



АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ

Ступінь вищої освіти:	бакалавр
Спеціальність:	123 «Комп'ютерна інженерія», 122 «Комп'ютерні науки»
Рік підготовки:	1
Семестр викладання:	весняний
Кількість кредитів ЄКТС:	5
Мова(-и) викладання:	українська
Вид семестрового контролю	іспит

Автор курсу та лектор:

к. ф.-м. н., доцент Хорошун Г. М.			
вчений ступінь, вчене звання, прізвище, ім'я та по-батькові			
доцент кафедри комп'ютерних наук та інженерії			
посада			
horoshun@snu.edu.ua	+38 (066) 1719306	Viber	за розкладом
електронна адреса	телефон	месенджер	консультації

Викладач лабораторних занять:*

к. ф.-м. н., доцент Хорошун Г. М.			
вчений ступінь, вчене звання, прізвище, ім'я та по-батькові			
доцент кафедри комп'ютерних наук та інженерії			
посада			
horoshun@snu.edu.ua	+38 (066) 1719306	Viber	за розкладом
електронна адреса	телефон	месенджер	консультації

Викладач практичних занять:*

вчений ступінь, вчене звання, прізвище, ім'я та по-батькові			
посада			
електронна адреса	телефон	месенджер	консультації

Анотація навчального курсу

Цілі вивчення курсу:

Наведені в курсі матеріали спрямовані на формування у здобувачів вищої освіти знань і навичок в питаннях щодо основ математичної теорії алгоритмів, засобів формалізації задач та процесів їхнього рішення. Розглядаються основні положення теорії алгоритмів, принципами побудови та аналізу алгоритмів, математичне обґрунтування їх застосування, принципи побудови ефективних і надійних програм, машини Тюрінга та машини Посту, основні алгоритми, що використовуються для обробки даних різноманітного характеру.

Курс може бути корисним здобувачам вищої освіти за спеціальностями в галузі знань «12. Інформаційні технології», а також майбутнім фахівцям з розробки математичного, інформаційного та програмного забезпечення інформаційних систем, у галузі інформаційних технологій.

Результати навчання:

Знати: теоретичні основи алгоритмізації і проектування програм; методи структурного програмування; засоби використання статичних та динамічних структур даних; принципи розробки ефективних алгоритмів; методи дослідження і теорію складності алгоритмів; набір базових алгоритмів.

Вміти: правильно вибрати структуру даних для конкретної задачі; розробити відповідно до структури даних алгоритм; оцінювати комп'ютерні алгоритми з використанням комплексних критеріїв якості, в тому числі оцінювати ресурсну ефективність алгоритмів; використовувати рекурсивні структури даних, рекурсивні алгоритми, оцінювати трудомісткість алгоритмів в їх ітераційній і рекурсивній реалізації.

Передумови до початку вивчення:

Базові знання та уявлення з вищої математики, дискретної математики, програмування.

Мета курсу (набуті компетентності)

В наслідок вивчення даного навчального курсу здобувач вищої освіти набуде наступних компетентностей:

1. Здатність до абстрактного мислення, аналізу і синтезу.
2. Здатність застосовувати знання на практиці.
3. Здатність розв'язувати поставлені задачі та приймати відповідні рішення.
4. Здатність оцінювати та забезпечувати якість виконуваних робіт.
5. Здатність розробляти алгоритмічне та програмне забезпечення, компоненти комп'ютерних систем та мереж, Інтернет додатків, кіберфізичних систем з використанням сучасних методів і мов програмування, а також засобів і систем автоматизації проектування тощо.

Структура курсу

№	Тема	Години (Л/ЛБ/ПЗ)	Стислий зміст	Інструменти і завдання
1.	Вступ до теорії алгоритмів	2/4/0	Історичний огляд. Цілі і задачі теорії алгоритмів. Практичне застосування результатів теорії алгоритмів. Формалізація поняття алгоритму. Способи запису алгоритму. Основні алгоритмічні конструкції.	Лабораторна робота «Алгоритми та алгоритмізація»
2.	Машина Поста. Машина Тюрінга та алгоритмічно невирішені проблеми.	2/0/0	Основні питання та операції. Фінітний 1 – процес. Спосіб завдання проблеми і формулювання 1. Машина Тюрінга. Алгоритмічно невирішені проблеми. Проблема відповідностей Поста над алфавітом Σ .	Тести
3.	Вступ в аналіз алгоритмів. Трудомісткість алгоритмів і часові оцінки.	2/4/0	Порівняльні оцінки алгоритмів. Система позначень в аналізі алгоритмів. Класифікація алгоритмів із вигляду функції трудомісткості. Асимптотичний аналіз функцій. Елементарні операції у мові запису алгоритмів. Приклади аналізу простих алгоритмів. Перехід до часових оцінок. Приклад поопераційного часового аналізу.	Лабораторна робота «Дослідження алгоритмів роботи з числами в різних системах числення»
4.	Теорія складності обчислень та складні класи задач. Приклад повного аналізу алгоритму вирішення задачі про суму.	2/0/0	Теоретична границя трудомісткості задач. Складні класи задач. Проблема $P = NP$. Клас NPC (NP – повні задачі). Приклади NP – повних задач. Формулювання завдання та асимптотична оцінка. Алгоритм точного вирішення задачі про суму (метод перебору). Аналіз алгоритму точного рішення задачі про суму.	Тести
5.	Алгоритми організації ітераційних циклів	2/4/0	Основні поняття. Обчислення суми нескінченного ряду. Знаходження коренів рівнянь.	Лабораторна робота «Дослідження ітераційних алгоритмів»
6.	Алгоритми пошуку	4/4/0	Пошук у неупорядкованих множинах даних: лінійний, послідовний пошук, метод транспозиції, метод переміщення в початок. Пошук у впорядкованих множинах даних: двійковий пошук, метод інтерполяції, індексно-послідовний пошук.	Лабораторна робота «Дослідження алгоритмів пошуку»
7.	Алгоритми сортування	4/4/0	Задача сортування. Сортування методом прямого включення. Сортування методом прямого вибору. Бульбашкове сортування. Шейкер-сортування. Сортування включеннями зі спадаючими приростами. Сортування за допомогою дерева. Пірамідаліне сортування. Швидке сортування.	Лабораторна робота «Дослідження алгоритмів сортування»
8.	Рекурсивні функції та алгоритми, методи їх аналізу	4/4/0	Рекурсивні функції. Рекурсивна реалізація алгоритмів. Аналіз трудомісткості механізму виклику процедури. Аналіз трудомісткості алгоритму обчислення факторіала. Логарифмічні тотожності. Методи вирішення рекурсивних співвідношень. Рекурсивні алгоритми. Основна теорема про рекурентні співвідношення.	Лабораторна робота «Розробка Windows-форми»

№	Тема	Години (Л/ЛБ/ПЗ)	Стислий зміст	Інструменти і завдання
9.	Прямий аналіз рекурсивного дерева викликів	2/4/0	Алгоритм сортування злиттям. Злиття відсортованих частин (Merge). Підрахунок вершин у дереві рекурсивних викликів. Аналіз трудомісткості алгоритму сортування злиттям.	Лабораторна робота «Створення Windows-додатка для обробки масиву та матриці»
10.	Алгоритми стандартної бібліотеки	2/0/0	Заголовний файл – Algorithm. Опис алгоритмів. Алгоритми, що не змінюють послідовні операції. Алгоритми, що змінюють послідовні операції. Операції сортування та бінарні операції пошуку. Операції злиття. Операції відносин. Алгоритм <code>std :: find ()</code> і пошук елемента за значенням. Алгоритм <code>std :: sort ()</code> і користувацьке сортування.	Тести
11.	Приклади застосування алгоритмів для практичних задач	2/0/0	Алгоритми на графах: задача пошуку найкоротшої відстані. Теорія чисел: решето Ератосфена.	Тести

Рекомендована література

1. Kleinberg J. and Tardos E.(2006). Algorithm Design (Pearson Education).
2. Головешкин В. А., Ульяно М. В. Теория рекурсии для программистов. — М.: Издательство «Наука ФИЗМАТЛИТ», 2006. — 296 с.
3. Bozoki S. and Rapsak T.(2008). On saaty's and koczkodaj's inconsistencies of pairwise comparison matrices, Journal of Global Optimization 42, 2, p. 157–175, doi:10.1007/s10898-007-9236-z.
4. Zingaro D.(2008). Invariants: A generative approach to programming.
5. Cormen T. H., Leiserson C. E., Rivest R. L. and Stein C.(2009). Introduction to Algorithms, 3rd edn. (McGraw-Hill Book Company), third Edition.
6. Austrin P.(2010). How hard is unshuffling a string (reply), CS Theory Stack Exchange reply to [Erickson (2010)] // <http://cstheory.stackexchange.com/q/692>.
7. Buss S. R. and Soltys M.(2013). Unshuffling a square is NP-hard, Journal of Computer and System Sciences 80, 4, p. 766–776, doi: <http://dx.doi.org/10.1016/j.jcss.2013.11.002>.
8. Fernandez A. G. and Soltys M. (2013). Feasible combinatorial matrix theory, in K. Chatterjee and J. Sgall (eds.), Mathematical Foundations of Computer Science 2013, Lecture Notes in Computer Science, Vol. 8087 (Springer Berlin Heidelberg), ISBN 978-3-642-40312-5, p. 777–788, doi: 10.1007/978-3-642-40313-2_68.
9. Клейнберг Дж., Тардос Е. Алгоритмы: разработка и применение. Классика Computers Science / Пер. с англ. Е. Матвеева. — СПб.: Питер, 2016. — 800 с.: ил. — (Серия «Классика computer science»).
10. Christian B. and Griffiths T.(2016). Algorithms to Live By: the Computer Science of Human Decisions (Henry Holt and Company, LLC).

11. Деркач М.В. Застосування алгоритму Форда-Фалкерсона для карти доріг міста Сєверодонецьк / М.В. Деркач, В.О. Хишев // Вісник Східноукраїнського національного університету імені Володимира Даля. – Сєверодонецьк: СНУ ім.В.Даля, 2017. – № 8 (238). – С.25-27.
12. Skarga-Bandurova I. A Framework for Real-Time Public Transport Information Acquisition and Arrival Time Prediction Based on GPS Data / I. Skarga-Bandurova, M. Derkach, A. Velykzhanin // Dependable IoT for Human and Industry: Modeling, Architecting, Implementation (Eds. V. Kharchenko, Ah L. Kor, A. Rucinski). – River Publishers Series in Information Science and Technology, 2018. – P. 411-431.
13. Солтис М. Введение в анализ алгоритмов / пер. с англ. А. В. Логунова. – М.: ДМК Пресс, 2019. – 278 с.: ил.
14. Деркач М.В. Algorithm for dynamic route assignment in complex street layouts / М.В. Деркач, І.С. Скарга-Бандурова // Вісник Східноукраїнського національного університету імені Володимира Даля. – Сєверодонецьк: СНУ ім.В.Даля, 2019. – № 5 (253). – С.37-40.
15. Деркач М.В. Розробка алгоритму управління рухом мобільного робота / М.В. Деркач, Д.С. Матюк // Наукові вісті Дніпровського університету. Електронне наукове фахове видання. - 2019. - №17.
16. Lysak V. Implementation of Dijkstra's algorithm for Golang / V. Lysak, M. Derkach // Advanced Technologies in Research and Education: collection of research materials of the Third International Conference. – Severodonetsk: Publishing House of the Volodymyr Dahl East Ukrainian National University, 2020.

Методичне забезпечення

1. Конспект лекцій з дисципліни «Розробка та аналіз комп'ютерних алгоритмів» (для здобувачів вищої освіти 1 курсу денної та заочної форми навчання за спеціальностями 122 "Комп'ютерні науки", 123 "Комп'ютерна інженерія") / Уклад.: Сафонова С.О., Деркач М.В. – Сєверодонецьк: вид-во СНУ ім. В. Даля, 2020. - 127 с. - електронне видання.
2. Методичні вказівки до лабораторних робіт з дисципліни «Розробка та аналіз комп'ютерних алгоритмів» (для здобувачів вищої освіти 1 курсу денної та заочної форми навчання за спеціальностями 122 "Комп'ютерні науки", 123 "Комп'ютерна інженерія")/ Уклад.: Сафонова С.О., Деркач М.В. – Сєверодонецьк: вид-во СНУ ім. В. Даля, 2020. - 124 с. - електронне видання.

Оцінювання курсу

За повністю виконані завдання здобувач вищої освіти може отримати визначену кількість балів:

Інструменти і завдання	Кількість балів
Лабораторні звіти	40
Електронне тестування	20
Іспит	40
Разом	100

Шкала оцінювання здобувачів вищої освіти

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов’язковим повторним вивченням дисципліни	не зараховано з обов’язковим повторним вивченням дисципліни

Політика курсу

Плагіат та академічна доброчесність:

Здобувач вищої освіти може пройти певні онлайн-курси, які пов'язані з темами дисципліни, на онлайн-платформах. При поданні документу про проходження курсу здобувачу вищої освіти можуть бути перезараховані певні теми курсу та нараховані бали за завдання. Під час виконання завдань здобувач вищої освіти має дотримуватись політики академічної доброчесності. Запозичення мають бути оформлені відповідними посиланнями. Списування є забороненим.

Завдання і заняття:

Всі завдання, передбачені програмою курсу мають бути виконані своєчасно і оцінені в спосіб, зазначений вище. Аудиторні заняття мають відвідуватись регулярно. Пропущені заняття (з будь-яких причин) мають бути відпрацьовані з отриманням відповідної оцінки не пізніше останнього тижня поточного семестру. В разі поважної причини (хвороба, академічна мобільність тощо) терміни можуть бути збільшені за письмовим дозволом декана.

Поведінка в аудиторії:

На заняття здобувачі вищої освіти вчасно приходять до аудиторії відповідно до діючого розкладу та обов'язково мають дотримуватися вимог техніки безпеки.

Під час занять здобувачі вищої освіти:

- не вживають їжу та напої;
- не залишають аудиторію без дозволу викладача;
- не заважають викладачу проводити заняття.

Під час контролю знань здобувачі вищої освіти:

- є підготовленими відповідно до вимог даного курсу;
- розраховують тільки на власні знання (не шукають інші джерела інформації або «допомоги» інших осіб);
- не заважають іншим;
- виконують усі вимоги викладачів щодо контролю знань.