



Проектування та аналіз обчислювальних алгоритмів

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 «Інформаційні технології»</i>
Спеціальність	<i>122 Комп'ютерні науки</i>
Освітня програма	<i>Системи і методи штучного інтелекту</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>очна(денна)/дистанційна/змішана</i>
Рік підготовки, семестр	<i>2курс, осінній семестр</i>
Обсяг дисципліни	<i>4 кредити</i>
Семестровий контроль/ контрольні заходи	<i>залік</i>
Розклад занять	<i>Rozklad.kpi.ua</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Лектор: <i>к.т.н. Селін Юрій Миколайович, selinyurij@online.ua</i> Практичні / Семінарські /Лабораторні: : <i>к.т.н. Селін Юрій Миколайович, selinyurij@online.ua</i>
Розміщення курсу	https://classroom.google.com/c/MTUyMDY0MDUxMTUy?cjc=bklafnl

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Метою навчальної дисципліни «Проектування та аналіз обчислювальних алгоритмів» вивчення фундаментальних та удосконалених алгоритмів для їх подальшого застосування при вирішенні алгоритмічних задач, опанування основних методів аналізу алгоритмів, набуття вміння будувати та здійснювати раціональний вибір алгоритму для конкретної задачі з відомих класів алгоритмів на основі обраних критеріїв.

Предметом навчальної дисципліни є сучасні та ефективні алгоритми оброблення інформації, а також методи їх дослідження та аналізу.

Об'єктом навчальної дисципліни є інформаційні системи та процеси, що відбивають різні аспекти їх функціонування.

Вивчення навчальної дисципліни націлено на формування, розвиток та закріплення у здобувачів таких загальних та фахових **компетентностей**: ЗК 1 Здатність до абстрактного мислення, аналізу та синтезу; ЗК 2 Здатність застосовувати знання у практичних ситуаціях; ЗК 3 Знання та розуміння предметної області та розуміння професійної діяльності; ЗК 6 Здатність вчитися і оволодівати сучасними знаннями; ФК 3 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; ФК 8 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.

Внаслідок вивчення курсу студент повинен бути здатний продемонструвати такий **програмний результат навчання ОПП**: 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; 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; ПР9 Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

У кінці вивчення курсу студент повинен **знати**: основ алгоритмізації, методи побудови алгоритмів, абстрактних структур даних та їх ролі при побудові алгоритмів, як вибір структури даних впливає на складність розв'язку задачі; **уміти**: розробляти структуровані алгоритми розв'язку задач, перевіряти відповідність вхідних даних екземпляру задачі, доводити коректність та ефективність побудованих алгоритмів, оцінювати час роботи алгоритму, порівнювати різні способи реалізації за складністю та часом роботи; **досвід**: розробки алгоритмів розв'язку задач, ефективного використання абстрактних структур даних, складання супровідної документації відповідно до вимог чинного законодавства України зі стандартизації (ДСТУ 3008:2015 Документація. Звіти у сфері науки і техніки. Структура і правила

оформлення).

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Вивчення дисципліни «Проектування та аналіз обчислювальних алгоритмів» ґрунтується на знаннях та вміннях, які студенти отримали під час вивчення дисциплін математичного циклу та циклу програмування. Водночас вона забезпечує базові та вибіркові дисципліни, що входять до циклів проектування інформаційних систем, аналізу даних та просунутого програмування.

Попередні дисципліни	Наступні дисципліни
Математичний аналіз	Об'єктно-орієнтоване програмування
Дискретна математика	Математична логіка і теорія алгоритмів
Алгоритмізація та програмування	

3. Зміст навчальної дисципліни

Тема	Лекції	Назва	Години		КП, теми
			4	5	
1	2	3	4	5	6
I		Змістовий модуль 1. Математичні основи аналізу алгоритмів та основні фундаментальні алгоритми		16	
	1	Ефективність алгоритмів. Характеристики продуктивності алгоритмів.	2		
	2	Алгоритмічні стратегії. Формалізація алгоритмів. Покрокове проектування алгоритмів	2		
	3	Тимчасова та просторова складності алгоритмів. Асимптотична складність алгоритмів	2		
	4	Структури даних. Вибір структури даних	2		
	5	Алгоритми сортування, злиття та пошуку.	2		
	6	Удосконалені (модифіковані) алгоритми сортування. Рекурсивні алгоритми. Порівняльний аналіз характеристик швидкодії ітераційних та рекурсивних алгоритмів.	2		1.Сортування даних
	7	Геометричні алгоритми. Алгоритм Клі. Метод Грехема. Метод вертикальної декомпозиції.	2		
	8	Комбінаторні алгоритми.	2		
II		Змістовий модуль 2. Алгоритми на графах та паралельні алгоритми.		20	

	9	Фундаментальні алгоритми на графах і деревах. Впорядкування масивів бінарним деревом. Задачі обробки графів. Методи та алгоритми обходу графів: DSF-метод, BFS-метод. Визначення характеристик алгоритмів обходу графів, їх аналіз.	2		2. Алгоритми роботи з деревами пошуку.
	10	Евристичні алгоритми Задача комівояжера. Генетичні алгоритми.	2		
	11	Алгоритми на підрядках. Алгоритм Рабіна-Карпа. Пошук підрядків за допомогою скінчених алгоритмів. Алгоритм Кнута-Моріса-Прата. Алгоритм Бойєра-Мура.	2		
	12	NP-повні граматики. Поліноміальний час. Перевірка належності класу та клас NP. NP- повнота та звідність. NP-повні задачі.	2		3. Евристичні алгоритми. Задача комівояжера.
	13	Граф алгоритму та паралельні обчислення. Паралельні форми графу. Ярус та висота. Концепція необмеженого паралелізму. Принцип здвоєння. Обмеженість концепції.	2		
	14	Внутрішній паралелізм. Його переваги. Переходи по вказівникам. CRCW та ERCW алгоритми. Ефективна паралельна обробка префіксів.	2		
	15	Ймовірнісні розрахунки. Метод Монте-Карло.	2		
	16	Дерева (TREE). Алгоритми обходу і пошуку в деревах.	2		4. Алгоритми пошуку шляхів в графах.
	17	Алгоритм Форда — Беллмана. Алгоритм Дейкстри	2		
	18	Метод Флойда — Варшалла. Алгоритм Джонсона	2		
	ВСЬОГО				
8	18		36		5

Теми лабораторних робіт

- | | |
|--|--|
| Лабораторна робота № 1 | Просунуті методи сортування масивів.. |
| Лабораторна робота № 2 | Двійкові дерева. |
| Лабораторна робота № 3
Монте-Карло. | Пошук об'єму 3-D фігури за допомогою методу Монте-Карло. |
| Лабораторна робота № 4 | Пошук шляхів в графах. |

Кожна лабораторна робота виконується студентами самостійно за індивідуальними завданнями. Мета роботи – розвиток навичок практичного дослідження як класичних алгоритмів, так і самостійно розроблених алгоритмів рішення індивідуальних завдань.

4. Навчальні матеріали та ресурси

Базова

1. Горлова Т. М. Теорія алгоритмів: конспект лекцій [для студентів напряму підготовки 6.050101 «Комп'ютерні науки» денної та заочної форм навчання] [Електронний ресурс] / Т. М. Горлова, К. Є. Бобрівник, Н. В. Ліманська. — Київ : НУХТ, 2015. — 95 с. — Режим доступу: <http://library.nuft.edu.ua/ebook/file/M51.21.pdf>
2. Новотарський М. А. Алгоритми та методи обчислень : навч. посіб. [для студ. спеціальностей 121 «Інженерія програмного забезпечення», спеціалізації «Програмне забезпечення високопродуктивних комп'ютерних систем та мереж» та 123 «Комп'ютерна інженерія», спеціалізації «Комп'ютерні системи та мережі»] [Електронний ресурс] / Новотарський М. А. — Київ : КПІ ім. Ігоря Сікорського, 2019. — 407 с. — Режим доступу: https://ela.kpi.ua/bitstream/123456789/27864/1/Alhorytmy_ta_metody_obchislenn.pdf
3. Кублій Л. І. Алгоритмізація та програмування. Практикум : навч. посіб. [для здобувачів ступеня бакалавра за спеціальністю 122 «Комп'ютерні науки»] [Електронний ресурс] / Л. І. Кублій. — Київ : КПІ ім. Ігоря Сікорського, 2019. — 209 с. — Режим доступу: ela.kpi.ua/handle/123456789/28216
4. Кренивч А.П. Алгоритми і структури даних. Підручник. – К.: ВПЦ "Київський Університет", 2021. – 200 с.
5. Матвієнко М.П. Теорія алгоритмів. – К. : Ліра-К 2019 – 344 с.
6. Бородкіна І. Л. Теорія алгоритмів: посіб. для студентів вищих навчальних закладів / І. Л. Бородкіна, Г. О. Бородкін. — Київ : Національний університет біоресурсів та природокористування України, 2018. — 231 с.

Інформаційні ресурси

1. Електронний кампус НТУУ «КПІ ім. І. Сікорського» [сайт] / Єдине інформаційне середовище НТУУ «КПІ ім. І. Сікорського», 2011-2017. – Режим доступу: <http://login.kpi.ua>
2. Online-додаток для створення блок-схем та діаграм [Електронний ресурс] – Режим доступу: <https://www.draw.io>
3. Сайт для створення діаграм online [Електронний ресурс] – Режим доступу: <https://www.lucidchart.com>
4. Git Pro [Електронний ресурс] – Режим доступу: <https://git-scm.com/book/uk/v2>

Навчальний контент

5. Методика опанування навчальної дисципліни(освітнього компонента)

Базові теоретичні відомості викладаються в ході лекцій. Приблизно половина матеріалу вивчається в ході самостійної роботи студентів. Контроль засвоєння знань виконується шляхом поточного опитування студентів , контрольних робіт та за допомогою підсумкових контрольних питань на заліку.

Практичні навички з використання теоретичних знань закріплюються в ході виконання лабораторних робіт. Освоєнню матеріалу допомагає деякий програмістський досвід, одержаний у 1-му семестрі, де розглядаються питання роботи з різними структурними даними в дисципліні «Програмування», а також окремі питання з курсу «Дискретна математика».

6. Самостійна робота студента

№	Назва видів самостійної роботи	Кількість годин
1.	Опрацювання лекційного матеріалу	20
2.	Підготовка до лабораторних занять	10
3.	Самостійне вивчення тем та питань, що не викладаються на лекційних заняттях	18
		48

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

- **Політика щодо відвідування:** Відвідування занять є обов'язковим. За об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання може відбуватись індивідуально (в он-лайн формі за погодженням із деканом факультету)
- **Політика щодо дедлайнів та перескладання:** Всім роботам, що треба виконати під час опанування учбового курсу назначається граничний термін виконання. Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку. Перескладання модулів відбувається із дозволу лектора за наявності поважних причин (наприклад, лікарняний).
- **Політика щодо академічної доброчесності:** Списування під час контрольних робіт та залікових занять заборонені (в т.ч. із використанням мобільних пристроїв). Всі роботи мають бути оформлено згідно правил та повинні мати коректні текстові посилання на використану літературу.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

Семестровий контроль: залік.

Умови допуску до семестрового контролю: мінімально позитивна оцінка за контрольні роботи, а також зарахування усіх лабораторних робіт.

Під час вивчення теоретичного курсу «Проектування та аналіз обчислювальних алгоритмів» треба виконати і захистити 4 лабораторні роботи, написати 3 поточних контрольних робіт, здати загальний залік по теоретичному курсу.

Бальне оцінювання за кожен вид роботи:

Лабораторні роботи – 10 балів кожна

Контрольні роботи - 10 балів кожна

Загальний залік - 10 балів.

Таким чином за весь курс можна набрати

$4 \text{ л.р.} \times 10 \text{ балів} + 5 \text{ к.р.} \times 10 \text{ балів} + \text{залік} \times 10 \text{ балів} = 100 \text{ балів.}$

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

<i>Кількість балів</i>	<i>Оцінка</i>
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, які виносяться на семестровий контроль.

Перерахуйте відомі вам методи розробки алгоритмів. Докладніше розкажіть про один з них.

Перерахуйте переваги та недоліки наступних методів розробки алгоритмів: методу часткових (проміжних) цілей, методу підйому (локального пошуку), методу відпрацювання назад.

Який тип алгоритмів називають «жадібними» і чому?

Дайте характеристику евристичним алгоритмам. В яких випадках доцільно використовувати цей тип алгоритмів? Опишіть загальний підхід до побудови евристичних алгоритмів.

Проаналізуйте, що спільного мають та чим відрізняються алгоритми, що використовують пошук з поверненням, та алгоритми, що використовують метод гілок та границь.

Поясніть, для чого можна використовувати метод альфа-бета відсікання.

Поясніть термін «структурне програмування». Для чого воно застосовується?

Методологія структурного програмування ґрунтується на трьох методах: низхідного проектування, модульного програмування і структурування програм. Розкрийте суть методу низхідного проектування.

Методологія структурного програмування ґрунтується на трьох методах: низхідного проектування, модульного програмування і структурування програм. Розкрийте суть методу модульного програмування.

Методологія структурного програмування ґрунтується на трьох методах: низхідного проектування, модульного програмування і структурування програм. Розкрийте суть методу структурування програм

Поняття обчислювальної складності алгоритму.

Асимптотичний аналіз складності, O-нотація.
Верхня та середня оцінка складності.
Емпіричні вимірювання ефективності алгоритмів.
Вплив структур даних і особливостей реалізації на ефективність алгоритмів.
Основні алгоритмічні стратегії.
Методи розробки алгоритмів
Властивості обчислювальних функцій.
Розв'язні множини і їх властивості.
Поняття складності обчислення.
Теорема Поста.
Теорема про графік обчислювальної функції.
Класи складності.
Опис класів P і NP.
Ототожнення класу P з класом реально обчислювальних функцій.
Приклади NP-повних задач.
Застосування теорії NP-повноти для аналізу складності завдань.
Переваги і недоліки простих сортувань.
Покращені алгоритми сортування.
Алгоритм сортування Шелла.
Алгоритм сортування Хоара (швидке сортування).
Алгоритм сортування злиттям.
Переваги і недоліки складних сортувань.
Порівняння простих та складних сортувань.
Лінійний пошук у масиві.
Бінарний пошук у масиві.
Основні поняття теорії графів.
Матричне подання графів.
Матриця зв'язності графу.
Матриця відстаней на графі.
Алгоритми пошуку найкоротших шляхів та оптимальних маршрутів у графах. 6.
Алгоритм Дейкстри.
Знаходження мінімального остовного дерева графа за алгоритмом Прима – Краскала.
Гамільтонові шляхи на графі.
Дерева, способи подання.
Пошук у ширину на графах та деревах.
Пошук в глибину на графах та деревах.
Задача комівояжера.

Робочу програму навчальної дисципліни (силабус):

Складено доцент кафедри ММСА, к.т.н., Селін Ю.М.

Ухвалено кафедрою ММСА (протокол № 9 від 24.06.2020)

Погоджено Методичною комісією ІІСА (протокол № 9 від 25.06.2020)