

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

Затверджено

На засіданні кафедри програмування
факультету прикладної математики та
інформатики
Львівського національного університету
імені Івана Франка
(протокол № 1 від 29 серпня 2023 р.)



Зав. кафедри к. ф.-м. н., доц. Ярошко С. А.

Силабус навчальної дисципліни
«Теорія алгоритмів»,
що викладається в межах ОПП Інформатика
першого (бакалаврського) рівня вищої освіти
для здобувачів зі спеціальності 122 Комп'ютерні науки

Львів 2023 р.

Назва дисципліни	Теорія алгоритмів
Адреса викладання дисципліни	Львівський національний університет імені Івана Франка, вул. Університетська 1, м. Львів, Україна, 79000
Факультет та кафедра, за якою закріплена дисципліна	Факультет прикладної математики та інформатики, кафедра програмування
Галузь знань, шифр та назва спеціальності	12 – інформаційні технології 122 – комп'ютерні науки
Викладачі дисципліни	Сибіль Юрій Миколайович, к. ф.-м. н., доцент, доцент кафедри програмування
Контактна інформація викладачів	Електронна пошта: yuriy.sybil@lnu.edu.ua
Консультації з питань навчання по дисципліні відбуваються	Консультації проводять в день проведення практичних занять (за попередньою домовленістю та за умови проведення аудиторних занять). Можливі он-лайн консультації через Microsoft Teams. Для погодження часу он-лайн консультацій слід писати на електронну пошту викладача.
Сторінка курсу	https://ami.lnu.edu.ua/course/prykladna-teoria-alg
Інформація про дисципліну	Курс “Теорія алгоритмів” є нормативною дисципліною з спеціальності 122 – комп'ютерні науки для освітньої програми Інформатика, яка викладається в 4-му семестрі в обсязі 4 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Мета та цілі дисципліни	Метою нормативної дисципліни “Теорія алгоритмів” є навчити студента: <ul style="list-style-type: none"> • оцінювати різні типи алгоритмів стосовно їх часової та ємнісної складності; • використовувати різні типи композицій алгоритмів для побудови ефективних алгоритмів розв’язування практично важливих задач; • використовувати класичні алгоритмічні системи (нормальні алгоритми Маркова, рекурсивні функції, машини Тьюрінга) для аналізу розв’язності різних типів задач; • на основі відповідних підходів (метод „поділяй і володарюй”, ”жадібні алгоритми”, динамічне програмування) розробляти та реалізовувати алгоритми для конкретних задач.
Коротка анотація дисципліни	Курс охоплює наступні розділи теорії алгоритмів: необхідність формалізації поняття алгоритму, оцінювання алгоритмів та використання оцінки складності для порівняння алгоритмів, алфавітні оператори та алгоритми, властивості алгоритмів, способи композиції алгоритмів, класичні алгоритмічні системи – нормальні алгоритми Маркова, рекурсивні функції, машини Тьюрінга, важкорозв’язні задачі, методи розробки ефективних алгоритмів.
Література для вивчення дисципліни	<i>Основна література</i> <ol style="list-style-type: none"> 1. Клакович Л.М., Левицька С.М., Костів О.В. Теорія алгоритмів. Львів, Вид-во Львів ун-ту, 2014.- 116 с. 2. Троцько В.В. Теорія алгоритмів: Навчально - методичний посібник. – Київ: Університет економіки та права «КРОК», 2023 – 123 с. 3. Томас Кормен, Чарльз Лейзерсон, Рональд Рівест, Кліффорд Стайн. Вступ до алгоритмів. К: – К.І.С., 2019. – 1288 с. 4. Ярошко С.А. Методи розробки алгоритмів. Програмування мовою С++: Навчальний посібник / С.А. Ярошко, О.С. Ярошко – Львів: ЛНУ імені Івана Франка, 2022. – 248 с. [електронна версія: https://lnuittutor.github.io/]

	<p><i>Додаткова література</i></p> <p>1. Крєневич А.П. Алгоритми і структури даних. Підручник.– К.:ВПЦ "Київський Університет", 2021. – 200 с.</p>
Обсяг курсу	4 кредити ЄКТС – 120 годин. З них 32 годин лекцій, 32 години лабораторних занять та 56 годин самостійної роботи
Очікувані результати навчання	<p>Після завершення цього курсу студент буде:</p> <p><i>знати</i></p> <p>різні підходи до формалізації поняття алгоритму, класичні алгоритмічні системи, методи розробки алгоритмів, основні алгоритмічно нерозв'язні задачі;</p> <p><i>вміти</i></p> <ul style="list-style-type: none"> визначати оцінки складності алгоритмів і застосовувати їх для порівняльної характеристики алгоритмів, складати алгоритми в різних алгоритмічних системах, будувати ефективні алгоритми різними методами.
Компетентності	<p><i>Загальні:</i></p> <p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК2. Здатність застосовувати знання у практичних ситуаціях.</p> <p><i>Спеціальні:</i></p> <p>СК1. Здатність до математичного формулювання та досліджування неперервних та дискретних математичних моделей, обґрунтування вибору методів і підходів для розв'язування теоретичних і прикладних задач у галузі комп'ютерних наук, аналізу та інтерпретування</p> <p>СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.</p> <p>СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.</p>
Програмні результати навчання	<p>ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.</p> <p>ПР5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.</p>
Ключові слова	Алфавітні оператори, класичні алгоритмічні системи, нормальні алгоритми Маркова, рекурсивні функції, машини Тьюрінга, методи побудови ефективних алгоритмів.
Формат курсу	Очний.
Теми	<p>Лекції</p> <ul style="list-style-type: none"> Базові поняття алгоритмів та їх складності. Оцінювання алгоритмів. Побудова алгоритмів із заданими оцінками. Алфавітні оператори та алгоритми. Властивості алгоритмів. Способи композиції алгоритмів. Граф-схеми алгоритмів. Класичні алгоритмічні системи. Система нормальних алгоритмів

	<p>Маркова. Універсальний нормальний алгоритм.</p> <ul style="list-style-type: none"> • Рекурсивні функції. Найпростіші функції та способи їх композиції. • Оператори суперпозиції та примітивної рекурсії. Примітивно-рекурсивні функції. • Оператор мінімізації. Частково-рекурсивні функції. • Машина Тюрінга (МТ). Варіанти МТ. Детерміновані МТ. Теза Тюрінга. • Зв'язок МТ з іншими алгоритмічними системами. • Поліноміальні алгоритми та важкорозв'язні задачі. Приклади важкорозв'язних задач. • NP-повні задачі. Недетерміновані машини Тюрінга. Приклади NP-повних задач. • Класи P та NP. Доведення результатів про NP-повноту. • Застосування теорії NP-повноти для аналізу задач. • Основні методи розробки та побудови ефективних алгоритмів. • Алгоритми "розділяй і володарюй". • Евристичні алгоритми. Метод гілок і меж. • "Жадібні" алгоритми. • Динамічне програмування. <p>Лабораторні</p> <ul style="list-style-type: none"> • Розробка і аналіз алгоритмів. Побудова та оцінювання алгоритмів. • Дослідження часової та ємнісної складності алгоритмів сортування. • Побудова алгоритмів із заданими оцінками • Способи композиції алгоритмів. Приклади. • Побудова нормальних алгоритмів Маркова. • Рекурсивні функції. Найпростіші функції • Оператори суперпозиції, примітивної рекурсії та мінімізації. • Побудова машин Тюрінга для типових алгоритмів. • Побудова недетермінованої машини Тюрінга. Аналіз NP-повноти деяких задач. • Алгоритми "розділяй і володарюй". • Метод гілок і меж. • "Жадібні" алгоритми". • Динамічне програмування.
Підсумковий контроль, форма	Екзамен
Пререквізити	Для вивчення курсу студенти потребують базових знань з дисциплін "Математика", "Програмування", "Дискретна математика".
Навчальні методи та техніки, які використовують під час викладання курсу	Лекції з мультимедійними презентаціями та з демонстрацією прийомів практичного використання середовища програмування; лабораторні заняття у вигляді проектування алгоритмів і програм, виконання практичних завдань; самостійне опрацювання навчальних матеріалів: підручників, конспектів лекцій. Обговорення теоретичного та практичного матеріалу в онлайн сервісах, формулювання творчих завдань для студентів, виконання яких готує до вивчення нового теоретичного матеріалу.
Необхідне обладнання	Для проведення лекцій: комп'ютер, проектор, доступ до мережі інтернет. Для проведення лабораторних та виконання завдань: комп'ютер, ОС Windows/Linux, доступ до інтернету, середовище програмування мовою C++ (Microsoft Visual Studio, Code Blocks тощо). Уся література, яку студенти не можуть 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>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</p>	<p>Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням:</p> <ul style="list-style-type: none"> • протягом семестру – виконання домашніх завдань за варіантами: 6 домашніх завдань (максимальна кількість балів за кожне 5). Максимальна кількість балів 30. • контрольні заміри (модулі): 2 модулі у вигляді тестів (максимальна кількість балів за кожен 10); максимальна кількість балів 20 • в кінці семестру – іспит. Максимальна кількість балів 50. <p>Підсумкова максимальна кількість балів 100</p> <p>Відвідування занять є важливою складовою навчання. Очікується, що всі студенти відвідують усі лекції і лабораторні заняття курсу. Активність під час проведення лекцій і лабораторних заохочується балами. У будь-якому випадку студенти зобов'язані дотримуватися усіх строків визначених для виконання всіх видів робіт, передбачених курсом. Виконані роботи завантажують у відповідне хмарне сховище. Активність на лекціях і лабораторних враховують при оцінюванні відповідного лабораторного завдання.</p> <p>Академічна доброчесність: очікується, що роботи студентів будуть їхнім оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів, здавання чужих комп'ютерних програм як своїх становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою для її не зарахування викладачем, незалежно від масштабів плагіату чи обману.</p> <p>Жодні форми порушення академічної доброчесності не толеруються.</p>
<p>Опитування</p>	<p>Анкету-оцінку з метою оцінювання якості курсу буде надано після завершення курсу.</p>

Схема курсу

Тиж.	Тема, план	Форма діяльності (заняття)	Література	Завдання, год	Термін виконання
1.	Тема 1. Базові поняття алгоритмів та їх складності. Оцінювання алгоритмів. Побудова алгоритмів із заданими оцінками.	лекція, самостійна робота	[1,4]	2 2	1 тиждень
	Тема 1. Розробка і аналіз алгоритмів. Побудова та оцінювання алгоритмів.	лаб.	[1,4]	2	1 тиждень
2.	Тема 2. Алфавітні оператори та алгоритми. Властивості алгоритмів.	лекція, самостійна робота	[1,4]	2 4	1 тиждень
	Тема 2. Дослідження часової та ємнісної складності складності алгоритмів сортування.	лаб.	[1,4]	2	1 тиждень
3.	Тема 3. Способи композиції алгоритмів. Граф-схеми алгоритмів.	лекція, самостійна робота	[1,4]	2 4	1 тиждень
	Тема 3. Побудова алгоритмів із заданими оцінками	лаб.	[1,4]	2	1 тиждень

4.	Тема 4. Класичні алгоритмічні системи. Система нормальних алгоритмів Маркова. Універсальний нормальний алгоритм.	лекція, самостійна робота	[1,4]	2 4	1 тиждень
	Тема 4. Способи композиції алгоритмів. Приклади.	лаб.	[1,4]	2	1 тиждень
5.	Тема 5. Рекурсивні функції. Найпростіші функції та способи їх композиції.	лекція, самостійна робота	[1,4]	2 2	1 тиждень
	Тема 5. Побудова нормальних алгоритмів Маркова.	лаб.	[1,4]	2	1 тиждень
6.	Тема 6. Оператори суперпозиції та примітивної рекурсії. Примітивно-рекурсивні функції.	лекція, самостійна робота	[1,4]	2 4	1 тиждень
	Тема 6. Рекурсивні функції. Найпростіші функції.	лаб.	[1,4]	2	1 тиждень
7.	Тема 7. Оператор мінімізації. Частково-рекурсивні функції.	лекція, самостійна робота	[1,4]	2 4	1 тиждень
	Тема 7. Оператори суперпозиції та примітивної рекурсії.	лаб.	[1,4]	2	1 тиждень
8.	Тема 8. Машини Тюрінга (МТ). Варіанти МТ. Детерміновані МТ. Теза Тюрінга.	лекція, самостійна робота	[1,4]	2 2	1 тиждень
	Тема 8. Оператор мінімізації.	лаб.	[1,4]	2	1 тиждень
9.	Тема 9. Зв'язок МТ з іншими алгоритмічними системами.	лекція, самостійна робота	[1,4]	2 2	1 тиждень
	Тема 9. Побудова машин Тюрінга для типових алгоритмів.	лаб.	[1,4]	2	1 тиждень
10.	Тема 10. Поліноміальні алгоритми та важкорозв'язні задачі. Приклади важкорозв'язних задач.	лекція, самостійна робота	[1,4]	2 4	1 тиждень
	Тема 10. Побудова недетермінованої машини Тюрінга для деяких задач.	лаб.	[1,4]	2	1 тиждень
11.	Тема 11. NP-повні задачі. Недетерміновані машини Тюрінга. Приклади NP-повних задач.	лекція, самостійна робота	[1,4]	2 4	1 тиждень
	Тема 11. Аналіз NP-повноти задачі про кліку для неорієнтованих графів та задачі комівояжера.	лаб.	[1,4]	2	1 тиждень
12.	Тема 12. Основні методи розробки та побудови ефективних алгоритмів.	лекція, самостійна робота	[1,4]	2 4	1 тиждень
	Тема 12. Реалізація алгоритму рекурсивного розв'язування задачі знаходження максимального і мінімального елементів.	лаб.	[1,4]	2	1 тиждень
13.	Тема 13. Алгоритми "розділяй і володарюй". Алгоритм множення двох двійкових чисел.	лекція, самостійна робота	[1,4]	2 4	1 тиждень
	Тема 13. Алгоритми "розділяй і володарюй". Реалізація алгоритму множення двох двійкових чисел.	лаб.	[1,4]	2	1 тиждень

14.	Тема 14. Евристичні алгоритми. Метод гілок і меж. Задача "комівояжера". Алгоритм GTS.	лекція, самостійна робота	[1,4]	2 4	1 тиждень
	Тема 14. Реалізація алгоритму GTS для оптимального розв'язування задачі "комівояжера".	лаб.	[1,4]	2	1 тиждень
15.	Тема 15. «Жадібні» алгоритми. Алгоритм Дейкстри – Пріма знаходження мінімального каркаса. Задача про ранець.	лекція, самостійна робота	[1,4]	2 4	1 тиждень
	Тема 15. Реалізація алгоритму Дейкстри – Пріма знаходження мінімального каркаса.	лаб.	[1,4]	2	1 тиждень
16.	Тема 16. Динамічне програмування. Задача про перемноження матриць.	лекція, самостійна робота	[1,4]	2 4	1 тиждень
	Тема 16. Реалізація алгоритму задачі про перемноження матриць з використанням засобів динамічного програмування.	лаб.	[1,4]	2	1 тиждень