**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**
**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ**
**ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

**ЗАТВЕРДЖЕНО**
на засіданні кафедри
інформаційних систем.
Протокол № 1 від 22.08.2023 р.

**ПОГОДЖЕНО**
Проректор з навчально-методичної роботи

_______________ Каріна НЕМАШКАЛО

# АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ

**робоча програма навчальної дисципліни (РПНД)**

| | |
|---|---|
| Галузь знань | **12 "Інформаційні технології"** |
| Спеціальність | **121 "Інженерія програмного забезпечення "** |
| Освітній рівень | **перший (бакалаврський)** |
| Освітня програма | **"Інженерія програмного забезпечення"** |

| | |
|---|---|
| Статус дисципліни | **обов'язкова** |
| Мова викладання, навчання та оцінювання | **англійська** |

Розробник:
д.п.н, професор підписано КЕП Людмила ГРИЗУН

Завідувач кафедри
інформаційних систем _______________ Дмитро БОНДАРЕНКО

Гарант програми _______________ Олег ФРОЛОВ

**Харків**
**2024**

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
SIMON KUZNETS KHARKIV NATIONAL UNIVERSITY OF ECONOMICS

**APPROVED**
at the meeting of the department
information systems
Protocol № 1 of 22.08.2023

**AGREED**
Vice-rector for educational and methodical work

________________ Karina NEMASHKALO

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ СЕМЕНА КУЗНЕЦЯ
№02071211

# ALGORITHMS AND DATA STRUCTURES

## Program of the course

| | |
|---|---|
| Field of knowledge | **12 "Information Technology"** |
| Specialty | **121 "Software engineering"** |
| Study cycle | **first (bachelor)** |
| Study programme | **"Software Engineering"** |

| | |
|---|---|
| Course status | **mandatory** |
| Language | **English** |

Developers:

| | | |
|---|---|---|
| Doctor (Pedagogical sciences), Professor | digital signature | Liudmyla GRYZUN |
| Head of Information systems department: Ph.D. (Technical sciences), associate professor | ________________ | Dmytro BONDARENKO |
| Head of Study Programme: Ph.D. (Technical sciences), associate professor | ________________ | Oleg FROLOV |

**Kharkiv**
**2024**

# INTRODUCTION

Wide spread of information technologies, scientific and technical progress, penetration of information and communication technologies into all spheres of human activity put forward new, increased requirements for the training of specialists in the field of information technologies. A modern professional in this field must possess a number of competencies, among which general scientific and general technical competencies and fundamental knowledge occupy a special place.

The course "Algorithms and data structures" considers such issues as the formalization of the concepts of "algorithm", "algorithm complexity" and the study of formal algorithmic systems; general principles of building effective algorithms; modern methods of algorithm research and analysis; methods and mechanisms of implementation of effective algorithms in specific applications; classification of tasks, definition and study of classes of complexity; asymptotic analysis of algorithm complexity; obtaining explicit functions of labor intensity for comparative analysis of algorithms; development of criteria for comparative evaluation of the quality of algorithms; a variety of effective data structures and the peculiarities of their application in the implementation of various algorithms.

The purpose of studying the course "Algorithms and Data Structures" is to obtain thorough mathematical training and knowledge of the theoretical, methodical and algorithmic foundations of information technologies for their use when solving applied and scientific tasks in the field of information systems and technologies, providing theoretical and engineering training of specialists in the field of design, implementation and use of information systems in business; familiarization with modern and effective algorithms of computer information processing, as well as methods of their research and analysis.

The task of the said course is to form students a systematized idea of the concepts of the theory of algorithms, methods and means of solving applied algorithmic problems, as well as to acquire practical skills in algorithmization and programming, necessary for the effective design of information systems.

The object of the educational componnet is the study of data structures and typical algorithms for working with them.

The subject of the course is the theoretical concepts of building and analyzing algorithms, as well as the use of various data structures for solving applied algorithmic problems.

The learning outcomes and competencies formed by the course are defined in the table. 1.

Table 1

**Learning outcomes and competencies formed by the course**

| Learning outcomes | Competencies |
|---|---|
| LO 13 | GC 01, GC 02, GC 02, SC 08, SC 14 |
| LO 25 | GC 01, GC 02, SC 14 |

where, LO 13. Know and apply methods of developing algorithms, designing software and data and knowledge structures.

LO 25. Have the skills to develop software taking into account distributed data processing, parallel calculations on several processors, cores and with the use of graphic adapters (accelerators).

GC 01. Ability to abstract thinking, analysis and synthesis.

GC 02. Ability to apply knowledge in practical situations.

SC 02. Ability to participate in software design, including modeling (formal description) of its structure, behavior and functioning processes.

SC 08. Ability to apply fundamental and interdisciplinary knowledge to successfully solve software engineering tasks.

SC 14. Ability to algorithmic and logical thinking.

# COURSE CONTENT

## Content module 1. Analysis of algorithms and algorithmic strategies

### Topic 1. Concept of algorithm. Basic properties of algorithms.

The purpose and tasks of the course, its place in the educational process. The structure of the course, recommendations for its study. Organizational and methodological support of the course.

Algorithm. Intuitive concept of algorithm. Recording methods and properties of algorithms. The need to clarify the concept of algorithm. Requirements for the general algorithmic model.

Practical methods of developing the simplest algorithms. The concept of algorithm complexity. Analysis of algorithms regarding their complexity.

Algorithm development methods: structural programming, recursion, tree traversal, “divide and rule”, balancing, dynamic programming, backtracking, branch-and-bound method, heuristic and approximate algorithms.

### Topic 2. Algorithms for work with integers

The relevance of studying and applying integer processing algorithms. The greatest common devisor. Least common multiple. Euclid's algorithm.

Eratosthenes Sieve. Sundaram Sieve. Atkin sieve. Testing numbers for simplicity.

Basic theorem of arithmetic. Decomposing a number into prime factors.

Euler's function.

### Topic 3. Search and sorting algorithms

Search algorithms, problems of their implementation, analysis of complexity.

The importance of sorting when implementing algorithms. Sorting classification. Sorting characteristics. Simple sorting as a way to quickly implement the algorithm. Examples of simple sorts (simple inclusion method, simple exchange method (bubble sort), shaker sort, insertion sort, counting sort). Advantages and disadvantages of simple sorting.

Complex sorting as a way to create efficient algorithms. Examples of complex sorts (Shell sort, Hoare sort (quick sort), merge sort). Advantages and disadvantages of complex sorting. Comparison of simple and complex sorting.

## Content module 2. Fundamental data processing algorithms

### Topic 4. Basic data structures. Features and implementation

Data type concept. Classifications of data types. Methods of presenting the data structure in computer memory. Types of memory and features of data presentation methods in different types of memory.

Simple and complex data structures. Characteristics of arrays, structures and lists.

Stacks, queues, and linked lists. Peculiarities of their implementation.

### Topic 5. Fundamental algorithms on graphs

Basic concepts of graph theory. Matrix representation of the graph. Connectivity matrix and distance matrix on a graph.

Search for optimal routes in the graph. Dijkstra's algorithm. Bellmon's algorithm.

Finding the minimum spanning tree of a graph using the Prim-Kraskal algorithm. The problem of finding Hamiltonian paths.

### Topic 6. Combinatorial and recursive algorithms

Basic concepts of combinatorics. The concept of a combinatorial problem. Permutations. Counting the number of possible permutations. Organization of 10 permutations.

Accommodation and connections. Counting the quantity. Organization of finding all possible accommodations and connections. Methods of organizing a complete search.

Method of branches and boundaries. Limitation of search options. Search algorithms with return. The problem of placing parentheses.

The list of laboratory studies in the course is given in table 2.

Table 2

**The list of laboratory studies**

| Name of the topic and/or task | Content |
|---|---|
| Topic 1. Laboratory study 1. | Mathematical analysis of algorithm complexity. |
| Topic 2. Laboratory study 2. | Integer processing algorithms. |
| Topic 3. Laboratory study 3.<br>Topic 3. Laboratory study 4. | Array search algorithms.<br>Array sorting algorithms . |
| Topic 4. Laboratory study 5. | Stacks, queues, linear lists. Features of implementation. |
| Topic 5. Laboratory study 6. | Graphs and their processing algorithms. |
| Topic 6. Laboratory study 7. | Combinatorial algorithms. |

The letter of self-studies in the course is given in table 3.

Table 3

## List of self-studies

| Name of the topic and/or task | Content |
|---|---|
| Topic 1. | Processing of lecture material. The role of algorithmization and programming in the design of information systems. Basic strategies for developing algorithms. Mathematical foundations of algorithm complexity assessment. Preparation for the laboratory session. |
| Topic 2. | Processing of lecture material. The role of prime number processing algorithms in cryptography. Problems of generating large prime numbers. Preparation for laboratory classes. |
| Topic 3. | Підготовка до лабораторних занять. Processing of lecture material. Classification of sorting algorithms. Implementation of various algorithm development strategies when implementing search and sorting algorithms. Basic approaches to solving problems by the method of dynamic programming. Preparation for laboratory classes. |
| Topic 4. | Processing of lecture material. Classifications of data types. Methods of presenting the data structure in computer memory. Hash tables and hash functions, their role in cryptography. Preparation for laboratory classes. |
| Topic 5. | Processing of lecture material. Problems of graph storage as a data structure and its visualization. The applied value of graph theory and its traversal algorithms. Preparation for laboratory classes. Development of individual projects. |
| Topic 6. | Processing of lecture material. Advantages and disadvantages of recursive algorithms. Features of implementation of combinatorial algorithms. Basic search algorithms in strings. String processing algorithms. Preparation for laboratory classes. Development of individual projects. |

The number of hours of lecture and laboratory studies and hours of self-study is given in the technological card of the course..

# TEACHING METHODS

In the process of teaching the course, in order to acquire certain learning outcomes, to activate the educational process, it is suggested using the following teaching methods:

Verbal (lecture (Topic 1, 2), problem-based lecture (Topic 3 - 6)).

Visual (demonstration (Topic 1 - 6)).

Practical (laboratory work (Topic 1 - 6).

Mini-conference with discussion of current problems of the course and defense of individual projects (Topic 5-6).

# FORMS AND METHODS OF ASSESSMENT

The University uses a 100-point cumulative system for assessing the learning outcomes of students.

**Current control** is carried out during lectures, laboratory classes and is aimed at checking the level of readiness of the student to perform a specific job and is evaluated by the amount of points scored:

– for courses with a form of semester control as an exam: maximum amount is 60 points; minimum amount required is 35 points.

**The final control** includes current control and an exam.

**Semester control** is carried out in the form of a semester exam.

***The final grade in the course*** is determined:

– for courses with a form of exam, the final grade is the amount of all points received during the current control and the exam grade.

During the teaching of the course, the following control measures are used:

Current control:

defense of laboratory work (50 points);

defense of individual project (10 points).

Semester control: Grading including Exam (40 points)

## An example of an examination card

Simon Kuznets Kharkiv National University of Economics
First (bachelor) level of higher education
Specialty "Software Engineering"
Study programme "Software engineering"
Semester III
Course "Algorithms and data structures"

### EXAM CARD

Task 1. Testing integers for simplicity: conduct comparative characteristics of simplicity tests.

Task 2. Develop a program that provides input from a file in the form of a graph in the form of an adjacency matrix, displaying the graph on the screen and searching in depth. Explain the idea of such a search and the features of its implementation. Run the program in step-by-step mode for a graph with 5 vertices. Illustrate on the appropriate geometric graph.

Protocol No. ____ dated "___"__________20___ was approved at the meeting of the Department of Information Systems

| | |
|---|---|
| Examiner | Doc.Ped.Sciences, Professor Gryzun L. |
| Chief department | PhD, Associate Professor Bondarenko D. |

### Assessment criteria

The final points for the exam consist of the sum of the points for the completion of all tasks, rounded to a whole number according to the rules of mathematics.

The exam card consists of two tasks to test knowledge of the basics of algorithmization and the ability to solve practical tasks related to the application of basic algorithms, analysis of their complexity, and the use of appropriate data structures.

The structure of the examination card is built according to the following example.

1. Give a detailed answer to the theoretical question.

2. Solve an algorithmic problem using the specified algorithm and (or) a certain data structure.

The duration of the exam is 90 minutes, while the estimated time for preparing answers to individual questions is as follows: task 1 – 45 minutes, task 2 – 45 minutes.

The answer to the theoretical question should be clear and detailed, with relevant practical examples.

The solution to the problem must contain an explanation of the used algorithm (its description), the features of its application, an analysis of its complexity, etc.; program code; a screenshot with the code and test results of the program; analysis of results; conclusions.

The evaluation of the exam result is formed according to the following rule. Each task of the examination card is evaluated for a maximum of 20 points. The number of points obtained from the answers to each task of the examination card is summed up.

20-18 points - awarded for in-depth knowledge of the educational material of the course contained in the main and additional recommended literary sources, the ability to analyze the algorithms being studied, in their relationship and development, to give apt examples, to answer clearly, succinctly, logically and consistently to the questions asked, the ability to apply theoretical provisions when solving practical problems;

17-15 points - awarded for solid knowledge of the educational material of the course, including calculations, reasoned answers to the questions, giving apt examples, the ability to apply theoretical provisions when solving practical problems;

14-11 points - awarded for solid knowledge of the educational material of the course, including giving examples, reasoned answers to the questions, which, however, contain insignificant inaccuracies, for the ability to apply theoretical provisions when solving practical problems;

10-8 points are awarded for sufficient knowledge of the educational material of the the course, answers to the questions and solving the problem, which, however, contain certain inaccuracies;

7-5 points - awarded for mediocre knowledge of the educational material of the course, poorly reasoned answers, mediocre application of theoretical provisions when solving practical problems, but there is significant progress in solving practical problems;

4-2 points - awarded for weak knowledge of the educational material, inaccurate or poorly reasoned answers, with a violation of the sequence of its presentation for weak application of theoretical provisions when solving practical problems;

1-0 points - awarded for ignorance of a significant part of the educational material, significant mistakes in answering questions, inability to apply theoretical provisions when solving practical problems.

As a result of such calculation, the applicant may receive from 0 to 40 points for two tasks on the exam.

# RECOMMENDED LITERATURE

## Main

1. Матвієнко М.П. Алгоритми та структури даних: навчальний посібник. / М. П. Матвієнко. – Київ: Видавництво Ліра-К, 2018. — 340 с.

2. Основи алгоритмізації. Методичні рекомендації до виконання лабораторних робіт для студентів спеціальності 121 "Інженерія програмного забезпечення" освітньої програми "Інженерія програмного забезпечення" першого (бакалаврського) рівня [Електронний ресурс] / уклад. О. В. Щербаков, О. В. Фролов; Харківський національний економічний університет ім.

С. Кузнеця. — Електрон. текстові дан. (3,95 МБ). — Харків : ХНЕУ ім. С. Кузнеця, 2023. — 67 с. : іл. — Загол. з титул. екрану. — Бібліогр.: с. 65-66. – Режим доступу : http://repository.hneu.edu.ua/handle/123456789/29584

3. Щербаков О. В. Порівняльний аналіз алгоритмів та методів сортування в сучасних мовах програмування / О. В. Щербаков, Ю. І. Скорін // Проблеми і перспективи розвитку ІТ-індустрії: матеріали міжнар. науково-практ. конф., 19 – 20 квітня 2018 р. : тези допов. – Х.: ХНЕУ ім. Семена Кузнеця, 2018. – С. 31. – Режим доступу : http://repository.hneu.edu.ua/handle/123456789/16453

4. Щербаков О. В. Порівняльний аналіз алгоритмів та методів сортування в сучасних мовах програмування / О. В. Щербаков, Ю. І. Скорін // Проблеми і перспективи розвитку ІТ-індустрії: матеріали міжнар. науково-практ. конф., 19 – 20 квітня 2018 р. : тези допов. – Х.: ХНЕУ ім. Семена Кузнеця, 2018. – С. 31. – Режим доступу : http://repository.hneu.edu.ua/handle/123456789/19019

### Additional

5. Bilousova L., Gryzun L., Zhytienova N., Pikalova V. (2019) Search algorithms learning based on cognitive visualization. ICT in Education, Research, and Industrial Applications: Integration, Harmonization, and Knowledge Transfer. Conference proceedings (2387). P. 472-478. ISSN 1613-0073. URL: http://ceur-ws.org/Vol-2387/20190472.pdf.

6. Gryzun, L., Shcherbakov, O., Parfonov, Y., & Bodnar, L. (2022). Visualization of algorithms on graphs with a large number of vertices: The features of applications design. Development management, 20(4), 36-44. http://doi.org/ 10.57111/devt.20(4).2022.36-44. – Режим доступу : https://devma.com.ua/uk/journals/t-20-4-2022/vizualizatsiya-algoritmiv-na-grafakh-z-velikoyu-kilkistyu-vershin-osoblivosti-proyektuvannya-zastosunkiv

7. Gryzun L., Shcherbakov O. and S. Lytvynova Computer modeling of the tournament of game algorithms in the process of learning of basics of algorithmization and programming by pre-service IT-specialists /L. Gryzun, O. Shcherbakov and S. Lytvynova // Kiv, A.E., Semerikov S.O., Shyshkina, M.P. (Eds.): Cloud Technologies in Education. Proceedings of the 9 th Workshop CTE 2021, Kryvyi Rih, Ukraine, December 17, 2021, CEUR-WS.org, Vol-3085 pp 28-38. URL: http://ceur-ws.org/Vol-3085/paper14.pdf

### Information resources

8. Алгоритми та структури даних (121) ПНС курс - Режим доступа: https://pns.hneu.edu.ua/course/view.php?id=10173

9. Algorithms Fundamentals. E-resourse. - available at: https://brilliant.org/courses/computer-science-algorithms/

10. Bhargava A. Y Grokking Algorithms - An illustrated guide for programmers and other curious people, 2016, available at: https://edu.anarcho-copy.org/Algorithm/grokking-algorithms-illustrated-programmers-curious.pdf

11. Algotester Online Course [Електронний ресурс]. – Режим доступа: https://www.youtube.com/watch?v=25wE3dBKx8s

12. Introduction to algorithms / Thomas H. Cormen . . . [et al.].—3rd ed, https://www.inf.ufpr.br/andre/textos-CI1165/Introduction%20to%20Algorithms%20-%203rd%20Edition.pdf

13. Knuth, D E The Art of Computer Programming (TAOCP). Vol. 1.Retrieved May 20, 2012, available at: https://cs.stanford.edu/~knuth/taocp.html

Sorting Algorithm Animations [Електронний ресурс]. – Режим доступа: http://www.sorting-algorithms.com.