ISSN 2450-6486
http://ehs.eeipsy.org
DOI: https://doi.org/10.38014/ehs-ss.2022.3.06
UDC 004

Ghryghorij KOROTENKO,
Leonid KOROTENKO,
Aliona KHAR,
Artem SHYRIN

# Ways to improve algorithmic understanding of sorting methods while studying the "Algorithms and data structures" course by the university students

KOROTENKO GHRYGHORIJ, KOROTENKO LEONID, KHAR ALIONA, SHYRIN ARTEM. **Ways to improve algorithmic understanding of sorting methods while studying the "Algorithms and Data Structures" Course by the university students.** *The paper considers peculiarities of the material teaching while learning algorithms of data sorting in terms of the "Algorithms and Data Structures" course and the ways to reduce possible complexities while its mastering. Active progress of software and information systems towards the formation of their convergent and hyper-convergent modifications results in rapid*

ARTICLE INFO
Received 12.08.2022
Peer-reviewed 12.12.2022
Revised version received 12.13.2022
Accepted 12.14.2022



© Ghryghorij KOROTENKO, Leonid KOROTENKO, Aliona KHAR, Artem SHYRIN, 2022

*development of the amount of all types of interconnected data. Along with that, there is still a constantly reported burning problem as for catastrophic lack of high-skilled specialists in all computing spheres including the data processing one. A competence of IT-specialists is based on the ability of solving various algorithmic problems that they have not met before. One of the most widespread problems is the necessity in data sorting. Consequently, it is proposed to consider the ways to improve algorithmic understanding of sorting methods in terms of the "Algorithms and Data Structures" course and possible methods for reducing the complexity while its studying.*

***Keywords:*** *sorting algorithm, data structure, data processing, programming, tabular graphical form of algorithm representation, Mind Maps method.*

**Statement of the problem.** IDC (International Data Corporation), the international company dealing with marketing studies and consultation services in the sphere of information technology, states that currently a process of digitalization, which is often called digital transformation, is changing completely a business form, influencing companies and consumers in all spheres all over the worlds. Digital transformation is not just the evolution of devices (though they will change anyway); it is integration of intellectual data into every aspect of our everyday life [1, p. 2]. Nowadays, the companies are facing great flow of bits and bytes controlled by real-time interferences from large business clusters. Working with digital transformation requires not only new technologies but also new skills [1, p. 2].

**Analysis of recent studies and publications.** Analysis of electronic resources and literature sources makes it possible to conclude that the "Algorithms and Data Structures" course is one of the foundation stones in the educational programmes in the field of computing. Moreover, it is important not only from the viewpoint of professional basics but also from the viewpoint of preparation for an interview for a position of programmer including large IT-organizations (Google, Microsoft, Apple etc.) or some prospective startup, when an applicant is to answer some questions concerning data sorting [2, p. 145-149]. As a rule, this course differs with great amount of complex data structures and various algorithms of their processing. To ensure proper training of future IT-specialists, the paper describes one of the possible approaches while teaching the materials concerning formation of algorithmic representation of sorting methods

for the university students of Ukraine in terms of the "Algorithm and Data Structures" course.

**Formulation of the paper purposes.** Basics of information science, supporting all software design processes, are the integral part of computing. Despite constantly growing number of educational programmes and their constituent elements, algorithms are the fundamentals for computer science and software development [3, p. 55-61].

In terms of Standards of Higher Education of Ukraine, determining basic competences included in the training outcomes of bachelors in terms of field 12 "Information Technology", the latter are implemented while studying the disciplines (courses) being a part of curricula of the educational and professional programmes. As a rule, the curricula for all six specialties of this field of knowledge (121 Software Engineering [4], 122 Computer Science [5], 123 Computer Engineering [6], 124 System Analysis [7], 125 Cybersecurity [8], 126 Information Systems and Technology [9]) include learning algorithms and data structures in terms of three courses: "Algorithmization and Programming", "Algorithms and Data Structures", and "Object-Oriented Programming". The first one usually means solution of educational and practical problems by means of any commonly used programming languages (PL). The second one is implemented as the description of using one of widely used PL (C++, Java, C#, Python etc.) to implement abstract data types basing on various integrated data structures as well as to use them as the basis for designing processing algorithms for data items stored in them with the evaluation of algorithm complexity. The third course considers the concepts of object-oriented programming (incapsulation, inheritance, polymorphism etc.) basing on the formation and use of classes and objects, operator overloading, multiple inheritance of virtual functions and classes as well as abstract classes.

Any course taught at the university functions within the structure of endless limitations and influences. On the one hand, a corresponding course is the component of certain undergraduate programme; it is a block being either inside or beyond the body of knowledge of a specific specialty. On the other hand, this course experiences certain influence of the specialized constituents of certain rapidly growing body of knowledge, for which future specialists are being trained. Thus, methodological support of the course is an important constituent of the academic process; its recommends the most effective rational variants and examples of actions

concerning the type of activity learnt during the course. Consequently, to study data sorting algorithms, the paper proposes tabular graphical representation of the corresponding processes based on the approaches proposed in [10].

**Statement of the main material.** Stage one of the research involved detalization of the computing field, related to data processing, involving the Mind Maps method [11] (Fig.1). In this context, all possible tendencies and spheres of servicing the data processing execution are taken into consideration including the processes of algorithmization, software design, and programming itself [12, 13, 14].

As a rule, data processing is "data collection and handling to get sensitive information". In this context, it can be considered as the subset of a "data collection" process, which in its turn is defined as "information altering (processing) by any method identified with the help of an observer" [15].

Data processing includes performing several basic processes.

- Validation – checking accuracy and actuality of the represented data.
- Sorting – "arranging the objects in certain order and / or in different sets (groups)".
- Generalization – reducing the detailed data to the main general moments.
- Aggregation – uniting several parts of the data.
- Analysis – "data collection, organization, analysis, interpretation, and representation".
- Reporting – a list of detailed or reduced data or calculated information.
- Classification – data division into different categories.

While considering the components of this representation based on Mind Map, it can be concluded that one of the most important elements of algorithm implementation is a process development of data structures and algorithms of their processing based on the applied PL and their use while sorting and searching for the data required to develop the corresponding software and information systems.

Analysis of Fig.1 helps formulate following range of problems to be solved while forming the approaches to learning the sorting methods:

- selection of the necessary sorting method;

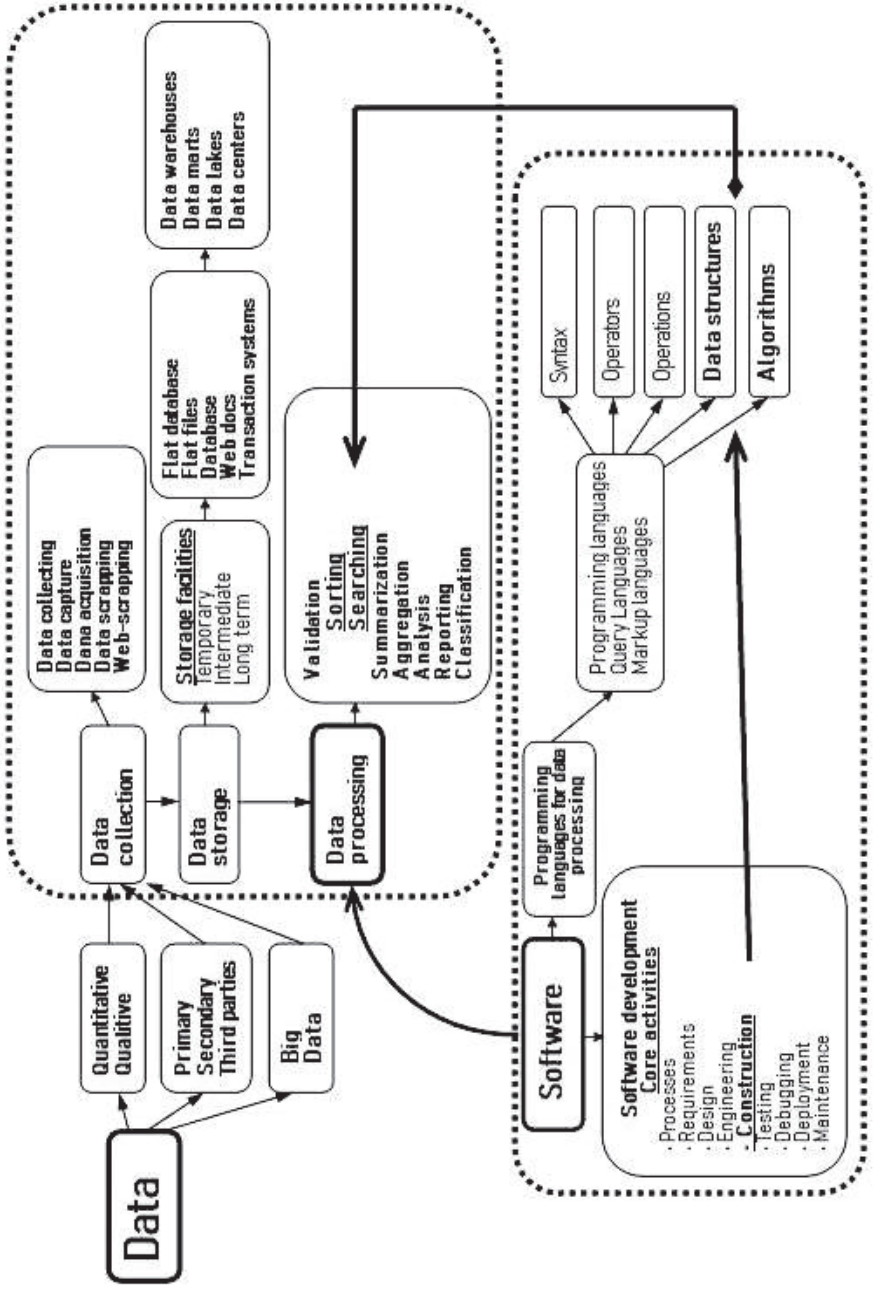



**Fig. 1. Mind Map of the data processing domain**

- design of an algorithm to implement the selected sorting method;
- selection of the corresponding programming languages for the implementation;
- selection of the built-in data types peculiar for the selected language, which makes it possible to store and process data being in compliance with the problem under solution (purposes, real numbers, text etc.);
- selection of the data structures meant for storage and operation with the elements of the processed elements.

Along with that, the requirement for the first-year students to know different sections of mathematics along with certain experience of mathematical transformations and mastering high-level PL [4-9] are rather problematic as usually students deal with more profound PL learning during their third and further years of study.

While using PL, one should take into account following important factors determining programmer's thinking style and many other additional complexities arising at different stages of implementing algorithms and data structures.

1. The available programming paradigms, with more than 70 names up to now [16].
2. Considerable number of PL, including quite new ones (Rust, Swift, Go etc.) [17].
3. Intra-language features relying on the peculiarities of the paradigms they belong to.
4. Different types of operations and different levels of their execution priorities [18].

Mastering of the "Algorithms and Data Structures" discipline is the complexity of the types of organization of means to represent various data and possible operating and algorithmic capacities of specific PL at different stages of solving the specified problems. In its essence, PL is an intermediate between certain structures of storage of data elements and their application area (development of Web-applications, databases of information systems, e-commerce systems etc.). On the other hand, apart from constantly growing number of PL, we can observe an increasing number of:

- constantly progressing hardware components;
- information-technological and digital platforms.

Thus, it is proposed to form in students the unified approaches to

modelling different algorithms basing on the study of those algorithms in terms of representation in two forms: programme and tabular graphical. Further, the examples of tabular graphical representation for four methods are given [19, p. 243-358]:

1. Bubble sort (Fig. 3).
2. Insertion sort (Fig.4).
3. Shell sort (Fig. 5).
4. Quick sort (Fig. 6).

Sorting methods 2-4 apply standard sequence of integer numbers (Fig.2).


**Fig. 2. Standard sequence of integer numbers for sorting**

Thus, tabular-graphic representation of sorting stages makes it possible to represent a procedure of applying different sorting methods.

**Conclusions.** The paper has considered the methods to improve a level of algorithmic understanding of sorting methods in terms of the "Algorithm and Data Structures" course along with the means for reducing the complexity during the studies.

The Mind Maps method was applied to detalize the computing area related to data processing. In this context, all possible tendencies and spheres of servicing of data processing execution were considered including the processes of algorithmization, software design, and the programming itself.

A range of problems was formulated to be solved while forming the approaches to learning the sorting methods. It is proposed to form in students certain unified approaches to modelling different algorithms by their studying not only by a standard method (description with the help of natural language, flowcharts, and pseudo-codes [20]) but also by two more forms: programme and tabular graphical.

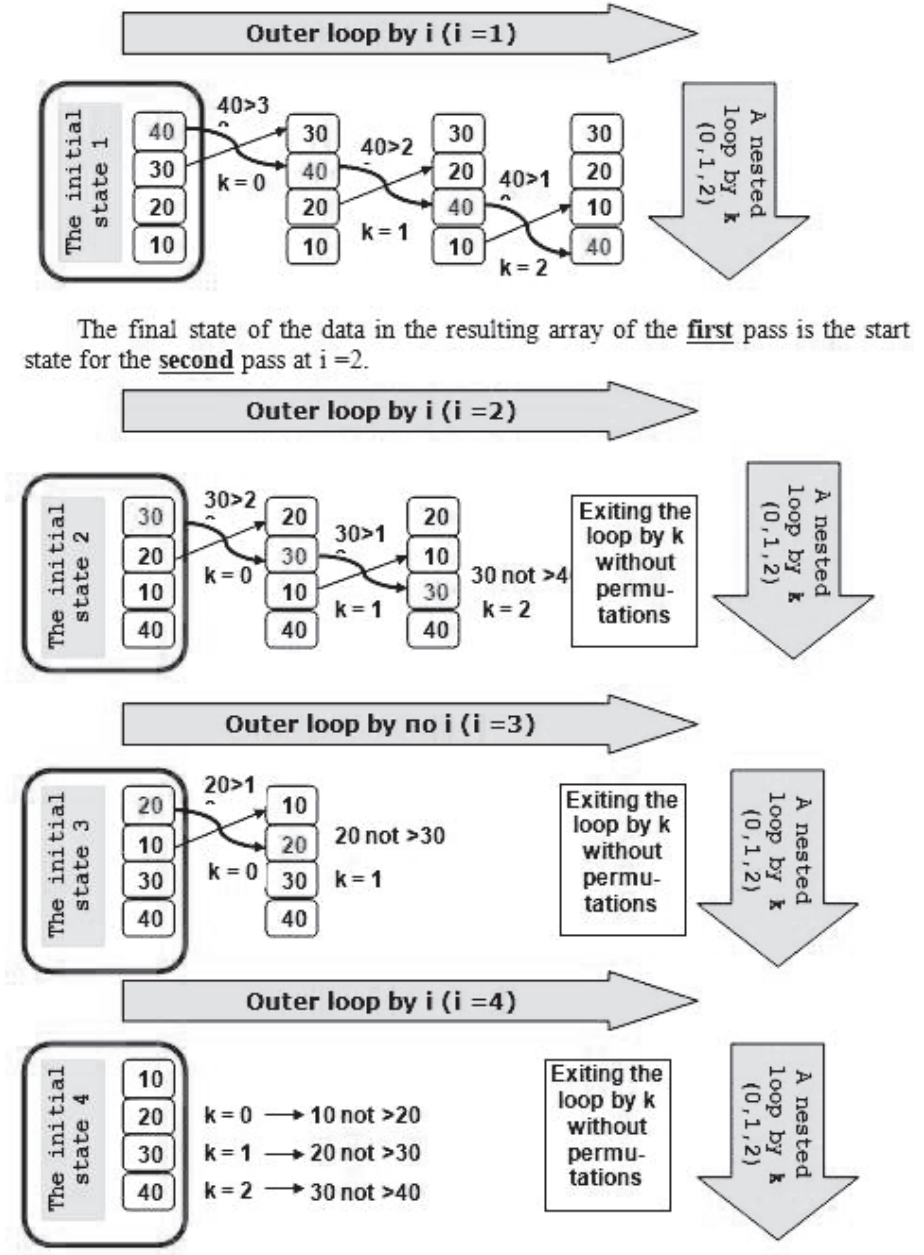



**Fig. 3. Tabular graphical scheme of applying a bubble sort method**

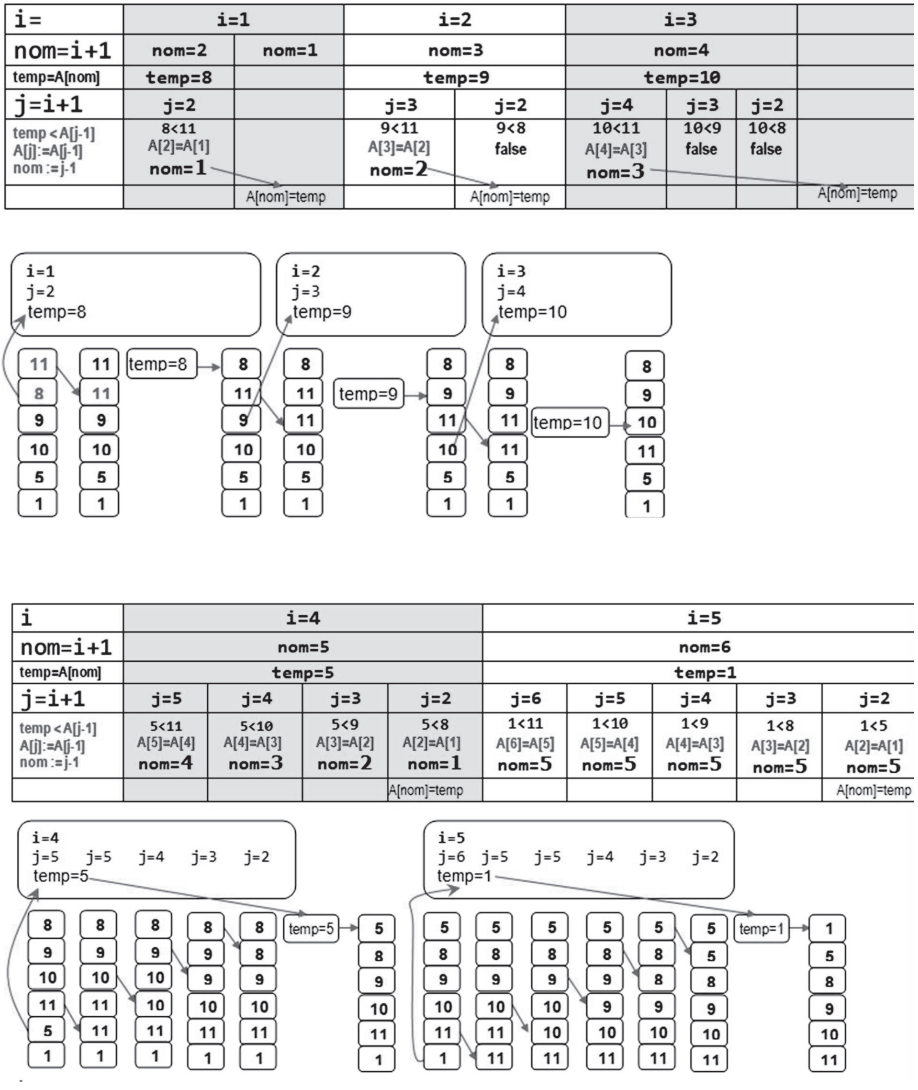
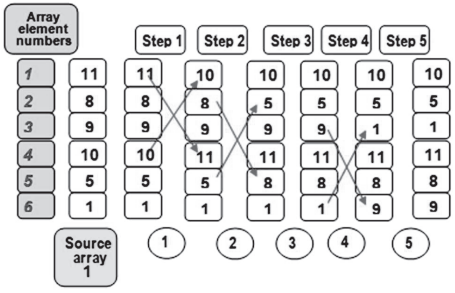



**Fig. 4. Tabular graphical scheme of representing the insertion sort stages**

| step number | 1 | 2 | 3 | 4 | 5 | | |
|---|---|---|---|---|---|---|---|
| n=6 | n-d=3 | | | | | | |
| d=6 | d=3 | | | | | | |
| i= | i=1 | i=2 | | i=3 | | | |
| j=i | j=1 | j=2 | j=1 | j=3 | j=2 | | |
| j+d | j+d=4 | j+d=5 | j+d=4 | j+d=6 | j+d=5 | | |
| A[j] > A[j+d] | A[1] > A[4] | A[2] > A[5] | A[1] > A[4] | A[3] > A[6] | A[2] > A[5] | | |
| | 11>10 | 8>5 | 10>11 | 9>1 | 5>8 | | |
| | true | true | false | true | false | | |
| temp = A[j]<br>A[j] =A[j+d]<br>A[j+d] = temp | A[1] = A[4]<br>A[4] = A[1] | A[2] = A[5]<br>A[5] = A[2] | no exchange | A[3] = A[6]<br>A[6] = A[3] | no exchange | | |
| | j=j-1=0 | j=j-1=1 | | j=j-1=2 | | | |




| step number | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|
| n=6 | n-d=5 | | | | | | | |
| d=6 | d=1 | | | | | | | |
| i= | i=1 | i=2 | | i=3 | i=4 | | | i=5 |
| j=i | j=1 | j=2 | j=j-1=1 | j=3 | j=4 | j=j-1=3 | j=j-1=2 | j=5 |
| j+d | j+d=2 | j+d=3 | j+d=2 | j+d=4 | j+d=5 | j+d=4 | j+d=3 | j+d=6 |
| A[j] > A[j+d] | A[1] > A[2] | A[2] > A[3] | A[1] > A[2] | A[3] > A[4] | A[4] > A[5] | A[3] > A[4] | A[2] > A[3] | A[5] > A[6] |
| | 10>5 | 10>1 | 5>1 | 10>11 | 11>8 | 9>8 | 5>8 | 11>10 |
| | true | true | true | false | true | true | false | true |
| temp = A[j]<br>A[j] =A[j+d]<br>A[j+d] = temp | A[1] = A[2]<br>A[2] = A[1] | A[2] = A[3]<br>A[3] = A[2] | A[1] = A[2]<br>A[2] = A[`] | no exchange | A[4] = A[5]<br>A[5] = A[4] | A[3] = A[4]<br>A[4] = A[3] | no exchange | A[5] = A[6]<br>A[6] = A[5] |
| | j=j-1=0 | j=j-1=1 | j=j-1=0 | | j=j-1=3 | j=j-1=2 | | j=j-1=4 |

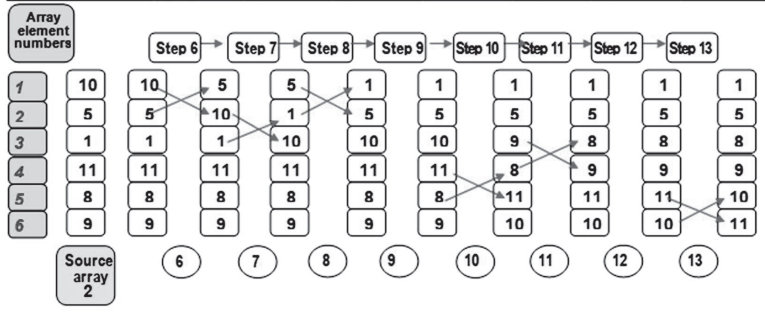



**Fig. 5. Sorting by Shell sort**

| step number | ❶ | | ❷ | quicksort (A, 1, 4) | ❸ | quicksort (A, 1, 3) | | |
|---|---|---|---|---|---|---|---|---|
| first=1 | f=1 | | | f=1 | | f=1 | | |
| last=6 | t=6 | | | t=4 | | t=3 | | |
| mid = A[(f + t) div 2] | mid=A[3] 9 | | | mid=A[2] 8 | | mid=A[2] 5 | | |
| repeat | | | | | | | | |
| while | | | | | | | | |
| *inc* | f=1 | f=2 | f=3 | f=1 | | f=1 | | |
| | A[f] < 9 | A[f] < 9 | A[f] < 9 | A[f] < 8 | | A[f] < 5 | | |
| | A[1] < 9 | A[2] < 9 | A[3] < 9 | A[1] < 8 | | A[1] < 5 | | |
| | 11<9 | 8<9; f=3 | 9<9 f=3 | 1<8; f=2 | | 1<5; f=2 | | |
| | false | | false | 8<8; f=2 | f=2 | 5<5 | | |
| while | | | | | | | | |
| *dec* | t=6 | t=5 | t=5 | t=4 | t=3 | t=3 | | |
| | A[t] > 9 | | A[t] > 9 | A[t] > 8 | A[t] > 8 | A[t] > 5 | | |
| | A[6] >9 | | A[5] > 9 | A[4] >8 | A[3] >8 | A[3] >5 | | |
| | 1 > 9 | | 5>9 | 10>8; t=3 | 5>8 | 5>5 | | |
| | false | | false | | false | false | | |
| if f<=t | 1<=6 | | 3<=5 | 1<=6 | 2<=3 | 2<=3 | | |
| temp = A[f] A[f] =A[t] A[t] = temp | A[1] = A[6] A[6] = A[1] | | A[3] = A[5] A[5] = A[3] | | A[3] = A[2] A[2] = A[3] | A[3] = A[2] A[2] = A[3] | | |
| inc | f=2 | | f=4 | | f=3 | | | |
| dec | t=5 | | t=4 | | t=3 | | | |
| first < t | | | 1 < 4 | | 1 < 3 | | | |
| f < last | | | | | | | | |
| | | | | | | | | |

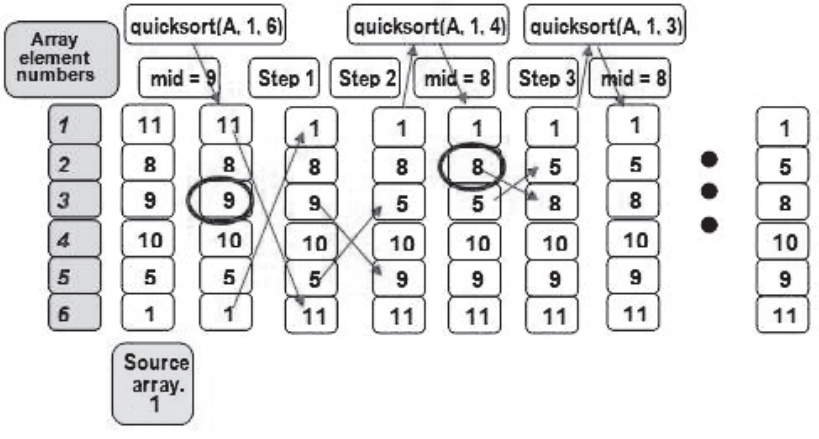



**Fig 6. Sorting by quick sort**

Examples of tabular graphical representation for four methods were given:

1. Bubble sort.
2. Insertion sort.
3. Shell sort.
4. Quick sort.

Has been formed an approach to increase the level of algorithmic understanding for sorting methods within the course "Algorithms and Data Structures".

**References:**


1. David Reinsel, John Gantz, John Rydning. The Digitization of the World From Edge to Core. URL: https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf (access date: 02.02.2023).
2. Gayle Laakmann McDowell. Cracking the Coding Interview. 6th Edition. 189 Programming Questions and Solutions. Published by CareerCup, LLC, Palo Alto, CA. Compiled Feb 10, 2016. – 696 p. ISBN-13: 978-0984782857
3. Computer Science Curricula 2013 (CS 2013); Curriculum Guidelines for Undergraduate Degree Programs in Computer Science, Association for Computing Machinery (ACM), IEEE Computer Society. URL: https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf (access date: 02.02.2023).
4. Затверджені стандарти вищої освіти. 121 Інженерія програмного забезпечення. URL: https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/12/21/121-inzhener.programn.zabezp.bakalavr-1.pdf (дата звернення: 02.02.2023).
5. Затверджені стандарти вищої освіти. 122 Комп'ютерні науки. URL: file:///C:/Users/%D0%93%D0%9C/Downloads/5d5aa2f2db1bb222307639.pdf (дата звернення: 02.02.2023).
6. Затверджені стандарти вищої освіти. 123 Комп'ютерна інженерія. URL: https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/123-kompyuterna-inzheneriya.pdf (дата звернення: 02.02.2023).
7. Затверджені стандарти вищої освіти. 124 Системний аналіз. URL: https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/12/21/124-sistemn.analiz-bakalavr-1.pdf (дата звернення: 02.02.2023).
8. Затверджені стандарти вищої освіти. 125 Кібербезпека. URL: https://mon.gov.ua/storage/app/uploads/public/5bb/626/1a8/5bb6261a84776166409164.pdf (дата звернення: 02.02.2023).
9. Затверджені стандарти вищої освіти. 126 Інформаційні системи і технології. URL: https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/12/21/126-inform.sist.tekhnol.bakalavr-1.pdf (дата звернення: 02.02.2023).

10. Microsoft Office 2019 Step by Step / Joan Lambert, Curtis Frye. 1st Edition. Microsoft Press: 2018. – 560 p.
11. The Ultimate Book of Mind Maps. URL: https://ia802601.us.archive.org/20/items/pdfy-MEuyCwZKGT3fqH56/The%20Ultimate%20Book%20of%20Mind%20Maps%20-%20Tony%20Buzan.pdf (access date: 02.02.2023).
12. Advanced Data Structure and Algorithms. EXCEL BOOKS PRIVATE LIMITED, Phagwara Punjab, India, 2011.
13. Software engineering. URL: https://en.wikipedia.org/wiki/Software_engineering (access date: 03.02.2023).
14. Software construction. URL: https://en.wikipedia.org/wiki/Software_construction (access date: 03.02.2023).
15. Data processing. URL: https://en.wikipedia.org/wiki/Data_processing (access date: 03.02.2023).
16. Korotenko , G., & Korotenko , L. (2021). Paradigms of programming languages and the difficulty of organizing the Algorithms and Data Structures course. Technium: Romanian Journal of Applied Sciences and Technology, Vol. 3, No. 4, 25–37. Issue Vol. 3 No. 4 (2021): Sustainable Future and Technology Development. DOI: https://doi.org/10.47577/technium.v3i4
17. Korotenko G, Korotenko L. Formation of a Programming Languages Stack and a methodology of teaching to students specialized in Computer Science at Technical Universities in the context of interdisciplinarity. Technium Sustainability. Vol. 1 No. 1 (2021): Sustainability. Published: 2021-10-07. P. 21-33.
18. Algorithms and Data Structures course multicomponent complexity and interdisciplinary connections. Chapter-5 / Gregory Korotenko and Leonid Korotenko. Dnipro University of Technology, Ukraine / Educational Developments (Volume 3). Publisher: Bhopal, India: Innovare Academics Sciences Pvt Ltd. 2022. – 69-83.
19. R. Sedgewick, K. Wayne: Algorithms. Fourth Edition. Addison-Wesley, 2011.
20. Expressing an algorithm. URL: https://www.khanacademy.org/computing/ap-computer-science-principles/algorithms-101/building-algorithms/a/expressing-an-algorithm (access date: 03.02.2023).

## Transliteration of References:

1. D. Reinsel, J. Gantz, J. Rydning, The Digitization of the World From Edge to Core. URL: https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf.
2. G. L. McDowell, Cracking the Coding Interview. 189 Programming Questions and Solutions, 6th ed., CareerCup, LLC, Palo Alto, CA, 2016.
3. Computer Science Curricula 2013 (CS 2013); Curriculum Guidelines for Undergraduate Degree Programs in Computer Science, Association for Computing Machinery (ACM), IEEE Computer Society. URL: https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf.

4. Approved standards of higher education 121 Software engineering [Затверджені стандарти вищої освіти 121 Інженерія програмного забезпечення]. URL: https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/12/21/121-inzhener.programn.zabezp.bakalavr-1.pdf.
5. Approved standards of higher education 122 Computer science [Затверджені стандарти вищої освіти 122 Комп'ютерні науки]. URL: https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/2019/07/12/122-kompyut.nauk.bakalavr-1.pdf
6. Approved standards of higher education 123 Computer Engineering [Затверджені стандарти вищої освіти 123 Комп'ютерна інженерія]. URL: https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/123-kompyuterna-inzheneriya.pdf.
7. Approved standards of higher education 124 System analysis [Затверджені стандарти вищої освіти 124 Системний аналіз]. URL: https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty /12/21/124-sistemn.analiz-bakalavr-1.pdf.
8. Approved standards of higher education 125 Cyber security [Затверджені стандарти вищої освіти 125 Кібербезпека]. URL: https://mon.gov.ua/storage/app/uploads/public/5bb/626/1a8/5bb6261a84776166409164.pdf.
9. Approved standards of higher education 126 Information systems and technologies [Затверджені стандарти вищої освіти 126 Інформаційні системи і технології]. URL: https://mon.gov.ua/storage/app/media/vishcha-osvita/zatverdzeni%20standarty/12/21/126-inform.sist.tekhnol.bakalavr-1.pdf.
10. J. Lambert, C. Frye, Microsoft Office 2019 Step by Step, 1st ed, Microsoft Press, 2018.
11. The Ultimate Book of Mind Maps. URL: https://ia802601.us.archive.org/20/items/pdfy-MEuyCwZKGT3fqH56/ The%20Ultimate%20Book%20of%20Mind%20Maps%20-%20Tony %20Buzan.pdf.
12. Advanced Data Structure and Algorithms, Excel Books Private Limited, Phagwara Punjab, India, 2011.
13. Software engineering. URL: https://en.wikipedia.org/wiki/Software_engineering.
14. Software construction. URL: https://en.wikipedia.org/wiki/Software_construction.
15. Data processing. URL: https://en.wikipedia.org/wiki/Data_processing.
16. G. Korotenko, L. Korotenko. (2021). Paradigms of programming languages and the difficulty of organizing the Algorithms and Data Structures course. Technium. Romanian Journal of Applied Sciences and Technology 3.4: Sustainable Future and Technology Development (2021) 25–37. DOI: https://doi.org/10.47577/technium.v3i4
17. G. Korotenko, L. Korotenko. (2021). Formation of a Programming Languages Stack and a methodology of teaching to students specialized in Computer Science at Technical Universities in the context of interdisciplinarity. Technium Sustainability. 1.1 (2021) 21-33.

18. G. Korotenko, L. Korotenko. (2022). Algorithms and Data Structures course multicomponent complexity and interdisciplinary connections. Educational Developments Vol. 3, Innovare Academics Sciences Pvt Ltd, Bhopal, India, 2022, pp. 69-83.

*Authors*

Ghryghorij Korotenko
*Doctor of Engineering Sciences*
*Associate Professor,*
*Department «Information technology and Computer engineering»,*
*Dnipro University of Technology, Dnipro, Ukraine*
*https://orcid.org/0000-0003-3774-5260*
*E-mail: gkorotenko@gmail.com*
*korotenko.g.m@nmu.one*

Leonid Korotenko
*Candidate of Engineering Sciences,*
*Associate Professor,*
*Department «Computer Systems Software»,*
*Dnipro University of Technology*
*Dnipro, Ukraine*
*https://orcid.org/0000-0002-2236-0205*
*E-mail: leonid_korotenko@ukr.net*
*korotenko.l.m@nmu.one*

Aliona Khar
*Assistant, Depepartment «Software Engineering»,*
*Dnipro University of Technology,*
*Dnipro, Ukraine*
*https://orcid.org/0000-0003-3176-7792*
*E-mail: khar.a.t@nmu.one*

Artem Shyrin
*Candidate of Engineering Sciences, Associate Professor,*
*Department «Software Engineering»,*
*Dnipro University of Technology,*
*Dnipro, Ukraine*
*https://orcid.org/0000-0003-0026-2767*
*E-mail: shyrin.a.l@nmu.one*

*Abstracts*

ГРИГОРІЙ КОРОТЕНКО, ЛЕОНІД КОРОТЕНКО, АЛЬОНА ХАРЬ, АРТЕМ ШИРІН. **Шляхи покращення алгоритмічного розуміння методів сортування під час вивчення курсу «Алгоритми та структури даних» студентами університету.** *У статті розглянуто особливості викладання матеріалу при вивченні алгоритмів сортування даних в рамках курсу «Алгоритми і структури даних» та шляхи зниження рівня існуючих складнощів при його освоєнні. Активний розвиток програмного забезпечення та інформаційних систем в напрямку формування їх конвергентних та гіперконвергентних модифікацій призводить до вибухового зростання кількості взаємопов'язаних даних самої різної природи. Разом з тим, продовжують надходити повідомлення про постійну (катастрофічну) нестачу кваліфікованих фахівців у всіх сферах проникнення комп'ютингу (computing) і, в тому числі, в сфері обробки даних. В основі компетентностей IT-фахівців лежить здатність вирішувати алгоритмічні завдання, що не зустрічалися їм раніше. Одним з досить поширених завдань є необхідність сортування даних. Тому пропонується розглянути шляхи підвищення рівня алгоритмічного розуміння методів сортування в рамках курсу «Алгоритми і структури даних» та можливі засоби зниження рівня складнощів (reducing the complexity) при його вивченні.*

***Ключові слова:*** *алгоритми сортування, структури даних, обробка даних, програмування, таблично-графічна форма уявлення алгоритмів, метод Mind Maps.*

HRYHORII KOROTENKO, LEONID KOROTENKO, ALIONA HAR, ARTEM SZYRIN. **Drogi doskonalenia algorytmicznego rozumienia metod sortowania w trakcie przejścia przez studentów kursu „Algorytmy i struktury danych".** *W artykule przeanalizowano specyfikę nauczania materiału podczas nauki algorytmów sortowania danych na kursie „Algorytmy i struktury danych" oraz sposoby zmniejszenia poziomu istniejących trudności w jego przyswajaniu. Aktywny rozwój oprogramowania i systemów informatycznych w kierunku tworzenia ich konwergentnych i hiperkonwergentnych modyfikacji prowadzi do gwałtownego wzrostu liczby powiązanych ze sobą danych o najróżniejszym charakterze. Jednocześnie wciąż pojawiają się informacje o permanentnym (katastrofalnym) niedoborze wykwalifikowanych*

*specjalistów we wszystkich obszarach informatyki (computing), w tym dziedzina przetwarzania danych. Podstawą kompetencji informatyków jest umiejętność rozwiązywania zadań algorytmicznych, z którymi wcześniej się nie spotkali. Jednym z dość powszechnych zadań jest konieczność sortowania danych. Dlatego sugeruje się rozważenie sposobów zwiększenia poziomu algorytmicznego rozumienia metod sortowania w ramach przedmiotu „Algorytmy i struktury danych" oraz możliwych sposobów zmniejszenia poziomu złożoności (reducing the complexity) podczas jego studiowania.*

***Słowa kluczowe:*** *algorytmy sortowania, struktury danych, przetwarzanie danych, programowanie, tabelaryczne i graficzne przedstawienie algorytmów, metoda Mind Maps.*