

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут _____ ІКТА _____
/назва навчально-наукового інституту/

Кафедра _____ ЕОМ _____
/назва /

«ЗАТВЕРДЖУЮ»

Голова науково-методичної комісії
Спеціальності Комп'ютерна інженерія

/назва /
_____ / А.О.Мельник /
/підпис/ /ініціали та прізвище /
« 30 » _____ серпня 2017 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

_____ Алгоритми та методи обчислень _____
/код і назва навчальної дисципліни/
_____ бакалавр _____
/рівень вищої освіти/
галузь знань _____ 12 “Інформаційні технології” _____
/шифр і назва/
спеціальність _____ 123 “Комп'ютерна інженерія” _____
/шифр і назва /
Спеціалізація _____
/шифр і назва /
вид дисципліни _____ обов'язкова _____
(обов'язкова / за вибором)
мова викладання _____ українська _____

Львів – 2017 рік

1. Структура навчальної дисципліни

Найменування показників	Всього годин	
	Денна форма навчання	Заочна форма навчання
Кількість кредитів/год.	120	-
Усього годин аудиторної роботи, у т.ч.:	64	-
• лекційні заняття, год.	32	-
• семінарські заняття, год.	-	-
• практичні заняття, год.	16	-
• лабораторні заняття, год.	16	-
Усього годин самостійної роботи, у т.ч.:	56	-
• контрольні роботи, к-сть/год.	-	-
• розрахункові (розрахунково-графічні) роботи, к-сть/год.	4	-
• індивідуальне науково-дослідне завдання, к-сть/год.	-	-
• підготовка до навчальних занять та контрольних заходів, год.	52	-
Екзамен	+	-
Залік	-	-

Частка аудиторного навчального часу студента у відсотковому вимірі:
денної форми навчання – _____; заочної форми навчання – _____.

2. Перелік попередніх та супутніх і наступних навчальних дисциплін

№ з/п	Попередні навчальні дисципліни	Супутні і наступні навчальні дисципліни
1.	Дискретна математика	Архітектура комп'ютерів
2.	Програмування, частина 1 (Основи алгоритмізації та програмування)	
3.	Програмування, частина 2 (Об'єктно орієнтоване програмування)	
4.	Програмування, частина 3 (Структури даних та алгоритми)	

3. Опис навчальної дисципліни

Основу дисципліни «Алгоритми та методи обчислень» становлять:

- Теорія алгоритмів (*в зарубіжній літературі фігурує термін теорія обчислень - англ. Theory of Computation*) [1]. Тут розглядаються:
 - Теорія автоматів (*англ. Automata theory*)
 - Теорія формальних мов (*англ. Formal language theory*)
 - Теорія обчислюваності (*англ. Computability theory*)
 - Теорія складності обчислень (*англ. Computational complexity theory*)
- Моделі обчислень (*англ. Models of Computation*) [2, 3].
- Методи розробки алгоритмів(алгоритмічні стратегії) (*англ. Algorithm Design Paradigms, Algorithmic Paradigm, Algorithmic Technique, Algorithmic Strategy*) [4].

Відповідні теоретичні розділи дисципліни посилаються на класичні літературні джерела[1, 2, 3, 4] та змістовно відповідають наступним курсам Массачусетського технологічного інституту:

- Theory of Computation(*MIT Course Number 18.404J / 6.840J*) [5];
- Automata, Computability, and Complexity(*MIT Course Number 6.045J / 18.400J*) [6].

Додатково до курсу включений розділ присвячений квантовим обчисленням, які становлять найбільш перспективний напрям розвитку комп'ютерних технологій.

Для базових алгоритмів обробки інформації та бібліотек популярних мов програмування, які їх реалізують, розглядаються зразки коду програм, в яких велику увагу приділено використанню:

- узагальненого програмування;
- метапрограмування;
- регулярних виразів та нотації Бекуса-Наура.

Також окрім імперативного(в основному процедурного та об'єктно-орієнтованого) програмування, в наведених зразках коду показано застосування:

- парадигми функційного програмування;
- парадигми реактивного програмування;
- парадигми подійно-орієнтованого програмування.

3.1. Лекційні заняття

№ з/п	Назви тем	Кількість годин	
		ДФН	ЗФН
1.	Вступ до теорії алгоритмів. 0.1. Неформальне тлумачення алгоритму. 0.1.1. Історія поняття алгоритму. 0.1.2. Визначення алгоритму. 0.1.3. Основні характеристики алгоритму. 0.1.4. Базові структури алгоритму. 0.2. Формалізація поняття алгоритму. 0.2.1. Введення в теорію алгоритмів. 0.2.2. Формалізація поняття алгоритму. Формальні алгоритмічні системи(ФАС). 0.2.3. Скінченний автомат. Детермінований скінченний автомат(DFA). Недетермінований скінченний автомат(NFA). Магазинний автомат(PDA). 0.2.4. Формальні граматики та формальні мови. 0.2.5. Машина Тюрінга.	4	-
2.	Частина 1. Тема 1. Методи відображення та синтез алгоритмів. 1. 1. Методи відображення алгоритмів. 1.1.1. Вербальне подання алгоритму. 1.1.2. Подання алгоритму псевдокодом з використанням формальних мов. 1.1.3. Схематичне подання алгоритму. 1.1.3.1. Графічне подання алгоритму. Блок-схеми. ПГА. UML. 1.1.3.2. Структурограма. Діаграма Нассі-Шнайдермана. 1. 2. Синтез алгоритмів. 1.2.1. Покрокове проектування алгоритмів. 1.2.2. Підходи при синтезі алгоритмів(алгоритмічні стратегії). 1.2.2.1. Повний перебір. 1.2.2.2. Бектрекінг (перебір з поверненням). 1.2.2.3. Метод “розділяй і пануй”. 1.2.2.4. Метод гілок і границь. 1.2.2.5. Скупі алгоритми. 1.2.2.6. Динамічне програмування.	4	-
3.	Частина 1. Тема 2. Основи аналізу алгоритмів. 2.1. Оцінка розміру вхідних даних. Складність по пам'яті. Складність по часу виконання алгоритму. 2.2. Порівняння найкращих, середніх та найгірших оцінок. 2.3. Ріст функцій. O-, o-, W-, ω-, Q- нотації. Стандартні класи ефективності алгоритмів. 2.4. Математичний аналіз нерекурсивних алгоритмів. 2.5. Математичний аналіз рекурсивних алгоритмів. 2.7. P- і NP-класів складності. NP-повнота (теорема Кука). Стандартні NP-повні проблеми. 2.8. ZPP-, RP- і BPP-класи складності. 2.6. Емпіричний аналіз алгоритмів.	2	-

4.	<p>Частина 1. Тема 3. Базові алгоритми обробки інформації.</p> <p>3.1. Алгоритми пошуку. Послідовний пошук. Бінарний пошук. Пошук в лінійних списках. Задача вибору. Дерева бінарного пошуку. Збалансовані дерева. Вичерпний пошук. Хешування. Розв'язання колізій при хешуванні відкритою адресацією та методом ланцюжків.. Порозрядний пошук. Зовнішній пошук.</p> <p>3.2. Алгоритми сортування даних. Сортування вибором. Сортування вставками. Сортування обміном. Сортування злиттям. Сортування Шелла. Швидке сортування. Пірамідальне сортування. Порозрядне та бітове сортування. Мережі сортування. Зовнішнє сортування.</p> <p>3.3. Алгоритми порівняння зі взірцем. Алгоритм Рабіна-Карпа. Пошук підрядків за допомогою скінчених автоматів. Алгоритм Кнута-Морріса-Пратта. Алгоритм Бойєра-Мура. Наближене порівняння рядків.</p> <p>3.4. Чисельні алгоритми. Матриці та дії з ними. Множення матриць по Винограду та по Штрассену. Робота з довгими числами. Алгебраїчні системи. Розв'язок систем лінійних рівнянь. Розв'язання нелінійних рівнянь. Многочлени та швидке перетворення Фур'є. Алгоритми апроксимації і інтерполяція чисельних функцій.</p> <p>3.5. Графи та мережеві алгоритми. Пошук у графі. Породження всіх каркасів графа. Каркас мінімальної ваги. Метод Дж. Краскала. Метод Р. Пріма. Досяжність. Визначення зв'язності. Двозв'язність. Ейлерові цикли. Гамільтонові цикли. Фундаментальна множина циклів. Алгоритм Дейкстри. Алгоритм Флойда. Метод генерації всіх максимальних незалежних множин графа. Задача про найменше покриття. Задача про найменше розбиття. Розфарбування графа. Пошук мінімального розфарбування вершин графа. Використання задачі про найменше покриття при розфарбуванні вершин графа. Потoki в мережах. Метод побудови максимального потоку в мережі. Методи наближеного рішення задачі комівояжера(метод локальної оптимізації, алгоритм Ейлера, алгоритм Крістофідеса). Аналіз алгоритмів на графах.</p> <p>3.6. Паралельні та розподілені алгоритми. Модель паралельного виконання програми зі спільною пам'яттю і модель передачі повідомлень. Організація паралельних обчислень відповідно до принципу консенсусу і на основі вибору. Методи визначення завершення паралельних обчислень. Паралельний пошук, паралельне сортування, паралельні чисельні алгоритми, паралельні алгоритми на графах.</p>	6	-
----	---	---	---

5.	<p>Частина 1. Тема 4. Бібліотеки основних алгоритмів обробки інформації для популярних мов програмування.</p> <p>5.1. Застосування базових алгоритмів при узагальненому програмуванні на C++ засобами STL(Standart Template Library).</p> <p>5.2. Застосування базових алгоритмів при узагальненому програмуванні на Java засобами JCL(Java Class Library).</p> <p>5.3. Застосування базових алгоритмів при узагальненому програмуванні на C# засобами FCL(Framework Class Library).</p> <p>5.4. Застосування алгоритмів лінійної алгебри при програмуванні на C++ засобами uBLAS.</p> <p>5.5. Застосування алгоритмів обробки сигналів при програмуванні на C++ засобами OpenCV(Open Source Computer Vision Library).</p>	6	
6.	<p>Частина 2. Тема 5. Моделі обчислень.</p> <p>5.1. Послідовні формальні алгоритмічні системи еквівалентні машині Тюрінга. Теза Черча.</p> <p>5.1.1. Нормальні алгоритми Маркова.</p> <p>5.1.2. Регістрова машина.</p> <p>5.1.3. РАМ-машина.</p> <p>5.2. Функційні моделі обчислень та парадигма функційного програмування.</p> <p>5.2.1. Функційні моделі обчислень.</p> <p>5.2.1.1. Лямбда числення.</p> <p>5.2.1.2. Типізоване лямбда числення.</p> <p>5.2.1.3. Рекурсивні функції.</p> <p>5.2.1.4. Комбінація логіки(як функційна модель обчислень).</p> <p>5.2.1.5. Клітковий автомат.</p> <p>5.2.1.6. Абстрактна перезаписуюча система.</p> <p>5.2.1. Парадигма функційного програмування.</p> <p>5.3. Паралельні моделі обчислень та парадигма реактивного програмування. Паралельне програмування.</p> <p>5.3.1. Паралельні моделі обчислень.</p> <p>5.3.1.1. ПРАМ-машина. ПРАМ-машина на основі ПВДН.</p> <p>5.3.1.2. Мережа процесів Кана.</p> <p>5.3.1.3. Мережа Петрі.</p> <p>5.3.1.4. Синхронний потік даних та парадигма реактивного програмування.</p> <p>5.3.1.5. Мережа взаємодій.</p> <p>5.3.2. Парадигма реактивного програмування.</p> <p>5.3.3. Паралельне програмування.</p>	8	-
7.	<p>Частина 3. Тема 6. Квантові обчислення.</p> <p>6.1. Основні поняття і принципи квантових обчислень. Квантова машина Тюрінга і BQP-клас складності.</p> <p>6.2. Кубіти, квантові вентиля та квантові регістри.</p> <p>Вимірювання значень кубітів. Система кубітів та квантова запутаність. Вентиль тотожного перетворення. Вентиль заперечення. Вентиль фазового зміщення. Вентиль перетворення Адамара. Прямий керуючий вентиль. Вентиль контрольованого заперечення. Вентиль Тоффолі. Вентиль Фредкіна.</p> <p>6.3. Функціонування квантової системи.</p> <p>Задача Дойча. Задача Дойча-Джозі. Задача Бернштайна-Вазірані. Задача Саймона.</p> <p>6.4. Важливі алгоритми квантових обчислень.</p> <p>Алгоритм Шора. Алгоритм Гровера.</p>	2	-
Усього годин		32	-

3.2. Практичні заняття

№ з/п	Назви тем	Кількість годин	
		ДФН	ЗФН
1.	Практичне заняття №1. Вступне заняття. Огляд матеріалу практичних занять. Поняття алгоритму.	2	-
2.	Практичне заняття №2. Алгоритм; властивості, параметри та характеристики складності алгоритму. <u>Завдання:</u> Порівняти складність арифметичних операцій в римській та десятковій системах числення. <i>(виконується як домашня робота)</i>	2	-
3.	Практичне заняття №3. Вплив правила безпосереднього перероблення на характеристики складності алгоритму. <u>Завдання:</u> порівняти часову складність трьох алгоритмів знаходження НСД. <i>(виконується як домашня робота)</i>	2	-
4.	Практичне заняття №4. Параметри алгоритму. Правило безпосереднього перероблення. Асимптотичні характеристики складності алгоритму. Алгоритми з поліноміальною та експоненціальною складністю. <u>Завдання:</u> Визначити часову складність заданого алгоритму в “найгіршому випадку”. <i>(виконується як контрольна робота в кінці пари)</i>	2	-
5.	Практичне заняття №5. Використання потокового графу алгоритму при проектуванні паралельних обчислень. <u>Завдання:</u> розпаралелити алгоритм шляхом формування потокового графу алгоритму. <i>(виконується як контрольна робота в кінці пари)</i>	2	-
6.	Практичне заняття №6. Формальні алгоритмічні системи (ФАС). Машина Тюрінга (МТ). <u>Завдання:</u> Побудувати алгоритм для МТ(сформувати “слід” МТ). Підрахувати часову, програмну та місткісну складність. <i>(виконується як контрольна робота в кінці пари)</i>	2	-
7.	Практичне заняття №7. Побудова алгоритмів ефективних за часовою складністю. <u>Завдання:</u> застосувати метод «гілок і границь» при побудові алгоритму для вирішення задачі квадратичного призначення. <i>(виконується як контрольна робота в кінці пари)</i>	2	-
8.	Захист домашніх завдань.	2	-
Усього годин		16	-

3.3. Лабораторні заняття.

№ з/п	Назви тем	Кількість годин	
		ДФН	ЗФН
1.	Вступне заняття. Інструктаж. Видача завдань.	2	-
2.	Виконання лабораторної №1. Алгоритм; властивості, параметри та характеристики складності алгоритму.	2	-
3.	Виконання лабораторної №2. Асимптотичні характеристики складності алгоритму; алгоритми з поліноміальною та експоненціальною складністю.	2	-
4.	Виконання лабораторної №3. Використання потокового графу алгоритму при проектуванні паралельних обчислень.	2	-
5.	Виконання лабораторної №4. Побудова алгоритмів ефективних за часовою складністю; задача квадратичного призначення.	2	-
6.	Виконання лабораторної №5. Функційне програмування.	2	-
7.	Виконання лабораторної №6. Реактивне програмування.	2	-
8.	Заключне заняття. Кінцевий термін захисту лабораторних робіт.	2	-
Усього годин		16	-

3.4. Самостійна робота

№ з/п	Найменування робіт	Кількість годин	
		ДФН	ЗФН
1.	Опрацювання лекційного матеріалу	20	-
2.	Опрацювання матеріалу практичних занять	12	
3.	Підготовка до лабораторних робіт	12	-
4.	Виконання розрахункової роботи	4	
5.	Підготовка до екзамену	8	-
Усього годин		56	-

4. Методи діагностики знань

Для оцінки знань студентів передбачається екзаменаційний контроль та поточний контроль. Поточний контроль передбачає виконання лабораторних робіт, домашніх завдань і контрольних робіт. Разом за поточний контроль студент може отримати максимум 30 балів. Екзаменаційний контроль складається з письмової компоненти(60 балів) та усної компоненти(10 балів).

Також необхідною умовою допуску до екзамену є виконання усіх лекційних контрольних робіт та лекційних домашніх завдань(починаючи з дев'ятої пари), за які оцінка не виставляється, а проставляється відмітка про успішне виконання. У випадку не зарахування виконання кожного домашнього завдання можна здати повторно, а контрольні роботи написати повторно на наступній лекції.

5. Критерії оцінювання результатів навчання студентів

Максимальна оцінка в балах															
Разом за дисципліну				Екзаменаційний контроль		Підготовка до екзамену				Поточний контроль (ПК)					
				усна компонента						письмова компонента					
								Разом за ПК				Виконання лабораторних робіт			
												Практичних заняття			

Література

- 1) Michael Sipser (2013). *Introduction to the Theory of Computation*. 3rd. Cengage Learning. ISBN 978-1-133-18779-0
- 2) Savage, John E. (1998). *Models Of Computation: Exploring the Power of Computing*. ISBN 978-0-201-89539-1
- 3) Fernández, Maribel (2009). *Models of Computation: An Introduction to Computability Theory*. Undergraduate Topics in Computer Science. Springer. ISBN 978-1-84882-433-1.
- 4) Anany Levitin (2012). *Introduction to the design & analysis of algorithms*. 3rd. ISBN-13: 978-0-13-231681-1.
- 5) <https://ocw.mit.edu/courses/mathematics/18-404j-theory-of-computation-fall-2006/>
- 6) <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-045j-automata-computability-and-complexity-spring-2011/>