

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики
Кафедра прикладної математики

Оновлено та затверджено
на засіданні
кафедри прикладної математики
факультету прикладної математики та
інформатики
Львівського національного університету
імені Івана Франка
(протокол № 5 від 19.11 2023 р.)



Завідувач кафедри

Юрій ЯЩУК

Силабус з навчальної дисципліни
«Алгоритми та структури даних (англійською мовою)»,
що викладається в межах ОНП Прикладна математика
другого (магістерського) рівня вищої освіти для здобувачів
з спеціальності 113 – прикладна математика

Львів 2023 р.

Назва дисципліни	Алгоритми та структури даних (англійською мовою)
Адреса викладання дисципліни	Головний корпус ЛНУ ім. І. Франка м. Львів, вул. Університетська 1
Факультет та кафедра, за якою закріплена дисципліна	Факультет прикладної математики та інформатики Кафедра прикладної математики
Галузь знань, шифр та назва спеціальності	11 – математика та статистика 113 – прикладна математика
Викладачі дисципліни	Білецький Василь, кандидат фізико-математичних наук, доцент кафедри прикладної математики
Контактна інформація викладачів	vasyl.biletskyi@lnu.edu.ua ; https://ami.lnu.edu.ua/employee/biletskyj Головний корпус ЛНУ ім. І. Франка, каб. 278 м. Львів, вул. Університетська, 1
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекцій/лабораторних занять (за попередньою домовленістю)
Сторінка курсу	https://ami.lnu.edu.ua/course/alhorytmy-ta-struktury-danykh-anhliyskoiu-movoju-trykkladna-matematyka-1-9
Інформація про дисципліну	Дисципліна «Алгоритми та структури даних» є дисципліною на вибір зі спеціальності 113 – прикладна математика для освітньо-наукової програми «Прикладна математика», яка викладається в 1-му семестрі в обсязі 6-и кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	Вивчення дисципліни «Алгоритми та структури даних» сформує у студентів уявлення про загальні принципи вирішення алгоритмічно складних задач, що передбачають використання та поєднання алгоритмів і структур даних. Курс розділений на 16 модулів. На лабораторних заняттях студенти розв'язують практичні задачі, які є складними та передбачають декілька можливих рішень та їхніх реалізацій. Тому виконання завдань потребує оцінки ефективності потенційних рішень, вибору та реалізації оптимального з них. Розв'язання практичної задачі передбачає написання коду програми на одній з об'єктно-орієнтованих мов програмування, використовуючи необхідні структури даних, та подальше тестування розв'язку задачі.
Мета та цілі дисципліни	Метою є формування у студентів цілісного уявлення про загальні принципи вирішення алгоритмічно складних задач. Цілі – сформулювати у студентів систему знань щодо принципів вирішення алгоритмічно складних задач, використання та поєднання алгоритмів та структур даних.
Література для вивчення дисципліни	<ol style="list-style-type: none"> 1. Кормен Т. Вступ до алгоритмів / Томас Кормен, Чарльз Лейзерсон, Рональд Рівест, Кліффорд Стайн. — К.І.С., 2019 2. Sussman, Gerry, Harold Abelson, and Julie Sussman. Structure and interpretation of computer programs — MIT Press, 2022. 3. Cormen, Thomas H. Algorithms unlocked. — Mit Press, 2013 4. Donald, E. Knuth. The art of computer programming. — Addison-Wesley 5. Skiena, Steven S. The algorithm design manual: Text. Vol. 1. — Springer Science & Business Media, 1998 6. Kao, Ming-Yang. Encyclopedia of algorithms. — Springer Science & Business Media, 2008. 7. Sedgewick, Robert, and Philippe Flajolet. An introduction to the analysis of algorithms. — Addison-Wesley, 2013.

Обсяг курсу	Загальний обсяг: 180 годин. Аудиторних занять: 64 год., з них 32 години лекцій та 32 години лабораторних робіт. Самостійної роботи: 116 год.					
Очікувані результати навчання	У результаті вивчення даного курсу студент повинен <ul style="list-style-type: none"> • знати: принципи вирішення алгоритмічних задач та необхідні для цього структури даних; • вміти: поєднувати алгоритми та структури даних для вирішення практичних задач, аналізувати характеристики практичних задач та оцінювати ефективність можливих шляхів вирішення, реалізувати розв'язок практичної задачі. 					
Ключові слова	алгоритм, структури даних, обчислювальна складність					
Формат курсу	очний					
Теми	Подано нижче у таблиці Схема курсу “Алгоритми та структури даних”					
Підсумковий контроль, форма	Залік					
Пререквізити	Для вивчення дисципліни студенти потребують базових знань з <ul style="list-style-type: none"> • основ програмування • програмного забезпечення 					
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Лекції, презентації, модульний контроль. Домашні та індивідуальні завдання					
Необхідне обладнання	Комп'ютер із програмним забезпеченням Visual Studio Code, доступ до інтернету.					
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	Оцінювання проводиться за 100-бальною шкалою.					
	Оцінка за шкалою ECTS		Оцінка в балах	Оцінка за національною шкалою		
				Екзамен, диференційований залік	залік	
	A	Відмінно	100 - 90	Відмінно	5	зараховано
	B	Дуже добре	81- 89	Добре	4	
	C	Добре	71 -80			
	D	Задовільно	61 - 70	Задовільно	3	
	E	Достатньо	51- 60			
FX (F)	Незадовільно	0 - 50	Незадовільно	2	не зараховано	
<p>Впродовж семестру студент може отримати 100 балів. З них:</p> <ul style="list-style-type: none"> • індивідуальні завдання (50 балів, 5 завдань по 10 балів кожне) • контрольні тести на платформі https://algotester.com/ (50 балів) <p>Загалом протягом семестру 100 балів.</p> <p>Критерії оцінювання індивідуальних завдань:</p> <p>10 балів – студент повністю виконав завдання, самостійно підібрав структури даних та реалізував необхідні алгоритми, впевнено відповідає на всі пов'язані запитання під час захисту;</p> <p>7-9 балів – студент повністю виконав завдання з незначними неточностями, використані алгоритми та структури даних оптимальні або близькі до таких, відповідає на більшість</p>						

	<p>запитань, які пов'язані з відповідною тематикою;</p> <p>4-6 балів – студент в основному виконав завдання, отримав рекомендації щодо використання тих чи інших структур даних та алгоритмів, демонструє посередні знання, відповідає на певні пов'язані запитання;</p> <p>1-3 бали – студент не повністю виконав завдання, використовує неоптимальні алгоритми та структури даних, демонструє слабкі знання, відповідає лише на окремі запитання;</p> <p>0 балів – студент не виконав завдання і під час захисту не може відповісти на жодне запитання за відповідною тематикою.</p> <p>Критерії оцінювання контрольних тестів: пропорційно до кількості завдань, котрі студент успішно здав протягом тесту на платформі https://algotester.com/</p> <p>Академічна доброчесність: Очікується, що роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.</p> <p>Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Студенти мають інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися усіх строків визначених для виконання усіх видів письмових робіт та індивідуальних завдань, передбачених курсом.</p> <p>Література. Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.</p> <p>Політика виставлення балів. Враховуються бали набрані за індивідуальні завдання та контрольні тести. При цьому обов'язково враховуються присутність на заняттях та активність студента під час лабораторного заняття; недопустимість пропусків та запізнь на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях не пов'язаних з навчанням; списування та плагіат; несвочасне виконання поставленого завдання і т. ін.</p> <p>Жодні форми порушення академічної доброчесності не толеруються.</p>
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

Схема курсу «Алгоритми та структури даних»

Тиж.	Тема, план, короткі тези	Форма заняття	Ресурси	Завдання, год.	Термін виконання
1	Вступ. Поняття алгоритму. Аналіз складності.	Лекція (2 год.)	[1-7]		1 тиждень
	О-символіка, асимптотика, час роботи алгоритмів, швидкість росту функцій: логарифм, поліном, експонента.	Лабораторне заняття (2 год.)	[1-7]		1 тиждень
2	Алгоритми сортування.	Лекція (2 год.)	[1-7]	Опрацювання лекційного матеріалу (4 год.)	1 тиждень
	Алгоритми сортування бульбашкою та вставками. Швидке сортування.	Лабораторне заняття (2 год.)	[1-7]		1 тиждень
3	Двійковий пошук, метод двох вказівників.	Лекція (2 год.)	[1-7]	Опрацювання лекційного матеріалу (8 год.)	1 тиждень
	Задачі на двійковий пошук, k-та порядкова статистика.	Лабораторне заняття (2 год.)	[1-7]		1 тиждень
4	Піраміди.	Лекція (2 год.)	[1-7]	Опрацювання лекційного матеріалу (4 год.)	1 тиждень
	Бінарна піраміда, k-арна піраміда, нижні оцінки на час сортування та на час побудови піраміди.	Лабораторне заняття (2 год.)	[1-7]		1 тиждень
5	Структури даних.	Лекція (2 год.)	[1-7]	Опрацювання лекційного матеріалу (8 год.)	1 тиждень
	Стек, дек, черга, список, двозв'язний список, динамічний масив, хеш-таблиця, структури, що розширюються.	Лабораторне заняття (2 год.)	[1-7]		1 тиждень
6	Метод повного перебору. Жадібні алгоритми.	Лекція (2 год.)	[1-7]	Опрацювання лекційного матеріалу (4 год.)	1 тиждень
	Задачі, які можна розв'язати методом повного перебору. Оптимізації повного перебору. Жадібні та наближені алгоритми.	Лабораторне заняття (2 год.)	[1-7]		1 тиждень

7	Динамічне програмування.	Лекція (2 год.)	[1-7]	Опрацювання лекційного матеріалу (16 год.)	1 тиждень
	Задачі про рюкзак, найдовшу спільну підпоследовність. Динаміка на підмножинах, підвідрізках, деревах, прямих та розірваних профілях	Лабораторне заняття (2 год.)	[1-7]		1 тиждень
8	Алгоритми на графах.	Лекція (2 год.)	[1-7]	Опрацювання лекційного матеріалу (16 год.)	1 тиждень
	Обхід в глибину, пошук цикла в графі, топологічне сортування, компоненти сильної звязності, розфарбування в два кольори. Алгоритм Дейкстри, алгоритм A*, алгоритм Флойда. Алгоритм Форда-Белмана та його оптимізації, пошук цикла відємної ваги.	Лабораторне заняття (2 год.)	[1-7]		1 тиждень
9	Алгоритми побудови мінімального каркасу.	Лекція (2 год.)	[1-7]	Опрацювання лекційного матеріалу (4 год.)	1 тиждень
	Система множин, що не перетинаються. Алгоритми Крускала та Пріма.	Лабораторне заняття (2 год.)	[1-7]		1 тиждень
10	Дерева пошуку та дерева відрізків.	Лекція (2 год.)	[1-7]	Опрацювання лекційного матеріалу (16 год.)	1 тиждень
	Двійкові дерева пошуку. Дерево відрізків, операції в точці і на відріжку, модифікації на відріжку. Дерево Фенвіка.	Лабораторне заняття (2 год.)	[1-7]		1 тиждень
11	Паросполуки та потоки.	Лекція (2 год.)	[1-7]	Опрацювання лекційного матеріалу (8 год.)	1 тиждень
	Максимальна паросполука в дводольному графі. Мінімальне покриття вершинами. Максимальна незалежна множина. Алгоритм Куна та його оптимізації. Потоки. Теорема та алгоритм Форда-	Лабораторне заняття (2 год.)	[1-7]		1 тиждень

	Фалкерсона. Алгоритми пошуку потоків мінімальної вартості.				
12	Рядкові алгоритми.	Лекція (2 год.)	[1-7]	Опрацювання лекційного матеріалу (8 год.)	1 тиждень
	Пошук підрядка в рядку. Суфіксий масив. Суфіксий автомат. Суфіксий дерево.	Лабораторне заняття (2 год.)	[1-7]		1 тиждень
13	Теорія ігор.	Лекція (2 год.)	[1-7]	Опрацювання лекційного матеріалу (4 год.)	1 тиждень
	Гра Німа, числа Гранді, ретроспективний аналіз.	Лабораторне заняття (2 год.)	[1-7]		1 тиждень
14	Хешування та бітове стиснення.	Лекція (2 год.)	[1-7]	Опрацювання лекційного матеріалу (4 год.)	1 тиждень
	Універсальне хешування, подвійне хешування. Бітове стиснення.	Лабораторне заняття (2 год.)	[1-7]		1 тиждень
15	Обчислювальна геометрія.	Лекція (2 год.)	[1-7]	Опрацювання лекційного матеріалу (8 год.)	1 тиждень
	Задача про перетин відрізків на площині. Обчислення площі багатокутника. Побудова опуклої оболонки.	Лабораторне заняття (2 год.)	[1-7]		1 тиждень
16	Алгебра та теорія чисел.	Лекція (2 год.)	[1-7]	Опрацювання лекційного матеріалу (4 год.)	1 тиждень
	Метод Гауса та лінійна алгебра. Перетворення Фур'є. Тест на простоту, факторизація чисел.	Лабораторне заняття (2 год.)	[1-7]		1 тиждень

Discipline name	Algorithms and Data Structures (in English)
Address where the discipline is taught	The main building of Ivan Franko National University of Lviv 1, Universytetska st., Lviv
The faculty and department under which the discipline is established	Faculty of Applied Mathematics and Informatics Department of Applied Mathematics
Field of knowledge, code and name of specialty	11 – Mathematics and Statistics 113 – Applied Mathematics
Teachers of the discipline	Biletskyy Vasyl, Candidate of Physical and Mathematical Sciences, Associate Professor of the Department of Applied Mathematics
Contact information of teachers	vasyl.biletsky@lnu.edu.ua ; https://ami.lnu.edu.ua/employee/biletskyj Room 262, The main building of Ivan Franko National University of Lviv, 1, Universytetska st., Lviv
Consultations on issues of training in the discipline are taking place	Consultations on the day of lectures/laboratory sessions (by prior agreement)
Course page	https://ami.lnu.edu.ua/course/alhorytmy-ta-struktury-danykh-anhliyskoii-movoiu-prykladna-matematyka-1-9
Information about the discipline	The discipline "Algorithms and data structures" is a discipline of choice from specialty 113 - applied mathematics for the educational and scientific program "Applied Mathematics", which is taught in the 1st semester in the amount of 6 credits (according to the European Credit Transfer System ECTS).
Brief abstract of the discipline	The discipline "Algorithms and data structures" will give students an idea of the general principles of solving algorithmically complex problems, which involve usage and combination of algorithms and data structures. The course is divided into 16 modules. During laboratory classes, students solve practical problems that are complex and might have several possible solutions as well as their implementations. Therefore, assignments and tasks require an assessment of the effectiveness of potential solutions, the proper selection and implementation of the optimal solution. Solving a practical problem involves writing the program code in one of the object-oriented programming languages, using the related data structures, and further thorough testing.
Goal and objectives of the discipline	The goal is to form students' holistic idea of the general principles of solving algorithmically complex problems. The objectives are to form students' knowledge system of complex algorithmic problem solution principles, usage and combination of algorithms as well as data structures.
Literature for the discipline	<ol style="list-style-type: none"> 1. Кормен Т. Вступ до алгоритмів / Томас Кормен, Чарльз Лейзерсон, Рональд Рівест, Кліффорд Стайн. — К.І.С., 2019 2. Sussman, Gerry, Harold Abelson, and Julie Sussman. Structure and interpretation of computer programs — MIT Press, 2022. 3. Cormen, Thomas H. Algorithms unlocked. — Mit Press, 2013 4. Donald, E. Knuth. The art of computer programming. — Addison-Wesley 5. Skiena, Steven S. The algorithm design manual: Text. Vol. 1. — Springer Science & Business Media, 1998 6. Kao, Ming-Yang. Encyclopedia of algorithms. — Springer Science &

	<p>Business Media, 2008.</p> <p>7. Sedgewick, Robert, and Philippe Flajolet. An introduction to the analysis of algorithms. — Addison-Wesley, 2013.</p>
Scope of the course	Total volume: 180 hours. Classroom classes: 64 hours, including 32 hours of lectures and 32 hours of laboratory work. Independent work: 116 hours.
Expected learning outcomes	<p>Upon completion of this course, the student will</p> <ul style="list-style-type: none"> • know: the principles of solving algorithmic problems and related data structures; • be able to: combine algorithms and data structures to solve practical problems, analyze the properties of practical problems and evaluate the effectiveness of possible solutions, implement the solution of a practical problem.
Keywords	Algorithm, Data Structures, Computational Complexity
Course format	Face-to-face
Topics	See below in the table Scheme of the course "Algorithms and data structures"
Final control, form	Test
Prerequisites	<p>To study the course, students need basic knowledge of</p> <ul style="list-style-type: none"> • fundamentals of programming • software development
Teaching methods and techniques that will be used during the teaching of the course	Lectures, presentations, modular control. Homework and individual tasks
Required equipment	A computer with Visual Studio Code, internet access.
Evaluation criteria (separately for each type of educational activity)	<p>The assessment is carried out on a 100-point scale. During the semester a student can earn 100 points according to the following distribution:</p> <ul style="list-style-type: none"> • individual tasks (50 points, 5 tasks, 10 points per task) • control tests on Algotester platform https://algotester.com/ (50 points) <p>A total of 100 points during the semester.</p> <p>Individual task evaluation criteria:</p> <p>10 points – the student fully completed the task, independently selected data structures and implemented the necessary algorithms, confidently answered all related questions during the defense;</p> <p>7-9 points – the student fully completed the task with minor inaccuracies, the used algorithms and data structures are optimal or close to such, they answered most of the questions related to the topic;</p> <p>4-6 points – the student basically completed the task, receiving recommendations on the use of certain data structures and algorithms, demonstrated mediocre knowledge, answered certain related questions;</p> <p>1-3 points – the student did not complete the task completely, used suboptimal algorithms and data structures, demonstrated weak knowledge, answered only some questions;</p> <p>0 points – the student did not complete the task and during the defense cannot answer any question on the relevant topic.</p> <p>Control test evaluation criteria: proportional to the number of tasks that the student successfully passed during the test on the Algotester platform https://algotester.com/</p> <p>Academic Integrity: Students' works are expected to be their own original research or reasoning. Lack of references to used sources, fabrication of sources, plagiarism, interference in the work of other students are, but are not limited to, examples of possible academic dishonesty. The detection of</p>

	<p>signs of academic dishonesty in a student's written work is a reason for its rejection by the teacher, regardless of the scale of plagiarism or deception.</p> <p>Attending classes is an important part of learning. All students are expected to attend all lectures and laboratory sessions of the course. Students must inform the teacher about the impossibility to attend classes. In any case, students are obliged to adhere to the deadlines set for homework and individual assignments provided by the course.</p> <p>Literature. All literature that students cannot find on their own will be provided by the teacher for educational purposes only, without the right to transfer it to third parties. Students are also encouraged to use other literature and sources that are not among the recommended ones.</p> <p>Scoring policy. Points scored during control tests and for individual tasks are taken into account. At the same time, attendance of classes and the student's activity during laboratory sessions must be taken into account; inadmissibility of absences and lateness to classes; using a mobile phone, tablet or other mobile devices during class for purposes not related to education; plagiarism; untimely performance of the assigned task, etc. Any form of breach of academic integrity will not be tolerated.</p>
Poll	An evaluation questionnaire for the purpose of assessing the quality of the course will be provided upon completion of the course.

Scheme of the course "Algorithms and data structures"

Week	Topic, plan, short theses	Form of activity	Resources	Assignment, hours	Deadline
1	Introduction. The concept of an algorithm. Complexity analysis.	Lecture (2 hours)	[1-7]		1 week
	O-notation, asymptotics, running time of algorithms, growth rate of functions: logarithm, polynomial, exponent.	Laboratory class (2 hours)	[1-7]		1 week
2	Sorting algorithms.	Lecture (2 hours)	[1-7]	Processing of lecture material (4 hours)	1 week
	Bubble sort, insertion sort, quick sort.	Laboratory class (2 hours)	[1-7]		1 week
3	Binary search, method of two pointers.	Lecture (2 hours)	[1-7]	Processing of lecture material (8 hours)	1 week
	Binary search problems, k-th order statistics.	Laboratory class (2 hours)	[1-7]		1 week
4	Heaps.	Lecture (2 hours)	[1-7]	Processing of lecture material (4 hours)	1 week
	Binary heap, k-heap, sort time and heap construction time.	Laboratory class (2 hours)	[1-7]		1 week
5	Data structures.	Lecture (2 hours)	[1-7]	Processing of lecture material (8 hours)	1 week

	Stack, deck, queue, list, doubly linked list, dynamic array, hash table, expanding structures.	Laboratory class (2 hours)	[1-7]		1 week
6	Full search method. Greedy algorithms.	Lecture (2 hours)	[1-7]	Processing of lecture material (4 hours)	1 week
	Problems that can be solved by the full search method. Optimization of the full search. Greedy and approximate algorithms.	Laboratory class (2 hours)	[1-7]		1 week
7	Dynamic programming.	Lecture (2 hours)	[1-7]	Processing of lecture material (16 hours)	1 week
	Knapsack problems, longest common subsequence. Dynamics on subsets, subsegments, trees, straight and broken profiles.	Laboratory class (2 hours)	[1-7]		1 week
8	Graph algorithms.	Lecture (2 hours)	[1-7]	Processing of lecture material (16 hours)	1 week
	Depth first search, finding a cycle in a graph, topological sort, strongly connected components, two-coloring problem. Dijkstra's algorithm, A* algorithm, Floyd's algorithm. Ford-Belman algorithm and its optimization, finding a negative weight cycle.	Laboratory class (2 hours)	[1-7]		1 week
9	Minimal spanning tree algorithms.	Lecture (2 hours)	[1-7]	Processing of lecture material (4 hours)	1 week
	Disjoint set union. Kruskal's and Prim's algorithms.	Laboratory class (2 hours)	[1-7]		1 week
10	Search trees and segment trees.	Lecture (2 hours)	[1-7]	Processing of lecture material (16 hours)	1 week
	Binary search trees. Segment tree, operations at a point and on a segment, segment updates. Fenwick Tree.	Laboratory class (2 hours)	[1-7]		1 week
11	Matchings and network flows.	Lecture (2 hours)	[1-7]	Processing of lecture material (8 hours)	1 week
	The maximum matching in a bipartite graph. Minimal vertex coverage. Maximum independent set. Kuhn's algorithm and its	Laboratory class (2 hours)	[1-7]		1 week

	optimizations. Ford-Falkerson theorem and algorithm. Algorithms for finding minimum cost flows.				
12	String algorithms.	Lecture (2 hours)	[1-7]	Processing of lecture material (8 hours)	1 week
	Search for a substring in a string. Suffix array. Suffix automaton. Suffix tree.	Laboratory class (2 hours)	[1-7]		1 week
13	Game theory.	Lecture (2 hours)	[1-7]	Processing of lecture material (4 hours)	1 week
	Nim game, Sprague-Grundy numbers, retrospective analysis.	Laboratory class (2 hours)	[1-7]		1 week
14	Hashing and bit-compression.	Lecture (2 hours)	[1-7]	Processing of lecture material (4 hours)	1 week
	Universal hashing, double hashing. Bit compression.	Laboratory class (2 hours)	[1-7]		1 week
15	Computational geometry.	Lecture (2 hours)	[1-7]	Processing of lecture material (8 hours)	1 week
	Intersection of segments on a plane. Calculating the area of a polygon. Construction of a convex hull.	Laboratory class (2 hours)	[1-7]		1 week
16	Algebra and number theory.	Lecture (2 hours)	[1-7]	Processing of lecture material (4 hours)	1 week
	Gauss method and linear algebra. Fourier transformation. Prime test, integer number factorization.	Laboratory class (2 hours)	[1-7]		1 week