresolution Recurrent Neural Networks: An Application to Dialogue Response Generation, *AAAI*, 2017.

Статью рекомендовала к опубликованию д.т.н., профессор М.А. Бутакова.

Куликова Яна Владимировна – Волгоградский государственный технический университет; e-mail: cul.jana@yandex.ru; г. Волгоград, Россия; кафедра программного обеспечения автоматизированных систем; старший преподаватель.

Литвиненко Владимир Алексеевич – e-mail: vladimirlit00.00@mail.ru; бакалавр.

Kulikova Yana Vladimirovna – Volgograd State Technical University; e-mail: cul.jana@yandex.ru; Volgograd, Russia; the department software engineering, senior lecturer.

Litvinenko Vladimir Alekseevich – e-mail: cul.jana@yandex.ru; the department software engineering; student.

УДК 004.421.6

DOI 10.18522/2311-3103-2023-5-104-118

И.И. Левин, К.Н. Алексеев

ПРЕОБРАЗОВАНИЕ СОРТИРУЮЩИХ СЕТЕЙ ДЛЯ РАЗНОЙ СТЕПЕНИ ПАРАЛЛЕЛИЗМА

Одни известные алгоритмы сортировки могут быть эффективнее других по какому-либо из основных критериев: число выполняемых операций, время выполнения элементарных операций, объем используемой памяти, степень параллелизма, функциональная регулярность связей в информационном графе алгоритма и т.д. При этом, имеется возможность выбрать такой алгоритм сортировки, который после выполнения операции редукции производительности вычислительной структуры, будет занимать минимум аппаратного ресурса. Выбор конкретного алгоритма напрямую зависит степени его распараллеливания, заданного временем обработки данных, коэффициента редукции и латентности вычислительной структуры, количества и разрядности сортируемых элементов. Алгоритмы сортировки являются информационно-эквивалентными, так как они выполняют одну и ту же математическую функцию. Однако каждый из алгоритмов рассматривается как автономный и независимый подход к решению задачи упорядочивания данных. Известно, что алгоритмам сортировки «пузырьком», «вставками» и «выбором» соответствует одна и та же сортирующая сеть, однако переход от одного алгоритма к другому до сих пор не описан в виде математических преобразований. Можно утверждать, что в настоящее время математический аппарат для описания различных алгоритмов сортировки и сортирующих сетей не формализован в полной мере, из-за чего не существует методологических основ перехода от одного алгоритма к другому. Иным способом описания алгоритма решения задачи является его представление в виде информационного графа, где выполняемые операции являются вершинами, которые объединены дугами, отражающими информационную зависимость между операциями. Преобразование информационного графа может приводить к получению иных информационно-эквивалентных алгоритмов. Пре-

имуществом подобного подхода к описанию алгоритмов является сравнительная простота используемого понятийного аппарата. В данной работе рассмотрены правила преобразования сортирующих сетей, на основе которых выполнен переход от одной сети к другой. Каждая из полученных сортирующих сетей может быть эффективна при разных коэффициентах распараллеливания и разном темпе обработки данных, от которых напрямую зависит коэффициент редукции производительности реализуемой вычислительной структуры. Автоматизация предложенных методов преобразования может позволить использовать разные алгоритмы сортировки, полученные из единого описания задачи в виде информационного графа, и зависящие от заданной скорости обработки данных.

Алгоритмы сортировки; сортирующие сети; информационный граф; ярусно-параллельная форма; преобразование алгоритмов.

I.I. Levin, K.N. Alekseev

TRANSFORMATION OF SORTING NETWORKS FOR DIFFERENT DEGREES OF PARALLELISM

Some well-known sorting algorithms may be more efficient than others according to any of the main criteria: the number of performed operations, the execution time of elementary operations, the used memory capacity, the parallelism degree, the functional regularity of connections in the information graph of algorithm, etc. At the same time, it is possible to choose such sorting algorithm, which will take up a minimum hardware resource after performance reducing operation of computational structure. The choice of a particular algorithm directly depends on it parallelization degree, the specified by the data processing time, the coefficient of reduction and latency of the computational structure and the number and bit depth of the sorted elements. Sorting algorithms are information-equivalent, since they perform the same mathematical function. However, each of algorithms is considered as an autonomous and independent approach to solving the data ordering problem. It is known that the same sorting network correspond to the "bubble", "inserts" and "selection" sorting algorithms. However, the transition from one algorithm to another has not yet been described in the form of mathematical transformations. It can be argued that the mathematical tools for describing various sorting algorithms and sorting networks is not fully formalized for today. Because of this, there is no methodological basis for the transition from one algorithm to another. Another method to describe the algorithm for solving the problem is its representation in the form of an information graph. In it, the performed operations are vertices that are combined by arcs reflecting the information dependence between operations. Transformation of the information graph can lead to obtaining other information and equivalent algorithms. The advantage of similar approach for the algorithm description of is the comparative simplicity of the used conceptual tools. In this paper, the transformation rules of sorting networks are considered, on the basis of which the transition from one network to another is performed. Each of the resulting sorting networks can be effective at different parallelization coefficients and data processing rates, on which the performance reduction coefficient of the implemented computational structure directly depends. Automation of the proposed transformation methods can allow the use the different sorting algorithms, derived from a single problem description in the form of an information graph, and depending on the specified data processing rate.

Sorting algorithms; sorting networks; information graph; tiered-parallel form; algorithm transformation .

Введение. Сортировка данных является одной из базовых операций, которую выполняют для облегчения последующей обработки информации при решении множества прикладных вычислительно-трудоемких задач, таких как: подавление шумов в изображении с помощью медианной фильтрации сигнала; кодирование и сжатие данных различными методами, например, методом Хаффмана; обработка данных сейсмической разведки; молекулярный докинг; анализ и обработка больших массивов данных, в том числе различные операции над матрицами и графами; и т.д.

Существует множество различных подходов к решению задачи сортировки данных, причем на основе каждого подхода построено большое количество алгоритмов [1–4]. К примеру, сортировка методом «пузырька», «вставок», «выбором»

и некоторые другие известные алгоритмы используют принцип попарного сравнения элементов сортируемого множества; быстрая сортировка, сортировка слиянием и сортирующие сети Батчера помимо сравнений используют принцип «разделяй и властвуй» для упорядочивания частей множества и их последующего слияния; сортировка подсчетом использует операцию сложения для нахождения одинаковых элементов множества, а также вместо сравнения чисел между собой выполняет их сравнение с полным набором вариантов их кодов; поразрядная сортировка предполагает выполнение последовательных сравнений каждого из разрядов кодов символов, и т.д.

На практике, различные алгоритмы сортировки целесообразно использовать в зависимости от требуемой степени параллелизма или скорости обработки данных, а также от параметров сортируемого множества [5]. Так в случае сортировки небольших объемов данных, порядка десятков мегабайт, на вычислительных системах традиционной архитектуры зачастую используется последовательный алгоритм быстрой сортировки. Однако при увеличении объема входных данных, который в некоторых задачах может достигать нескольких терабайт информации, время сортировки данным алгоритмом увеличивается нелинейно [6]. Для ускорения процесса сортировки больших объемов информации используют параллельные аппаратные реализации сортирующих сетей.

На сегодняшний день существует множество разнообразных сортирующих сетей [1, 7, 8], отличающихся количеством сортирующих элементов (числом сравнений), латентностью вычислительной структуры (временем получения результата на выходе конвейерной схемы), и регулярностью структуры (организация связей между сортируемыми элементами), причем эффективность каждой конкретной сети существенно зависит от требуемой скорости обработки. Так, при необходимости обеспечить наибольшую степень параллелизма, эффективной будет сеть с меньшим числом сортирующих элементов в схеме, а при возможности параллельной реализации лишь некоторой части сети – эффективно использовать сеть с функционально-регулярной структурой, где из-за наличия симметрии можно использовать меньшее число устройств коммутации [5, 9–11].

Так как использование различных алгоритмов сортировки приводит к получению упорядоченного множества, можно утверждать, что выполняемые ими математические функции эквивалентны. В тоже время все алгоритмы сортировки обычно рассматриваются как независимые и несвязанные между собой. Известно, что из алгоритма сортировки методом пузырька можно получить сортировку выбором, однако переход от одного алгоритма к другому не формализован в виде математических преобразований, а основан лишь на эмпирических наблюдениях и логике построения последовательных программ. Можно утверждать, что на сегодняшний день не существует методологических основ, позволяющих переходить от одного алгоритма к другому, в связи с чем нет возможности автоматизировать процесс получения наиболее эффективных алгоритмов в зависимости от критериев задачи.

Вместе с этим, существует иной подход к формализации и выведению параллельных алгоритмов, основанный на анализе и преобразовании информационных графов. В работах [5, 12-19] было рассмотрено преобразование графов, состоящих из ассоциативных и дистрибутивных алгебраических операций, которые приводили к получению графов, соответствующих другим известным алгоритмам решаемой задачи. Исследования показывают, что задача преобразования информационного графа обладает меньшей алгоритмической сложностью по сравнению последовательным выполнением математических преобразований, что может обеспечить возможность ее дальнейшую автоматизацию.

Для выполнения операций на информационных графах, необходимо сформулировать правила элементарных преобразований. В рамках данной статьи описаны методы преобразования сортирующих сетей, основанные на базовых преобразованиях порядка следования операций сравнения и их числа. Работоспособность методов проиллюстрирована на примере преобразования простейшей функционально-регулярной сортирующей сети, соответствующей последовательному алгоритму сортировки методом «пузырька» и состоящей из $n^2/2$ сортирующих элементов SE , и латентностью равной $2n - 3$. На примере преобразования данной сети показана принципиальная возможность получения иных известных сортирующих сетей, обладающих лучшими характеристиками для реализации при разном темпе поступления данных.

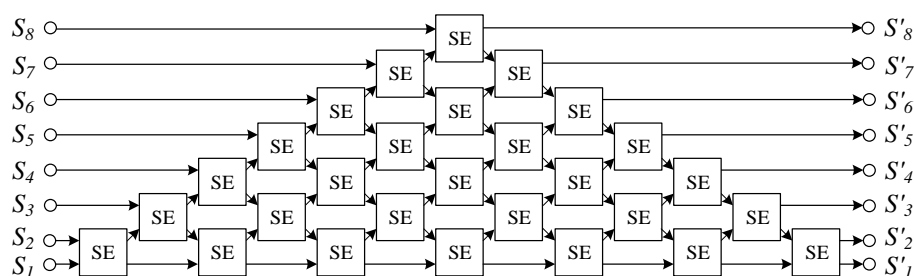


Рис. 1. Схема простейшей сортирующей сети при $n = 8$, последовательная реализация которой соответствует алгоритму сортировки «пузырьком»

Простейшие сортирующие сети. Пусть необходимо отсортировать (упорядочить) в порядке возрастания множество $A = \{A_1, A_2, A_3, \dots, A_n\}$, где n – число элементов. Для упорядочивания используются сортирующие элементы SE , структурная схема которых представлена на рисунке 2-а. Граф, полученный путем объединения множества сортирующих элементов SE согласно принципу «голова-хвост» (рисунок 2-б), является графом операции выделения максимума, где $X = X_n > \{X_1, X_2, X_3, \dots, X_{n-1}\}$ представляют результат обработки множества A : $X_n = \max(A)$. Нетрудно заметить, что из объединения подобных подграфов и состоит сортирующая сеть методом «пузырька», представленная на рис. 1. Здесь и далее данные подграфы будем называть «ступенями» сортирующей сети.

Исследования показали, что нахождение максимума является не единственной функцией, которую выполняют ступени графы, построенный путем объединения сортирующих элементов SE согласно принципу «голова-хвост». Положим, на вход ступени подается упорядоченное подмножество $\langle A_2, A_3, \dots, A_n \rangle$, в котором $n - 1$ элементов расположены в порядке возрастания, а также элемент A_1 , место которого не определено. В первом блоке SE выполняется сравнение элемента A_1 и элемента A_2 , который является минимальным в подмножестве $\langle A_2, A_3, \dots, A_n \rangle$. В случае если $A_1 \leq A_2$, на выход \min ступени X_1 будет передано значение A_1 , а на следующий блок сравнения SE будет передан элемент A_2 для сравнения с элементом A_3 . Очевидно, что дальнейшие сравнения не приведут к изменению порядка следования элементов подмножества $\langle A_2, A_3, \dots, A_n \rangle$. Так как $X_1 \leq \langle X_2, X_3, \dots, X_n \rangle$, можно утверждать, что множество X будет упорядоченным по возрастанию.

Если же $A_1 \geq A_2$, на выход \min ступени X_1 будет передано значение A_2 , а на следующий блок сравнения SE будет передан элемент A_1 для сравнения с элементом A_3 . Положим, что существует такой элемент A_i , который будет больше элемента A_1 , тогда справедливо: $\langle A_2, \dots, A_{i-1} \rangle \leq A_1 \leq \langle A_i, \dots, A_n \rangle$. Очевидно, что множество X , сформированное на выходе ступени, будет упорядочено по возрастанию.

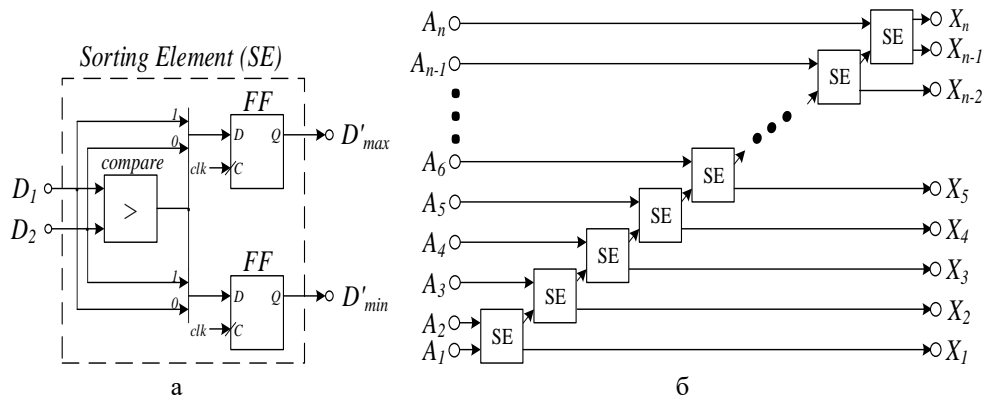


Рис. 2. а – структурная схема сортирующего элемента SE;
 б – граф, построенный путем объединения сортирующих элементов SE согласно принципу «голова-хвост»

Рассмотрим случай, когда на вход ступени подается неупорядоченное множество $\{A_1, \dots, A_k\}$ и упорядоченное подмножество $\langle A_{k+1}, \dots, A_n \rangle$, элементы которого расположены в порядке возрастания. Легко заметить, что на выходе max блока сравнения SE_{k-1} будет получен максимальный элемент неупорядоченного множества: $max(A_1, \dots, A_k)$. Все следующие блоки сравнения $SE_k \div SE_n$ можно рассматривать как ступень, на которую приходит упорядоченное подмножество $\langle A_{k+1}, \dots, A_n \rangle$ и элемент $max(A_1, \dots, A_k)$, место которого не определено. Очевидно, что результатом выполнения данной операции будет упорядоченное подмножество $\langle X_k, \dots, X_n \rangle$, при этом отношение элементов $\{X_1, \dots, X_{k-1}\}$ останется неопределенным.

Таким образом, можно утверждать, что ступень, сортирующие элементы в которой объединены согласно принципу «голова-хвост», помимо функции нахождения максимума также выполняет функцию вставки элемента A_1 в частично упорядоченное подмножество $\langle A_{k+1}, \dots, A_n \rangle$. В связи с этим, в зависимости от последовательности ступеней в сортирующей сети, при последовательном выполнении операций, она может соответствовать иным известным алгоритмам сортировки, таким как сортировка «вставками» и «выбором» (рис. 3,а,б соответственно).

Принципы объединения ступеней сети сортировки. Пусть граф, в котором ступень S_1 с меньшим числом операций следует за ступенью S_2 (рис. 4,а), описывает функцию F , а граф, в котором ступени S_1 и S_2 объединены в обратном порядке (рис. 4,б), описывает функцию F' . Докажем, что $F \Leftrightarrow F'$ при объединении ступеней S_1 и S_2 .

Очевидно, что элемент множества X_3 на рис. 4,а является максимальным элементом множества: $X_3 = max(A)$. Остальные два выхода ступени S_2 никак не относятся между собой и поступают на вход ступени S_1 . Элемент множества X_2 будет являться вторым максимальным элементом множества A . Так как после выделения двух максимумов остается только один элемент, то множество X будет полностью упорядоченным: $X_3 \geq X_2 \geq X_1$. Таким образом, ступени S_1 и S_2 объединенные в порядке уменьшения количества операций, выполняют функцию упорядочивания $F = X_3 \geq X_2 \geq X_1$.

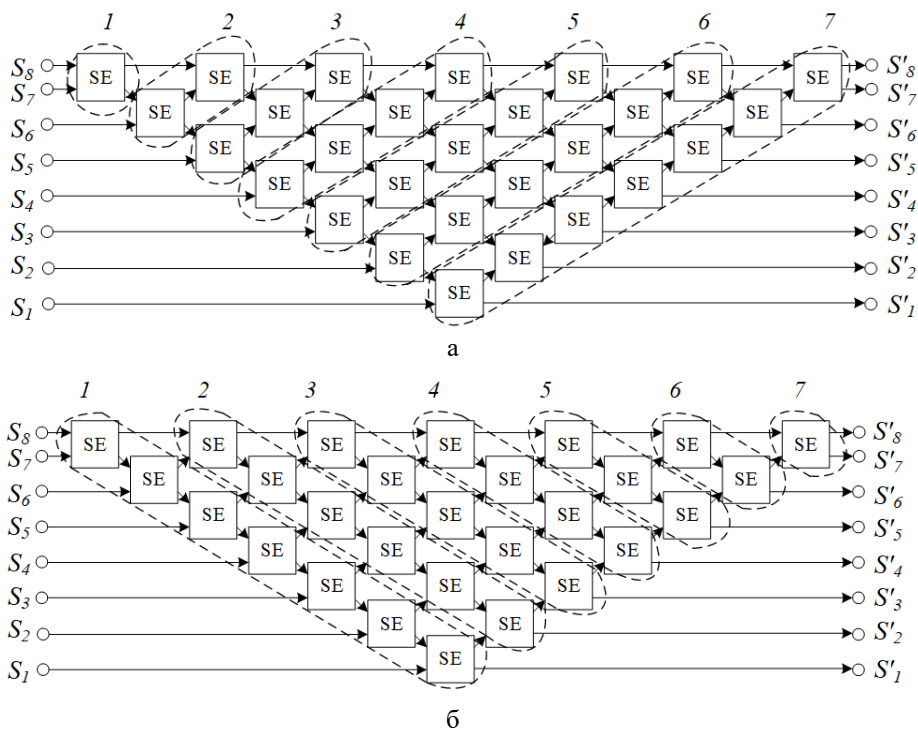


Рис. 3. Схемы сортирующих сетей, соответствующие:
а – сортировке вставками; б – сортировке выбором

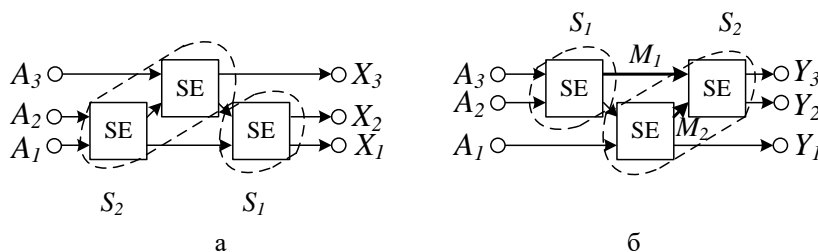


Рис. 4. Объединение ступеней S_1 и S_2 : а – в порядке уменьшения количества операций; б – в порядке увеличения количества операций

На рис. 4,б ступени S_1 и S_2 объединены в порядке увеличения количества операций. На выходе ступени S_1 будет сформировано упорядоченное подмножество, которое поступает на вход ступени S_2 . Также на вход ступени S_2 поступает элемент A_1 , место которого не определено. В этом случае на выходе ступени S_2 будет сформировано упорядоченное по возрастанию множество: $Y_3 \geq Y_2 \geq Y_1$. Можно сказать, что ступени S_1 и S_2 объединенные в порядке увеличения количества операций, выполняют функцию упорядочивания $F' = Y_3 \geq Y_2 \geq Y_1$.

Так как в результате обработки множества A и в первом и во втором варианте объединения ступеней сортирующей сети получены упорядоченные по убыванию множества, функции F и F' являются эквивалентными.

Положим, что множество $A = \{A_1, A_2, A_3, A_4\}$ является неупорядоченным. Сравним функции, выполняемую ступенями, объединенными в различном порядке и в которых на один элемент больше по сравнению с графами, приведенными на рис. 4. Докажем, что любом при подобном объединении ступеней функции F и F' являются эквивалентными.

На рис. 5,а ступени S_2 и S_3 объединены в порядке уменьшения количества операций. Выходы ступени S_3 представляют собой неупорядоченное множество $B = \{B_1, B_2, B_3\}$ и элемент $X_4 = \max(A)$. В свою очередь, выходы ступени S_2 будут представлять собой неупорядоченное множество $\{X_1, X_2\}$ и элемент $X_3 = \max(B)$. Довольно очевидно, что результирующая функция рассматриваемого объединения ступеней S_2 и S_3 будет иметь следующий вид: $F = X_4 \geq X_3 \geq \{X_1, X_2\}$.

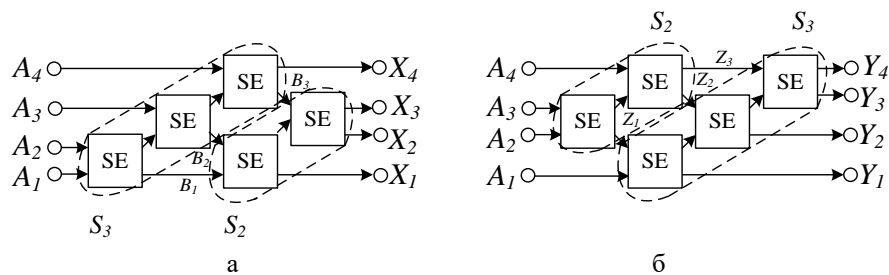


Рис. 5. Объединение ступеней S_2 и S_3 : а – в порядке уменьшения количества операций; б – в порядке увеличения количества операций

На рис. 5,б ступени S_2 и S_3 объединены в порядке увеличения количества операций. Выходы ступени S_2 представляют собой частично упорядоченное множество $Z_3 \geq \{Z_1, Z_2\}$, которое поступает в качестве входных данных на входы ступени S_3 . Также на вход ступени S_3 поступает элемент исходного множества A_1 . Тогда на вход последнего элемента SE ступени S_3 поступит два элемента: $Z_3 = \max(A_2, A_3, A_4)$ и $\max(A_1, Z_1, Z_2)$. Очевидно, что два старших выхода ступени S_3 имеют следующие отношения: $Y_4 = \max(A)$ и $Y_4 \geq Y_3$. В общем виде Y_3 можно представить как:

$$Y_3 = \min(\max(Z_2, Z_1, A_1), Z_3) = \min(\max(Z_2, Z_1, A_1), \max(A_2, A_3, A_4)),$$

или

$$Y_3 = \min\left(\max\left(\left(\{A_2, A_3, A_4\} - \max(A_2, A_3, A_4)\right), A_1\right), \max(A_2, A_3, A_4)\right).$$

Из приведенного отношения очевидно, что Y_3 является вторым по величине элементом множества A , тогда $F' = Y_4 \geq Y_3 \geq \{Y_1, Y_2\}$. В связи с этим можно утверждать, что ступени S_2 и S_3 , объединенные в различном порядке, выполняют эквивалентные преобразования: $F \Leftrightarrow F'$.

На рис. 6,а ступени S_{n-1} и S_n объединены в порядке уменьшения количества операций. Очевидно, что элемент множества X_{n+1} является максимальным элементом множества: $X_{n+1} = \max(A)$. Остальные выходы ступени S_n никак не относятся между собой и поступают на входы ступени S_{n-1} . Очевидно, что элемент множества X_n будет являться вторым максимальным элементом множества A . Тогда множество X будет частично упорядоченным, где функция упорядочивания $F = X_{n+1} \geq X_n \geq \{X_1, \dots, X_{n-1}\}$.

На рис. 6,б ступени S_{n-1} и S_n объединены в порядке увеличения количества операций. Так как на выходе ступени S_{n-1} будет найден максимальный элемент подмножества: $\max(A_2, \dots, A_{n+1})$, на вход ступени S_n поступит частично упорядоченное подмножество, а также элемент A_1 , место которого не определено. Отметим, что в качестве частично упорядоченного множества будет выступать один максимальный элемент, найденный в ступени S_{n-1} . Так как ступень S_n позволяет выделить максимальный элемент множества A , на вход Z_4 либо поступит элемент A , либо второй по величине максимум из подмножества $\{A_2, \dots, A_{n+1}\}$. В любом из этих случаев на выходе ступени S_n будет сформировано частично упорядоченное множество Y , где функция упорядочивания: $F' = Y_{n+1} \geq Y_n \geq \{Y_1, \dots, Y_{n-1}\}$. Очевидно, что функции F и F' , эквивалентны.

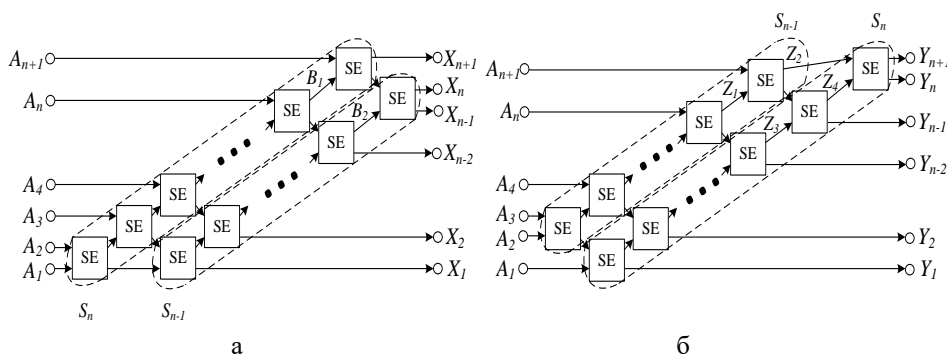


Рис. 6. Объединение ступеней S_{n-1} и S_n : а – в порядке уменьшения количества операций; б – в порядке увеличения количества операций

Таким образом доказано, что выражение $F \Leftrightarrow F'$ будет верно для ступеней, состоящих из любого числа сортирующих элементов SE при объединении ступеней S_{n-1} и S_n , и не зависит от порядка их следования. Основываясь данным свойстве, докажем, что функция, выполняемая двумя ступенями, не зависит не только от порядка их следования, но и от количества операций в них.

На рис. 7 показано два варианта последовательного объединения ступеней S_n и S_m , где $n > m$, в которых операции объединены по принципу «голова-хвост». На рис. 7,а ступени S_n и S_m объединены в порядке уменьшения количества операций. Очевидно, что элемент множества X_n является максимальным элементом множества A : $X_n = \max(A)$. Остальные выходы ступени S_n никак не относятся между собой и коммутируются ко входам ступени S_m . Для элемента множества X_{n-1} , который будет являться максимальным элементом части множества A , справедливо:

$$X_{n-1} = \max(A - \max(A)); X_{n-1} \geq \{X_{n-2}, X_{n-3}, \dots, X_{n-m+1}\}.$$

Множество X будет являться частично упорядоченным, где выделен максимальный элемент X_n и выделено подмножество $\{X_{n-2}, X_{n-3}, \dots, X_{n-m+1}\}$, для которого определен максимальный элемент X_{n-1} , причем отношения элементов $X_{n-1} > \{X_{n-2}, X_{n-3}, \dots, X_{n-m+1}\}$ и $\{X_{n-m}, \dots, X_1\}$ не определены между собой. Так, функцию F , которую выполняют ступени S_n и S_m , объединенные, как показано на рис. 5,а, можно описать как:

$$F = X_n > \{\{X_{n-1} > \{X_{n-2}, X_{n-3}, \dots, X_{n-m+1}\}\}, X_{n-m}, \dots, X_1\}.$$

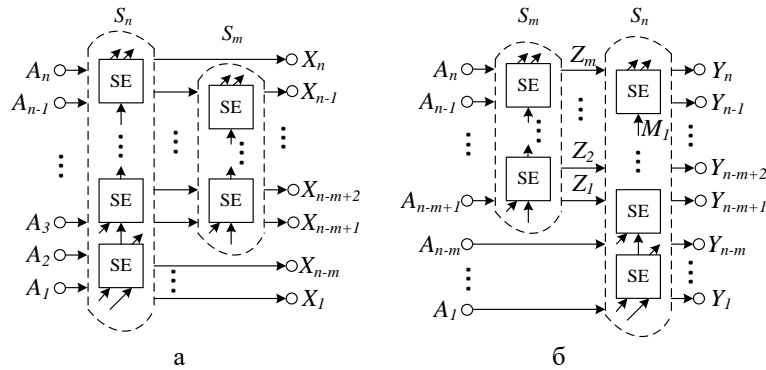


Рис. 7. Объединение ступеней S_n и S_m : а – в порядке уменьшения количества операций; б – в порядке увеличения количества операций

На рис. 7,б ступени S_n и S_m объединены в порядке увеличения количества операций. На вход ступени S_m поступают элементы исходного множества $\{A_{n-m+1}, \dots, A_n\}$, а на выходе формируется частично отсортированное множество $Z = \{Z_1, Z_2, \dots, Z_m\}$, для которого справедливо:

$$\begin{aligned} Z_m &= \max(A_{n-m+1}, \dots, A_n); \\ Z_m &> \{Z_1, Z_2, \dots, Z_{m-1}\}; \\ Z_m &> \{A_{n-m+1}, \dots, A_n\} - \max(A_{n-m+1}, \dots, A_n). \end{aligned}$$

Оставшаяся часть исходного множества $\{A_1, \dots, A_{n-m}\}$, а также частично отсортированное множество Z , поступают на вход ступени S_n . Для максимального элемента ступени S_n справедливо: $Y_n = \max(A)$, где:

$$Y_n = \max(A_1, A_2, \dots, A_{n-m}, Z_m).$$

Так как ступень S_n позволяет выделить максимальный элемент множества A , на вход M_1 либо поступит элемент $\max(A_1, A_2, \dots, A_{n-m})$, либо второй по величине максимум из подмножества $\{A_{n-m+1}, \dots, A_n\}$. В связи с этим, на выходе ступени S_n будет сформировано частично упорядоченное множество Y , со следующей функцией упорядочивания:

$$F' = Y_n > \{Y_{n-1} > \{Y_{n-2}, Y_{n-3}, \dots, Y_{n-m+1}\}, Y_{n-m}, \dots, Y_1\}.$$

Очевидно, что функции F и F' , эквивалентны. Таким образом, доказано, что результат упорядочивания входного множества A будет одинаков для обоих вариантов расположения ступеней относительно друг друга, операции в которых объединены согласно принципу «голова-хвост» и не зависит как от порядка их следования, так и от числа используемых сортирующих элементов.

В то же время стоит отметить, что для выполнения отношения $F \Leftrightarrow F'$ необходимо, чтобы ступени были объединены корректно: при порядке следования от большей ступени к меньшей, после большей ступени необходимо подавать на меньшую старшие выходы, кроме максимального элемента, тогда как при обратном порядке следования ступеней требуется подавать все выходы меньшей ступени к старшим входам большей ступени.

Построение и преобразование сортирующих сетей. Сортирующая сеть, соответствующая последовательным алгоритмам сортировки «пузырьком», «вставками» и «выбором» (рисунок 3) имеет различное количество сортирующих эле-

ментов в ступенях, из-за чего когда коэффициент редукции [20] находится в интервале $1 < r^0 < n - 1$ при редукции по операциям в итерациях, аппаратный ресурс, занимаемый вычислительной структурой задачи, будет уменьшаться нелинейно [5, 14]. Наиболее эффективной будет являться предельная редукция сортирующей сети по операциям в ступенях ($r^0 = n - 1$), как это показано на рис. 8: остается $n - 1$ изоморфных итераций, состоящих из одного сортирующего элемента.

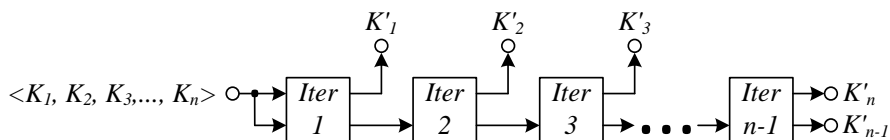


Рис. 8. Вычислительная структура сортирующей сети, соответствующая последовательным алгоритмам сортировки «пузырьком», «вставками» и «выбором», после выполнения редукции по количеству операций

Основываясь на принципах объединения ступеней нетрудно показать, что порядок их следования в структуре сортирующей сети может быть любым при корректной коммутации ступеней между собой. Так, иные архитектуры, представленные на рис. 9, также являются сортирующими сетями с одинаковым числом сортирующих элементов, а их разбиение на ступени соответствует различным известным последовательным алгоритмам сортировки.

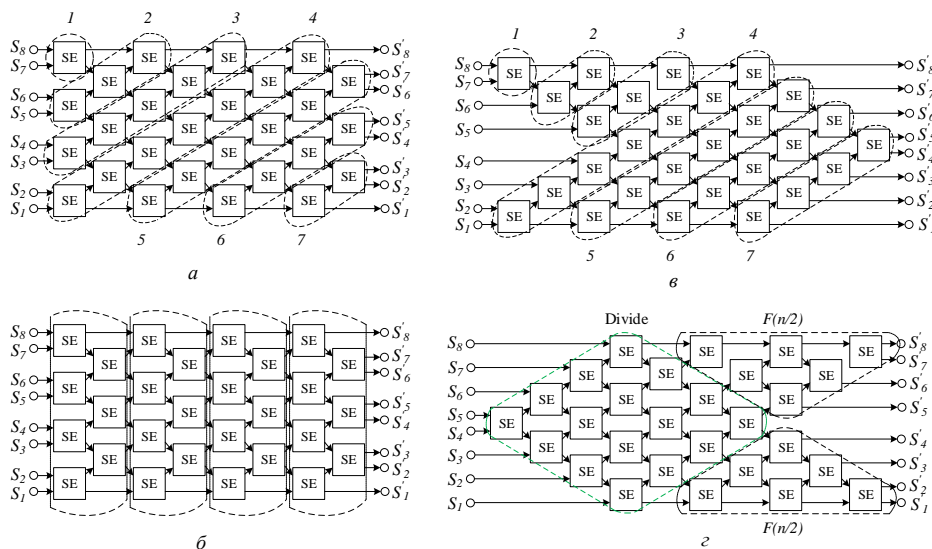


Рис. 9. Схемы иных простейших сортирующих сетей, соответствующие: а – смешанной сортировке «пузырьком», вставками и слиянием; б – нечетно-четной сортировке; в – «шейкерной» сортировке; г – «ведерной» сортировке

Нечетно-четная сортирующая сеть (рис. 9,а), построенная путем изменения порядка следования ступеней, имеет четкое разбиение на итерации, содержащие одинаковое количество сортирующих элементов (рис. 9,б). Когда коэффициент редукции $r^0 < n/2$, количество итераций нечетно-четной сортирующей сети со-

кращается в r^0 раз. Так как граф нечетно-четной сортирующей сети имеет регулярную структуру, при выполнении редукции по итерациям в данном случае полностью сохраняется межитерационная коммутация и возникает обратная связь по данным: обеспечивается коммутация входов первой итерации с выходами последней, а при редукции производительности будет наблюдаться линейное сокращение используемого аппаратного ресурса.

Если коэффициент редукции $r^0 = n/2$, нечетно-четную сортирующую сеть необходимо сократить до одной итерации. В этом случае для организации обратной связи необходимо обеспечить коммутацию входов единственной итерации с ее же выходами.

Все ступени сортирующих сетей, соответствующих алгоритмам «шейкерной» и «ведерной» сортировки (рис. 8, в, г), содержат различное количество сортирующих элементов, зависящее от количества обрабатываемых данных, поэтому при выполнении редукции по операциям в итерациях прослеживается нелинейная зависимость аппаратного ресурса от коэффициента редукции. В данном случае наиболее эффективной является предельная редукция сортирующей сети по числу операций в итерациях при $r^0 = n - 1$, приводящая к тому, что остается $n/2$ изоморфных итераций, состоящих из двух сортирующих элементов.

Эффективность сортирующих сетей. Эффективность применения одной из множества существующих сетей зависит от заданного темпа обработки данных, коэффициента редукции r^0 , латентности алгоритма, количества элементов в массиве n [5].

При сортировке множества массивов данных, элементы которых поступают на обработку параллельно ($r^0 = 1$), можно использовать любую из представленных сортирующих сетей, так как все они содержат одинаковое число сортирующих элементов. Однако, наиболее эффективным является использование топологии нечетно-четной сортирующей сети, которая не требует дополнительной синхронизации потоков входных и выходных данных.

Когда коэффициент редукции вычислительной структуры лежит в пределах $1 < r^0 \leq n$, наиболее эффективной также является нечетно-четная сортирующая сеть, так как позволяет обеспечить меньшее число коммутирующих элементов между итерациями.

При сортировке множества массивов данных, элементы которых поступают на обработку последовательно ($r^0 \geq n$), рационально использовать либо сортирующую сеть, соответствующую алгоритмам сортировки «пузырьком», «вставками» и «выбором», либо сети, соответствующие алгоритмам «шейкером» или «ведерной» сортировки, так как они используют значительно меньше каналов данных по сравнению с нечетно-четной сортирующей сетью. Отметим, что выбор конкретной топологии зависит от доступного аппаратного ресурса: так, если возможно организовать два входных канала данных, более эффективной является сеть с наименьшей латентностью вычислительной структуры: соответствующая алгоритмам «шейкером» или «ведерной» сортировки. Если же доступен только один канал данных, наиболее эффективной является конвейерная реализация сортирующей сети, соответствующей алгоритмам сортировки «пузырьком», «вставками» и «выбором».

Заключение. В рамках данной работы рассмотрены принципы объединения ступеней сети сортировки и способы выбора конкретной сети в зависимости от темпа поступления данных.

Отметим, что помимо рассмотренных функций, ступень сортирующей сети, сортирующие элементы в которой объединены по принципу «голова-хвост», дополнительно выполняет перенос нескольких больших элементов ближе к старшим выходам ступени. Так, в результате упорядочивания множества $A = \{7, 4, 3, 8, 6, 2, 9, 3\}$ ступенью

S_7 , состоящей из семи SE , будет получено множество $X = 9 \geq \{4,3,7,6,2,8,3\}$. Из представленных данных видно, что помимо найденного максимального элемента 9, значения 7 и 8 передвинулись на 2 позиции ближе к старшим выходам ступени.

Так как основным принципом работы каждой ступени в простейших сортирующих сетях, соответствующих последовательным алгоритмам сортировки «пузырьком» или «выбором», является именно нахождение максимума в неотсортированном множестве элементов, перемещение не максимальных элементов очевидно не имеет никакой необходимости. Также очевидно, что для сортировки элементов массива достаточно единожды переместить каждый элемент на новое место, для чего необходимо выполнить всего n операций перемещения элементов, что используется, например, в последовательном алгоритме сортировки «выбором». Исходя из вышесказанного, можно сделать вывод, что сортирующие сети, построенные путем объединения ступеней согласно принципу «голова-хвост», обладают избыточностью выполняемых операций перемещения элементов.

В следующих работах будет показано, что на основе принципов построения сортирующих сетей, а также некоторых иных базовых преобразований, можно устранить избыточность выполняемых операций и получить более эффективную нечетно-четную сеть Батчера, состоящую из значительно меньшего числа операций и имеющую функционально-регулярную структуру [1, 21]. На основе формализации принципов построения и преобразования сортирующих сетей имеется возможность разработать методику их преобразования, позволяющую оптимизировать число операционных вершин и латентность вычислительной структуры.

Потенциально, автоматизация процесса преобразования сортирующих сетей позволит увеличить эффективность используемого оборудования без больших временных затрат на разработку и отладку кода параллельной программы при изменении темпа поступления и обработки данных.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кнут Д.Э. Искусство программирования. Т. 3. Сортировка и поиск. – 2-е изд.: пер. с англ. – М.: Издательский дом «Вильямс», 2001.
2. Дасгутта С., Пападимитриу Х., Вазирани У. Алгоритмы: пер. с англ. / под ред. А. Шеня. – М.: МЦНМО, 2014. – 320 с.
3. Кормен Томас Х. и др. Алгоритмы: построение и анализ. – 3-е изд.: пер. с англ. – М.: ООО «И.Д. Вильямс», 2013. – 1328 с.
4. Скиена С.С. Алгоритмы. Руководство по разработке. – 3-е изд.: пер. с англ. – СПб.: БХВ-Петербург, 2022. – 848 с.
5. Алексеев К.Н. Методы и средства создания параллельно-конвейерных программ для решения задач реального времени на реконфигурируемых вычислительных системах: дисс. ... канд. техн. наук, по специальности 05.13.11 “Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей”, научный руководитель: д.т.н., проф. Левин И.И. – Таганрог, 2020. – 213 с.
6. Скворцов С.В., Пюрова Т.А. Параллельные алгоритмы сортировки данных и их реализация на платформе CUDA // Вестник РГРТУ. – 2016. – № 58. – С. 42-48.
7. Сортировочные сети с особыми свойствами. – Режим доступа: https://neerc.ifmo.ru/wiki/index.php?title=%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BE%D1%87%D0%BD%D1%8B%D0%B5_%D1%81%D0%B5%D1%82%D0%B8_%D1%81_%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%BC%D0%B8_%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0%D0%BC%D0%B8 (дата обращения: 01.11.2023).
8. Сортирующие сети для квадратичных сортировок. – Режим доступа: https://neerc.ifmo.ru/wiki/index.php?title=%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D1%83%D1%8E%D1%89%D0%B8%D0%B5_%D1%81%D0%B5%D1%82%D0%B8_%D0%B4%D0%BB%D1%8F_%D0%BA%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%B8%D1%87%D0%BD%D1%8B%D1%85_%D1%81%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BE%D0%BA (дата обращения: 01.11.2023).

9. *Алексеев К.Н., Левин И.И., Сорокин Д.А.* Структурно-процедурная реализация на реконфигурируемых вычислительных системах кодирования Хаффмана в реальном масштабе времени // Вестник компьютерных и информационных технологий. – 2018. – № 9. – С. 3-10. – ISSN 1810-7206. – DOI: 10.14489/vkit.2018.09.pp.003-010. – <http://www.vkit.ru/index.php/current-issue-rus/751-003-010>.
10. *Алексеев К.Н., Сорокин Д.А., Семерникова Е.Е.* Структурно-процедурная реализация кодирования алгоритмом Хаффмана в задачах управления // Матер. 10-й Всероссийской мультikonференции по проблемам управления (МКПУ-2017), 11-16 сентября 2017 г. – Ростов-на-Дону: Изд-во ЮФУ, 2017. – Т. 3. – С. 126-127.
11. *Dudnikov E.A., Alekseev K.N., Sorokin D.A.* High-Speed Huffman encoding of dense data flows on FPGA // German International Journal of Modern Science / Deutsche internationale Zeitschrift für zeitgenössische Wissenschaft. – 2021. – No. 9, Vol. 1. – P. 36-40. – <https://dizwz.com/wp-content/uploads/2021/05/Deutsche-internationale-Zeitschrift-für-zeitgenössische-Wissenschaft-№9-part-1-2021-37-41.pdf>.
12. *Каляев А.В., Левин И.И.* Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. – М.: Янус-К, 2003. – 380 с.
13. *Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И.* Реконфигурируемые мультikonвейерные вычислительные структуры. – 2-е изд., перераб. и доп. / под общ. ред. И.А. Каляева. – Ростов-на-Дону: Изд-во ЮНЦ РАН, 2009. – 344 с. – ISBN 978-5-902982-61-6.
14. *Писаренко И.В., Алексеев К.Н., Мельников А.К.* Ресурснезависимое представление сортирующих сетей на языке программирования Set@1 // Вестник компьютерных и информационных технологий. – 2019. – № 11 (185). – С. 53-60. – DOI: 10.14489/vkit.2019.11.pp.053-060. – <http://vkit.ru/index.php/current-issue-rus/857-053-060>.
15. *Писаренко И.В., Касаркин А.В., Алексеев К.Н.* Рекурсивное описание графов с ассоциативными операциями на языке программирования Set@1 // Всероссийская научная конференция "Суперкомпьютерные технологии (СКТ) 2020", 05-10 октября 2020 г.: Крым, Алушта, Россия. – Ростов-на-Дону; Таганрог: Изд-во ЮФУ, 2020. – С. 79-83.
16. *Михайлов Д.В.* Методы создания параллельно-конвейерных программ для задач с последовательным информационным графом: дисс. ... канд. техн. наук, по специальности 05.13.11 "Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей", научный руководитель: д.т.н., проф. Левин И.И. – Таганрог, 2022. – 213 с.
17. *Михайлов Д.В., Левин И.И., Дордопуло А.И., Писаренко И.В.* Представление графов с ассоциативными операциями на языке программирования SET@L // Известия ЮФУ. Технические науки. – 2020. – № 3 (213). – С. 98-109. – DOI: 10.18522/2311-3103-2020-3-98-111.
18. *Михайлов Д.В.* Преобразование некоторых видов последовательных информационных графов в параллельно-конвейерную форму // Известия ЮФУ. Технические науки. – 2020. – № 7 (217). – С. 78-93. – DOI: 10.18522/2311-3103-2020-7-78-93.
19. *Михайлов Д.В.* Преобразование последовательного информационного графа метода прогонки в параллельную форму // Известия ЮФУ. Технические науки. – 2021. – № 7 (224). – С. 177-188. – DOI: 10.18522/2311-3103-2021-7-177-188.
20. *Сорокин Д.А., Дордопуло А.И.* Методика сокращения аппаратных затрат в сложных системах при решении задач с существенно-переменной интенсивностью потоков данных // Известия ЮФУ. Технические науки. – 2012. – № 4. – С. 213-219.
21. *Batcher K.E.* Sorting Networks and their Applications // Proc. AFIPS Spring Joint Comput. Conf. – 1968. – Vol. 32. – P. 307314.

REFERENCES

1. *Knut D.E.* Искусство программирования. Т. 3. Сортировка и поиск [The art of programming. Volume 3. Sorting and searching]. 2nd ed.: transl. from engl. Moscow: Izdatel'skiy dom «Vil'yams», 2001.
2. *Dasgupta S., Papadimitriou Kh., Vazirani U.* Algoritmy [Algorithms]: trans. from engl., ed. by A. Shenyu. Moscow: MTSNMO, 2014, 320 p.
3. *Kormen Tomas Kh. i dr.* Algoritmy: postroenie i analiz [Algorithms: construction and analysis]. 3rd ed.: transl. from engl. Moscow: OOO «I.D. Vil'yams», 2013, 1328 p.
4. *Skienna S.S.* Algoritmy. Rukovodstvo po razrabotke [Algorithms. Development Guide]. 3rd ed.: transl. from engl. Saint Petersburg: BKhV-Peterburg, 2022, 848 p.

5. *Alekseev K.N.* Metody i sredstva sozdaniya parallel'no-konveyernykh programm dlya resheniya zadach real'nogo vremeni na rekonfiguriruemyykh vychislitel'nykh sistemakh: diss. ... kand. tekhn. nauk, po spetsial'nosti 05.13.11 "Matematicheskoe i programmnoe obespechenie vychislitel'nykh mashin, kompleksov i komp'yuternykh setey", nauchnyy rukovoditel': d.t.n., prof. Levin I.I. [Methods and means for creating parallel-pipeline programs for solving real-time problems on reconfigurable computing systems: cand. of eng. sc. diss., specialty 05.13.11 "Mathematical and software support for computers, complexes and computer networks", scientific supervisor: dr. of eng. sc., professor Levin I.I.]. Taganrog, 2020, 213 p.
6. *Skvortsov S.V., Pyurova T.A.* Parallelnye algoritmy sortirovki dannykh i ikh realizatsiya na platforme CUDA [Parallel data sorting algorithms and their implementation on the CUDA platform], *Vestnik RGRU* [Vestnik of Ryazan State Radio Engineering University], 2016, No. 58, pp. 42-48.
7. Sortirovochnye seti s osobymi svoystvami [Sorting networks with special properties]. Available at: https://neerc.ifmo.ru/wiki/index.php?title=%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BE%D1%87%D0%BD%D1%8B%D0%B5_%D1%81%D0%B5%D1%82%D0%B8_%D1%81_%D0%BE%D1%81%D0%BE%D0%B1%D1%8B%D0%BC%D0%B8_%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0%D0%BC%D0%B8 (accessed 01 November 2023).
8. Sortiruyushchie seti dlya kvadrachnykh sortirovok [Sorting networks for quadratic sorts]. Available at: https://neerc.ifmo.ru/wiki/index.php?title=%D0%A1%D0%BE%D1%80%D1%82%D0%B8%D1%80%D1%83%D1%8E%D1%89%D0%B8%D0%B5_%D1%81%D0%B5%D1%82%D0%B8_%D0%B4%D0%BB%D1%8F_%D0%BA%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%B8%D1%87%D0%BD%D1%8B%D1%85_%D1%81%D0%BE%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%BE%D0%BA (accessed 01 November 2023).
9. *Alekseev K.N., Levin I.I., Sorokin D.A.* Strukturno-protsedurnaya realizatsiya na rekonfiguriruemyykh vychislitel'nykh sistemakh kodirovaniya KHaffmana v real'nom masshtabe vremeni [Structural and procedural implementation on reconfigurable computing systems of Huffman coding in real time], *Vestnik komp'yuternykh i informatsionnykh tekhnologiy* [Bulletin of computer and information technologies], 2018, No. 9, pp. 3-10. ISSN 1810-7206. DOI: 10.14489/vkit.2018.09.pp.003-010. Available at: <http://www.vkit.ru/index.php/current-issue-rus/751-003-010>.
10. *Alekseev K.N., Sorokin D.A., Semernikova E.E.* Strukturno-protsedurnaya realizatsiya kodirovaniya algoritmom Khaffmana v zadachakh upravleniya [Structural and procedural implementation of coding by the Huffman algorithm in control problems], *Mater. 10-y Vserossiyskoy mul'tikonferentsii po problemam upravleniya (MKPU-2017), 11-16 sentyabrya 2017 g.* [Proceedings of the 10th All-Russian Multi-Conference on Control Problems (MCPU-2017), September 11-16, 2017]. Rostov-on-Don: Izd-vo YuFU, 2017, Vol. 3, pp. 126-127.
11. *Dudnikov E.A., Alekseev K.N., Sorokin D.A.* High-Speed Huffman encoding of dense data flows on FPGA, *German International Journal of Modern Science / Deutsche internationale Zeitschrift für zeitgenössische Wissenschaft*, 2021, No. 9, Vol. 1, pp. 36-40. Available at: <https://dizww.com/wp-content/uploads/2021/05/Deutsche-internationale-Zeitschrift-für-zeitgenössische-Wissenschaft-№9-part-1-2021-37-41.pdf>.
12. *Kalyaev A.V., Levin I.I.* Modul'no-narashchivaemye mnogoprotsessornyye sistemy so strukturno-protsedurnoy organizatsiyey vychisleniy [Modularly scalable multiprocessor systems with structural and procedural organization of calculations]. Moscow: Yanus-K, 2003, 380 p.
13. *Kalyaev I.A., Levin I.I., Semernikov E.A., Shmoylov V.I.* Rekonfiguriruemyye mul'tikonveyernyye vychislitel'nye struktury [Reconfigurable multi-pipeline computing structures]. 2nd ed., under general ed. I.A. Kalyaeva. Rostov-on-Don: Izd-vo YuNTS RAN, 2009, 344 s. ISBN 978-5-902982-61-6.
14. *Pisarenko I.V., Alekseev K.N., Mel'nikov A.K.* Resursonezavisimoe predstavlenie sortiruyushchikh setey na yazyke programmirovaniya Set@l [Resource-independent representation of sorting networks in the Set@l programming language], *Vestnik komp'yuternykh i informatsionnykh tekhnologiy* [Bulletin of computer and information technologies], 2019, No. 11 (185), pp. 53-60. DOI: 10.14489/vkit.2019.11.pp.053-060. Available at: <http://vkit.ru/index.php/current-issue-rus/857-053-060>.

15. *Pisarenko I.V., Kasarkin A.V., Alekseev K.N.* Rekursivnoe opisanie grafov s assotsiativnymi operatsiyami na yazyke programmirovaniya Set@l [Recursive description of graphs with associative operations in the Set@l programming language], *Vserossiyskaya nauchnaya konferentsiya "Superkomp'yuternye tekhnologii (SKT) 2020", 05-10 oktyabrya 2020 g.: Krym, Alushta, Rossiya* [All-Russian scientific conference "Supercomputer Technologies (SCT) 2020", October 05-10, 2020: Crimea, Alushta, Russia]. Rostov-on-Don; Taganrog: Izd-vo YuFU, 2020, pp. 79-83.
16. *Mikhaylov D.V.* Metody sozdaniya parallel'no-konveyernykh programm dlya zadach s posledovatel'nyim informatsionnym grafom: diss. ... kand. tekhn. nauk, po spetsial'nosti 05.13.11 "Matematicheskoe i programmnoe obespechenie vychislitel'nykh mashin, kompleksov i komp'yuternykh setey", nauchnyy rukovoditel': d.t.n., prof. Levin I.I. [Methods for creating parallel-pipeline programs for problems with a sequential information graph: cand. of eng. sc. diss., specialty 05.13.11 "Mathematical and software support for computers, complexes and computer networks", scientific supervisor: dr. of eng. sc., professor Levin I.I.]. Taganrog, 2022, 213 p.
17. *Mikhaylov D.V., Levin I.I., Dordopulo A.I., Pisarenko I.V.* Predstavlenie grafov s assotsiativnymi operatsiyami na yazyke programmirovaniya SET@L [Representation of graphs with associative operations in the SET@L programming language], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2020, No. 3 (213), pp. 98-109. DOI: 10.18522/2311-3103-2020-3-98-111.
18. *Mikhaylov D.V.* Preobrazovanie nekotorykh vidov posledovatel'nykh informatsionnykh grafov v parallel'no-konveyernuyu formu [Transformation of some types of sequential information graphs into a parallel-pipeline form], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2020, No. 7 (217), pp. 78-93. DOI: 10.18522/2311-3103-2020-7-78-93.
19. *Mikhaylov D.V.* Preobrazovanie posledovatel'nogo informatsionnogo grafa metoda progonki v parallel'nyuyu formu [Transformation of a sequential information graph of the sweep method into parallel form], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2021, No. 7 (224), pp. 177-188. DOI: 10.18522/2311-3103-2021-7-177-188.
20. *Sorokin D.A., Dordopulo A.I.* Metodika sokrashcheniya apparatnykh zatrat v slozhnykh sistemakh pri reshenii zadach s sushchestvenno-peremennoy intensivnost'yu potokov dannykh [Methodology for reducing hardware costs in complex systems when solving problems with significantly variable intensity of data flows], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 4, pp. 213-219.
21. *Batcher K.E.* Sorting Networks and their Applications, *Proc. AFIPS Spring Joint Comput. Conf.*, 1968, Vol. 32, pp. 307314.

Статью рекомендовал к опубликованию д.т.н., профессор А.В. Боженюк.

Левин Илья Израилевич – Южный федеральный университет; e-mail: iilevin@sfedu.ru; г. Таганрог, Россия; тел.: 88634612111; зав. кафедрой; д.т.н.; профессор.

Алексеев Кирилл Николаевич – e-mail: alexseev91@mail.ru; тел.: 9283536268; к.т.н.; доцент.

Levin Ilya Izrailevich – Southern Federal University; e-mail: iilevin@sfedu.ru; Taganrog, Russia; phone: 88634612111; head of department; dr. of eng. sc.; professor.

Alekseev Kirill Nikolayevich – e-mail: alexseev91@mail.ru; phone: +79283536268; cand. of eng. sc.; professor.