УДК 004+51
DOI: 10.31866/2617-796X.5.1.2022.261288

Овсяк Олександр,
доктор технічних наук, доцент, доцент кафедри мистецтв,
Київський національний університет культури і мистецтв,
Київ, Україна
ovsjak@ukr.net
https://orcid.org/0000-0003-2620-1938

Окончено Ігор,
викладач кафедри мистецтв,
Київський національний університет культури і мистецтв,
Київ, Україна
archiglaz@gmail.com
https://orcid.org/0000-0002-1812-0006

МЕТОДИ ВПОРЯДКУВАНЬ У КОМП'ЮТЕРНИХ НАУКАХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЯХ

Метою статті є аналіз відомих методів упорядкувань у комп'ютерних науках.

Методи дослідження: методи упорядкованих пар, відношень, нумерація Геделя, система алгоритмічних алгебр Глушкова.

Наукова новизна полягає у створенні методу як опису, так і перетворень упорядкувань.

Висновки. У методах упорядкованих пар, відношень, нумерації Геделя, математичної логіки, машинах Поста і Тюрінга, модифікованої системи алгоритмічних алгебр Глушкова відсутня можливість еквівалентних перетворень упорядкувань, які наявні в алгоритмах.

Ключові слова: впорядкування; комутативна операція; алгебра; алгоритм.

Вступ. У комп'ютерних науках та інформаційних технологіях упорядкування є фундаментальним поняттям, на якому базуються як теоретичні, так і прикладні результати. Методи впорядкувань застосовують починаючи від теорії множин і закінчуючи будь-якими прикладними апаратурними та програмними комп'ютерними системами й інформаційними технологіями. Особливе значення методи впорядкувань мають у теорії алгоритмів, яка в самій своїй суті і є теорією впорядкувань.

Фахівці з комп'ютерних наук та інформаційних технологій вважають, що історично першим алгоритмом є алгоритм Евкліда, призначений для знаходження найбільшого спільного дільника двох натуральних чисел x і y, таких, що $x>y$ (Нікітченко, 2010). Якщо потрібно написати комп'ютерну програму, яка б реалізовувала алгоритм Евкліда з вводом даних (x і y) з клавіатури комп'ютера, то очевидно, що в разі ручного вводу не виключені помилки. Наприклад, замість цифри набрано й введено інший знак. Чи навіть набрано та введено натуральні числа, але їхні значення є більшими за допустимі для певного комп'ютера. У зв'язку з цим після вводу даних необхідно виконати перевірки. По-перше, потрібно перевірити, чи введені дані є числовими. По-друге, у разі наявності числових даних необхідно перевірити, чи вони належать до діапазону значень комп'ютера (наприклад, 32- чи 64-розрядний), на якому буде програмно реалізовано алгоритм Евкліда. По-третє, перевірка виконання умови *x*>*y* має сенс тільки в разі виконання перших двох умов.

Результати дослідження. Алгоритм Евкліда ілюструє те, що врахування впорядкувань виконуваних операцій має суттєве значення для обчислень. У зв'язку з цим робота присвячена аналізу й оцінці найбільш використовуваних методів упорядкувань, які застосовують у комп'ютерних науках та інформаційних технологіях. Результати роботи можуть бути використані для створення нових більш адекватних й ефективних методів упорядкувань у порівнянні з відомими методами.

Метод упорядкованих пар

Упорядкування. Відомо, що декартовим добутком (Виноградов ред., 1985; Кривий, 2014) $A_1 \times A_2 \times \ldots \times A_n$ множин $A_1, A_2, \ \ldots \ , A_n$ називається сукупність послідовностей (сукупність упорядкованих *n*-ок елементів) виду $(a_1, a_2, \ \ldots \ , a_n)$, де $a_i \in A_i, \ 1 \le i \le n$.

У частковому випадку, коли *n* = 2, декартовий добуток

$A \times B = \ \{(a,b) \,|\, a \in A, b \in B\} \ = C$ є множиною *C*, утвореною упорядкованими парами (*a*, *b*).

Заведено вважати (Виноградов ред., 1985; Кривий, 2014), що в упорядкованій (впорядкованій) парі (*a*, *b*) елемент *a* є першим, а елемент *b* – другим.

Відповідно в упорядкованій *n*-ці елементів $(a_1, a_2, \ \ldots \ , a_n)$ елемент a_1 – перший, a_2 – другий, ..., a_n – *n*-тий.

Відомі (Ordered pair; Куратовский и Мостовский, 1970) описи впорядкованих пар з використанням множин:

– Н. Вінера:

$(a,b): \ = \{\{\{a\}, \varnothing\}, \ \{\{b\}\}\}$.

– Ф. Хаусдорфа:

$(a,b): \ = \ \{\{a, \ 1\}, \ \{b, \ 2\}\}$,

де 1 і 2 – два різних об'єкти, відмінні від *a* і *b*.

– К. Куратовського впорядкована пара (*a*, *b*) множини {*a*, *b*} описується як:

$(a,b): \ = \{\{a\}, \ \{a,b\}\}$.

– Крім вищенаведених, також відомі (Ordered pair) описи упорядкованих пар Кантора – Фреге, Куайна – Россера і Морзе.

Метод відношень

Упорядковування. Для здійснення впорядковування множин застосовують відношення часткового порядку, суворого порядку та відношення квазіпорядку (Виноградов, 1985: Кривий, 2014).

Бінарне відношення *R*, визначене на множині *A*, називається частковим порядком на *A*, якщо воно рефлексивне, транзитивне й антисиметричне, тобто якщо для довільних елементів *a*, *b*, *c* із *A* виконуються властивості:

aRa (рефлексивність);

aRb і *bRc*, то *aRc* (транзитивність);

aRb і *bRa*, то *a* = *b* (антисиметричність) (Виноградов ред., 1985; Кривий, 2014).

Частковий порядок на множині *A*, як правило, позначають символом $\leq$, а саму частково упорядковану множину *A* називають скорочено чум і позначають (A, $\leq$) (Виноградов ред., 1985).

Крім відношення часткового порядку, ще застосовують відношення суворого порядку (<) і відношення квазіпорядку.

На частково упорядкованій множині $(A,\leq)$ введені поняття мінімального (максимального), найбільшого (найменшого) елемента *A*, лінійного порядку на *A* (коли довільні два елементи з *A* порівнюються відносно $\leq$), лінійно упорядкованої множини (ланцюга) і добре упорядкованої множини (коли довільна її непуста підмножина має найменший елемент) (Виноградов ред., 1985; Кривий, 2014).

Відома (Виноградов ред., 1985) аксіома доброго упорядкування (довільну непусту множину можна добре упорядкувати).

Практичне застосування теоретико-множинної моделі впорядковування елементів *a*, *b*, *c*, ..., *t* довільної скінченної множини $M = \{a, b, c, ..., t\}$ присвоєне кожному елементу множини числа. В ідеальному випадку кожне приписане довільному елементу множини число має бути унікальним. Наприклад, у вигляді нижніх індексів елементам *a*, *b*, *c*, ..., *t* припишемо такі числа як 2, 0, 1, ..., *n*. У результаті отримаємо множину з пронумерованими елементами $M^1 = \{a_2, b_0, c_1, ..., t_n\}$. Застосувавши відношення часткового порядку ($\leq$) впорядковуємо елементи в порядку зростання номерів й отримуємо таке впорядкування елементів множини $M^1 = \{b_0, c_1, a_2, ..., t_n\}$.

Дещо інший і часто застосовуваний спосіб упорядковування полягає у використанні впорядкованості літер алфавіту.

Дані способи впорядковувань можуть бути застосовані для впорядкування елементів скінченних множин з відносно невеликою кількістю впорядковуваних елементів.

Упорядкування в математичній логіці

Загальновідомо, що алгоритм Евкліда призначений для знаходження найбільшого спільного дільника двох натуральних чисел x і y, таких що x>y. Потрібно написати комп’ютерну програму, яка б реалізовувала алгоритм Евкліда з вводом даних (x і y) із клавіатури комп’ютера. Очевидно, що під час ручного введення даних не виключені помилки. Наприклад, замість цифри набрано та введено інший знак. Чи навіть набрано й введено натуральні числа, але їхні значення є більшими за допустимі для певного комп’ютера. У зв’язку з цим після вводу даних необхідно виконати перевірки, які запишемо як $x\in N$ і $y\in N$. Якщо, наприклад, на початку перевіряється умова $x\in N$ і результатом перевірки є значення «хибно» (0), то здійснювати перевірку виконання умови $y\in N$ немає потреби. І тільки після виконання обох умов ($x\in N$ і $y\in N$) має сенс виконати наступну перевірку (x>y). Послідовність виконання перевірок $x\in N$ і $y\in N$, чи $y\in N$ і $x\in N$ суттєвого значення не має. Перед тим як здійснити перевірку виконання умови x>y, необхідно перевірити виконання умов $x\in N$ і $y\in N$ чи $y\in N$ і $x\in N$. Інакше кажучи, перевірка виконання умов $x\in N$ і $y\in N$ має бути першою, а перевірка виконання умови x>y – другою. Значеннями як умов $x\in N$ і $y\in N$, так і умови x>y є логічні значення «істинно» (1) або «хибно» (0).

Послідовність цих трьох умов застосуванням операції кон’юнкції класичної математичної логіки (Виноградов ред., 1985; Кривий, 2014) описується виразом $[(x\in N)\&(y\in N)]\&(x>y)$. Ураховуючи комутативність й асоціативність кон’юнкції, цей вираз рівноважний, наприклад, кон’юнкціям $[(x\in N)\&(x>y)]\&(y\in N)$ і $[(x>y)\ \&(x\in N)]\&(y\in N)$, тобто $[(x\in N)\&(y\in N)]\&(x>y) = [(x\in N)\&(x>y)]\&(yN) = [(x>y)\&(x\in N)]\&(y\in N)$.

Однак кон’юнкції $[(x\in N)\&(x>y)]\&(y\in N)$ і $[(x>y)\&(x\in N)]\&(y\in N)$, які рівноважні кон’юнкції $[(x\in N)\&(y\in N)]\&(x>y)$, уже адекватно не описують необхідну послідовність перевірки умов $x\in N$, yN і $x>y$.

Операції класичної математичної логіки, якими є кон’юнкція (&), диз’юнкція () та інвертування ($\neg$), не враховують упорядкувань логічних змінних і сталих.

Нумерація Геделя

У математичній логіці нумерація Геделя – це функція, яка присвоює кожному символу та чітко сформованій формулі якоїсь формальної мови унікальне натуральне число, яке називається числом Геделя (Gödel numbering).

Нумерацію Геделя можна інтерпретувати як кодування, в якому кожному символу математичного позначення присвоюється число, після чого *послідовність* натуральних чисел може представляти *послідовність* символів (Gödel numbering).

Машинні методи

Упорядкування обчислень має фундаментальне значення і в теорії алгоритмів. Найбільш відомими є методи машини Поста (Post, 1936), машини Тюрінга (Turing, 1937), машини Колмогорова (1953), машини Шонгаге (Schönhage, 1970), машини Ахо (Aho, Hopcroft and Ullman, 1974), числення лямбда (Church, 1936), рекурсивних функцій (Kleene, 1981), алгоритмів Маркова (1951), універсальних алгоритмів Крініцкого (Криницкий, 1988).

У методах машин Поста і Тюрінга впорядкування інструкцій програми базується на заданні кожній інструкції номера, якого не має жодна інша інструкція програми. Номера інструкції, з якої починається виконання програми машини, та номера інструкції, яка має виконуватися наступною після виконуваної інструкції.

Машина Поста є абстрактною машиною, а її умовне позначення представлене на рис. 1 і складається з поділеної на комірки нескінченної стрічки та головки. За допомогою машини виконують операції переміщення головки на одну комірку вправо, яку позначимо $\rightarrow$, переміщення головки на одну комірку вліво – $\leftarrow$, запису в комірку мітки – @, усунення з комірки мітки – ×, розпізнавання наявності чи відсутності в комірці мітки – ? та зупинки функціонування машини – #.

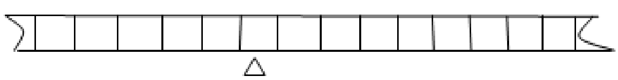


Рис. 1. Умовне позначення машини Поста

Операціями та номерами, які є натуральними числами, утворюються інструкції машини Поста. Є два типи інструкцій. Інструкції першого типу утворено трьома записуваними горизонтально полями. У першому полі записується номер поточної інструкції, у другому – операція і в третьому – номер інструкції, до якої відбувається перехід після виконання машиною поточної інструкції. Другий тип інструкцій складається з двох полів. У першому полі записується номер поточної інструкції, а в другому – операція зупинки функціонування машини Поста. Між полями інструкцій записується хоча б один знак пропуску.

Дві інструкції й більше записуються у вигляді стовпчика, який є послідовністю інструкцій. Програмою є послідовність інструкцій, яка відповідає таким вимогам:

– серед інструкцій є інструкція з номером 1;

– є інструкції, номери яких записано в третіх полях інструкцій.

Представлення даних у машині Поста, наприклад, може бути у вигляді кількості зірочок (*). Нехай записана на стрічці одна зірочка є числом 1, а дві зірочки – числом 2 тощо. Два і більше пропусків між зірочками є розділювачем чисел. Наприклад (рис. 2), записані на стрічці числа 3 і 2 матимуть такий вигляд:

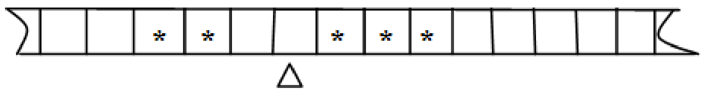


Рис. 2. Стрічка із записаними на ній числами 3 і 2

Програма додавання чисел 2 і 3 має такий вигляд. У програмі перша інструкція описує переміщення головки на одну комірку вправо та перехід до виконання другої інструкції. Видалення мітки з комірки й перехід до третьої інструкції описано в другій інструкції. У третій інструкції описано переміщення головки на одну комірку вправо. Наступними інструкціями описано видалення мітки з комірки, переміщення головки на одну комірку вправо, видалення мітки, переміщення головки на чотири комірки вліво, запис у комірку мітки, переміщення на одну комірку вправо, запис у комірку мітки, знову переміщення на одну комірку вправо і знову запис у комірку мітки та завершення функціонування машини відповідно.

1 → 2

2 × 3

3 → 4

4 × 5

5 → 6

6 × 7

7 ← 8

8 ← 9

9 ← 10

10 ← 11

11 @ 12

12 → 13

13 @ 14

14 → 15

15 @ 16

16 #

Після виконання програми стан машини Поста буде таким, як це показано на рис. 3.

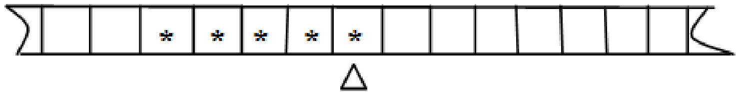


Рис. 3. Стрічка із записаним на ній результатом додавання чисел 3 і 2

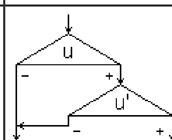
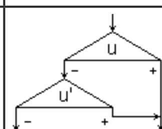
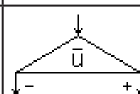
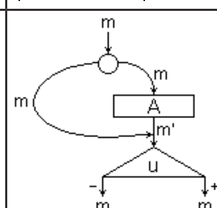
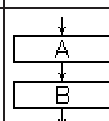
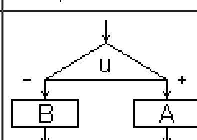
Модифікована система алгоритмічних алгебр Глушкова

Модифікована система алгоритмічних алгебр (Глушков, Цейтлин и Ющенко, 1989) утворена розширеною алгеброю логіки й операторною алгеброю.

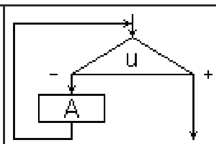
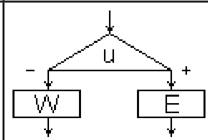
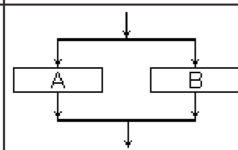
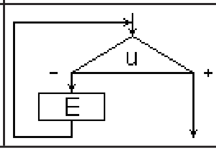
Основні операції модифікованої системи алгоритмічних алгебр наведено в таблиці 1.

Таблиця 1

Основні операції, що входять до сигнатури модифікованої системи алгоритмічних алгебр (САА-М)

Тип	**Назва операції**	**Форма**		
		аналітична	**природно-лінгвістична**	**графова**
Логічні	Кон’юнкція	$u \wedge u'$ $u\ .\ u'$	*u* I *u’*	
	Диз’юнкція	$u \vee u'$ $u + u'$	*u* АБО *u’*	
	Заперечення	$\overline{u}$	НЕ(*u*)	
	Прогнозування (ліве множення умови на оператор)	$A \cdot u$	ПІСЛЯ *A* умова *u*	
Операторні	Композиція	$A * B$	*A* ПОТІМ *B*	
	Альтернатива	$([u]\ A, B)$	ЯКЩО *u* ТО *A* ІНАКШЕ *B*	

Продовження табл. 1

Операторні	Цикл	$\{[u]\ A\}$	ПОКИ НЕ *u* ЦИКЛ *A*	
	Фільтрація	$\underline{u}$	ФІЛЬТР(*u*)	
	Асинхронна диз’юнкція	$A \dot{\vee}$ B	А ПАРАЛЕЛЬНО В	
	Синхронізатор	$S(u)$	ЧЕКАТИ(*u*)	

Операції виконуються над логічними й операторними змінними.

Будь-якими операторами є: *P, R, Q, …*

Змінні логічні: *α, β, γ, ...*

Узагальнені логічні операції виконуються над змінними, які приймають три значення, а саме 0, 1, *μ*:

Узагальнена кон’юнкція:

α &β	0	1	μ
0	0	0	0
1	0	1	μ
μ	0	μ	μ

Узагальнена диз’юнкція:

α ˅β	0	1	μ
0	0	1	μ
1	1	1	1
μ	μ	1	μ

Узагальнене інвертування:

α	*α*
0	1
1	0
μ	μ

Операції узагальнених кон’юнкції, диз’юнкції та інвертування мають такі самі властивості, як і операції кон’юнкції, диз’юнкції та інвертування математичної логіки першого порядку, крім властивостей виключного третього $(\alpha \vee \overline{\alpha} = 1)$ і заперечення $(\alpha \,\&\, \overline{\alpha} = 0)$.

Операція фільтрації:

$$\underline{\alpha} = \begin{cases} E, \text{ якщо } u = \alpha, \\ N, \text{ в інших випадках}; \end{cases}$$

де *E* – тотожний оператор і *N* – неозначений оператор.

Операція фільтрації має такі властивості:

$$\underline{\alpha \,\&\, \beta} = \underline{\alpha} \,\&\, \underline{\beta},$$

$$\underline{\alpha \vee \beta} = \underline{\alpha} \,||\, \underline{\beta},$$

$$\underline{\alpha} \,\&\, \underline{\beta} = \underline{\beta} \,\&\, \underline{\alpha},$$

$$\underline{\alpha} \,||\, \underline{\beta} = \underline{\beta} \,||\, \underline{\alpha}.$$

Асинхронна диз'юнкція має властивості асоціативності та комутативності:

P|| (Q|| R) = (P||Q) || R,

P|| Q = Q|| R.

Асинхронна диз'юнкція також має властивості дистрибутивності й поглинання:

P(Q|| R) = PQ || PR,

(P || Q)R = PR || QR,

P || P = P.

Операція композиції *P * Q* (послідовне застосування операторів *P* і *Q*) асоціативна:

(P * Q) * R = P * (Q * R),

не є комутативна й одночасно для неї виконуються рівності:

P * E = E * P = P,

P * N = N * P = N.

Отож *E* і *N* є сталими операторної алгебри.

Як було описано вище, операції системи алгоритмічних алгебр відповідають усім законам Булевої алгебри, крім законів виключного третього ($x \vee \neg x = 1$) і суперечності ($x \& \neg x = 0$). У зв'язку з цим у системі алгоритмічних алгебр кон'юнкція $[(x \in N)\&(y \in N)]\&(x>y)$ теж рівноважна кон'юнкціям $[(x \in N)\&(x>y)]\&(y \in N)$ і $[(x>y)\&(x \in N)]\&(y \in N)$.

Крім того, у системі алгоритмічних алгебр для впорядкування операторів (*P*, *Q*, *R*, …) введена операція композиції (позначена знаком *), яка є асоціативною:

(P * Q) * R = P * (Q * R).

Розглянемо приклад застосування модифікованої системи алгоритмічних алгебр. Алгоритм Евкліда є такою формулою модифікованої системи алгоритмічних алгебр:

$x{=}\leftarrow * ([x \in N]\ y{=}\leftarrow * ([y \in N]\ ([x>y]\ \{(r{=}\%(x,\ y)) * [r \neq 0]\ x{=}y * y{=}r\} * n{=}y,\ K_3),\ K_2),\ K_1).$

Можливо також спочатку ввести дані, а потім виконувати їхню перевірку. Формула алгоритму Евкліда з урахуванням властивості асоціативності операції композиції матиме такий вигляд:

x=← * (y=← * ([x∈N] ([y∈N] ([x>y] {(r=%(x, y)) * [r≠0] x=y * n= r} * n=y, K_3), K_2), K_1)).

Звідси бачимо, що насамперед має бути виконана формула:

(y=← * ([x∈N] ([y∈N_1] ([x>y] {(r=%(x, y)) * [r≠0] x=y * y= r} * n=y, K_3), K_2), K_1)),

а потім – введене значення змінної (x=←). Очевидно, що така послідовність дій загалом дасть неправильний результат.

Висновки. У методах упорядкованих пар, відношень, нумерації Геделя, математичної логіки, машинах Поста і Тюрінга, модифікованої системи алгоритмічних алгебр Глушкова відсутня можливість еквівалентних перетворень упорядкувань, які наявні в алгоритмах. У зв'язку з цим є актуальним створення методу як опису, так і перетворень упорядкувань.

СПИСОК ПОСИЛАНЬ

Виноградов, И.М. ред., 1985. *Математическая энциклопедия.* Москва: Советская энциклопедия.

Глушков, В.М., Цейтлин, Г.Е. и Ющенко, Е.Л., 1989. *Алгебра. Языки. Программирование.* Киев: Наукова думка.

Колмогоров, А.Н., 1953. О понятии алгоритма. *Успехи математических наук*, 8 (4), с.175-176.

Кривий, С.Л., 2014. *Дискретна математика.* Чернівці: Київ: Букрек.

Криницкий, Н.А., 1988. *Алгоритмы вокруг нас.* Москва: Мир.

Куратовский, К. и Мостовский, А., 1970. *Теория множеств.* Перевод с английского М.И. Кратко. Москва: Мир.

Марков, А.А., 1951. Теория алгорифмов. В: *Труды Математического института им. В.А. Стеклова АН СССР*, 38, с.176-189.

Нікітченко, М.С., 2010. *Теоретичні основи програмування.* Ніжин: Видавництво НДУ імені Миколи Гоголя.

Aho, A.V., Hopcroft, J.E. and Ullman, J.D., 1974. *The design and analysis of computer algorithms*. Addison-Wesley Publishing Company,

Church, A., 1936. An unsolvable problem of elementary number theory. *American Journal of Mathematics,* 58, pp.345-363.

Gödel numbering. *Wikipedia*. [online] Avialable at: <https://en.wikipedia.org/wiki/G%C3%B6del_numbering> [Accessed 15 April 2022].

Kleene, S.C., 1981. Origins of recursive function theory. *Annals of the History of Computing*, 3 (1), pp.52-67.

Ordered pair. *Wikipedia*. [online] Avialable at: <https://en.wikipedia.org/wiki/Ordered_pair> [Accessed 15 April 2022].

Post, E.L., 1936. Finite Combinatory Processes Formulation 1. *Journal of Symbolic Logic*, 1, pp.103-105.

Schönhage, A., 1970. Universelle Turing Speicherung. In: J. Dörr and G. Hotz eds. *Automatentheorie und Formale Sprachen, Bibliogr. Institut, Mannheim*, pp.369-383.

Turing, A.M., 1937. On computable numbers, with an application to the Entscheidungs problem. *Proceedings of London Mathematical Society, series 2*, 42, pp.230-265.

REFERENCES

Aho, A.V., Hopcroft, J.E. and Ullman, J.D., 1974. *The design and analysis of computer algorithms.* Addison-Wesley Publishing Company,

Church, A., 1936. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58, pp.345-363.

Gluschkow, W.M., Zeitlin, G.E. and Justchenko, J.L. 1980. *Algebra. Sprachen. Programmierung.* Berlin: Akademie-Verlag.

Glushkov, V.M., Tceitlin, G.E. and Iushchenko, E.L., 1989. *Algebra. Iazyki. Programmirovanie* [Algebra. Languages. Programming]. Kyiv: Naukova dumka.

Gödel numbering. Wikipedia. [online] Avialable at: <https://en.wikipedia.org/wiki/G%C3%B6del_numbering> [Accessed 15 April 2022].

Kleene, S.C., 1981. Origins of recursive function theory. *Annals of the History of Computing,* 3 (1), pp.52-67.

Kolmogorov, A.N., 1953. O poniatii algoritma [On the concept of an algorithm]. *Uspekhi matematicheskikh nauk*, 8 (4), pp.175-176.

Krinitckii, N.A., 1988. *Algoritmy vokrug nas* [Algorithms around us]. Moscow: Mir.

Kryvyi, S.L., 2014. *Dyskretna matematyka* [Discrete Mathematics]. Chernivtsi: Kyiv: Bukrek.

Kuratovskii, K. and Mostovskii, A., 1970. *Teoriia mnozhestv* [Theory of sets]. Translation from English M.I. Kratko. Moscow: Mir.

Markov, A.A., 1951. Teoriia algorifmov [Theory of algorithm]. In: *Trudy Matematicheskogo instituta im. V.A. Steklova AN SSSR* [Proceedings of the Mathematical Institute. V.A. Steklov Academy of Sciences of the USSR], 38, pp.176-189.

Nikitchenko, M.S., 2010. *Teoretychni osnovy prohramuvannia* [Theoretical bases of programming]. Nizhyn: Vydavnytstvo NDU imeni Mykoly Hoholia.

Ordered pair. *Wikipedia*. [online] Avialable at: <https://en.wikipedia.org/wiki/Ordered_pair> [Accessed 15 April 2022].

Post, E.L., 1936. Finite Combinatory Processes Formulation 1. *Journal of Symbolic Logic,* 1, pp.103-105.

Schönhage, A., 1970. Universelle Turing Speicherung. In: J. Dörr and G. Hotz eds. *Automatentheorie und Formale Sprachen,* Bibliogr. Institut, Mannheim, pp.369-383.

Turing, A.M., 1937. On computable numbers, with an application to the Entscheidungs problem. *Proceedings of London Mathematical Society, series* 2, 42, pp.230-265.

Vinogradov, I.M. ed., 1985. *Matematicheskaia entciklopediia* [Mathematical Encyclopedia]. Moscow: Sovetskaia entciklopediia.

UDC 004+51

Ovsiak Oleksandr
Doctor of Technical Sciences,
Associate Professor at the Department of Arts,
Kyiv National University of Culture and Arts,
Kyiv, Ukraine
ovsjak@ukr.net
https://orcid.org/0000-0003-2620-1938

Okonchenko Ihor,
Senior Lecturer at the Department of Arts,
Kyiv National University of Culture and Arts,
Kyiv, Ukraine
archiglaz@gmail.com
https://orcid.org/0000-0002-1812-0006

METHODS OF ORDERING IN COMPUTER SCIENCES AND INFORMATION TECHNOLOGIES

The purpose of the article is to analyze the known methods of ordering in computer science.

The research methodology consists of methods of ordered pairs, relations, Gödel numbering, and Glushkov's system of algorithmic algebras.

The scientific novelty is to create a method of both description and transformation of orders.

Conclusions. In the methods of ordered pairs, relations, Gödel numbering, mathematical logic, Post and Turing machines, and the modified system of Glushkov's algorithmic algebras there is no possibility of equivalent transformations of orders which are available in algorithms.

Keywords: ordering; commutative operation; algebra; algorithm.

11.01.2022