

Міністерство освіти і науки України
Сумський державний університет



4636 Методичні вказівки
до лабораторних робіт
із дисциплін **«Інформаційна безпека телекомунікаційних
мереж», «Основи кібербезпеки в інформаційних мережах»**
та **«Основи протидії до хакерських атак»**
для студентів
спеціальності 172 *«Телекомунікації та радіотехніка»*
всіх форм навчання

Частина 1

Суми
Сумський державний університет
2019

Методичні вказівки до лабораторних робіт із дисциплін «Інформаційна безпека телекомунікаційних мереж», «Основи кібербезпеки в інформаційних мережах» та «Основи протидії до хакерських атак» / укладачі: О. В. Бережна, Т. О. Протасова, О. В. Д'яченко. – Суми : Сумський державний університет, 2019 – Ч. 1. – 15 с.

Кафедра електроніки і комп'ютерної техніки

ЗМІСТ

	С.
Передмова.....	4
Теоретичні відомості.....	4
Лабораторна робота 1 «LZW-стиснення».....	8
Список літератури.....	12
Додаток А.....	13

ПЕРЕДМОВА

Інформаційна безпека – одна з головних проблем, з якою стикається сучасне суспільство. Причиною загострення цієї проблеми є широкомасштабне використання автоматизованих засобів накопичення, зберігання, оброблення та передавання інформації. Вирішення проблеми інформаційної безпеки пов'язане з гарантованим забезпеченням трьох її головних складових: доступності, цілісності і конфіденційності.

Під час виконання лабораторної роботи 1 передбачається ознайомлення з принципами стиснення інформації з використанням методу LZW (Lempel – Ziv – Welch).

ТЕОРЕТИЧНІ ВІДОМОСТІ

LZW-стиснення. Одним із найбільш широко використовуваних сьогодні методів стиснення без втрат є алгоритм Лемпела – Зива – Уелча (Lempel – Ziv – Welch).

Метод стиснення був запропонований у 1977 р., вдосконалений (Terry Welch) варіант був опублікований у 1984 р. Надалі цей метод неодноразово модернізувався. Цей алгоритм використовується, зокрема, стандартною процедурою UNIX Compress. Алгоритми групи LZ (LZ77, LZ78, LZSS) лежать в основі майже всіх сучасних програмних і апаратних засобів стиснення інформації.

Архіватори PKZIP, LHA, Stacker, SuperStor, Dblspace і багато інших використовують ту або іншу модифікацію алгоритму LZ.

Ідея стиснення, запропонована Лемпелом і Зивом, полягає в такому: якщо в тексті повідомлення з'являється послідовність із двох символів, які раніше вже траплялися, то ця послідовність оголошується новим символом, для неї призначається код, який за певних умов може бути значно коротше за вихідну послідовність. Надалі в стислому повідомленні замість вихідної послідовності записується призначений код. Під час декодування повторюються аналогічні дії.

Процес кодування:

1. Кожному символу вихідного алфавіту присвоюється певний код (тут код – порядковий номер, починаючи з нуля).

2. Обирається перший символ повідомлення і замінюється на цей код.

3. Обираються наступні два символи і замінюються своїми кодами. Одночасно цій двосимвольній комбінації присвоюється свій код (зазвичай це порядковий номер, що дорівнює кількості вже використаних кодів). Так, якщо алфавіт містить вісім символів, що мають коди від 000 до 111, то перша двосимвольна комбінація отримає код 1000, наступна – 1001 й т.д.

4. Обираються з вихідного тексту чергові $Z, D \dots V$ –символьні комбінації доти, поки не утворюється комбінація, що ще не траплялася. Тоді цій комбінації присвоюється черговий код, а оскільки сукупність із перших $N-1$ символів вже траплялася, то вона має свій код, який і присвоюється замість цих $N-1$ символів. Кожен акт введення нового коду назвемо кроком кодування.

5. Процес триває до вичерпання вихідного коду.

Процес декодування

Під час декодування код першого символу, а потім другого та третього замінюються на символи алфавіту. Одночасно стає відомим код комбінації другого та третього символів. У наступній позиції можуть бути лише коди вже відомих символів та їх комбінацій. Процес декодування триває до вичерпання стислого тексту.

Приклад 1. Нехай вихідний текст являє собою двійковий код (перший рядок табл. 1), тобто вихідний алфавіт $A = \{1,0\}$. Коди цих символів відповідно також 0 і 1.

Утворений за методом Лемпела – Зива LZ-код показаний у другому рядку табл. 1. У третьому рядку відзначені кроки кодування, після яких відбувається перехід на подання кодів A зі збільшеною кількістю розрядів g .

Таблиця 1 – Ілюстрація процесу кодування

Вихідний текст	0.00.000.01. 11. 111.1111. 110. 0000.00000. 1101. 1110
LZ-код	0.00.100.001.0011.1011.1101. 1010.00110.10010.10001.10110
r	2 3 4
Коди, що вводяться	– 10 11 100 101 110 111 1000 1001 1010 1011 1100

Так, на першому кроці вводиться код 10 для комбінації 00 і з цього на наступний двох кроках $r = 2$, після третього кроку $r = 3$, після сьомого кроку $r = 4$. Загалом $r = k$ після кроку $2^{k-1}-1$.

У наведеному прикладі LZ-код виявився навіть довшим за вихідний код, оскільки зазвичай короткі тексти з невеликою кількістю символів, що повторюються, не дають ефекту стиснення. Ефект стиснення проявляється в досить довгих текстах і особливо помітний, наприклад, у графічних файлах.

Удосконалений алгоритм стиснення і розпакування за методом Лемпела – Зива – Уелча наведений нижче.

Алгоритм компресії

```

СТРОКА = черговий символ із вхідного потоку
WHILE вхідний потік не порожній DO
СИМВОЛ = черговий символ із вхідного потоку
IF СТРОКА + СИМВОЛ у таблиці рядків THEN
СТРОКА = СТРОКА + СИМВОЛ
ELSE
вивести у вихідний потік код для СТРОКА
додати в таблицю рядків СТРОКА + СИМВОЛ
СТРОКА = СИМВОЛ
END of IF
END of WHILE
вивести у вихідний потік код для СТРОКА
    
```

Алгоритм декомпресії

Читати СТАРИЙ_КОД

вивести СТАРИЙ_КОД

WHILE вхідний потік не порожній DO

читати НОВИЙ_КОД

СТРОКА = перевести НОВИЙ_КОД

вивести РЯДОК

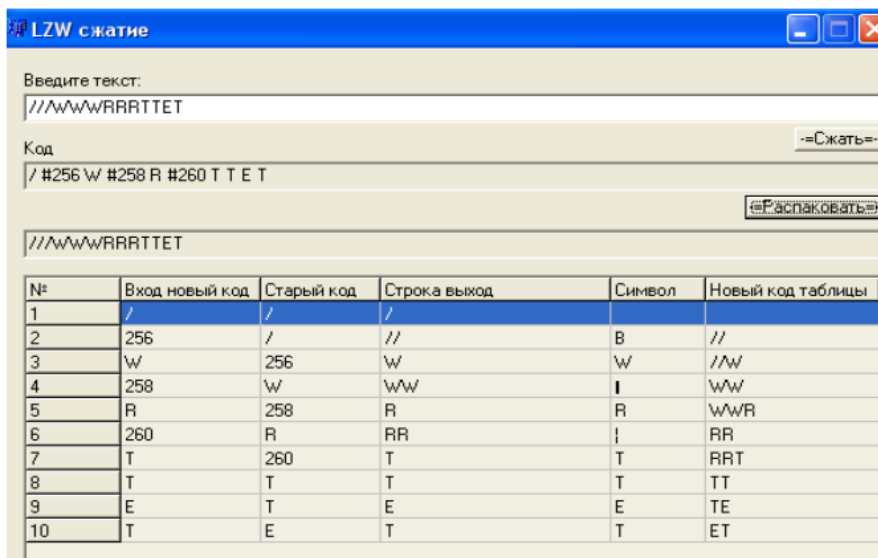
СИМВОЛ = перший символ РЯДКИ

додати в таблицю перекладу СТАРИЙ_КОД + СИМВОЛ

СТАРИЙ_КОД = НОВИЙ_КОД

END of WHILE

На рисунку 1 наведений приклад програмної реалізації алгоритму LZW.



Введите текст:
///wWwRRRTTET

Код
/ #256 w #258 R #260 T T E T

///wWwRRRTTET

№	Вход новый код	Старый код	Строка выкод	Символ	Новый код таблицы
1	/	/	/		
2	256	/	//	В	//
3	w	256	w	w	//w
4	258	w	wW	!	wW
5	R	258	R	R	wWR
6	260	R	RR	!	RR
7	T	260	T	T	RRT
8	T	T	T	T	TT
9	E	T	E	E	TE
10	T	E	T	T	ET

Рисунок 1 – Приклад програмної реалізації алгоритму LZW

ЛАБОРАТОРНА РОБОТА 1

«LZW-Стиснення»

Мета роботи: ознайомлення з принципами стиснення інформації методом LZW (Lempel – Ziv – Welch).

Примітка. Для виконання лабораторної роботи на комп'ютері необхідно встановити файл LZW.exe.

Описання лабораторної роботи. Лабораторна робота виконується на персональному комп'ютері в середовищі програми LZW.exe.

Під час запуску програми LZW.exe з'являється вікно, зображене на рис. 2, яке містить такі компоненти:

- рядок тексту, де вводиться текст для стиснення;
- рядок коду, де виводиться код стислого тексту;
- рядок коду, де виводиться розпакований текст;
- таблиця виконання кодування програмою.

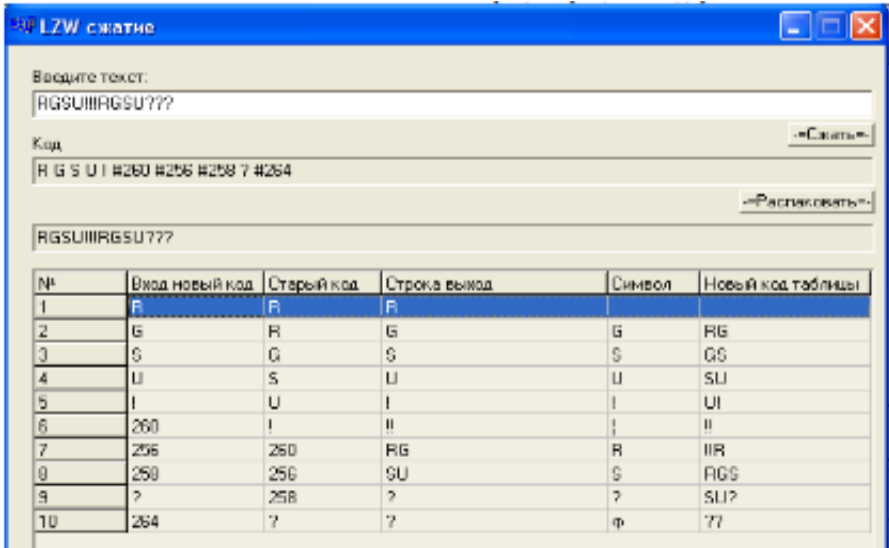


Рисунок 2 – Головне вікно програми

Програма LZW.exe дозволяє стискати та розпаковувати текст будь-якої довжини, записаний як кирилицею, так і латиницею, а також переглядати процес стиснення і розпаковування тексту.

Для стиснення будь-якого тексту потрібно виконати наступні дії:

- 1) У рядок тексту «Введіть текст» ввести текст для стиснення;
- 2) натиснути кнопку «Стиснути»; у рядку «Код» з'явиться код стислого тексту;
- 3) натиснути кнопку «Розпакувати»; у наступному рядку з'явиться розпакований текст;
- 4) переглянути таблицю виконання програми стиснення і розпаковування тексту.

Приклад 2

Оригінальний текст: RGSU!!!RGSU???

Код стислого тексту: R G S U ! # 260 # 256 # 258 ? # 264

Розпакований текст: RGSU!!!RGSU???

Процес кодування поданий на рис. 2 та 3.

№	Вкод новий код	Старий код	Строка вкод	Символ	Новый код таблицы
1	R	R	R		
2	G	R	RG	G	RG
3	S	G	SG	S	GS
4	U	S	SU	U	SU
5	!	U	U!	!	U!
6	260	!	U!!	!	U!!
7	256	260	RG260	R	U!R
8	258	256	SU256	S	RG256
9	?	258	SU?258	?	SU?258
10	264	?	SU?264	Ф	SU?264

Рисунок 3 - Процес кодування

Примітка. LZW-стиснення виділяється серед інших. Коли стикається з потоком даних, що містить повторювані рядки будь-якої структури, рівень стиснення може досягати 50 % і вище.

Завдання

1. Стиснути і потім розпакувати вихідні рядки символів латинського алфавіту, які набираються безпосередньо у вікні програми.

Зберегти у звіті екранні форми, що демонструють процес стиснення і розпаковування інформації. Проаналізувати зроблену роботу і зробити висновки про ефективність алгоритму стиснення і розпаковування.

2. Стиснути і потім розпакувати вихідні рядки символів кириличного алфавіту, які набираються безпосередньо у вікні програми. Зберегти у звіті екранні форми, що демонструють процес стиснення і розпаковування інформації. Проаналізувати зроблену роботу і зробити висновки про ефективність алгоритму стиснення і розпаковування.

3. Стиснути і потім розпакувати вихідні рядки чисел, які набираються безпосередньо у вікні програми. Зберегти у звіті екранні форми, що демонструють процес стиснення і розпаковування інформації. Проаналізувати зроблену роботу і зробити висновки про ефективність алгоритму стиснення і розпаковування.

4. Стиснути і потім розпакувати вихідні рядки математичних функцій і формул, які набираються безпосередньо у вікні програми. Зберегти у звіті екранні форми, що демонструють процес стиснення і розпаковування інформації. Проаналізувати зроблену роботу і зробити висновки про ефективність алгоритму стиснення і розпаковування.

5. Стиснути і потім розпакувати вихідні рядки довільних символів, які набираються безпосередньо у вікні програми. Зберегти у звіті екранні форми, що демонструють процес стиснення і розпаковування інформації. Проаналізувати зроблену роботу і зробити висновки про ефективність алгоритму стиснення і розпаковування.

6. Додати до звіту:

– результати стиснення і розпаковування текстів (варіанти вказані в додатку А) за допомогою програми LZW-стиснення;

- зберегти у звіті кодову таблицю у вигляді таблиці або рисунка;
- проаналізувати отримані результати і сформулювати аргументовані висновки.

7. Навести у звіті відповіді на контрольні питання відповідно до номера варіанта, зазначеного викладачем (табл. 2).

Таблиця 2 – Контрольні питання

Номер варіанта	Контрольні питання
1,5, 7,3, 9, 18, 28	У чому полягає принцип LWZ-стиснення?
2 ,4 ,6 , 8, 20, 22, 24, 26, 30	Які формати файлів найбільш ефективно архівувати за допомогою LWZ-стиснення?
И , 13, 15, 10, 17, 19, 27	У чому полягає особливість процесу розпакування в разі використання алгоритму LWZ?
12, 14, 16 21,23,25,29	Які достоїнства та недоліки має алгоритм LWZ?

СПИСОК ЛИТЕРАТУРИ

1. Информационная безопасность. Лабораторный практикум (+CD) : учебное пособие / А. В. Бабаш, Е. К. Баранова, Ю. Н. Мельников. – 2-е изд. – Москва : КНОРУС, 2013. – 136 с.
2. Баранова Е. К. Эффективное кодирование и защита информации : текст лекций для студентов специальности 510200. – Москва : МГУЛ, 2002. – 88 с.
3. Вернер М. Основы кодирования : учебник для вузов. – Москва : Техносфера, 2006. – 288 с.
4. Крушный В. В. Основы теории информации и кодирования. – Снежинск : СГФТА, 2005. – 69 с.
5. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. – Москва : Техносфера, 2006. – 320 с.
6. Теория информации и кодирования. / Филоненков А. И., Самсонов Б. Б., Кречет Т. В., Плохов Е. М. – Москва : ФЕНИКС, 2002. – 288 с.
7. Хохлов Г. И. Основы теории информации : учеб. пособие для студ. высш. учеб. заведений / Г. И. Хохлов. – Москва : Академия, 2008. – 176 с.
8. Шеннон К. Математическая теория связи. Работы по теории информации и кибернетике / К. Шеннон. пер. с англ. ; под ред. Р. Л. Добрушина и О. Б. Лупанова. – Москва : ИЛ, 1963. – 512 с.

ДОДАТОК А
(обов'язковий)

Таблиця А.1 – Вихідні дані для виконання завдання

Номер варіанта	Номер завдання до лабораторної роботи				
	1	2	3	4	5
1	Program	інформація	612534712	$X+Y+Z-12=5$!!A/b/C/dE@?
2	Processor	Математика	954723890	$2*X+Y+Z=2$	ЪС<л>*ш/dE?
3	Operation	Мова	410125935	$X*X+2*Y-3$	[4Pu/t%/3\$1\$
4	Information	обчислення	181952471	$Y*Y+Z=4*X$	Ё1?ОяЧ!/Д/;
5	Mathematics	Суспільство	365412389	$X+4Y+Z-2+1$	ЙрТиV{G}%
6	Language	технологія	113212598	$X+2Y+5*Z=3$	ёЮВ&<h>@
7	Company	кодування	725847391	$5*X+Y+8*Z=5$	ЖО[Д]m/я/12
8	Encoding	Інформатика	820191856	$X-6*Y-3*Z=2$	ВН!(И)13№
9	Informatics	забезпечення	100258741	$X-2*Y+7*Z-9$	Б!15лЛ2ыф»
10	Software	операція	101296752	$8*X/Y+Z/2=5$	y4?-+/25/8?8
11	Editor	редактор	496123578	$5X/2+Y/12+Z$	uФ/z-1/2+3#
12	Table	процесор	123457890	$X/5+5Y+Z/12$	Г@14^5D/G
13	Security	Таблиця	154789123	$7*X>Y+2*8Z$	Г13#&7Fоэ
14	Cybernetics	Безпека	571236984	$X*X4*Y<Z+5$	ЩП/'ы'45*7

Продовження таблиці А.1

Номер варіанта	Номер завдання до лабораторної роботи				
	1	2	3	4	5
15	Knowledge	знання	432987067	$X+3*Y>Z+27$	i5(B)<px/Й
16	Computer	кібернетика	433129851	$X/2+2*Y>3*Z$	шЁ8впо/тп5?
17	System	комп'ютер	184569347	$7*X+Y-Z/2>45$	bРлOas156+
18	Swindle	система	189242578	$Y<5*X/3+Z-2$	Z~+258 =[f]
19	Logic	шахрайство	213459632	$2*Z>4*X+5*Y$	C8&8вАъ/ё
20	Programming	логіка	214357980	$X/5+Y/4+Z=5$	n-q6&8%^
21	Hardware	програмування	896124587	$Y/4-2X+3Z=6$	S?dkошй^*
22	Intellec	техніка	654932140	$X+2>Z/4Y+2$	P5%jkGhgdku
23	Internet	інтелект	781258743	$X+7*Z<2Y+3$	J5%ojkЖЁ%
24	Translator	інтернет	834107892	$2*X-7Z=2+Y$	Fd&5&%№
25	Driver	транслятор	417625877	$6X*X+2*Y-3$	Dfi&5*6-2

Навчальне видання

Методичні вказівки
до лабораторних робіт
із дисциплін **«Інформаційна безпека телекомунікаційних
мереж», «Основи кібербезпеки в інформаційних
мережах»** та **«Основи протидії до хакерських атак»**
для студентів
спеціальності 172 *«Телекомунікації та радіотехніка»*
всіх форм навчання

Частина 1

Відповідальний за випуск А. С. Опанасюк
Редактор І. О. Кругляк
Комп'ютерне верстання О. В. Бережної

Формат 60x84/16. Ум. друк. арк. 0,93. Обл.-вид. арк. 0,97.

Видавець і виготовлювач
Сумський державний університет,
вул. Римського-Корсакова, 2, м. Суми, 40007
Свідоцтво суб'єкта видавничої справи ДК № 3062 від 17.12.2007.