Syllabus
Course Program

Computer discrete mathematics

Specialty
113 Applied mathematics

Institute
Computer science and information technology.

Educational program
Intelligent Data Analysis

Department
Computer mathematics and data analysis.

Level of education
Bachelor's level

Course type
Special (professional), Mandatory .

Semester
2

Language of instruction
Ukrainian

Lecturers and course developers

Oleksii Haluza

oleksii.haluza@khpi.edu.ua
Doctor of Science (Physics&Mathematics), Full Professor, Professor of Computer Mathematics and Data Analysis Department.

Work experience – more than 20 years. The author of many scientific, educational, and methodological works. Leading lecturer in the courses: «Higher mathematics », «Optimization Methods», «Machine Learning», etc.

More about the lecturer on the department's website

Oleh Tonitsa

oleh.tonitsa@khpi.edu.ua
Candidate of Science (Physics&Mathematics), Associate professor, Associate professor of Computer Mathematics and Data Analysis Department.

Work experience – more than 20 years. The author of many scientific, educational, and methodological works. Leading lecturer in the courses: «Algorithmization and Programming», «Optimization Methods», «Machine Learning», etc.

More about the lecturer on the department's website

General information

Summary

The basic concepts of discrete mathematics, in particular, graph theory and issues related to the development of algorithms and complex data structures, as well as their implementation using the C++ programming language, are considered

Course objectives and goals

The discipline is aimed at studying and mastering the theoretical foundations of discrete mathematics and basic algorithmic constructions and types, and methods of their implementation in the C++ language. Methods of constructing complex programs are considered. Particular attention is paid to typical complex dynamic data structures and algorithms for working with them, as well as methods for their implementation in the C++ language. Particular attention is paid to the study of techniques and methods for the development of effective applications in the C++ language and the formation of students' theoretical knowledge and practical skills in using computers in engineering activities.

Format of classes

Lectures, practical classes, consultations, self-study. Final control in the form of an exam.

Competencies

GC 1. Ability to learn and master modern knowledge.
GC 2. Ability to apply knowledge in practical situations.
GC 3. Ability to generate new ideas (creativity).
GC 4. Ability to be critical and self-critical.
GC 6. Ability to abstract thinking, analysis, and synthesis.
GC 7. Ability to search, process and analyze information from various sources.
GC 8. Knowledge and understanding of the subject area and understanding of professional activities.
GC 10. Skills in the use of information and communication technologies.
SC 2. Ability to perform tasks formulated in mathematical form.
SC 3. Ability to choose and apply mathematical methods for solving applied problems, modeling, analysis.
SC 5. Ability to develop algorithms and data structures, software tools, and program documentation.
SC 7. Ability to solve professional problems with the help of computer equipment, computer networks and the Internet, in the environment of modern operating systems, using standard office applications.
SC 14. Ability to understand the statement of the task, formulated in the language of a certain subject area, to search and collect the necessary initial data.
SC 15. Ability to formulate a mathematical formulation of the problem, based on the statement in the language of the subject area, and choose a method for its solution, which provides the necessary accuracy and reliability of the result.

Learning outcomes

PH 1. Demonstrate knowledge and understanding of the basic concepts, principles, theories of applied mathematics and use them in practice.
PH 3. Formalize tasks formulated in the language of a particular subject area; formulate their mathematical formulation and choose a rational method of solution; solve the obtained problems by analytical and numerical methods, evaluate the accuracy and reliability of the results obtained.
PH 11. Be able to apply modern technologies of programming and software development, software implementation of numerical and symbolic algorithms.
PH 14. Demonstrate the ability to self-learn and continue professional development.
PH 15. Be able to organize your own activities and get results within a limited time.

Student workload

The total volume of the course is 180 hours (6 ECTS credits): lectures – 42 hours, practical classes – 32 hours, laboratory classes – 16 hours, self-study – 90 hours.

Course prerequisites

School courses of mathematics and computer science, discipline "Algorithmization and programming".

Features of the course, teaching and learning methods, and technologies

When teaching this discipline, such methods of learning and teaching as gamification and peer-to-peer are used. LMS (learning management systems) are used in the learning process.

Program of the course

Topics of the lectures

Topic 1. Recursion.
The concept of recursion. Recursion implementation mechanism.
Topic 2. Composite data types.
Structure. Union. Enumeration.
Topic 3. Files.
Structure of text files. Input/output of textual information to the file. Binary file structure. Input/output of information into a binary file. Formatted input and output to the stream.
Topic 4. Classification of data structures.
Definition of the data structure; An overview of the main types of data structures.
Topic 5. Linear data structures.
Singly linked lists. Double-linked lists. Cyclic lists. Stack. Queue. Deck.
Topic 6. Trees.
Basic concepts about trees. Classification of trees. Methods of representing trees in memory. Basic algorithms for working with trees. Methods and algorithms for bypassing trees.
Topic 7. Linear and binary search. Binary search trees.
Linear search. Binary search and its properties. Binary search trees. Basic algorithms for working with binary search trees.
Topic 8. Basic Concepts of Graph Theory.
Graph definition. History of Graph Theory. Tasks of graphs. Graphs with numerical characteristics of edges. Graph connectivity. The concept of a tree, a forest. A tree of minimum weight.
Topic 9. Operations on graphs.
Addition and multiplication of graphs. Joining, intersecting and unary operations on graphs.
Topic 10. Transitive closures.
Reachability and counter-reach.
Topic 11. Types of graphs.
Types of graphs: complete, symmetric, antisymmetric, planar, dicotyledonous, tree. Types of subgraphs. Connectivity components.
Topic 12. Splitting the graph.
Methods for splitting a graph into maximum strongly connected components. Malgrange method. Matrix method.
Topic 13. Paths and cycles in graphs.
Topic 14. The shortest paths in the graph.
Dijkstra's algorithm for finding the shortest paths in a graph.
Topic 15. Independent vertex sets.
Constructing independent vertex sets. Search with returns.
Topic 16. Dominant sets.
Planarity, graph stacking. The five-color theorem.
Topic 17. Network.
Streams in networks. Algorithm for finding the maximum flow. Ford-Falkerson theorem.
Topic 18. Multipole maximum flow.
The multipole maximum flux problem. Gomori-Hu algorithm.
Topic 19. Application of methods and algorithms of graph theory for solving applied problems.

Topics of the workshops

Topic 1. Lists
Topic 2. Stack. Queue. Deck.
Topic 3. Basic algorithms for working with trees.
Topic 4. Methods and algorithms for bypassing trees.
Topic 5. Basic algorithms for working with binary search trees.
Topic 6. Balancing binary search trees.
Topic 7. Priority queue.
Topic 8. Methods for representing graphs in memory.

Topic 9. Exploring the ways of defining graphs. The sum and product of the graphs. Joining and intersecting graphs. Circular sum of graphs. Unary operations: removal of a vertex or edge, closure, contraction.
Topic 10. Search for forward and backward transitive closures. Building reachability and counter-reach matrices. Matrix method of finding paths in graphs.
Topic 11. Determination of components of strong cohesion.
Topic 12. The Malgrange method is the partitioning of a graph into maximum strongly connected components. Matrix method.
Topic 13. Solving problems using the concepts of route, chain and cycle. The shortest paths in the graph.
Topic 14. Problems that use the construction of independent vertex sets. Search with returns. Constructing dominant sets. Methods and algorithms for constructing a minimum connecting tree and paths.
Topic 15. Algorithm for searching for databases and antibases of a graph. Algorithms for traversing graphs in depth and by levels. Solving the problem of maximum flow and minimum section.
Topic 16. Graphs with many sources and drains. Search for multipole maximum flux. Solving typical problems using graph theory.

Topics of the laboratory classes

Topic 1. Recursion
Topic 2. Composite data types
Topic 3. Working with text files
Topic 4. Working with Binary Files
Topic 5. Linear data structures
Topic 6. Tree Bypass Methods
Topic 7. Binary search
Topic 8. Binary Search Trees

Self-study

The course involves the performance of individual tasks, the result of which is checked automatically by LMS tools and controlled and evaluated by teachers. Students are also recommended additional materials (videos, articles) for self-study.

Course materials and recommended reading

Basic literature

1. Кормен Т., Лейзерсон Ч., Рівест Р., Стайн К. Вступ до алгоритмів. - К.: К. І. С., 2019. - 1288 с. ISBN 978-617-684-239-2
2. Новотарський М. А.. Дискретна математика – Київ : КПІ ім. Ігоря Сікорського, 2020. – 278 с.
3. Коцовський В. М. Основи дискретної математики: навч. посіб. – Ужгород : ПП «АУТДОР-ШАРК», 2020. – 128 с.
4. Кублій Л. І. Комп'ютерна дискретна математика – Київ КПІ ім. Ігоря Сікорського 2020, –165 с.
5. Sriraman Sridharan, R. Balakrishnan. Discrete Mathematics Graph Algorithms, Algebraic Structures, Coding Theory, and Cryptography. – Chapman and Hall/CRC, 2019. – 340 p.
6. Ryan T. White , Archana Tikayat Ray. Practical Discrete Mathematics. – Packt, 2021. – 330 p.

Additional literature

1. James A. Anderson , Jerome Lewis, O. Dale Saylor . Discrete Mathematics With Combinatorics. – Prentice Hall; Subsequent edition, 2003. – 960 p.
2. Sarah-Marie Belcastro. Discrete Mathematics with Ducks. – Chapman and Hall/CRC, 2020. – 700 p.

Internet resources

1. https://www.hackerrank.com - база задач з програмування
2. https://projecteuler.net - база алгоритмічних задач

Assessment and grading

Criteria for assessment of student performance, and the final score structure

Description of the final score structure, course requirements, and necessary steps to earn points, especially paying attention to self-study and individual assignments.

Grading scale

Total points	National	ECTS
90–100	Excellent	A
82–89	Good	B
75–81	Good	C
64–74	Satisfactory	D
60–63	Satisfactory	E
35–59	Unsatisfactory (requires additional learning)	FX
1–34	Unsatisfactory (requires repetition of the course)	F

Norms of academic integrity and course policy

The student must adhere to the Code of Ethics of Academic Relations and Integrity of NTU «KhPI»: to demonstrate discipline, good manners, kindness, honesty, and responsibility. Conflict situations should be openly discussed in academic groups with a lecturer, and if it is impossible to resolve the conflict, they should be brought to the attention of the Institute's management.
Regulatory and legal documents related to the implementation of the principles of academic integrity at NTU «KhPI» are available on the website: http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/

Approval

Approved by	Date, signature 29.08.2024		Head of the Department Olena AKHIIEZER
	Date, signature 29.08.2024		Guarantor of the Educational Program Olena AKHIIEZER