

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Алгоритмы и структуры данных»
Тема: Метод Шеннона-Фано

Студент гр. 9304

Краев Д.В.

Преподаватель

Филатов А.Ю.

Санкт-Петербург

2020

Цель работы.

Изучить методы кодирования. Реализовать алгоритмы кодирования и декодирования Шеннона-Фано.

Задание.

Вариант 2

Метод Шеннона-Фано

Выполнение работы.

1) Функции

Для реализации алгоритма циклической сортировки были написаны несколько функций.

1.1) void comp(const std::shared_ptr<Node> &a, const std::shared_ptr<Node> &b)

Данная функция является компаратором для сортировки функцией std::sort. Функция принимает две ссылки на умные указатели на Node и возвращает 1, если a больше b, 0 в ином случае.

1.2) countLetters(const std::string text)

Функция принимает на вход текст и считает количество каждого символа в тексте. Возвращает вектор, состоящий из структур Code, содержащую имя символа и его количество в тексте.

1.3) createTree(std::vector<std::shared_ptr<Node>> &elems)

Функция создает бинарное дерево из символов текста по методу Шеннона-Фано. Возвращает умный указатель на корень дерева.

1.4) setCodes(const std::shared_ptr<Node> &nde, std::vector<Code> &codes, std::string code = "")

Функция принимает умный указатель на корень бинарного дерева, полученного предыдущей функцией и ссылку на вектор, в который будут

записаны символы и их коды. Функция рекурсивно высчитывает коды символов по методу Шеннона-Фано.

1.5) writeCodes(std::vector<Code> codes, std::string f)

Функция принимает на вход вектор кодов символов и название файла и выводит коды символов в данный файл.

1.6) encode(const std::string text, std::vector<Code> codes, std::string f)

Функция принимает на вход текст, который нужно закодировать, коды символов, и файл, в который нужно вывести закодированный текст. Функция производит кодирование текста по заданным кодам символов.

1.7) decode(std::string encodedText, std::string decoded, std::vector<Code> codes)

Функция принимает на вход текст, состоящий из 0 и 1, который нужно раскодировать, название файла, в который нужно записать раскодированный текст и коды символов. Функция производит раскодировку текста и записывает ее в заданный файл.

1.8) readText(std::string file)

Функция принимает название файла, считывает текст из этого файла и возвращает его.

1.9) readCodes(std::string file)

Функция принимает на вход название файла, в котором записаны коды символов, считывает их и возвращает вектор из структур Code.

2) Структуры

2.1) Code

Структура содержит 2 поля типа std::string: первое поле хранит сам название символа, второе поле хранит его код.

2.2) Node

Структура является узлом бинарного дерева, поэтому содержит указатели 2 поля, которые указывает на левого и правого потомка. Структура

так же содержит название символа и его количество повторений в заданном тексте.

2.3) args

Структура была создана для реализации CLI. Поэтому содержит информацию об аргументах, подаваемых программе. Структура содержит 5 полей. Первые два поля типа bool обозначают какой режим программы включен, так как должен быть включен только 1 режим, при равенстве этих 2 полей будет выведено сообщение об ошибке «ERROR: you must enable 1 of 2 modes». Следующие 3 поля типа std::string содержат названия различных файлов. Поле iFile содержит название файла ввода, поле oFile содержит название файла вывода, поле codes содержит название файла, содержащего кодировку символов. В поле oFile всегда должно быть определено, в ином случае будет выведено сообщение об ошибке «ERROR: you must set output file». Если поле iFile не было определено начнется считывание текста с терминала.

3) Command line interface

Для реализации программы был создан CLI. С помощью него можно совершить 4 команды:

- d (--decode) «файл, содержащий коды символов» — установить режим декодирования
- e (--encode) «файл, содержащий коды символов» — установить режим кодирования
- f (--file) «название файла» — установить файл ввода
- o (--out) «название файла» — установить файл вывода

Тестирование

Тестирование проводится с помощью скрипта, написанном на языке Python. Скрипт использует библиотеки unittest и subprocess. Скрипт проводит 3 теста. В первом тесте производится кодировка и декодировка текста,

декодированный текст сравнивается с изначальным. В тестах 2, 3 производится проверка обработки ошибок. Библиотека subprocess нужна для запуска программы с нужными входными данными, а библиотека unittest для проведения тестирования.

Скрипт можно запустить при помощи команды «make run_tests».

Таблица 1 — Результаты тестирования

№ п/п	Входные данные	Выходные данные
1.	<pre>./lab --encode Tests/codes.txt --file Tests/test.txt --out Tests/encoded_text.txt\n") ./lab --decode Tests/codes.txt --file Tests/encoded_text.txt --out Tests/decoded_text.txt\n")</pre>	<p>Around 1948, both Claude E. Shannon (1948) and Robert M. Fano (1949) independently proposed two different source coding algorithms for an efficient description of a discrete memoryless source. Unfortunately, in spite of being different, both schemes became known under the same name Shannon-Fano coding. There are several reasons for this mixup. For one thing, in the discussion of his coding scheme, Shannon mentions Fano's scheme and calls it "substantially the same" (Shannon, 1948, p. 17). For another, both Shannon's and Fano's coding schemes are similar in the sense that they both are efficient, but suboptimal prefix-free coding schemes with a similar performance.</p> <p>Around 1948, both Claude E. Shannon (1948) and Robert M. Fano (1949) independently proposed two different source coding algorithms for an efficient description of a discrete memoryless source. Unfortunately, in spite of being different, both schemes became known</p>

		under the same name Shannon-Fano coding. There are several reasons for this mixup. For one thing, in the discussion of his coding scheme, Shannon mentions Fano's scheme and calls it "substantially the same" (Shannon, 1948, p. 17). For another, both Shannon's and Fano's coding schemes are similar in the sense that they both are efficient, but suboptimal prefix-free coding schemes with a similar performance.
2.	./lab	ERROR: you must enable 1 of 2 modes
3.	./lab --encode Tests/codes.txt	ERROR: you must set output file

Вывод.

Изучили алгоритм Шеннона-Фано. Написали программу, реализующую данный алгоритм кодирования и декодирования. Провели тестирование данной программы.