https://doi.org/10.31891/2219-9365-2023-74-25
УДК 004.94.2

МАРТИНЮК Тетяна
Вінницький національний технічний університет
https://orcid.org/0000-0001-9952-9438
e-mail: martyniuk.t.b@gmail.com
КРУКІВСЬКИЙ Богдан
Вінницький національний технічний університет
https://orcid.org/0000-0003-0788-3259
e-mail: smiletex11@gmail.com
КУПЕРШТЕЙН Леонід
Вінницький національний технічний університет
https://orcid.org/0000-0001-6737-7134
e-mail: kupershtein.lm@gmail.com
КРЕНЦІН Михайло
Вінницький національний технічний університет
https://orcid.org/0000-0002-1792-9401
e-mail: mishatron98@gmail.com

ОСОБЛИВОСТІ ПРЕДСТАВЛЕННЯ МЕРЕЖНОГО АЛГОРИТМУ СОРТУВАННЯ З РАНЖУВАННЯМ

Вдосконалення відомих алгоритмів сортування та розроблення нових підходів до сортування пов'язано, в першу чергу, з їх широким застосуванням у найпоширеніших у теперішній час прикладних областях. Це стосується, наприклад, пошукових систем, систем управління БД, нейромережних та експертних технологій, попереднього оброблення сигналів і зображень. Поряд з розвинутими програмними засобами сортування масивів даних певний інтерес представляють апаратні реалізаційні моделі сортування, як однієї з найбільш розповсюджених асоціативно-логічних операцій. Особливо це стосується паралельних методів сортування, до яких належать варіанти їх мережного подання. У статті розглянуто особливості мережного алгоритму сортування лінійного числового масиву на базі відомого методу попарного обміну. Особливістю запропонованого підходу є використання сформованих рангів відповідних елементів масиву в процесі їх сортування. В результаті поступове перетворення (підлаштування) рангів елементів масиву дозволяє відмовитись від необхідності виконання складної процедури перекомутації самих елементів у сформованих парах. Ця трудомістка операція замінюється швидкісними операціями інкремента/декремента над відповідними рангами. Для порівняння у вигляді таблиці показано приклад по циклах двох процесів сортування: за класичним мережним методом попарного обміну і запропонованим підходом з формуванням відповідних рангів. Наведено класичний варіант покрокового опису алгоритму мережного сортування з ранжуванням. Для порівняння представлено опис цього алгоритму у термінах системи алгоритмічних алгебр (САА) Глушкова. Такий підхід свідчить про компактність подання запропонованого алгоритму, а також дозволяє показати значний рівень паралелізму оброблення, притаманний мережним алгоритмам сортування.

Ключові слова: сортування, мережний алгоритм, ранг, система алгоритмічних алгебр.

MARTYNIUK Tatiana, KRUKIVSKYI Bohdan,
KUPERSHTEIN Leonid, KRENTSIN Mykhailo
Vinnytsia National Technical University

PRESENTATION PECULIARITIES OF THE NETWORK SORTING ALGORITHM WITH RANKING

The improvement of known sorting algorithms and the development of new approaches to sorting is primarily due to their widespread use in the most common application areas today. This affects, for example, to search engines, DB control systems, neural network and expert technologies, pre-processing of signals and images. Along with developed software tools for sorting data arrays, hardware implementation models of sorting are of some interest, as one of the most widespread associative-logical operations. This especially applies to parallel sorting methods, which include variants of their network representation. The article considers the peculiarities of the network algorithm for sorting a linear numerical array based on the well-known method of pair exchange. A feature of the proposed approach is the use of formed ranks of the array corresponding elements in the process of sorting them. As a result, the gradual transformation (tuning) of the ranks of the array elements allows you to abandon the need to perform a complex procedure of commutation of the elements themselves in the formed pairs. This time-consuming operation is replaced by high-speed increment/decrement operations on the corresponding ranks. For comparison, an example of the cycles of two sorting processes is shown in the form of a table: according to the classic network method of pair exchange and the proposed approach with the formation of the corresponding ranks. A classic version of the step-by-step description of the network sorting algorithm with ranking is presented. For comparison, a description of this algorithm in terms of the system of algorithmic algebras (SAA) Glushkov is presented. This approach shows the compact presentation of the proposed algorithm, and also allows showing a significant level of processing parallelism inherent in network sorting algorithms.

Keywords: sorting, network algorithm, rank, system of algorithmic algebras.

Постановка проблеми у загальному вигляді та її зв'язок з важливими науковими чи практичними завданнями

Серед відомих паралельних методів сортування [1, 2], найбільш придатних для апаратної реалізації, наприклад, на асоціативному процесорі [3], необхідно виділити метод попарного обміну, який також називають методом парно-непарної перестановки (odd-even transposition) [1, 4]. Цей метод у публікації [1] представлено у вигляді сортувальної мережі, що визначає один із способів його реалізації. У публікації [5] саме мережний варіант методу попарного обміну і його трансформація компактно представлено з використанням базису системи алгоритмічних алгебр САА Глушкова.

Необхідно зауважити, що при апаратній реалізації паралельного методу попарного обміну можливе використання систолічних та матричних структур, в яких мережний процес сортування «розгорнутий» у двовимірному просторі [6]. Разом з тим, в асоціативних процесорах [7, 8] цей ітераційний метод «згорнутий» у вигляді циклічного процесу.

Аналіз досліджень та публікацій

Аналізуючи мережний алгоритм попарного обміну з ранжуванням [9, 10], можна відмітити наступні його властивості: простоту, стійкість, природну поведінку, обчислювальну складність O(m), де m – довжина масиву, що сортується [1, 4]. Крім того, цей метод сортування є порівняльним і нелінійним за відомою класифікацією методів і способів сортування [4].

Разом з тим, у методі попарного обміну можна в процесі сортування ефективно використовувати принцип індексації або присвоєння рангів елементам масиву, що сортується [11]. Такий підхід дозволяє оперувати з рангами елементів, змінюючи їх значення за результатами попарного порівняння серед відповідних елементів масиву [9]. Це дає можливість не змінювати розташування елементів у початковому масиві. Крім того, сортування з переупорядкуванням непрямих адрес (рангів чи індексів) відноситься до алгоритмів швидкого сортування [12].

Але разом з тим, варто відмітити, що використання рангів елементів при сортуванні потребує наявності додаткової регістрової пам'яті для перетворення рангів [11]. При цьому виграш у часі досягається за рахунок того, що виконуються швидкі операції інкремента/декремента над значно меншими за величиною числами – рангами [10, 11].

Формулювання цілей статті

Метою роботи є аналіз особливостей мережного алгоритму сортування з ранжуванням елементів лінійного масиву чисел з можливістю представлення алгоритму в термінах САА Глушкова.

Особливості ранжування елементів масиву чисел, що сортується

Початковими даними для мережного алгоритму сортування є початковий масив чисел у вигляді вектора

$$\mathbf{x}^0 = x_1,...,x_i,...,x_n$$

якому ставиться у відповідність початковий вектор рангів

$$\mathbf{g}^0 = g_1^0,...,g_i^0,...,g_m^0,$$

де m – довжина (кількість елементів) масиву, що сортується. Сортування виконується за зростанням значень елементів масиву і відповідних рангів, тобто за модифікованим методом «бульбашка» [1, 4].

Класичний варіант (покрокового) опису мережного алгоритму сортування з ранжуванням елементів масиву чисел має такий вигляд [9].

Крок 1. Ініціалізація шляхом присвоєння значенням рангів чисел з натурального ряду (1,…, m), тобто

$$g_i^0 = i,\ i = \overline{1, m}. \qquad (1)$$

Крок 2. Вибірка (селекція) пари елементів для сортування в непарному циклі з позиціями ($2k$ – 1, $2k$) і в парному циклі з позиціями ($2k$, $2k$ + 1) з урахуванням в t – му циклі вектора рангів $\boldsymbol{g}^{t-1}$:

$$\mathbf{x}^{t-1} \xrightarrow{\mathbf{g}^{t-1}} \mathbf{x}^{t}, \quad t=\overline{1,\ N}, \tag{2}$$

де N – кількість циклів сортування; $k=\overline{1, K}$, K=[m / 2] – ціла частина числа m / 2.

Крок 3. Формування вектора реакцій для кожної з K пар елементів, що сортуються в t-му циклі:

$$\mathbf{q}^t = q_1^t,...,q_K^t \tag{3}$$

з урахуванням співвідношень між елементами в кожній k-й парі:

$$q_k^t = \begin{cases} 1, \text{ якщо } \begin{array}{l} x_{2k-1}^t > x_{2k}^t \text{ у непарних циклах,} \\ x_{2k}^t > x_{2k+1}^t \text{ у парних циклах,} \end{array} \\ \\ 0, \text{ якщо } \begin{array}{l} x_{2k-1}^t \le x_{2k}^t \text{ у непарних циклах,} \\ x_{2k}^t \le x_{2k+1}^t \text{ у парних циклах.} \end{array} \end{cases} \tag{4}$$

Одночасно виконується перевірка умови:

$$\forall q_k^t = 0. \tag{5}$$

Якщо умова (5) не виконується, то процес сортування продовжується з переходом до кроку 4. Якщо умова (5) виконується і це не 1-й цикл, то перехід до кроку 6.

Крок 4. Поділ вектора реакцій (3) відповідно на два вектори для виконання операцій інкремента/декремента вигляду

$$\mathbf{q}^{t+} = q_1^{t+},\ ...,\ q_m^{t+}, \tag{6}$$

$$\mathbf{q}^{t-} = q_1^{t-},\ ...,\ q_m^{t-}, \tag{7}$$

для кожного рангу наступним чином:

а) у непарних циклах

$$q_{2k-1}^{t+} = \begin{cases} 1, \text{ якщо } q_k^t = 1, \\ 0, \text{ якщо } q_k^t = 0, \end{cases} \qquad q_{2k}^{t-} = \begin{cases} 1, \text{ якщо } q_k^t = 1, \\ 0, \text{ якщо } q_k^t = 0, \end{cases} \tag{8}$$

б) у парних циклах

$$q_{2k}^{t+} = \begin{cases} 1, \text{ якщо } q_k^t = 1, \\ 0, \text{ якщо } q_k^t = 0, \end{cases} \qquad q_{2k+1}^{t-} = \begin{cases} 1, \text{ якщо } q_k^t = 1, \\ 0, \text{ якщо } q_k^t = 0. \end{cases} \tag{9}$$

Крок 5. Ранжування (зміна рангів) в процесі виконання відповідних операцій інкремента/декремента, тобто формування поточного вектора рангів $\mathbf{g}^t$ з елементами вигляду:

$$g_i^t = (g_i^{t-1} + q_i^{t+}) \vee (g_i^{t-1} - q_i^{t-}), \;\; i = \overline{1,m}. \tag{10}$$

Перехід до кроку 2.

Крок 6. Формування (зчитування) відсортованого масиву $\mathbf{x}^N$ за виразом (2) з урахуванням встановленого вектора рангів $\mathbf{g}^{N-1}$, тобто

$$\mathbf{x}^0 \xrightarrow{\mathbf{g}^{N-1}} \mathbf{x}^N.$$

Приклад процесу сортування з ранжуванням

У табл. 1 наведено два варіанти сортування за зростанням значень 4-х елементів числового масиву, а саме, класичним методом попарного обміну і модифікованим методом попарного обміну з формуванням рангів у процесі сортування [10]. У табл. 1 квадратні дужки показують пари позицій елементів, що формуються для порівняння у кожному циклі у відповідності з методом попарного обміну, а також пари елементів у масиві, що порівнюються між собою. У нульовому циклі показано, що позиції елементів масиву та присвоєні їм ранги співпадають на початку сортування.

Таблиця 1

Приклади сортування методом попарного обміну

Цикли	Правило формування пар позицій елементів	Метод попарного обміну	Метод з формуванням рангів	
		Масив чисел	Масив чисел	Ранги
0	1	125	125	1
	2	312	312	2
	3	68	68	3
	4	483	483	4
1	1	125	125	1
	2	312	312	2
	3	68	68	3
	4	483	483	4
2	1	125	125	1
	2	312	312	2 +1
	3	68	68	3 -1
	4	483	483	4
3	1	125	125	1 +1
	2	68	312	3
	3	312	68	2 -1
	4	483	483	4
4	1	68	125	2
	2	125	312	3
	3	312	68	1
	4	483	483	4

За класичним методом нумерація позицій елементів масиву (другий стовпець) залишається незмінною, але елементи обмінюються у парах через виконання умов порівняння, що показано квадратними дужками зі стрілками (третій стовпець).

У методі сортування з ранжуванням елементи у масиві не змінюють свого положення (четвертий стовпець), а відповідні ранги у парах, що порівнюються, за певних умов збільшують/зменшують свої значення на одиницю (п'ятий стовпець), що відповідає обміну між елементами масиву у класичному методі сортування. Таким чином показано, як задіяно операції інкремента/декремента.

Отже, в обох випадках застосовуються такі основні операції, як формування пар сусідніх елементів масиву чисел та їх попарне порівняння (табл. 1). Відмінність полягає в тому, що за результатами попарного порівняння при класичному варіанті мережного сортування як у статті [5] виконується транспозиція елементів у парі, а у запропонованому – зміна (інкремент/декремент) значень відповідних рангів елементів і в подальшому пари елементів формуються з урахуванням відповідних сусідніх рангів (табл. 1).

Опис мережного алгоритму сортування в термінах САА Глушкова

Для порівняння із розглянутим класичним (покроковим) описом мережного алгоритму сортування з ранжуванням має сенс навести приклад його опису у базисі САА Глушкова. При цьому, в якості прикладу доцільно використати опис в термінах САА Глушкова синхронного мережного алгоритму сортування методом попарного обміну [5].

В результаті, для опису мережного алгоритму сортування з ранжуванням необхідно ввести два масиви однакової довжини m:

– масив $M = x_1, x_2, \dots, x_m$ елементів, що сортуються;
– масив $R = g_1, g_2, \dots, g_m$ рангів, які відповідають елементам, що сортуються.

На позначених масивах використовуються базисні умови (предикати) і оператори, як для алгоритмів сортування [13, 14], а також розширений базис САА Глушкова [15, 16] з орієнтацією на відоме позначення регулярних схем (РС):

– $(l > r)$ істинне, якщо виконується вказане співвідношення для двох елементів масиву M з сусідніми за значенням рангами;
– $\mathrm{OUT}(M')$ оператор виведення відсортованого масиву M;
– FIN оператор завершення роботи РС.

Крім того, необхідно розширити список предикатів наступним чином:

– $Z(K)$ істинне, якщо виконано умову (5), тобто $\forall q_k^t = 0$;
– q істинне, якщо виконано умову у формулах (8) та (9), тобто $q_k^t = 1$;
– θ істинне, якщо цикли непарні ($2t$ - 1).

До особливостей даного алгоритму сортування також слід віднести необхідність введення базисних операторів вибірки підмасивів із зоною p = 2 вигляду $\mathrm{SEL}(\overline{2k-1, 2k})$ і $\mathrm{SEL}(\overline{2k, 2k+1})$ для формування пар елементів масиву M із сусідніми рангами в непарному і парному циклах (на кроці 2). Оскільки до процесу сортування залучено всі K пари елементів масиву M і всі елементи масиву R, то покажчики і маркери для їхніх розміток не використовуються. Крім того, в даному алгоритмі сортування під зоною предикатів і операторів мають на увазі всі елементи масивів M та R, оскільки виконання операторів і перевірка істинності предикатів здійснюється одночасно над усіма елементами кожного з масивів.

Серед основних операцій, які входять до сигнатури розширених САА Глушкова [2, 14] в подальшому будуть використовуватися такі:

– диз'юнкція $\alpha \vee \beta$;
– кон'юкція $\alpha \wedge \beta$;
– композиція $A \times B$, тобто послідовне застосування операторів A та B;
– альтернатива $[\alpha](A \vee B)$, тобто якщо α істинне, то A, інакше B;
– цикл $[\alpha]\{A\}$, тобто виконувати A, поки α хибне, при α істинному кінець циклу.

Для паралельного застосування операторів A та B можна використовувати позначення A+B [5].

Крім того, необхідно використати такі базисні оператори:

– $\mathrm{INIT}(\overline{g_1^0, g_m^0})$ оператор ініціалізації на кроці 1 для задання значень початкового вектора рангів (1);
– $\mathrm{SET}(\overline{p_1, p_n})$ оператор встановлення в одиничний стан елементів підмасиву $(\overline{p_1, p_n})$ для виконання за виразом (4) на кроці 3.

В результаті складові оператори із застосуванням альтернативи можна представити наступним чином:

– формування вектора реакцій (3) для кожної пари елементів, що сортуються, з сусідніми рангами на кроці 3 з урахуванням непарності /парності циклів (4);

$$\mathrm{FORM}(\overline{q_1, q_K}) ::= [\theta](\mathrm{SET}(\overline{2k-1, 2k}) \vee \mathrm{SET}(\overline{2k, 2k+1}) \tag{11}$$

де використовуються базисні оператори встановлення вигляду:

$$\mathrm{SET}(\overline{2k-1, 2k}) ::= [l > r](q_k = 1 \vee q_k = 0),$$
$$\mathrm{SET}(\overline{2k, 2k+1}) ::= [l > r](q_k = 1 \vee q_k = 0)$$

– поділ вектора реакцій на два вектора (6), (7) на кроці 4 відповідно для операцій інкремента/декремента з урахуванням непарності/парності циклів (8), (9):

$$\mathrm{DIV}(\overline{\mathbf{q}^+, \mathbf{q}^-}) ::= \mathrm{SET}(\overline{q_1^+, q_m^+}) + \mathrm{SET}(\overline{q_1^-, q_m^-}) \tag{12}$$

де використовуються складові оператори встановлення вигляду:

$$\mathrm{SET}(\overline{q_1^+, q_m^+}) ::= [q]([\theta](q_i^+(2k-1) = 1 \vee q_i^+(2k) = 1))$$
$$\mathrm{SET}(\overline{q_1^-, q_m^-}) ::= [q]([\theta](q_i^-(2k) = 1 \vee q_i^-(2k+1) = 1))$$

– ранжування (зміна рангів) на кроці 5 шляхом застосування операцій інкремента/декремента (10):

$$\mathrm{RANK}(\overline{g_1, g_m}) ::= \mathrm{INC}(\overline{g_1, g_m}) + \mathrm{DEC}(\overline{g_1, g_m}), \tag{13}$$

де використовуються оператори вигляду:

$$\mathrm{INC}(\overline{g_1, g_m}) ::= [q](g_i + q_i^+ \vee E),$$
$$\mathrm{DEC}(\overline{g_1, g_m}) ::= [q](g_i - q_i^- \vee E).$$

Таким чином, мережний алгоритм сортування з ранжуванням елементів масиву, що сортується, на основі методу попарного обміну можна представити, використовуючи базисні та складові оператори за виразами (11) – (13), таким чином:

$$\begin{aligned} &\mathrm{EXCH} ::= \mathrm{INIT}(\overline{g_1^o, g_m^o}) \times [\theta](\mathrm{SEL}(\overline{2k-1, 2k}) \vee \mathrm{SEL}(\overline{2k, 2k+1})) \times \\ &\times \mathrm{FORM}(\overline{q_1, q_K}) \times [Z(K)]\{\mathrm{DIV}(\overline{\mathbf{q}^+, \mathbf{q}^-}) \times \mathrm{RANK}(\overline{g_1, g_m}) \times \\ &\times [\theta](\mathrm{SEL}(\overline{2k-1, 2k}) \vee \mathrm{SEL}(\overline{2k, 2k+1})) \times \mathrm{FORM}(\overline{q_1, q_K})\} \times \mathrm{OUT}(M') \times \mathrm{FIN}. \end{aligned} \tag{14}$$

У записі (14) вказано базисні та складові оператори, які відповідають крокам 1–6 наведеного раніше алгоритму сортування. Аналіз запису (14) підтверджує той факт, що основою усього процесу сортування є цикл, який завершується після виконання умови $Z(K)$, що відповідає нульовим значенням q_K у всіх K парах порівнянь елементів за виразом (5). У самому циклі послідовно виконуються всі дії покроково з 3-го по 5-й з поверненням до кроку 2.

Результати

Запропонований підхід до сортування з ранжуванням відрізняється від розглянутого у статті [17] тим, що в даному випадку ранги відсортованих елементів масиву чисел формуються не по завершенню процесу сортування, а безпосередньо приймають участь у цьому процесі, формуючи пари чисел для попарного порівняння, а також, перетворюючись у процесі виконання швидких операцій інкремента/декремента, заміняють трудомісткі операції можливої перекомутації елементів у парах [1, 7]. Враховуючи це, ранги можна розглядати як ваги відповідних елементів масиву, що змінюються (підлаштовуються) за певним правилом, що дозволяє вважати такий підхід нейромережним варіантом сортування [9].

Необхідно також відмітити, що представлення розглянутого мережного алгоритму сортування з формуванням рангів у вигляді запису за принципами САА Глушкова наочно свідчить про максимально припустимий паралелізм оброблення у межах як масиву даних, так і масиву рангів.

Крім того, особливістю алгоритмів сортування з формуванням рангів елементів масиву, що сортується, є те, що зчитувати відсортований масив можна у різних варіантах:

– починаючи зі старших чи молодших рангів, тобто в порядку спадання чи зростання значень елементів масиву;

– тільки елемент зі старшим/молодшим рангом, тобто з вибіркою максимального/мінімального елемента масиву;

– елемент із середнім значенням рангу.

Остання властивість може знайти застосування при медіанній фільтрації сигналів і зображень [18]. Крім того, у розглянутому алгоритмі сортування відмічено, що у початковому масиві не змінюється розташування його елементів, що є необхідною умовою, наприклад, при роботі з базами даних [19, 20].

Висновки з даного дослідження і перспективи подальшого розвитку у даному напрямі

У даній статті наведено два варіанти опису на основі процедурних уявлень синтезованого мережного алгоритму сортування з формуванням рангів елементів масиву чисел, що сортується. Цей алгоритм є трансформацією відомого паралельного алгоритму сортування попарним обміном, який, у свою чергу, є мережним варіантом реалізації алгоритму «бульбашка».

При цьому мережний алгоритм сортування з ранжуванням розроблено для апаратної реалізації у вигляді сортувальної мережі з паралелізмом оброблення пар елементів $[m/2]$, де m – розмірність числового масиву, що сортується. Крім того, обчислювальна складність такого методу сортування порядку $O(m)$, а паралелізм оброблення наочно підтверджено поданням цього алгоритму у базисі САА Глушкова.

Разом з тим, результат подання конкретного алгоритму сортування в рамках формалізму САА Глушкова підтверджує той факт, що компактність, інтерпретація та трансформація є основними властивостями при застосуванні системи алгоритмічних алгебр.

Література

1. Knuth D.E. The Art of Computer Programming. V.3, Sorting and Searching / D.E. Knuth. – Reading: Addison-Wesley Longman, Inc., 1998. – 800 p.
2. Цейтлин Г.Е. Распараллеливание алгоритмов сортировки / Г.Е. Цейтлин // Кибернетика. – 1989. – Т.24, №6. – С. 67-74.
3. Мартинюк Т.Б. Ассоциативный процессор для сортировки массива чисел / Т.Б. Мартинюк, М.М. Аль Хияри, В.П. Майданюк, Ш.М. Хилесь // Вимірювальна та обчислювальна техніка в технологічних процесах. – 2004. – №1. – С. 107-109.
4. Lorin H. Sorting and Sort Systems / H. Lorin. – Mass.: Addison-Wesley Publishing Company, 1975. – 373 p.
5. Кожемяко В.П. Особенности структурного программирования синхронных алгоритмов сортировки / В.П. Кожемяко, Т.Б. Мартынюк, В.В. Хомюк // Кибернетика и системный анализ. – 2006. – №5. – С. 122-133.
6. Kung S.Y. VLSI array processors / S.Y. Kung. – Reading: Prentice Hall, Englewood Cliffs, New Jersey, 1987. – 667 p.
7. Kohonen T. Content-Addressable Memories / T. Kohonen. – Springer-Verlag Berlin Heidelberg, 1987. –388 p.
8. Thurber K.J. Large Scale Computer Architecture: Parallel and Associative Processors / K.J. Thurber. – NJ.: Hayden Book Company, 1976. – 324 p.
9. Martyniuk T.B. Neural network approach to numeric array sorting / T.B. Martyniuk, A.V. Kozhemiako, L.M. Kupershtein, V.V. Khomyuk, Mohamed Salem Nasser Mohamed, A. Smolarz, A. Kozbakova // Proceedings of SPIE 11176: Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments, 111761N (6 November). 2019.
10. Martyniuk T. Features of sorting memory realization / T. Martyniuk, T. Vasilyeva, V. Suprigan, M. AL-Heyari // Proceedings of SPIE (The International Society for Optical Engineering), 2001. – V. 4425. – P. 89-91.

11. Sedgewick R. Algorithms in C++. Third Edition. Parts 1-4: Fundamentals, Data Structures, Sorting, Searching / R. Sedgewick. – Addison Wesley Longman, 1999. – 738p.
12. Blahut R.E. Fast algorithms for digital signal processing / R.E. Blahut. – Reading: Addison-Wesley Publishing Company, 1985. – 466 p.
13. Цейтлин Г.Е. Структурное программирование задач символьной мультиобработки /Г.Е. Цейтлин // Кибернетика. – 1983. – Т.18, №5. – С. 22-30.
14. Цейтлин Г.Е. Проектирование последовательных алгоритмов сортировки: классификация, трансформация, синтез / Г.Е. Цейтлин // Программирование. – 1989. – №3. – С. 3-24.
15. Андон Ф.И. Программирование высокопроизводительных параллельных вычислений: формальные модели и графические ускорители / Ф.И. Андон, А.Е. Дорошенко, К.А. Жереб // Кибернетика и системный анализ. – 2011. – Т.47, №4. – С. 176-187.
16. Андон Ф.И. Инструментальные средства автоматизации параллельного программирования на основе алгебры алгоритмов / Ф.И. Андон, А.Е. Дорошенко, А.Г. Бекетов, В.А. Иовчев, Е.А. Яценко // Кибернетика и системный анализ. – 2015. – Т.51, №1. – С. 162-170.
17. Мартинюк Т.Б. Особливості паралельного алгоритму сортування з формуванням рангів / Т.Б. Мартинюк, Б.І. Круківський // Кібернетика та системний аналіз. – 2022. – Т.58, №1. – С. 31-36.
18. Pratt W.K. Introduction to Digital image Processing / W.K. Pratt. – Reading: Taylor and Francis Group, Inc, 2014. – 371 p.
19. Дейт К. Дж. Введение в системы баз данных. 8-е издание / К. Дж. Дейт. – М.: Вильямс, 2005. – 1328 с.
20. Voitovych O. Multilayer Access for Database Protection / O. Voitovych, L. Kupershtein, V. Lukichov,
I. Mikityuk // Proceedings of 2018 International Scientific-Practical Conference on Problems of Infocommunications Science and Technology, PICS&T'2018. pp. 474-478.
doi: 10.1109/INFOCOMMST.2018.8632152.

References

1. Knuth D.E. The Art of Computer Programming. V.3, Sorting and Searching / D.E. Knuth. – Reading: Addison-Wesley Longman, Inc., 1998. – 800 p.
2. Cejtlin G.E. Rasparallelivanie algoritmov sortirovki / G.E. Cejtlin // Kibernetika. – 1989. – T.24, №6. – S. 67-74.
3. Martyniuk T.B. Associativnyj processor dlya sortirovki massiva chisel / T.B. Martyniuk, M.M. Al Hiyari, V.P. Majdanyuk, Sh.M. Hiles // Vimiryuvalna ta obchislyuvalna tehnika v tehnologichnih procesah. – 2004. – №1. – S. 107-109.
4. Lorin H. Sorting and Sort Systems / H. Lorin. – Mass.: Addison-Wesley Publishing Company, 1975. – 373 p.
5. Kozhemyako V.P. Osobennosti strukturnogo programmirovaniya sinhronnyh algoritmov sortirovki / V.P. Kozhemyako, T.B. Martyniuk, V.V. Homyuk // Kibernetika i sistemnyj analiz. – 2006. – №5. – S. 122-133.
6. Kung S.Y. VLSI array processors / S.Y. Kung. – Reading: Prentice Hall, Englewood Cliffs, New Jersey, 1987. – 667 p.
7. Kohonen T. Content-Addressable Memories / T. Kohonen. – Springer-Verlag Berlin Heidelberg, 1987. – 388 p.
8. Thurber K.J. Large Scale Computer Architecture: Parallel and Associative Processors / K.J. Thurber. – NJ.: Hayden Book Company, 1976. – 324 p.
9. Martyniuk T.B. Neural network approach to numeric array sorting / T.B. Martyniuk, A.V. Kozhemiako, L.M. Kupershtein, V.V. Khomyuk, Mohamed Salem Nasser Mohamed, A. Smolarz, A. Kozbakova // Proceedings of SPIE 11176: Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments, 111761N (6 November). 2019.
10. Martyniuk T. Features of sorting memory realization / T. Martyniuk, T. Vasilyeva, V. Suprigan, M. AL-Heyari // Proceedings of SPIE (The International Society for Optical Engineering), 2001. – V. 4425. – P. 89-91.
11. Sedgewick R. Algorithms in C++. Third Edition. Parts 1-4: Fundamentals, Data Structures, Sorting, Searching / R. Sedgewick. – Addison Wesley Longman, 1999. – 738p.
12. Blahut R.E. Fast algorithms for digital signal processing / R.E. Blahut. – Reading: Addison-Wesley Publishing Company, 1985. – 466 p.
13. Cejtlin G.E. Strukturnoe programmirovanie zadach simvolnoj multiobrabotki / G.E. Cejtlin // Kibernetika. – 1983. – T.18, №5. – S. 22-30.
14. Cejtlin G.E. Proektirovanie posledovatelnyh algoritmov sortirovki: klassifikaciya, transformaciya, sintez / G.E. Cejtlin // Programmirovanie. – 1989. – №3. – S. 3-24.
15. Andon F.I. Programmirovanie vysokoproizvoditelnyh parallelnyh vychislenij: formalnye modeli i graficheskie uskoriteli / F.I. Andon, A.E. Doroshenko, K.A. Zhereb // Kibernetika i sistemnyj analiz. – 2011. – T.47, №4. – S. 176-187.
16. Andon F.I. Instrumentalnye sredstva avtomatizacii parallelnogo programmirovaniya na osnove algebry algoritmov / F.I. Andon, A.E. Doroshenko, A.G. Beketov, V.A. Iovchev, E.A. Yacenko // Kibernetika i sistemnyj analiz. – 2015. – T.51, №1. – S. 162-170.
17. Martyniuk T.B. Osoblyvosti paralelnoho alhorytmu sortuvannia z formuvanniam ranhiv / T.B. Martyniuk, B.I. Krukivskyi // Kibernetyka ta systemnyi analiz. – 2022. – T.58, №1. – S. 31-36.
18. Pratt W.K. Introduction to Digital image Processing / W.K. Pratt. – Reading: Taylor and Francis Group, Inc, 2014. – 371 p.
19. Dejt K. Dzh. Vvedenie v sistemy baz dannyh. 8-e izdanie / K. Dzh. Dejt. – M.: Vilyams, 2005. – 1328 s.
20. Voitovych O. Multilayer Access for Database Protection / O. Voitovych, L. Kupershtein, V. Lukichov, I. Mikityuk // Proceedings of 2018 International Scientific-Practical Conference on Problems of Infocommunications Science and Technology, PICS&T'2018. pp. 474-478. doi: 10.1109/INFOCOMMST.2018.8632152.