

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ
КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ПРОГРАМНИХ СИСТЕМ**

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

_____ Кашпур О.Ф.

«___» _____ 2019 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
АЛГОРИТМИ ТА СКЛАДНІСТЬ
для студентів**

галузь знань	12 Інформаційні технології
спеціальність	121 Інженерія програмного забезпечення
освітній рівень	бакалавр
освітня програма	Програмна інженерія
вид дисципліни	обов'язкова

Форма навчання	денна
Навчальний рік	2020/2021
Семестр	3,4
Кількість кредитів ECTS	7
з них семестр 3	3
семестр 4	4
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладачі: **к.ф.-м.н., доц. Шкільняк О.С.** (лекції)
к.ф.-м.н., ас. Ходзінський О.М. (лабораторні заняття)
к.ф.-м.н., ас. Ліндер Я.М. (лабораторні заняття)

Пролонговано: на 20__/20__ н.р. _____ (_____) «__» __ 20__ р.
(підпис, ПІБ, дата)

на 20__/20__ н.р. _____ (_____) «__» __ 20__ р.
(підпис, ПІБ, дата)

КИЇВ 2019

Розробник: Шкільняк Оксана Степанівна, к. ф.-м. н., доцент, доцент кафедри інтелектуальних програмних систем.

ЗАТВЕРДЖЕНО

Завідувач кафедри інтелектуальних програмних систем

_____ О.І. Провотар

Протокол № __ від «__» _____ 2019 р.

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від «__» _____ 2019 року №__

Голова науково-методичної комісії _____ Л.Л. Омельчук

«__» _____ 2019 року

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «__» _____ 2019 року №__

Голова вченої ради факультету _____ А.В. Анісімов

1. Мета дисципліни – вивчення основних структур даних і алгоритмів, що використовуються в програмуванні та аналіз їх ефективності.

2. Попередні вимоги до опанування навчальної дисципліни

1. Знати: базові поняття алгебри, математичного аналізу, дискретної математики.
2. Вміти: програмувати на мові високого рівня.

3. Анотація навчальної дисципліни

Навчальна дисципліна «Алгоритми та складність» є складовою освітньо-професійної програми підготовки фахівців за першим (бакалаврським) рівнем вищої освіти галузі знань 12 Інформаційні технології за спеціальністю 121 Інженерія програмного забезпечення освітньо-професійної програми „Програмна інженерія”.

Дисципліна належить до обов’язкових навчальних дисциплін. Викладається у **3 та 4 семестрах 2 курсу в обсязі – 210 год. (7 кредитів ECTS)**, зокрема: лекції – 56 год., лабораторні заняття – 42 год., консультації – 4 год., самостійна робота – 108 год. У курсі передбачено 4 частини та 4 контрольні роботи, **залік в 3 семестрі та іспит в 4 семестрі**.

В результаті вивчення навчальної дисципліни студент повинен:

знати основні структури даних і алгоритми, що використовуються в програмуванні та їх ефективність; методи аналізу складності алгоритмів;

вміти проводити аналіз ефективності алгоритмів; вибирати оптимальні структури даних та алгоритми при розв’язанні конкретних задач; за необхідності створювати власні алгоритми, що базуються на модифікації відомих технік і обґрунтовувати їх ефективність.

Дисципліна є логічним продовженням, доповненням та розширенням дисциплін «Програмування» та «Дискретна математика» і виступає базовою для дисципліни «Обчислювальна геометрія та комп’ютерна графіка» блоку вибору «Програмна інженерія» однойменної освітньої програми.

4. Завдання (навчальні цілі). Основними завданнями дисципліни «Алгоритми та складність» є набуття знань, умінь та навичок (компетенцій) на рівні новітніх досягнень в галузі дослідження алгоритмів та їх складності відповідно до освітньої кваліфікації бакалавр з програмної інженерії.

Зокрема, розвивати:

- Здатність до абстрактного мислення, аналізу та синтезу (ЗК01).
- Здатність застосовувати знання у практичних ситуаціях (ЗК02).
- Здатність спілкуватися державною мовою як усно, так і письмово (ЗК03).
- Здатність вчитися і оволодівати сучасними знаннями (ЗК05).
- Здатність до пошуку, оброблення та аналізу інформації з різних джерел (ЗК06).
- Здатність до алгоритмічного та логічного мислення (СК14).

5. Результати навчання за дисципліною.

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
РН1.1	Знати основні структури даних і алгоритми, що використовуються в програмуванні та їх ефективність	Лекція, лабораторне заняття, самостійна робота	Контрольна робота, на 60% правильно виконана, іспит	50%
РН1.2	Знати методи аналізу складності алгоритмів	Лекція, лабораторне заняття, самостійна робота	Контрольна робота, на 60% правильно виконана, іспит	
РН2.1	Вміти проводити аналіз ефективності алгоритмів	Лекція, лабораторне заняття, самостійна робота	Контрольна робота, на 60% правильно виконана, захист лабораторної роботи, іспит	10%
РН2.2	Вміти вибирати оптимальні структури даних та алгоритми при розв'язанні конкретних задач; за необхідності створювати власні алгоритми, що базуються на модифікації відомих технік і обґрунтовувати їх ефективність; реалізовувати розроблені алгоритми.	Лабораторне заняття, самостійна робота	Захист лабораторної роботи, іспит	20%
РН3.1	Обґрунтовувати власний погляд на розв'язання задачі, спілкуватися з колегами з питань проектування та розробки програм, складати письмові звіти	Лабораторне заняття, самостійна робота	Поточне оцінювання, захист лабораторної роботи	10%
РН4.1	Організувати свою самостійну роботу для досягнення результату	Самостійна робота	Поточне оцінювання, захист лабораторної роботи	5%
РН4.2	Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість	Лабораторне заняття	Захист лабораторної роботи	5%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання.

Результати навчання дисципліни	РН1.1	РН1.2	РН2.1	РН2.2	РН3.1	РН4.1	РН4.2
Програмні результати навчання							
ПРН01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.	+	+				+	
ПРН05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.	+	+	+	+			
ПРН06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.			+	+	+		+
ПРН13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.	+			+	+		+

7. Схема формування оцінки.

7.1. Форми оцінювання студентів.

– семестрове оцінювання (максимальна кількість балів):

Семестр 3

1. Контрольна робота 1: РН 1.1, РН 1.2, РН 2.1 – **30/18 б.**
2. Контрольна робота 2: РН 1.1, РН 1.2, РН 2.1 – **20/12 б.**
3. Лабораторний проект: РН 1.1-1.2, РН 2.1-2.2, РН 3.1, РН 4.1-4.2 – **50/30 б.**

Підсумкове оцінювання (у формі заліку):

- Залікові бали визначаються як сума оцінок/балів за всіма успішно оціненими результатами навчання передбачених даною програмою.
- Оцінки нижче від мінімального порогового рівня не додаються.
- Мінімальний пороговий рівень для сумарної оцінки за всіма компонентами становить 60% від максимально можливої кількості балів.

Семестр 4

1. Контрольна робота 1: РН 1.1, РН 1.2, РН 2.1 – **20/12 б.**
2. Контрольна робота 2: РН 1.1, РН 1.2, РН 2.1 – **10/6 б.**
3. Лабораторний проект: РН 1.1-1.2, РН 2.1-2.2, РН 3.1, РН 4.1-4.2 – **30/18 б.**

Підсумкове оцінювання (у формі іспиту):

– максимальна кількість балів які можуть бути отримані студентом: 40 балів;

- підсумкова оцінка визначається як сума оцінок/балів за всіма успішно оціненими результатами навчання передбачених даною програмою (оцінки нижче від мінімального порогового рівня до підсумкової оцінки не додаються) та оцінка, що отримана під час іспиту;
- результати навчання які будуть оцінюватись: РН 1.1, РН 1.2, РН 2.1, РН 2.2;
- форма проведення і види завдань: письмова-усна форма, 5 завдань по 8 балів (20%) кожне;
- для отримання загальної позитивної оцінки з дисципліни оцінка за іспит **не може бути меншою 24 балів.**

Здобувач освіти може бути **недопущений** до підсумкового оцінювання, якщо під час семестру він:

- 1) не досяг мінімального порогового рівня (60%) оцінки тих результатів навчання, які не можуть бути оцінені під час підсумкового контролю;
- 2) набрав кількість балів, що є недостатньою для отримання позитивної оцінки навіть у випадку досягнення ним на підсумковому контролі максимально можливого результату.

7.2. Організація оцінювання.

Терміни проведення форм оцінювання в 3 семестрі:

1. Контрольна робота 1: до 9 тижня семестру.
2. Контрольна робота 2: до 14 тижня семестру.
3. Лабораторний проект: контроль на кожному лабораторному занятті, починаючи з другого.

Терміни проведення форм оцінювання в 4 семестрі:

1. Контрольна робота 1: до 8 тижня семестру.
2. Контрольна робота 2: до 14 тижня семестру.
3. Лабораторний проект: контроль кожні два тижні, до 14 тижня семестру.

Студент має право один раз перескласти контрольну роботу з можливістю отримати не більше 80% балів, призначених за роботу. Термін перескладання визначається викладачем.

У випадку відсутності студента з поважних причин відпрацювання та перездачі контрольних робіт здійснюються у відповідності до „Положення про порядок оцінювання знань студентів при кредитно-модульній системі організації навчального процесу” від 1 жовтня 2010 року.

У разі неякісного або невчасного виконання лабораторної роботи викладач має право не зараховувати завдання або знизити за нього бали.

7.3. Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59
Зараховано / Passed	60-100
Не зараховано / Fail	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій.

СЕМЕСТР 5

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лабор. заняття	С/р
Частина 1. Вступ до аналізу алгоритмів. Ефективні сортування				
1	Тема 1. Вступ до предмету. Поняття алгоритму та аналізу алгоритмів	2		2
2	Тема 2. Зростання функцій	2	2	4
3–4	Тема 3. Аналіз нерекурсивних алгоритмів	4	2	8
5–6	Тема 4. Аналіз рекурсивних алгоритмів. Рекурентні співвідношення	4	2	8
7–9	Тема 5. Ефективні сортування	6	2	6
Контрольна робота №1				2
		18	8	30
Частина 2. Застосування різних методів проектування алгоритмів				
10–11	Тема 6. Декомпозиція та зменшення розміру задачі	4	2	6
12–14	Тема 7. Покращення вхідних даних. Алгоритми пошуку підрядка	6	4	8
Контрольна робота №2				2
		10	6	16
ВСЬОГО		28	14	46

Загальний обсяг – **90 год.**, у тому числі:

Лекцій – **28 год.**,

Консультацій – **2 год.**,

Лабораторних занять – **14 год.**,

Самостійна робота – **46 год.**

Теми, винесені на самостійне вивчення

Використання границь для обчислення порядку зростання функцій.

Алгоритми сортування, що мають квадратичну складність (вставкою, вибором, бульбашковий).

Зовнішнє багатофазне сортування злиттям.

Варіанти алгоритмів Карацуби та Штрассена.

Пошук підрядків з використанням скінченних автоматів.

Z-функція та префікс-функція.

Перелік питань для підготовки до заліку

1. Поняття алгоритму.
2. Алгоритм як технологія: процес проектування та аналізу.
3. Основні базові структури та абстрактні типи даних.
4. Ефективність алгоритму в різних випадках: найкращому, найгіршому, в середньому.
5. Порядок зростання функцій.
6. Асимптотичні позначення: O , Ω , Θ , o та ω .

7. Порівняння функцій.
8. Основні асимптотичні класи ефективності.
9. P, NP та NP-повні задачі.
10. Загальна схема аналізу рекурсивних алгоритмів.
11. Поняття інваріанту циклу. Використання інваріантів циклу для доведення коректності алгоритму.
12. Метод декомпозиції (принцип "розділяй та владарюй").
13. Поняття рекурентного співвідношення. Розв'язання рекурентних співвідношень.
14. Методи розв'язання рекурентних співвідношень: методи підстановок, метод дерев рекурсії, основний метод, метод Акра-Баззі.
15. Сортування злиття, його коректність та оцінка складності. Переваги і недоліки.
16. Зовнішнє багатofазне сортування злиттям.
17. Бінарна піраміда як структура даних та її властивості.
18. Пірамідальне сортування, його коректність і оцінка складності.
19. Реалізація черги з пріоритетами на основі піраміди.
20. Швидке сортування: алгоритм та оцінка складності. Переваги і недоліки. Варіації алгоритму швидкого сортування.
21. Нижня оцінка алгоритмів сортування, що використовують порівняння. Асимптотично оптимальні алгоритми сортування.
22. Алгоритми сортування, що мають квадратичну складність.
23. Алгоритми сортування за лінійний час та їх аналіз: сортування підрахунком, порозрядне сортування, сортування черпаками.
24. Медіани та порядкові статистики, алгоритми їх пошуку за лінійний час. Порівняння цих алгоритмів з алгоритмами сортування.
25. Приклади алгоритмів зі змінним зменшенням розміру задачі.
26. Бінарний та інтерполяційних пошуки і методи дихотомії та хорд розв'язання нелінійних рівнянь: спільне та відмінне.
27. Алгоритм Карацуби множення великих чисел.
28. Метод Штрассена множення великих матриць.
29. Задача пошуку підрядка. Різноманіття підходів до її розв'язання.
30. Алгоритм Боера-Мура та його варіації: алгоритм Хорспула, алгоритм Райти.
31. Z-функція та префікс-функція.
32. Алгоритм Кнута-Морріса-Пратта.
33. Пошук підрядків з використанням скінченних автоматів.

СЕМЕСТР 4

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лабор. заняття	С/р
Частина 3. Структури даних				
15	Тема 8. Елементарні структури даних. Хеш-таблиці	2	2	4
16–18	Тема 9. Бінарні дерева пошуку. Збалансовані дерева пошуку. Розширення структур даних	6	6	12
19	Тема 10. В-дерева	2	2	4
20–21	Тема 11. Піраміди злиття	4	4	8
22	Тема 12. Дерева ван Емде Боаса	2	2	4
Контрольна робота №3				2
		16	16	34

Частина 4. Алгоритми на графах				
23	Тема 13. Представлення графів. Елементарні алгоритми на графах	2	2	4
24–25	Тема 14. Жадібні алгоритми. Пошук мінімального кістякового дерева	4	4	9
26	Тема 15. Пошук найкоротших шляхів з однієї вершини	2	2	4
27–28	Тема 16. Динамічне програмування. Пошук найкоротших шляхів між усіма парами вершин	4	4	9
Контрольна робота №4				2
		12	12	28
ВСЬОГО		28	28	62

Загальний обсяг– *120 год.*, у тому числі:

Лекцій – *28 год.*,

Консультацій – *2 год.*,

Лабораторних занять – *28 год.*,

Самостійна робота – *62 год.*

Теми, винесені на самостійне вивчення

Аналіз побудови універсальної множини хеш-функцій.

Аналіз ідеального хешування.

Випадкова побудова бінарних дерев пошуку.

Дерева відрізків.

В+-дерева.

2-3-4-піраміди.

Точки з'єднання, мости, двозв'язні компоненти графа. Ейлерові цикли.

Дерева Хаффмана і коди Хаффмана.

Доведення властивостей найкоротших шляхів.

Алгоритм Джонсона для розріджених графів.

Перелік питань для підготовки до іспиту

1. Динамічні множини та типові операції на них.
2. Елементарні структури даних (стеки, черги, зв'язані списки).
3. Реалізація вказівників та об'єктів, управління пам'яттю.
4. Способи представлення дерев з коренем.
5. Хеш-таблиці та їх застосування.
6. Таблиці з прямою адресацією.
7. Колізії та методи їх розв'язання.
8. Хеш-функції. Методи побудови хеш-функцій: метод поділу, метод множення, універсальне хешування.
9. Відкрита адресація: методи і їх аналіз.
10. Ідеальне хешування.
11. Дерево пошуку як структура даних, операції над ним.
12. Збалансовані дерева пошуку. Огляд основних їх різновидів.
13. Червоно-чорні дерева, їх властивості, представлення і аналіз. Повороти. Відновлення червоно-чорних властивостей.
14. Розширення структур даних.

15. Дерево порядкової статистики як розширення червоно-чорних дерев. Теорема про розширення червоно-чорних дерев.
16. Персистентні динамічні множини.
17. Структури даних у вторинній пам'яті.
18. В-дерева як узагальнення дерев пошуку: означення і властивості В-дерева.
19. Основні операції над В-деревом, їх складність. Розбиття вузла.
20. Різновиди В-дерев.
21. Піраміди злиття.
22. Біноміальні дерева, лема про їх властивості.
23. Біноміальна піраміда. Представлення біноміальних пірамід.
24. Основні операції над біноміальними пірамідами. Злиття біноміальних пірамід.
25. Піраміди Фібоначчі: означення і представлення. Функція потенціалу для пірамід Фібоначчі.
26. Невпорядковані біноміальні дерева, їх властивості.
27. Основні операції над пірамідами Фібоначчі.
28. Оцінка максимальної степені вузла в піраміді Фібоначчі.
29. Дерево ван Емде Боаса: представлення, основні операції та їх складність.
30. Переваги і недоліки дерева ван Емде Боаса.
31. Поняття графа. Способи представлення графів.
32. Пошук в глибину і пошук в ширину. Класифікація ребер.
33. Топологічне сортування.
34. Пошук сильно зв'язних компонент графа.
35. Поняття Ейлерового циклу.
36. Жадібна стратегія розв'язання задач. Особливості застосування підходу.
37. Приклади жадібних алгоритмів.
38. Коди Хаффмана.
39. Кістякове дерево і мінімальне кістякове дерево графа.
40. Алгоритми Прима та Крускала.
41. Постановки задач про найкоротші шляхи.
42. Властивості найкоротших шляхів.
43. Найкоротші шляхи та ослаблення (релаксація).
44. Алгоритм Беллмана-Форда.
45. Алгоритм Дейкстри.
46. Поняття про динамічне програмування та його особливості.
47. Основні задачі динамічного програмування.
48. Пошук найкоротших шляхів між усіма парами вершин. Алгоритм Флойда.
49. Пошук транзитивного замикання графа. Алгоритм Воршелла.
50. Динамічне програмування і жадібна стратегія: порівняння.

9. Рекомендовані джерела.

Основні

1. Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Стайн. Алгоритмы – построение и анализ. Третье издание. – М.: ИД "Диалектика-Вильямс", 2019.
2. А. Левитин. Алгоритмы: введение в разработку и анализ. – М.: ИД "Вильямс", 2006.
3. Д. Кнут. Искусство программирования. Т.1: Основные алгоритмы. – М.: Мир, 1976.
4. Д. Кнут. Искусство программирования. Т.2: Получисленные алгоритмы. – М.: Мир, 1976.

5. Д. Кнут Искусство программирования. Т.3: Сортировка и поиск. – М.: Мир, 1976.

Додаткові

6. А. Ахо, Дж. Хопкрофт, Дж. Ульман. Структуры данных и алгоритмы. – М.: ИД "Вильямс", 2000.

7. А. Ахо, Дж. Хопкрофт, Дж. Ульман. Построение и анализ вычислительных алгоритмов. – М.: Мир, 1979.

8. Дж. Макконнелл. Основы современных алгоритмов. Второе издание. – М.: Техносфера, 2004.

9. Р. Седжвик. Фундаментальные алгоритмы на C++. – М.: DiaSoft, 2001.

10. Д. Гасфилд. Строки, деревья и последовательности в алгоритмах. Информатика и вычислительная биология. – СПб.: ВHV-СПб, 2008.