

Міністерство освіти і науки України
Харківський національний університет імені В.Н. Каразіна
Кафедра штучного інтелекту та програмного забезпечення



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Теорія алгоритмів

рівень вищої освіти	<u>перший (бакалаврський) рівень</u>
галузь знань	<u>12 Інформаційні технології</u>
спеціальність	<u>122 Комп'ютерні науки</u>
освітня програма	<u>Комп'ютерні науки</u>
вид дисципліни	<u>обов'язкова</u>
факультет	<u>комп'ютерних наук</u>

Програму рекомендовано до затвердження Вченою радою факультету комп'ютерних наук
«29» серпня 2018 року, протокол № 9

РОЗРОБНИКИ ПРОГРАМИ:

кандидат технічних наук, доцент кафедри штучного інтелекту та програмного
забезпечення **Олешко Олег Іванович**, кандидат технічних наук, доцент кафедри
моделювання систем та технологій **Щебенюк Володимир Сергійович**.

Програму схвалено на засіданні кафедри штучного інтелекту та програмного забезпечення
Протокол від «26» червня 2018 року № 11

Завідувач кафедри штучного інтелекту та програмного забезпечення



Куклін В.М.

Програму погоджено методичною комісією факультету комп'ютерних наук
Протокол від «27» червня 2018 року № 7

Голова методичної комісії факультету комп'ютерних наук



(Васильєва Л.В.)



ВСТУП

Програма навчальної дисципліни "Теорія алгоритмів" складена відповідно до освітньо-професійної програми підготовки першого (бакалаврського) рівня вищої освіти, спеціальності 122 «Комп'ютерні науки».

1. Опис навчальної дисципліни

1.1. Метою викладання навчальної дисципліни є засвоєння студентами знань та навичок по використанню та складанню алгоритмів, спеціалізованих для вирішення різних класів задач.

1.2. Основними завданнями вивчення дисципліни є:

- формування та обробка складних структур даних на основі базових типів;
- ознайомлення з ефективними методами зберігання інформації;
- розгляд великого класу алгоритмів для сортування даних;
- ознайомлення з особливостями розв'язання задач із графами;
- вивчення методів складання алгоритмів та пошуку найбільш ефективного для вирішення того чи іншого типу задач.

1.3. Кількість кредитів – 6

1.4. Загальна кількість годин – 180

1.5. Характеристика навчальної дисципліни	
Нормативна / за вибором	
денна форма навчання	Заочна (дистанційна) форма навчання
Рік підготовки	
2-й	-й
Семестр	
3-й	-й
Лекції	
32 год.	_____ год.
Практичні, семінарські заняття	
_____ год.	_____ год.
Лабораторні заняття	
32 год.	_____ год.
Самостійна робота	
76 год.	_____ год.
Індивідуальні завдання	
40 год.	_____ год.

1.6. Заплановані результати навчання

знати:

- алгоритми складання словників та кодування тексту;
- ефективні алгоритми пошуку інформації у масивах даних;
- алгоритми сортування;
- алгоритми вирішення завдань, пов'язаних із графами;

- методи розробки алгоритмів.

вміти:

- застосовувати складні структури даних, методи та алгоритми для обробки різнорідної інформації та вирішування складних обчислювальних завдань.

2. Тематичний план навчальної дисципліни

Розділ 1. Складні структури даних.

Тема 1. Коди Хаффмана.

Аналіз текстів та складання кодів кожного символу за алгоритмом Хаффмана. Повторення вмінь по використанню складених структур даних (запис, дерево, динамічний масив дерев) та роботі з текстовими файлами.

Тема 2. Навантажене дерево.

Складання словника за даним набором слів. Повторення та закріплення вмінь по використанню складених структур даних (запис, дерево, односпрямований список) та роботі з текстовими файлами.

Тема 3. Хешування.

Зберігання інформації з швидким пошуком. Методи розв'язання колізій за допомогою закритого та відкритого хешування та методу ланцюгів (масив односпрямованих списків).

Тема 4. Піраміда.

Принцип черги з пріоритетом на базі алгоритму піраміди.

Розділ 2. Алгоритми сортування

Тема 5. Поняття сортування. Елементарні методи сортування. Швидке сортування.

Сортування масиву простими методами (вибором, вставками, бульбашковим сортуванням). Повторення рекурсивного методу для швидкого сортування.

Тема 6. Піромідалне сортування. Сортування злиттям.

Використання алгоритму піраміди для сортування масиву. Вирішення проблеми сортування великого масиву, складеного з двох відсортованих, за допомогою алгоритму злиття.

Тема 7. Кишенькове сортування. Сортування підрахунком. Зовнішнє сортування.

Інші методи сортування. Вирішення проблеми сортування масиву, що за розмірами перевищує об'єм динамічної пам'яті (зовнішнє сортування у файлах).

Розділ 3. Алгоритми на графах

Тема 8. Орієнтований, неорієнтований граф. Зважений граф.

Способи запису орієнтованих та неорієнтованих графів у комп'ютерній програмі. Поняття ваги ребра графа.

Тема 9. Пошук найкоротшого шляху в орієнтованому графі.

Алгоритми Дейкстри та Беллмана-Форда пошуку найкоротших шляхів від однієї вершини графа до інших.

Тема 10. Основні дерева мінімальної вартості.

Алгоритми Крускала та Пріма для пошуку остовних дерев мінімальної вартості в неорієнтованому графі.

Розділ 4. Методи розробки алгоритмів

Тема 11. Динамічне програмування.

Вирішення великої задачі шляхом виконання часткових завдань та зберігання відповідей.

Тема 12. "Жадібні" алгоритми.

Приклади "жадібних" алгоритмів з попередніх тем. Задача комівояжера.

Тема 13. Пошук з поверненням.

Пошук часткових рішень у задачах з перестановками.

3. Структура навчальної дисципліни

Назви розділів і тем	Кількість годин					
	Денна форма					
	Усього	у тому числі				
л		п	лаб	інд	сп	
1	2	3	4	5		7
Розділ 1. Складні структури даних						
Тема 1. Коди Хаффмана.	9	2		2		5
Тема 2. Навантажене дерево.	9	2		2		5
Тема 3. Хешування.	9	2		2		5
Тема 4. Піраміда.	9	2		2		5
Разом за розділом 1	36	8		8		20
Розділ 2. Алгоритми сортування						
Тема 5. Поняття сортування. Елементарні методи сортування. Швидке сортування.	16	4		4		8
Тема 6. Пірамідальне сортування. Сортування злиттям.	18	4		4		10
Тема 7. Кишенькове сортування. Сортування підрахунком. Зовнішнє сортування.	16	4		4		8
Разом за розділом 2	50	12		12		26
Розділ 3. Алгоритми на графах						
Тема 8. Орієнтований, неорієнтований граф. Зважений граф.	13	4		4		5
Тема 9. Пошук найкоротшого шляху в орієнтованому графі.	9	2		2		5
Тема 10. Остовні дерева мінімальної вартості.	9	2		2		5
Разом за розділом 3	31	8		8		15
Розділ 4. Методи розробки алгоритмів						
Тема 11. Динамічне програмування.	7	1		1		5
Тема 12. "Жадібні" алгоритми.	7	1		1		5
Тема 13. Пошук з поверненням.	9	2		2		5
Разом за розділом 4	23	4		4		15
Курсова робота	40				40	
Усього годин	180	32		32	40	76

4. Теми практичних (лабораторних) занять

№ з/п	Назва теми	Кількість годин
Розділ 1. Складні структури даних		
1	Коди Хаффмана.	2
2	Навантажене дерево.	2
3	Хешування.	2
4	Піраміда.	2
Розділ 2. Алгоритми сортування		
1	Елементарні методи сортування: вибором, вставками, бульбашкове.	4
2	Пірамідальне сортування. Сортування злиттям.	4
3	Кишенькове сортування. Сортування підрахунком.	4
Розділ 3. Алгоритми на графах		
1	Орієнтований та неорієнтований зважені графи.	4
2	Пошук найкоротшого шляху в графі (алгоритм Беллмана-Форда).	2
3	Остовні дерева мінімальної вартості (алгоритм Прима).	2
Розділ 4. Методи розробки алгоритмів		
1	Динамічне програмування.	1
2	"Жадібні" алгоритми.	1
3	Пошук з поверненням.	2
	Разом	32

5. Завдання для самостійної роботи

№ з/п	Види, зміст самостійної роботи	Кількість годин
Розділ 1. Складні структури даних		
1	Для заданого тексту розробити словник на основі алгоритму навантаженого дерева. Реалізувати функцію перевірки тексту на орфографію. Підготовка до контрольної роботи	20
Розділ 2. Алгоритми сортування		
1	Розробити програму з використанням внутрішніх алгоритмів сортування (вибіркове, вставкою, "бульбашкове", швидке). Порівняти їх ефективності за швидкістю сортування для різних вихідних даних.	10
2	Реалізувати алгоритм зовнішнього сортування на основі алгоритму злиття. Відсортувати значення у заданому типізованому файлі з використанням додаткової пам'яті та без неї.	16
Розділ 3. Алгоритми на графах		
1	Роздрукувати найкоротші шляхи в орієнтованому графі з однієї вершини до всіх інших за допомогою алгоритму Дейкстри.	15
Розділ 4. Методи розробки алгоритмів		
1	Використати метод пошуку з поверненням для вирішення задачі розташування ферзів на дошці заданого розміру.	15
	Курсова робота	40
	Разом	116

6. Індивідуальні завдання (курсова робота)

Як наукове та навчально-дослідне завдання кожен студент має виконати курсону роботу, яка передбачає розробку комп'ютерної програми, що реалізує алгоритм, обраний

студентом із списку запропонованих:

1. *Хешування.*
Реалізувати алгоритми відкритого, закритого хешування та хешування методом ланцюжків.
2. *2-3-дерево.*
Реалізувати основні операції для 2-3-дерева: додавання, видалення, друк, пошук значення.
3. *Червоно-чорне дерево.*
Реалізувати основні операції для червоно-чорного дерева: додавання, видалення, друк, пошук значення.
4. *АВЛ-дерево.*
Реалізувати основні операції для АВЛ-дерева: додавання, видалення, друк, пошук значення, балансування.
5. *Сильно пов'язані компоненти.*
В орієнтованому графі знайти всі пов'язані компоненти та їх кількість.
6. *Порядкові статистики.*
В заданому масиві шукати заданий за номером максимальний та заданий мінімальний елементи.
7. *Фільтр Блума.*
Реалізувати Фільтр Блума з декількома хеш-функціями. Проаналізувати роботу алгоритма для різних параметрів (розміру фільтра, кількості елементів, кількості хеш-функцій).
8. *Дераміда.*
Реалізувати основні операції для дераміди: додавання, видалення, друк, пошук значення, пошук пріоритету заданого значення.
9. *Алгоритм множення матриць Штрассена.*
Реалізувати алгоритми множення матриць звичайним способом та способом Штрассена. Порівняти алгоритми за швидкістю для матриць рідного розміру.
10. *Тест Міллера-Рабіна.*
Для заданого числа визначити, чи складне воно, чи просте.
11. *Алгоритм Кнута-Морріса-Пратта.*
У заданому рядку шукати всі входження іншого заданого рядку. Таким чином проаналізувати текстовий файл.
12. *Тренажер «Прості схеми сортування».*
Розробити програму, яка може навчити простим схемам сортування заданого масиву: вибіркового, вставкою, "бульбашковому". Передбачити наявність зрозумілого інтерфейсу.
13. *Тренажер «Швидке сортування».*
Розробити програму, яка може навчити алгоритму швидкого сортування заданого масиву. Передбачити наявність зрозумілого інтерфейсу.
14. *Тренажер «Пірамідальне сортування».*
Розробити програму, яка може навчити алгоритму пірамідального сортування заданого масиву. Передбачити наявність зрозумілого інтерфейсу.
15. *Тренажер «Сортування злиттям».*
Розробити програму, яка може навчити алгоритму сортування заданого масиву злиттям. Передбачити наявність зрозумілого інтерфейсу.
16. *Алгоритм швидкого сортування та його візуалізація.*
Реалізувати покрокове сортування заданого масиву методом швидкого сортування з візуалізацією основних етапів.
17. *Алгоритм пірамідального сортування та його візуалізація.*
Реалізувати покрокове сортування заданого масиву методом пірамідального сортування з візуалізацією основних етапів.

18. *Піраміда. Візуалізація основних операцій.*
Реалізувати покрокову візуалізацію операцій додавання та видалення значень в піраміді.
19. *Навантажене дерево. Візуалізація основних операцій.*
Реалізувати покрокову візуалізацію операцій додавання, видалення та пошуку слів у словнику, який створено на основі алгоритму навантаженого дерева.
20. *Алгоритм побудови кодів Хаффмана та його візуалізація.*
Реалізувати покрокову візуалізацію пошуку бінарних кодів для кожного символу з наданого рядка за допомогою алгоритму Хаффмана.
21. *2-3-дерево. Візуалізація основних операцій.*
Реалізувати покрокову візуалізацію операцій додавання, видалення та пошуку значень у 2-3-дереві.
22. *Червоно-чорне дерево. Візуалізація основних операцій.*
Реалізувати покрокову візуалізацію операцій додавання, видалення та пошуку значень у червоно-чорному дереві.
23. *АВЛ-дерево. Візуалізація основних операцій.*
Реалізувати покрокову візуалізацію операцій додавання, видалення та пошуку значень в АВЛ-дереві.
24. *Дераміда. Візуалізація основних операцій.*
Реалізувати покрокову візуалізацію операцій додавання, видалення та пошуку значень у дераміді.
25. *Неорієнтований граф. Візуалізація обходів в ширину та в глибину.*
Для заданого неорієнтованого графа реалізувати покрокову візуалізацію алгоритму обходів в ширину та в глибину.
26. *Візуалізація алгоритму Дейкстри для пошуку найкоротшого шляху в графі.*
Для заданого орієнтованого зваженого графа реалізувати покрокову візуалізацію алгоритму Дейкстри для пошуку найкоротшого шляху з однієї вершини до всіх інших.
27. *Візуалізація алгоритму Беллмана-Форда для пошуку найкоротшого шляху в графі.*
Для заданого орієнтованого зваженого графа реалізувати покрокову візуалізацію алгоритму Беллмана-Форда для пошуку найкоротшого шляху з однієї вершини до всіх інших.
28. *Візуалізація алгоритму Прима для побудови остовного дерева мінімальної вартості.*
Для заданого орієнтованого зваженого графа реалізувати покрокову візуалізацію алгоритму Прима для побудови остовного дерева мінімальної вартості.
29. *Візуалізація алгоритму Крускала для побудови остовного дерева мінімальної вартості.*
Для заданого орієнтованого зваженого графа реалізувати покрокову візуалізацію алгоритму Крускала для побудови остовного дерева мінімальної вартості.

7. Методи контролю

Контроль засвоєння навчального матеріалу здійснюється шляхом:

- прийому та оцінювання звітів з виконання індивідуальних завдань з курсової роботи;
- прийому та оцінювання завдань самостійної роботи;
- проведення контрольної роботи;
- проведення письмового підсумкового контролю знань.

8. Схема нарахування балів

Поточний контроль та самостійна робота																Залік	Сума
Розділ 1				Розділ 2			Розділ 3			Розділ 4			Контрольна робота, передбачена НП	Індивідуальне завдання (КП)	Разом		
T1	T2	T3	T4	T1	T2	T3	T1	T2	T3	T1	T2	T3					
2	2	2	2	2	2	2	2	4	4	2	2	2	10	20	60	40	100

Шкала оцінювання

Сума балів за всі види навчальної діяльності протягом семестру	Оцінка	
	для чотирирівневої шкали оцінювання	для дворівневої шкали оцінювання
90 – 100	відмінно	зараховано
70-89	добре	
50-69	задовільно	
1-49	незадовільно	не зараховано

Критерії оцінювання

Оцінка «**зараховано**» (50-100 балів) ставиться студентові, який виявив знання основного навчального матеріалу в обсязі, необхідному для подальшого навчання і майбутньої роботи за фахом, здатний виконувати завдання, передбачені програмою, ознайомлений з основною рекомендованою літературою; при виконанні завдань припускається помилок, але демонструє спроможність їх усувати.

Оцінка «**незараховано**» (1-49 балів) ставиться студентові, який допускає принципові помилки у виконанні передбачених програмою завдань, не може продовжити навчання чи розпочати професійну діяльність без додаткових занять з відповідної дисципліни.

9. Рекомендована література

Основна література

1. Кормен Т. Алгоритмы: построение и анализ. – М.: Издательский дом "Вильямс", 2013. – 1296 с.
2. Ахо А., Хопкрофт Дж., Ульман Дж. Структуры данных и алгоритмы. – М.: Издательский дом "Вильямс", 2003. – 384 с.
3. Дасгупта С., Пападимитриу Х., Вазирани У. Алгоритмы. – М.: МЦНМО, 2014. – 320 с.
4. Вирт Н. Алгоритмы и структуры данных. – М.: Мир, 1989.
5. Кнут Д. Искусство программирования. Т.1. Основные алгоритмы. – М.: Издательский дом "Вильямс". – 2000.
6. Кнут Д. Искусство программирования. Т.3. Поиск и сортировка. – М.: Издательский дом "Вильямс". – 2000.

Допоміжна література

1. Дейкстра Э. Дисциплина программирования. – М.: Мир, 1978.
2. Беллман Р. Динамическое программирование. – М., ИЛ., 1960.
3. Берж С. Теория графов и ее применение. – М., ИЛ., 1962.
4. Грин Д., Кнут Д. Математические методы анализа алгоритмов. – М.: Мир, 1987.

10. Посилання на інформаційні ресурси в Інтернеті, відео-лекції, інше методичне забезпечення

1. Алгоритмы сортировки: [Електроний ресурс] // Алгоритмы методы исходники, [Илья Кантор](http://algolist.manual.ru/sort), 2000-2017. Режим доступа: <http://algolist.manual.ru/sort>.
2. Алгоритмы на графах: [Електронний ресурс] // Wikipedia the free encyclopedia, 2001-2017. Режим доступа:
https://uk.wikipedia.org/wiki/%D0%9A%D0%B0%D1%82%D0%B5%D0%B3%D0%BE%D1%80%D1%96%D1%8F:%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%B8_%D0%BD%D0%B0_%D0%B3%D1%80%D0%B0%D1%84%D0%B0%D1%85.