МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних та практичних робіт

з дисципліни

«ОБРОБКА ВЕЛИКИХ МАСИВІВ ДАНИХ»

для студентів спеціальності 126 –

«Інформаційні системи та технології»

Міністерство освіти і науки України
Вінницький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних та практичних робіт з дисципліни «Обробка великих масивів даних» для студентів спеціальності 126 – «Інформаційні системи та технології»

Електронне видання
комбінованого (локального та мережного) використання

Вінниця
ВНТУ
2021

Рекомендовано до видання Методичною радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 10 від 20.05.2021 р.)

Рецензенти:
П. І. Кулаков, доктор технічних наук, професор ВНТУ
А. Я. Кулик, доктор технічних наук, професор ВНМУ ім. М. Пирогова

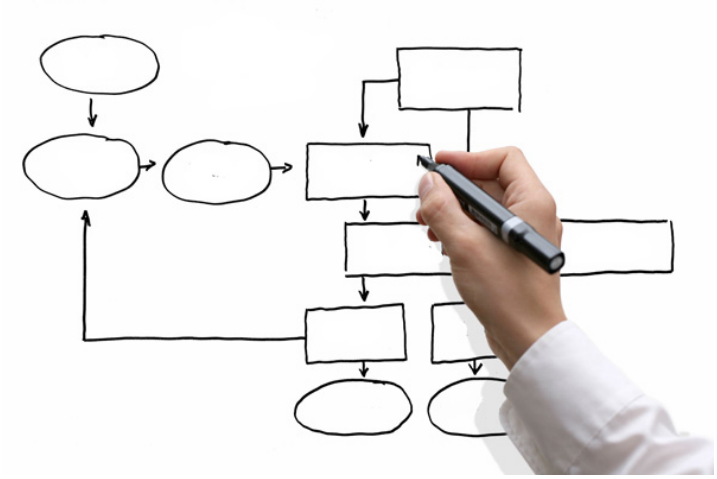
Методичні вказівки до виконання лабораторних та практичних робіт з дисципліни «Обробка великих масивів даних» для студентів спеціальності 126 – «Інформаційні системи та технології» [Електронний ресурс] / Уклад. Ю. Ю. Іванов, С. Г. Кривогубченко. – Вінниця : ВНТУ, 2021. – 29 с.

У методичних вказівках наведено завдання на лабораторні та практичні роботи з дисципліни «Обробка великих масивів даних», а також рекомендована для їхнього виконання література. Сформовано блоки контрольних питань, які можна використовувати для підготовки до контрольних заходів.

ЗМІСТ

Захист та структура звіту лабораторних робіт 4
Критерії оцінювання знань, умінь та навичок студентів 5
Вступ 6
Практична робота 8
Лабораторна робота № 1. Основи майнінгу криптовалют.
Задача хешування даних 10
Лабораторна робота № 2. Перевірка унікальності контенту.
Задача нечіткого пошуку зображень 13
Лабораторна робота № 3. Задача криптографічного закриття даних 16
Лабораторна робота № 4. Задача завадостійкого кодування даних 19
Рекомендована література 22
Таблиця символів ASCII 25
Оцінювання студентами матеріалів курсу та рівня викладання 28

ЗАХИСТ ТА СТРУКТУРА ЗВІТУ ЛАБОРАТОРНИХ РОБІТ



Студенти виконують *підготовчу роботу* відповідно до теми завдання, щоб зрозуміти теоретичні та практичні аспекти, математичну основу, застосування задач у професійній діяльності як тестових полігонів. Вони вивчають конспект лекцій, рекомендовані матеріали (література, аудіо-, відеофайли) тощо. Глибоке вивчення теоретичного матеріалу, набуті навички програмування та вміння використовувати пакети прикладних програм допоможуть успішно виконати поставлені задачі.

За результатами виконаної роботи на аркушах формату А4 оформлюється *звіт*, у якому відображаються всі результати роботи. Він має містити такі *елементи*: титульний аркуш, аналіз літератури та ключового матеріалу, формалізована постановка задачі, розв'язання задачі (числовий приклад) та алгоритми методів, програмне забезпечення, результати роботи, висновки, список використаної літератури. Оформлення звіту необхідно виконувати охайно за *правилами ДСТУ 3008:2015 та ДСТУ ГОСТ 7.1:2006* [1, 2]. Спочатку звіт необхідно подати викладачу, після виконання коригувань студент може захистити роботу. Також потрібно показати роботу програмного забезпечення на ЕОМ та відповісти на контрольні запитання, задані викладачем.

Студент має впевнено та чітко *презентувати* виконану ним роботу, показати її суть, знайомство з ключовою літературою; вміти розв'язати поставлену задачу та визначити методи її вирішення; володіти відповідним термінологічним апаратом; мати прийнятний рівень мовної грамотності, включаючи володіння функціональним стилем викладення матеріалу. Чітке виконання поставлених вимог та використання аргументованих відповідей на запитання викладача з посиланнями на роботи вчених допоможе підвищити оцінку.

КРИТЕРІЇ ОЦІНЮВАННЯ ЗНАНЬ, УМІНЬ ТА НАВИЧОК СТУДЕНТІВ

Рівень компетентності	За нац. шкалою	За шкалою ЕКТС	Критерії оцінювання
IV Високий (творчий) «5»	відмінно «5»	*A*	Виставляється, якщо під час відповіді на питання виявлено всебічні, систематизовані, глибокі знання матеріалу, який виноситься на контроль, уміння вільно виконувати завдання, передбачені програмою, знання основної і додаткової літератури, передбаченої програмою на рівні творчого використання.
III Достатній (конструктивний) «4»	добре «4+»	*B*	Повні знання з питань і задач, що стоять перед студентом. Уміння викладати основні ідеї. Вміння професійно відстоювати свою точку зору. Припускаються несуттєві неточності у викладенні матеріалу та у відповідях.
	добре «4»	*C*	Достатньо повні знання з поставлених питань і задач. Вміння викладати основні ідеї. Здатність самостійно застосовувати вивчений матеріал на рівні стандартних ситуацій, наводити власні приклади на підтвердження власних тверджень. Вміння доводити правильність своїх рішень. Несуттєві неточності у відповідях та деякі нераціональності при програмуванні задач.
II Середній (репродуктивний) «3»	задовільно «3+»	*D*	Студент може відтворити значну частину теоретичного матеріалу, виявляє знання та розуміння основних положень, з допомогою викладача може аналізувати матеріал, робити висновки та розробляти програмні блоки. Пояснення неповні, нелаконічні, не завжди точні. Відповіді на питання неповні, містять неточності, при програмуванні застосовуються не найраціональніші рішення.
	задовільно «3»	*E*	Задовільні знання програмного матеріалу на рівні вищому за початковий. Здатність за допомогою викладача логічно відтворювати значну частину матеріалу. Під час відповіді на запитання виникають труднощі у деяких положеннях, відповіді не повні, програми пишуться нераціонально, не використовуються всі ефективні засоби програмування.
I Низький «2»	незадовільно з можливістю повторного складання «2»	*FX*	Теорією володіє на рівні фрагментів, викладає матеріал уривчасто. Помітна складність в обґрунтуванні рішень. Самостійно, без допомоги викладача, не може сформувати алгоритм рішення задачі. Програми не раціональні та неефективні.
	незадовільно з обов'язковим повторним вивченням «2»	*F*	Теорією володіє на рівні фрагментів, викладає матеріал уривчасто. Не може обґрунтувати рішення задачі, на запитання викладача дає неправильні відповіді. Самостійно не може сформувати алгоритм рішення задачі.

ВСТУП

Хто володіє інформацією – той володіє світом!
Н. Ротшильд, В. Черчилль

За концепцією академіка В. М. Глушкова для подолання «*першого інформаційного бар'єру*» (можливість зберігати та передавати знання) людина приблизно в V тис. до н. е. винайшла *писемність*. Але на перешкоді суспільного прогресу виникає проблема («*другий бар'єр*») необхідності більшої кількості носіїв знань для надання людям освіти, яка потрібна для якісного управління процесом виробництва. Винайдення *книгодрукування* у XV ст., а пізніше й інших методів поширення і зберігання інформації (телеграф, телефон, фотографія, телебачення, кіно, магнітні записи) розв'язало цю проблему. Але оброблення інформації продовжував виконувати винятково мозок людини. У другій половині XX ст., коли загальні обсяги інформації зросли настільки, що сумарної пропускної здатності людського мозку виявилося недостатньо для її переробки, прогрес людства став залежати від того, чи вдасться вирішити проблему автоматизації оброблення інформації («*третій бар'єр*»). Людство знайшло вихід, розробивши електронну обчислювальну машину (комп'ютер), що дало можливість організувати збереження великих масивів даних і з розвитком технологій проводити їх ґрунтовний аналіз [3, 4].

Обробка великих масивів даних – дисципліна, яка займається збиранням, вивченням, збереженням та прийняттям рішень на основі аналізу великих обсягів даних з використанням обчислювальних потужностей комп'ютерної техніки, дозволяючи виявити непомітні закономірності і покращити процес прийняття рішень у різних галузях людської діяльності. Більшість напрямків сучасної науки, включаючи робототехніку, генну інженерію, біоніку, штучний інтелект тощо застосовують аналіз даних, які потрібно обробити та зберегти, причому ці дані можуть бути неструктурованими та досить великого обсягу [5–7]. З одного боку, застосовуються фундаментальні основи математики – теорія множин, математична логіка, теорія алгоритмів, теорія графів, криптографія та кодування інформації, статистика, машинне навчання тощо, з іншого боку, необхідно розробляти програмне забезпечення, яке дозволяє автоматизувати заданий алгоритм аналізу, використовуючи певну мову програмування. Тому знання, набуті студентами на попередніх курсах та самостійна робота, абсолютно необхідні для чіткого формулювання понять і постановок різних прикладних

інтелектуальних задач, їх формалізації і комп'ютеризації, а також для засвоєння і розробки сучасних інформаційних технологій.

У цьому курсі *студентам запропоновано у якості тестових полігонів низку теоретичних* та *практичних* задач з різних науково-практичних галузей. Варто зазначити, що в умовах обмеженого часу досить важко охопити всі теми такого об'ємного курсу. ***Метою дисципліни*** є ознайомлення студентів з основами обробки даних, формування у них уявлення про принципи і методи математичного програмування та моделювання систем, вироблення системи знань, умінь і навичок, що необхідні для ефективного застосування ними як майбутніми фахівцями спеціалізованих методів та алгоритмів в різних галузях професійної діяльності з використанням прикладних пакетів математичного моделювання або програм, які створені власноруч. Обов'язковим є виконання для кожної задачі числового прикладу (наприклад, у системах математичного моделювання – *MatLab*, *MathCad*, *Mathematica*, *Maple* тощо), підвищує бали розробка програмного забезпечення з обґрунтуванням вибору технологічної платформи для дослідження ефективності роботи методу розв'язання задачі.

Програмне забезпечення розробляється з використанням принципів структурного, функціонального, об'єктно-орієнтованого програмування. Програмний комплекс має бути гнучким, наприклад, надавати можливість змінювати параметри вхідних даних; містити графічне меню, яке забезпечує можливість введення початкових умов задачі, вибір параметрів та збереження отриманого розв'язку.

Таким чином, студенти мають розширити свої знання за цим курсом та вдосконалити навички програмування, що знадобиться їм у професійній діяльності.

ПРАКТИЧНА РОБОТА

Основне завдання дисципліни полягає у виконанні *самостійного пошуку* студентом (або за допомогою викладача) певної цікавої *науково-дослідницької задачі*, її аналізу, розв'язання (*теоретично* та *програмно*).

Студенти виконують *підготовчу роботу відповідно до теми завдання*, щоб зрозуміти теоретичні та практичні аспекти, математичну основу, застосування цих задач у професійній діяльності як тестових полігонів. Вони вивчають потрібні матеріали (література, аудіо-, відеофайли) тощо. Глибоке вивчення теоретичного матеріалу, набуті навички програмування та вміння використовувати пакети прикладних програм допоможуть успішно провести практичне заняття. Обов'язковим є виконання для кожної задачі числового прикладу (наприклад, у системах математичного моделювання – *MatLab, MathCad, Mathematica* тощо) та розроблення програмного забезпечення.

Студент має *оформити звіт* з виконаної роботи, *презентацію* та *виступити* перед аудиторією, яку представляє група студента та викладач дисципліни. Попередньо тема виступу обговорюється з викладачем курсу.

Спершу звіт необхідно подати викладачу, після виконання коригувань студент може захистити роботу. Оформлення звіту необхідно виконувати охайно за правилами *ДСТУ 3008:2015*, *ДСТУ ГОСТ 7.1:2006* [1, 2] для наукових робіт. Він має містити такі *елементи*: титульний аркуш, аналіз літератури та ключового матеріалу, формалізована постановка задачі, розв'язання задачі (числовий приклад) та алгоритми методів, програмне забезпечення, результати роботи, висновки, список використаної літератури.

У *презентації* студент наглядно показує результати своєї роботи. Для отримання *відмінної оцінки* необхідно виступити перед аудиторією, яку представляє група студента та викладач курсу. Студент має впевнено й точно презентувати виконану ним роботу.

Виконання такого завдання допоможе сформувати у студентів вміння використовувати методи та засоби викладання, організовувати навчальний процес, навчитися без сорому виступати перед аудиторією, уміти активізувати пізнавальну діяльність, стимулювати до критичного та творчого мислення, самостійного пошуку й дослідження за цим курсом. Студенти покращать свої знання та вдосконалять навички програмування, що знадобиться їм у професійній діяльності.

ЛАБОРАТОРНА РОБОТА № 1

Тема: Основи майнінгу криптовалют. Задача хешування даних.

Мета: ознайомитися з ключовими поняттями майнінгу на прикладі криптовалюти Bitcoin, навчитися знаходити дайджест повідомлення за допомогою алгоритмів стандарту SHA-2 та застосовувати відповідне програмне забезпечення.

Теоретичні відомості: для більш детального розуміння матеріалу необхідно проаналізувати джерела [8–12], у яких подано теоретичні основи майнінгу й хешування даних та проаналізовано відповідні задачі.

Завдання

Подати математичну постановку задач майнінгу криптовалют та хешування даних. Проаналізувати й теоретично обґрунтувати:

- особливості блокчейну;
- алгоритми майнінгу криптовалют (*Scrypt*, *EtHash*, *X*11, *CryptoNight*, *EquiHash*, *X*13, *Quark*, *NeoScrypt*);
- алгоритми хеш-функцій стандарту *SHA*-2.

Розв'язати наступні задачі:

Задача 1. Обчислити значення хеш-функції *H*:

а) *SHA*-2(«*ПІБ студента*»);

б) *SHA*-2(«*вашої улюбленої книги в форматі DOC*»);

в) *SHA*-2(«*зображення формату JPEG розміром 256 на 256*»).

Примітка.

1) текст повідомлення має зчитуватися з файлу;

2) отримане значення хеш-функції має подаватися в шістнадцятковому вигляді й зберігатися у файлі;

3) має бути можливість перегляду проміжних значень змінних та їхнє обчислення «онлайн» по ітераціях (рис. 1.1);

4) необхідно обчислювати час хешування даних.

Задача 2. Дослідити лавинний ефект при зміні одного біта на різних позиціях у повідомленні та побудувати графік залежностей кількості бітів, які змінилися в хеші *k*, залежно від раунду обчислення хеш-функції *I*.

```
            a         b         c         d         e         f         g         h

init:    6a09e667  bb67ae85  3c6ef372  a54ff53a  510e527f  9b05688c  1f83d9ab  5be0cd19
t =  0   5d6aebcd  6a09e667  bb67ae85  3c6ef372  fa2a4622  510e527f  9b05688c  1f83d9ab
t =  1   5a6ad9ad  5d6aebcd  6a09e667  bb67ae85  78ce7989  fa2a4622  510e527f  9b05688c
t =  2   c8c347a7  5a6ad9ad  5d6aebcd  6a09e667  f92939eb  78ce7989  fa2a4622  510e527f
…
t = 62   d39a2165  04d24d6c  b85e2ce9  b6ae8fff  fb121210  948d25b6  961f4894  b21bad3d
t = 63   506e3058  d39a2165  04d24d6c  b85e2ce9  5ef50f24  fb121210  948d25b6  961f4894

Block 1 has been processed. The values of {Hi} are

H1 = 6a09e667 + 506e3058 = ba7816bf
H2 = bb67ae85 + d39a2165 = 8f01cfea
H3 = 3c6ef372 + 04d24d6c = 414140de
H4 = a54ff53a + b85e2ce9 = 5dae2223
H5 = 510e527f + 5ef50f24 = b00361a3
H6 = 9b05688c + fb121210 = 96177a9c
H7 = 1f83d9ab + 948d25b6 = b410ff61
H8 = 5be0cd19 + 961f4894 = f20015ad.
```

Рисунок 1.1 – Приклад протоколу результатів досліджень

Примітка.

1) має існувати можливість змінювати біт або символ повідомлення, який буде інвертовано;

2) після кожного раунду обчислення хеш-функції необхідно підраховувати кількість бітів хешу, які змінилися, після інвертування одного біта / символа в тексті вхідного повідомлення;

3) необхідно зберігати у файл дані, потрібні для побудови графіка;

4) проаналізувати отриману залежність.

Задача 3. Згенерувати базу даних з кількістю користувачів $p = 100$, їхні паролі (можуть бути повтори) та знайти хеші. З використанням сервісів для злому хешів типу *https://crackstation.net* розшифрувати паролі. Додати до паролів та хешу статичну та динамічну сіль. Перевірити структуру хешів та розшифрувати їх.

Розробити алгоритмічне забезпечення та перевірити результати роботи в програмних середовищах (*MatLab*, *MathCAD, Mathematica, Maple, Python Anaconda* тощо [36–40]) або розробити програму самостійно. Отримати рішення заданих задач, подати результати роботи програми, зробити висновки.

Контрольні запитання

1. Історична ретроспектива. Видатні вчені.
2. Поняття хешування даних.
3. Математична постановка задачі хешування даних.
4. Властивості криптографічних хеш-функцій.
5. Колізії.
6. Методи пошуку колізій.
7. Поняття «солі» (статична, динамічна).
8. Майнінг криптовалют.
9. Особливості блокчейну.
10. Біткоїн. Структура блоку та особливості транзакцій.
11. Дерево Меркла.
12. Особливості хеш-функції *Scrypt*, *EtHash*, *X*11, *CryptoNight*, *EquiHash*, *X*13, *Quark*, *NeoScrypt*, *SHA*-1, *SHA*-2, *SHA*-3 (*Kessak*), *RIPEMD*, *MD*5.
13. Конструкція Меркла-Дамгарда.
14. Математична модель хеш-функцій стандарту *SHA*-2.
15. Технічні характеристики хеш-функцій сімейства *SHA*-2.
16. Які операції використовуються під час обчислення значень хешу для *SHA*-2 ?
17. Етап попередньої обробки даних для *SHA*-2.
18. Визначення початкових значень хешу для *SHA*-2.
19. Визначення констант для *SHA*-2.
20. Яким чином формується функція компресії для *SHA*-2?
21. Хеш-функції типу *SHA*-512/*t*.
22. Апаратна база для роботи з криптовалютами. Процесори та відеокарти.
23. Особливості обчислення хешів у середовищах програмування.

ЛАБОРАТОРНА РОБОТА № 2

Тема: Перевірка унікальності контенту. Задача нечіткого пошуку зображень.

Мета: ознайомитися з ключовими поняттями нечіткого пошуку зображень, видалення дублікатів та контамінантів у контенті, а також навчитися застосовувати методи перцептивного хешування даних та відповідне програмне забезпечення.

Теоретичні відомості: для більш детального розуміння матеріалу необхідно проаналізувати джерела [13–19], у яких подано теоретичні основи роботи з колекціями зображень та застосуванням перцептивного хешування, а також проаналізовано відповідні задачі.

Завдання

Подати математичну постановку задачі нечіткого пошуку у колекції зображень. Проаналізувати й теоретично обґрунтувати:

- алгоритми пошуку дублікатів у колекціях зображень на основі:
  - метаданих;
  - візуального пошуку (глобальні та локальні дескриптори типу *SIFT* та *SURF*, перцептивне хешування, нейронні мережі);
  - комбінованого пошуку;
- алгоритми ресайзингу зображень (найближчого сусіда, швидкого перетворення Фур'є, білінійної або бікубічної інтерполяції, «різьби по шву» (*seam carving* або *liquid resizing*) тощо);
- особливості перцептивного хешування на базі функцій на основі:
  - середнього значення *ahash*;
  - медіани *mhash*;
  - різниць *dhash*;
  - градієнта зображення *ghash*;
  - дискретного косинусного перетворення *DCT-Hash* або *phash*;
  - поблокового *bHash*;
  - радіальної дисперсії *RV-Hash*;
  - оператора Марра-Хілдрета *MHO-Hash*;

- алгоритм *HEngine* для прискореного пошуку рядків колекції зображень у заданих межах відстані Хеммінга з використанням теореми Лю-Шена-Торнга (*Liu*, *Shen*, *Torng*);
- застосування фільтра Блума для зменшення часу роботи бінарного пошуку.

Розв'язати такі задачі:

Задача 1. Реалізувати перцептивні хеші *ahash*, *mhash*, *dhash*, *ghash*, *phash*, *bHash*. Для аналізу та тестування роботи програми (пошук дублікатів, виділення унікального контенту) використати колекцію зображень типу:

– *Natgeo* (*https://www.instagram.com/natgeo/*) або подібні сторінки;

– *Content by topic* (*https://commons.wikimedia.org/wiki/Main_Page*);

– *Caltech* (*http://www.vision.caltech.edu/html-files/archive.html*), наприклад *Caltech*-101 або *Faces*-1999.

Змінюючи відстань Хеммінга, знайти кількість подібних пар та порівняти час роботи програми без використання перцептивних хеш-алгоритмів (побітове або побайтове порівняння контенту) та з їх застосуванням. Побудувати відповідні графіки.

Примітка.

1) скачати зображення зі сторінки *Natgeo* можна з використанням різних додатків, наприклад, *Instagram Downloader*.

Задача 2. Для прискорення пошуку відстані Хеммінга в заданих межах застосувати алгоритм *HEngine* з фільтром Блума. Порівняти час роботи програми із застосуванням такого алгоритму та без нього (повний перебір). Побудувати графіки.

Задача 3. На основі низки отриманих схожих зображень (шуми, розмиття, зміна контрасту, масштабування тощо) експериментальним шляхом визначити оптимальне значення відстані Хеммінга для порівняння зображень, яке дозволяє зменшити розміри вибірки (без значних втрат у точності пошуку).

Розробити алгоритмічне забезпечення та перевірити результати роботи в програмних середовищах (*MatLab*, *MathCAD, Mathematica, Maple, Python Anaconda* тощо [36–40]) або розробити програму самостійно. Отримати рішення заданих задач, подати результати роботи програми, зробити висновки.

Контрольні запитання

1. Історична ретроспектива. Видатні вчені.
2. Поняття контенту.
3. Математичні методи виділення унікального контенту. Метадані та візуальні характеристики контенту.
4. Глобальні та локальні дескриптори (*SIFT* та *SURF*), нейронні мережі.

5. Поняття перцептивного хешування даних. Галузі застосування.
6. Алгоритми ресайзингу зображень.
7. Метрики схожості (відстань Хеммінга, евклідова метрика, коефіцієнт бітових помилок тощо).
8. Що представляє собою зображення у разі використання перетворення Фур'є?
9. Загальний алгоритм перцептивного хешування.
10. Пряме та зворотне дискретне косинусне перетворення.
11. Перцептивні хеш-функції.
12. Алгоритм *HEngine* для прискорення пошуку відстані Хеммінга в заданих межах. Особливості попередньої обробки даних та пошуку ознак.
13. Теорема Лю-Шена-Торнга.
14. Бінарний пошук. Оцінка складності.
15. Лінійний пошук. Оцінка складності.
16. Фільтр Блума. Імовірність хибно позитивного спрацьовування. Оптимальна кількість хеш-функцій.
17. Особливості обробки контенту та пошуку перцептивних хешів у середовищах програмування та математичних пакетах.

ЛАБОРАТОРНА РОБОТА № 3

Тема: Задача криптографічного закриття даних.

Мета: ознайомитися з ключовими поняттями криптографічного закриття даних (криптографія, криптоаналіз, стеганографія), навчитися застосовувати класичні алгоритми шифрування даних та застосовувати відповідне програмне забезпечення.

Теоретичні відомості: для більш детального розуміння матеріалу необхідно проаналізувати джерела [20–25], у яких подано теоретичні основи криптології та проаналізовано відповідні задачі.

Завдання

Подати математичну постановку задачі криптографічного закриття інформації. Проаналізувати та теоретично обґрунтувати методи криптографічного закриття інформації *DES* та *RSA*, а також основи стеганографії.

Розв'язати такі задачі:

Задача 1. Зашифрувати ПІБ студента, а також текст вашої улюбленої книги у форматі *DOC*. Метод шифрування вибрати за вказівкою викладача на основі табл. 3.1. Використати випадкову генерацію ключа / ключів в алгоритмі *DES* та стартових чисел p і q в алгоритмі *RSA* з використанням зерна (*seed*). Зберегти *seed* та показати викладачу для перевірки.

Примітка.

1) текст повідомлення має зчитуватися з файлу;

2) ключ / ключі шифрування потрібно зберігати в інших файлах;

3) довжини чисел p і q для алгоритма *RSA* мають бути більші, ніж 2^{128};

4) зашифрований текст має зберігатися у файлі;

5) забезпечити можливість перегляду обчислень «онлайн» для більшості можливих операцій;

6) має бути можливість оцінити час шифрування / дешифрування.

Таблиця 3.1 – Варіанти завдань

<table>
<tr><th>Варіант</th><th>Симетричний шифр</th><th>Асиметричний шифр</th></tr>
<tr><td>1</td><td>*DES-EEE*2 (шифрування три рази трьома ключами k_1, k_2, k_1)</td><td rowspan="3">Шифр *RSA*</td></tr>
<tr><td>2</td><td>*DES-EEE*3 (шифрування три рази трьома різними ключами k_1, k_2, k_3)</td></tr>
<tr><td>3</td><td>*DES-EDE*2 (шифрування-дешифрування-шифрування з трьома ключами k_1, k_2, k_1)</td></tr>
<tr><td>4</td><td>*DES-EDE*3 (шифрування-дешифрування-шифрування з трьома різними ключами k_1, k_2, k_3)</td><td rowspan="2">Цифровий підпис на основі *RSA*</td></tr>
<tr><td>5</td><td>*DESX* (перед виконанням *DES* і після нього на дані операцією *XOR* накладаються різні 64-бітові фрагменти ключа)</td></tr>
</table>

Задача 2. Дослідити лавинний ефект для алгоритму *DES* при зміні одного біта на різних позиціях в повідомленні та побудувати графік залежностей кількості бітів, які змінилися в шифрі *l*, залежно від раунду шифрування *I*.

Примітка.

1) має існувати можливість змінювати біт повідомлення, який буде інвертовано;

2) після кожного раунду шифрування необхідно підраховувати кількість бітів шифру, які змінилися, після інвертування одного біта / символа в тексті вхідного повідомлення;

3) необхідно зберігати у файл дані, потрібні для побудови графіка.

4) проаналізувати отриману залежність.

Задача 3. Виконати шифрування кольорового зображення певного формату (наприклад *jpeg*) розміром 256×256.

Примітка. Робочі параметри обрати самостійно.

Розробити алгоритмічне забезпечення та перевірити результати роботи в програмних середовищах (*MatLab*, *MathCAD, Mathematica, Maple, Python Anaconda* тощо [36–40]) або розробити програму самостійно. Отримати рішення заданих задач, подати результати роботи програми, зробити висновки.

Контрольні запитання

1. Історична ретроспектива створення теорії інформації та кодування, методів криптографічного закриття інформації, стеганографії. Видатні вчені.
2. Поняття інформації, даних та знань.
3. Криптологія: криптографія та криптоаналіз.
4. Принцип Керкгоффса.
5. Узагальнена структура системи секретного зв'язку.
6. Математична постановка задачі криптграфічного закриття інформації.
7. Особливості алгоритмів шифрування даних (симетричні, асиметричні, перестановочні, потокові, блокові).
8. *SP*-мережі. Дифузія, конфузія, лавинний ефект.
9. Мережа Фейстеля.
10. На яких принципах ґрунтується алгоритм *DES*?
11. Алгоритми шифрування *multi-DES* (2*DES*, 3*DES*), *DESX*.
12. На яких принципах ґрунтується алгоритм *RSA*?
13. Алгоритм Евкліда та його розширена версія.
14. Стеганографічні методи закриття інформації (водяний знак, найменший значущий біт, ехо-, структурні й статистичні методи тощо).
15. Як зашифрувати зображення?
16. Як вбудувати та приховати зображення в зображенні?
17. Апаратна база для розроблення алгоритмів шифрування даних.
18. Особливості розв'язання такої задачі в пакетах математичного моделювання та середовищах програмування.

ЛАБОРАТОРНА РОБОТА № 4

Тема: Задача завадостійкого кодування даних.

Мета: ознайомитися з ключовими поняттями завадостійкого кодування даних, навчитися працювати з сучасними завадостійкими кодами та застосовувати відповідне програмне забезпечення.

Теоретичні відомості: для більш детального розуміння матеріалу необхідно проаналізувати джерела [3, 4, 20, 25-35], у яких подано теоретичні основи завадостійкого кодування даних та проаналізовано відповідні задачі.

Завдання

Подати математичну постановку задачі завадостійкого кодування / декодування даних. Проаналізувати й теоретично обґрунтувати сучасні методи завадостійкого кодування даних *турбо-подібного типу* (*turbo-like codes*):

- послідовна конкатенація згорткових кодів (*serial concatenated convolutional code*);
- паралельна каскадна конкатенація згорткових кодів або турбо-код (*turbo-code*);
- блоковий турбо-код-добуток або композиційний код (*turbo-product code*);
- гібридна конкатенація згорткових кодів (*hybrid concatenated convolutional code*);
- код з низькою щільністю перевірок на парність (*low-density parity-check code*, *LDPC*) або код Галлагера;
- *LOTUs*-код (комбінація турбо-коду та *LDPC*-коду).

Також розглянути відповідні *алгоритми декодування*:

- Вітербі з м’яким виходом *SOVA* (*soft-output Viterbi algorithm*);

- *SOVA* з апроксимацією Хагенауера;
- двонаправлений *SOVA* (*Bi-SOVA*);
- за максимумом апостеріорної імовірності символа *MAP* (*maximum a posteriori probability*, *BCJR*);
- за максимумом апостеріорної імовірності символа в логарифмічній області *log-MAP*;
- за максимумом апостеріорної імовірності символа в логарифмічній області без компенсуюча функції *max-log-MAP*;
- за максимумом апостеріорної імовірності символа в логарифмічній області з апроксимацією компенсуюча функції;
- Чейза-Піндайя для блокових кодів;
- розповсюдження довіри Перл *belief propagation* або *Sum-Product*;
- *Min-Sum.*

Розв'язати такі задачі:

Задача 1. Знайти експериментальні функціональні залежності частоти виникнення бітових помилок (*bit error rate, BER*) залежно від нормованого відношення сигнал / шум у *дБ*, тобто $BER = f(E_b/N_0)$, застосувавши комп'ютерне імітаційне моделювання.

Примітка. Робочі параметри коду вибрати самостійно. Вхідними даними є згенеровані за допомогою генератора випадкових бінарних послідовностей інформаційні блоки; параметри компонентних кодерів; кількість ненадійних бітів для декодера Чейза (якщо потрібно); правило роботи інтерлівера (якщо необхідно); кількість ітерацій декодування; перевірочна матриця (якщо потрібно); критерій зупинки ітеративного декодера (кількість помилок у пакеті або час роботи на даному відрізку); крок зміни, мінімальне та максимальне значення E_b/N_0. Вихідні дані – інформаційна послідовність на виході, значення *BER* залежно від відношення E_b/N_0. Варто зазначити, що для виконання цієї задачі використати *BPSK*-модем (двійкова фазова модуляція), а також канал з адитивним гаусівським шумом (*additive white Gaussian noise*, *AWGN*).

Задача 2. Використовуючи блоковий та згортковий код із приведеного вище списку, виконати процедуру передавання даних. Необхідно передати каналом зв'язку:

- рибосомну рибонуклеїнову кислоту (РНК) бактерії *Carsonella*. Дані можна отримати з банку національного центру біотехнологічної інформації (http://www.ncbi.nlm.nih.gov/nuccore/929558963?report=fasta). У ході експериментів змінювати нормоване відношення сигнал / шум від 0 до 3…5 *дБ* з кроком 0,25…0,5 *дБ*. Визначити частоту виникнення бітових (*bit error rate, BER*) та символьних (*symbol error rate, SER*) помилок, оцінити енергетичний виграш;
- кольорове зображення певного формату (наприклад *jpeg*) в умовах сильної шумової компоненти, наприклад E_b/N_0 = 1…1,5 *дБ*. Визначити кількості помилок у бітах та у символах, передавши його без кодування, а також із застосуванням певної кількості ітерацій декодування (1, 4, 8, …,

20). У результаті відобразити як змінювалося зображення на кожній ітерації та графічні залежності *BER / SER* залежно від кількості ітерацій декодування.

Примітка. Робочі параметри коду вибрати самостійно. Вхідними даними є параметри компонентних кодерів; кількість ненадійних бітів для декодера Чейза (якщо потрібно); правило роботи інтерлівера (якщо необхідно); кількість ітерацій декодування; перевірочна матриця (якщо потрібно); критерій зупинки ітеративного декодера (кількість помилок у пакеті або час роботи на даному відрізку); крок зміни, мінімальне та максимальне значення E_b/N_0. Вихідні дані – інформаційна послідовність на виході, значення *BER* / *SER* залежно від відношення E_b/N_0 та кількості ітерацій декодування. Варто зазначити, що для виконання такої задачі використати *BPSK*-модем, а також канал з адитивним гаусівським шумом.

Розробити алгоритмічне забезпечення та перевірити результати роботи в програмних середовищах (*MatLab*, *MathCAD, Mathematica, Maple, Python Anaconda* тощо [36–40]) або розробити програму самостійно. Отримати рішення заданих задач, подати результати роботи програми, зробити висновки.

Контрольні запитання

1. Історична ретроспектива створення теорії інформації та кодування, методів завадостійкого кодування. Видатні вчені.
2. Структура системи цифрового зв’язку.
3. Коди, які знаходять помилки.
4. Пряме коригування помилок.
5. Математична постановка задачі завадостійкого кодування даних.
6. Складність задачі декодування даних.
7. Блокові та згорткові коди. Особливості роботи. Алгоритми кодування та декодування.
8. Алгоритми декодування з жорстким та м’яким виходом.
9. Код Хеммінга, Ріда-Соломона, Боуза-Чоудхурі-Хоквінгема.
10. Згортковий код.
11. Алгоритм декодування Вітербі.
12. Зв’язок алгоритму Вітербі та динамічного програмування Беллмана.
13. Чому турбо-код вважають «кодом, що навчається»? Алгоритм Перл.
14. Алгоритми декодування *SOVA, SOVA* з апроксимацією Хагенауера, *Bi-SOVA*, *MAP, log-MAP, max-log-MAP* та інші апроксимації компенсуюча функції.
15. Особливості розв’язання цієї задачі в пакетах математичного моделювання та середовищах програмування.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки та техніки. Структура та правила оформлювання / В. Земцева, Ю. Поліщук, Р. Санченко, Л. Шрамко, А. Ямчук. – К. : ДП Український інститут науково-технічної і економічної інформації, 2016. – 31 с.
2. ДСТУ ГОСТ 7.1:2006. Бібліографічний запис, бібліографічний опис. Загальні вимоги та правила складання / О. К. Галевич, І. М. Штогрин. – Львів, 2008. – 20 с.
3. Глик Дж. Информация. История. Теория. Поток / Дж. Глик. – М.: АСТ: CORPUS, 2013. – 576 с.
4. Быховский М. А. Пионеры информационного века. История развития теории связи / Быховский М. А. – М. : Эко-Трендз, 2006. – 375 с.
5. Henderson H. Encyclopedia of Computer Science and Technology / H. Henderson. – New-York, 2003. – 593 p.
6. Левитин А. В. Алгоритмы: введение в разработку и анализ / А. В. Левитин. – М. : Вильямс, 2006. – 527 с.
7. Скиена С. Алгоритмы. Руководство по разработке / С. Скиена. – СПб. : БХВ-Петербург, 2011. – 720 с.
8. Криптовалюты. Блокчейн [Электронный ресурс]. – Режим доступа : https://ru.bitcoinwiki.org/wiki/Категория:Криптовалюты.
9. Bitcoin Protocol Documentation [Web resource]. – Access mode: https://en.bitcoin.it/wiki/Protocol_documentation#block.
10. Авезова Я. Е. Современные подходы к построению хеш-функций на примере финалистов конкурса SHA-3 / Я. Е. Авезова // Вопросы кибербезопасности. – № 3 (11). – 2015. – С. 60–67.
11. FIPS PUB 180-2. Federal Information Processing Standards Publication. Secure Hash Standard [Web resource]. – Access mode: https://csrc.nist.gov/csrc/media/publications/fips/180/2/archive/ 2002-08-01/documents/fips180-2.pdf.

12. FIPS PUB 180-4. Federal Information Processing Standards Publication. Secure Hash Standard [Web resource]. – Access mode: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf.
13. Gonzalez R.C. Digital Image Processing / R.C. Gonzalez, R.E. Woods. – Prentice Hall, 2008. – 956 p.
14. Zauner C. Implementation and Benchmarking of Perceptual Image Hash Functions / C. Zauner. – Hagenberg, 2010. – 107 p.
15. Beghdadi A. A Survey of Perceptual Image Processing Methods / A. Beghdadi, M.-C. Larabi, A. Bouzerdoum, K.M. Iftekharuddin // Signal Processing. Image. Communication. – 2014. – Vol. 28. – 45 p.
16. Нємкова О. А. Біометрична ідентифікація у кіберпросторі / О. А. Нємкова // Системи обробки інформації. – К., 2015. – № 7 (132). – С. 118–121.
17. The Discrete Cosine Transform (DCT) / Cardiff School of Computer Science & Informatics [Web Resource]. – Access mode: http://www.cs.cf.ac.uk/Dave/ Multimedia/node231.html.
18. Liu A.X. Large Scale Hamming Distance Query Processing / A. X. Liu, K. Shen, E. Torng // ICDE Conference. – 2011. – P. 553–564.
19. Bloom Filters by Example [Web Resource]. – Access mode: https://llimllib.github.io/bloomfilter-tutorial/.
20. Шеннон К. Е. Работы по теории информации и кибернетике. Математическая теория связи. Теория связи в секретных системах / К. Е. Шеннон. – М. : Издательство иностранной литературы, 1963. – С. 243–402.
21. Шнайер Б. Прикладная криптография / Б. Шнайер. – М. : Триумф, 2002. – 816 с.
22. Венбо М. Современная криптография / М. Венбо. – М. : Вильямс, 2005. – 768 с.
23. Смарт Н. Криптография / Н. Смарт. – М.: Техносфера, 2005. – 528 с.
24. Коутинхо С. Введение в теорию чисел. Алгоритм RSA. – М. : Постмаркет, 2001. – 328 с.
25. Скляр Б. Цифровая связь. Теоретические основы и практическое применение / Скляр Б. – М. : Вильямс, 2003. – 1004 с.
26. Банкет В. Л. Дискретная математика в задачах теории цифровой связи: учебное пособие / Банкет В. Л. – Одесса: ОНАС, 2008. – 118 с.
27. Зигангиров К. Ш. Современная теория кодирования / К. Ш. Зигангиров, Г. А. Кабатянский. – 2007. – 172 с.
28. Вернер М. Основы кодирования: учебник для ВУЗов / М. Вернер. – М. : Техносфера, 2004. – 288 с.
29. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение / Р. Морелос-Сарагоса. – М. : Техносфера, 2006. – 320 с.

30. Зубарев Ю. Б. Помехоустойчивое кодирование в цифровых системах передачи данных [Электронный ресурс] / Ю. Б. Зубарев, Г. В. Овечкин. – Режим доступа : http://www.mtdbest.iki.rssi.ru/pdf/obzor_ dvoichnie_kodi_2. pdf.
31. Іванов Ю. Ю. Особливості апаратно-програмної реалізації турбокодів: порівняльний аналіз складності реалізації на цифровому сигнальному процесорі / Ю. Ю. Іванов // Вісник Вінницького політехнічного інституту. – Вінниця : ВНТУ, 2016. – № 3(126). – С. 94–101.
32. Іванов Ю. Ю. Експериментальне дослідження завадостійкості турбокодів: числові оцінки та імітаційне моделювання нового субоптимального алгоритма PL-log-MAP / Ю.Ю. Іванов // Вісник Вінницького політехнічного інституту. – Вінниця: ВНТУ, 2016. – № 5. – С. 76–84.
33. Особливості оцінювання параметрів процесу передавання даних із використанням турбо-кодів / Р. Н. Квєтний, Ю. Ю. Іванов, С. Г. Кривогубченко, О. В. Стукач // Науковий журнал «Метрологія та прилади». – К. : ВКФ «Фавор ЛТД», 2017. – № 3 (65). – С. 25–32.
34. Johnson S. J. Iterative Error Correction. Turbo, Low-Density Parity-Check and Repeat-Accumulate Codes / S.J. Johnson. – New York: Cambridge University Press, 2009. – 356 p.
35. Moon T. K. Error Correction Coding: Mathematical Methods and Algorithms / Moon T. K. – John Wiley & Sons, Ltd, 2005. – 750 p.
36. Ануфриев И. Е. MATLAB 7 / Ануфриев И. Е., Смирнов А. Б., Смирнова Е. Н. – СПб. : БХВ-Петербург, 2005. – 1104 с.
37. Лутц М. Программирование на Python: 4-е издание / Лутц М. – СПб. : Символ-Плюс, 2010. – 1280 с.
38. Центр документации Wolfram Mathematica [Электронный ресурс] / Wolfram Language & System. – Режим доступа : http://reference.wolfram.com/ language/.
39. Numerical Recipes in C: The Art of Scientific Computing / Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P. – Cambridge : Cambridge University Press, 2002. – 949 p.
40. Макаров Е. Г. Инженерные расчеты в Mathcad 15. Учебный курс / Макаров Е. Г. – СПб. : Питер, 2011. – 400 с.

ТАБЛИЦЯ СИМВОЛІВ ASCII

Таблиця символів ASCII – це американський стандартний код для обміну даними.

Порядковий номер	Код	Символ	Порядковий номер	Код	Символ
0	00000000	код управління	32	00100000	пробіл
1	00000001	код управління	33	00100001	!
2	00000010	код управління	34	00100010	"
3	00000011	код управління	35	00100011	#
4	00000100	код управління	36	00100100	$
5	00000101	код управління	37	00100101	%
6	00000110	код управління	38	00100110	&
7	00000111	код управління	39	00100111	'
8	00001000	Backspace	40	00101000	(
9	00001001	код управління	41	00101001	)
10	00001010	код управління	42	00101010	*
11	00001011	код управління	43	00101011	+
12	00001100	код управління	44	00101100	,
13	00001101	код управління	45	00101101	-
14	00001110	код управління	46	00101110	.
15	00001111	код управління	47	00101111	/
16	00010000	код управління	48	00110000	0
17	00010001	код управління	49	00110001	1
18	00010010	код управління	50	00110010	2
19	00010011	код управління	51	00110011	3
20	00010100	код управління	52	00110100	4
21	00010101	код управління	53	00110101	5
22	00010110	код управління	54	00110110	6
23	00010111	код управління	55	00110111	7
24	00011000	код управління	56	00111000	8
25	00011001	код управління	57	00111001	9
26	00011010	код управління	58	00111010	:
27	00011011	код управління	59	00111011	;
28	00011100	код управління	60	00111100	<
29	00011101	код управління	61	00111101	=
30	00011110	код управління	62	00111110	>
31	00011111	код управління	63	00111111	?

Порядковий номер	Код	Символ	Порядковий номер	Код	Символ
64	01000000	@	96	01100000	`
65	01000001	A	97	01100001	a
66	01000010	B	98	01100010	b
67	01000011	C	99	01100011	c
68	01000100	D	100	01100100	d
69	01000101	E	101	01100101	e
70	01000110	F	102	01100110	f
71	01000111	G	103	01100111	g
72	01001000	H	104	01101000	h
73	01001001	I	105	01101001	i
74	01001010	J	106	01101010	j
75	01001011	K	107	01101011	k
76	01001100	L	108	01101100	l
77	01001101	M	109	01101101	m
78	01001110	N	110	01101110	n
79	01001111	O	111	01101111	o
80	01010000	P	112	01110000	p
81	01010001	Q	113	01110001	q
82	01010010	R	114	01110010	r
83	01010011	S	115	01110011	s
84	01010100	T	116	01110100	t
85	01010101	U	117	01110101	u
86	01010110	V	118	01110110	v
87	01010111	W	119	01110111	w
88	01011000	X	120	01111000	x
89	01011001	Y	121	01111001	y
90	01011010	Z	122	01111010	z
91	01011011	[	123	01111011	{
92	01011100	\	124	01111100	|
93	01011101	]	125	01111101	}
94	01011110	^	126	01111110	~
95	01011111	_	127	01111111	Delete

Порядковий номер	Код	Символ	Порядковий номер	Код	Символ
128	10000000	Ђ	160	10100000	
129	10000001	Ѓ	161	10100001	Ў
130	10000010	‚	162	10100010	ў
131	10000011	ѓ	163	10100011	Ј
132	10000100	„	164	10100100	¤
133	10000101	…	165	10100101	Ґ
134	10000110	†	166	10100110	¦
135	10000111	‡	167	10100111	§
136	10001000	€	168	10101000	Ё
137	10001001	‰	169	10101001	©
138	10001010	Љ	170	10101010	Є
139	10001011	‹	171	10101011	«
140	10001100	Њ	172	10101100	¬
141	10001101	Ќ	173	10101101	-
142	10001110	Ћ	174	10101110	®
143	10001111	Џ	175	10101111	Ї
144	10010000	ђ	176	10110000	°
145	10010001	‘	177	10110001	±
146	10010010	’	178	10110010	І
147	10010011	“	179	10110011	і
148	10010100	”	180	10110100	ґ
149	10010101	•	181	10110101	µ
150	10010110	–	182	10110110	¶
151	10010111	—	183	10110111	·
152	10011000	□	184	10111000	ё
153	10011001	™	185	10111001	№
154	10011010	љ	186	10111010	є
155	10011011	›	187	10111011	»
156	10011100	њ	188	10111100	ј
157	10011101	ќ	189	10111101	Ѕ
158	10011110	ћ	190	10111110	ѕ
159	10011111	џ	191	10111111	ї

Порядковий номер	Код	Символ	Порядковий номер	Код	Символ
192	11000000	А	224	11100000	а
193	11000001	Б	225	11100001	б
194	11000010	В	226	11100010	в
195	11000011	Г	227	11100011	г
196	11000100	Д	228	11100100	д
197	11000101	Е	229	11100101	е
198	11000110	Ж	230	11100110	ж
199	11000111	З	231	11100111	з
200	11001000	И	232	11101000	и
201	11001001	Й	233	11101001	й
202	11001010	К	234	11101010	к
203	11001011	Л	235	11101011	л
204	11001100	М	236	11101100	м
205	11001101	Н	237	11101101	н
206	11001110	О	238	11101110	о
207	11001111	П	239	11101111	п
208	11010000	Р	240	11110000	р
209	11010001	С	241	11110001	с
210	11010010	Т	242	11110010	т
211	11010011	У	243	11110011	у
212	11010100	Ф	244	11110100	ф
213	11010101	Х	245	11110101	х
214	11010110	Ц	246	11110110	ц
215	11010111	Ч	247	11110111	ч
216	11011000	Ш	248	11111000	ш
217	11011001	Щ	249	11111001	щ
218	11011010	Ъ	250	11111010	ъ
219	11011011	Ы	251	11111011	ы
220	11011100	Ь	252	11111100	ь
221	11011101	Э	253	11111101	э
222	11011110	Ю	254	11111110	ю
223	11011111	Я	255	11111111	я

Таблиця містить коди 128 символів, з яких 32 керівні (не виводяться ні на екран, ні на принтер), а інші 96 – літери латинського (англійського) алфавіту, розділові знаки, спеціальні знаки та цифри. Неважко побачити, що 8-й біт (відлік бітів – справа наліво) кожного коду цієї таблиці нульовий. Тому якщо інвертувати його значення, існує можливість додати до цієї таблиці ще одну, у якій також можна закодувати 128 символів. Так і було створено таблиці кодів національних символів різних країн. Наприклад, для роботи з кирилицею в *MS Windows* використовується кодова таблиця *win*1251.

ОЦІНЮВАННЯ СТУДЕНТАМИ МАТЕРІАЛІВ КУРСУ ТА РІВНЯ ВИКЛАДАННЯ

Після закінчення курсу необхідно дати відповіді на задані запитання, вписавши свої коментарі.

1. Наскільки Ви задоволені змістом дисципліни в цілому? Пропозиції до наповнення змісту курсу та критика.

Коментар

_____
_____
_____
_____

2. Як Ви оцінюєте якість викладання навчального матеріалу викладачем? Що б Ви запропонували змінити в методичному і змістовному плані для вдосконалення викладання цієї дисципліни?

Коментар

_____
_____
_____
_____

3. Що Ви вважаєте найскладнішим під час вивчення курсу (тема, задача, метод тощо)?

Коментар

_____
_____
_____
_____

4. Чи вважаєте Ви, що отримані знання будуть корисними у подальшому під час навчання / роботи? Що було найбільш корисним з точки зору подальшого навчання та / або застосування у практичній діяльності?

Коментар

_____
_____
_____
_____

Електронне навчальне видання
комбінованого використання
Можна використовувати в локальному та мережному режимах

Методичні вказівки до виконання лабораторних та практичних робіт з дисципліни «Обробка великих масивів даних» для студентів спеціальності 126 – «Інформаційні системи та технології»

Укладачі: *Іванов Юрій Юрійович*
Кривогубченко Сергій Григорович

Рукопис оформив *Ю. Іванов*

Редактор *О. Ткачук*

Оригінал-макет виготовлено *О. Кушнір*

Підписано до видання 24.06.2021 р.
Гарнітура Times New Roman.
Зам. № Р2021-012.

Видавець та виготовлювач
інформаційний редакційно-видавничий центр.
ВНТУ, ГНК, к. 114.
Хмельницьке шосе, 95,
м. Вінниця, 21021.
Тел. (0432) 65-18-06.
press.vntu.edu.ua;
E-mail: kivc.vntu@gmail.com.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.