­­МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

­­­

Лабораторна робота №2

з курсу «**Проблемно-орієнтовані методи та засоби**

**інформаційних тенологій**»

для студентів базового напрямку 6.08.04 "Комп’ютерні науки"

(заочна форма навчання)

Варіант 10

Виконав студент гр. КНз-3

Чалий Михайло

­­

Львів 2015

## Мета роботи

Вивчення методу Лемпеля-Зіва-Велча (LZW) для стиснення повідомлень та підходів до його програмної реалiзацiї для практичного використання.

## Теоретичні відомості

Відомі методи оптимального кодування різнозначними кодами (кодами різної довжини), які базуються на врахуванні різної частоти символів в тексті. Найбільш відомі з них: метод Шеннона-Фано та метод Хаффмана. Ці методи забезпечують достатньо ефективне стиснення без втрат. Для підвищення рівня стиснення даних доцільно перейти від оптимального кодування окремих символів до кодування буквосполучень. Саме таке кодування забезпечує метод LZW.

Метод базується на побудові таблиці фраз (словника), яка відображає стрічки символів повідомлення, що стискається, в коди фіксованої довжини (12 біт). Таблиця має властивість попередництва, тобто для кожної фрази словника, яка складається з деякої фрази w і символа K, фраза w теж міститься в словнику і представляється відповідним номером n.

Програмна реалізація методу здійснюється за наступним алгоритмом (ілюструється на прикладі кодування фрази МАМАМАЛАЛАМУ з представленням словника у вигляді таблиці:

* + - 1. У словник заносяться односимвольні фрази w, що відповідають символам вхідного алфавіту (ініціація словника).
      2. Біжучий вказівник встановлюється на перший символ тексту (фраза М).
      3. Здійснюється пошук фрази М у словнику (фраза є за номером 1).
      4. Фраза М доповнюється наступним символом А і здійснюється пошук розширеної фрази МА у словнику (фраза відсутня).
      5. Знайдена раніше частина розширеної фрази МА (фраза М) кодується її номером 1, а нова фраза МА заноситься у словник під номером 5 і представляється (у формі nK) кодом 1А.
      6. Після цього, вказівник встановлюється на останній символ А розширеної фрази МА, який розглядається і кодується як нова фраза, аналогічно як у п. 3.
      7. Фраза А є у словнику під номером 2.
      8. Фраза А доповнюється наступним символом М і здійснюється пошук розширеної фрази АМ у словнику (фраза відсутня).
      9. Знайдена частина розширеної фрази АМ (фраза а) кодується її номером 2, а нова фраза АМ заноситься в словник під номером 6 і представляється кодом 2М.
      10. Вказівник встановлюється на останній символ М розширеної фрази АМ, який у свою чергу розглядається і кодується як нова фраза.
      11. Здійснюється пошук фрази М у словнику (фраза є за номером 1).
      12. Фраза М доповнюється наступним символом А і здійснюється пошук розширеної фрази МА у словнику (фраза є за номером 5).
      13. Фраза МА доповнюється наступним символом М і здійснюється пошук розширеної фрази МАМ (фраза відсутня).
      14. Знайдена частина (фраза МА) розширеної фрази МАМ кодується відповідним номером 5, а відсутня фраза МАМ заноситься у словник під номером 7.
      15. Кроки циклічно повторюються до закінчення послідовності кодованих символів.

Результати кодування представлені на рис. 1 і таблицею словника.

## Завдання

1. Ознайомтесь з методом LZW.

2. Для заданого варіанту вручну проведіть стиснення тексту і визначіть коефіцієнт стиснення.

3. Розробiть алгоритм та реалiзуйте програму побудови коду LZW.

1. Реалiзуйте програму декодування стиснених даних.
2. Перевiрте роботоздатнiсть програм на тестових прикладах.
3. Дайте оцінку методу стиснення LZW.

|  |  |
| --- | --- |
| Варіант 10 | КАЛЕВАЛАВАЛКА |

## Реалізація

Лістінг 1. lzw.js

function initDict(input){

var dict = {};

// Add ABC

input.split('').forEach(function(sym){

if (!dict[sym]){

dict[sym] = '' + (Object.keys(dict).length + 1);

}

});

return dict;

}

function encode(input, init){

var init = init || initDict(input);

var phrases = {};

var output = '';

Object.keys(init).forEach(function(phrase){

phrases[phrase] = init[phrase];

});

var index = function(item){

return '' + (Object.keys(phrases).indexOf(item) + 1);

}

var phrase = '';

for(var i = 0; i < input.length; i++){

var k = input[i];

if (phrases[phrase + k]){

phrase += k;

} else {

var prevIndex = index(phrase);

phrases[phrase + k] = prevIndex + k;

output += phrases[phrase];

phrase = k;

}

}

if (phrase){

output += phrases[phrase];

}

return {

dictionary: phrases,

init: init,

output : output

}

}

function decode(input, init){

var codes = {};

var output = '';

Object.keys(init).forEach(function(phrase){

codes[init[phrase]] = phrase;

});

var index = function(code){

return '' + (Object.keys(codes).indexOf(code) + 1);

}

var code = '';

for(var i = 0; i < input.length; i++){

var n = input[i];

if (codes[code + n]){

code += n;

} else {

var prevPhrase = codes[code];

output += prevPhrase;

var k = codes[n];

codes[index(code) + k] = prevPhrase + k;

code = n;

}

}

if (code){

output += codes[code];

}

return output;

}

// var input = 'МАМАМАЛАЛАМУ';

// console.log(input);

// var res = encode(input);

// console.log(res);

//

// var res2 = decode(res.output, res.init);

// console.log(res2);

module.exports.encode = encode;

module.exports.decode = decode;

Лістінг 2. l2.js

var input = 'КАЛЕВАЛАВАЛКА';

console.log('Вхідний текст: ', input);

var lzw = require('./lib/lzw');

var result = lzw.encode(input);

console.log('LZW Dictionary:', result.dictionary);

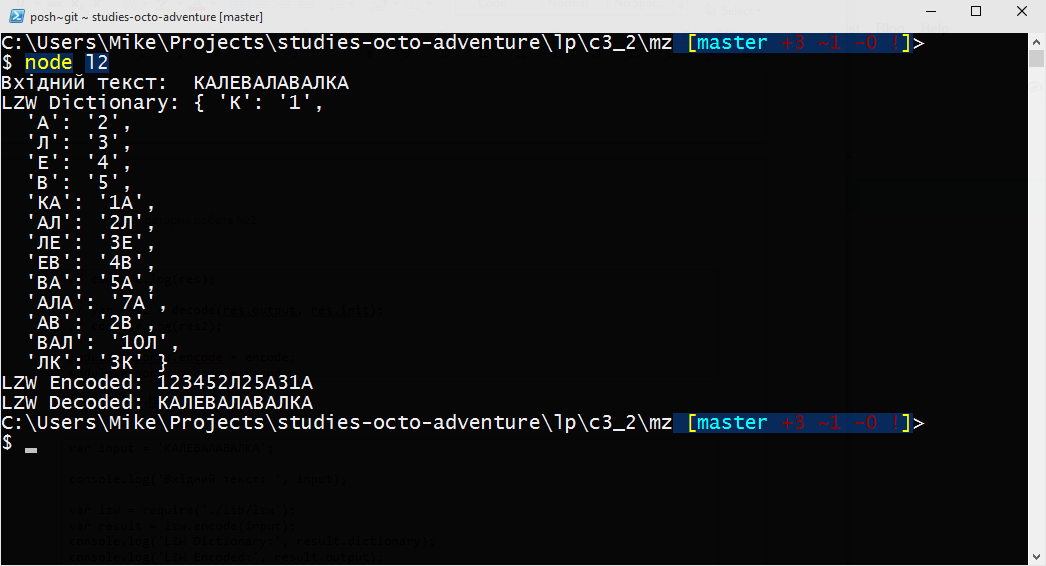
console.log('LZW Encoded:', result.output);

var decoded = lzw.decode(result.output, result.init);

console.log('LZW Decoded:', decoded);

## https://github.com/chaliy/studies-octo-adventure/tree/master/lp/c3\_2/mz

## Результат



## Висновки

Вивчив методу Лемпеля-Зіва-Велча (LZW) для стиснення повідомлень та підходів до його програмної реалiзацiї для практичного використання.