

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: Алгоритмы кодирования

Студент гр. 9303

Дюков В.А.

Преподаватель

Филатов А.Ю.

Санкт-Петербург

2020

Цель работы.

Реализовать алгоритм динамического кодирования по Хаффману.

Задание.

5 вариант. Кодирование: Динамическое Хаффмана.

Основные теоритические положения.

Адаптивное кодирование Хаффмана (также называемое динамическое кодирование Хаффмана) — адаптивный метод, основанный на кодировании Хаффмана. Он позволяет строить кодовую схему в поточном режиме (без предварительного сканирования данных), не имея никаких начальных знаний из исходного распределения, что позволяет за один проход сжать данные. Преимуществом этого способа является возможность кодировать на лету.

Существует несколько реализаций этого метода, наиболее примечательными являются «ФГК».

Он позволяет динамически регулировать дерево Хаффмана, не имея начальных частот. В ФГК дереве Хаффмана есть особый внешний узел, называемый *0-узел*, используемый для идентификации входящих символов. То есть, всякий раз, когда встречается новый символ — его путь в дереве начинается с нулевого узла. Самое важное — то, что нужно усекать и балансировать ФГК дерево Хаффмана при необходимости, и обновлять частоту связанных узлов. Как только частота символа увеличивается, частота всех его родителей должна быть тоже увеличена. Это достигается путём последовательной перестановки узлов, поддеревьев или и тех и других.

Важной особенностью ФГК дерева является принцип братства (или соперничества): каждый узел имеет два потомка (узлы без потомков называются листьями) и веса идут в порядке убывания. Благодаря этому

свойству дерево можно хранить в обычном массиве, что увеличивает производительность.

Выполнение работы.

Для выполнения работы были созданы классы:

- *Tree* – класс дерева
- *TreeList* – класс, отвечающий за обработку дерева: добавление ветвей, перестроение дерева, изменение весов и отрисовку дерева.

В бесконечном цикле из файла, который следует закодировать, программа считывает один символ при нажатии клавиши *enter* и вызывает метод *add* класса *TreeList*. Функция записывает кодировку символа в файл с результатом следующим образом. При поступлении нового символа в файл записывается сначала путь до зарезервированного листа, а затем записывается код самого символа. Если поступающий элемент ранее уже встречался, записывается его код в дерево. Далее происходит перестройка дерева с учётом поступившего элемента в методе *rebuild*. В списке (в котором хранятся все узлы дерева в порядке убывания), происходит обход всех узлов с конца. Если текущий элемент списка меньше предыдущего (1), то условие упорядоченности нарушено. В таком случае, обход в списке продолжается до тех пор, пока не найдётся первый элемент, который не меньше, чем (1), и элемент (1) меняются данными с предыдущим, только-что найденного.

Процедура считывание и обновления продолжается пока файл не достигнет конца.

Тестирование.

№	Входные данные	Результат
1	shhooowwweel	s 0 h 01 00 o 001 11 100 w 1001 111 1100 e 0001 0100 1
2	hello	h 0 e 00 1 101 110 o

Выводы.

В ходе выполнения работы был изучен динамический алгоритм Хаффмана. Были изучены его достоинства и недостатки. Был реализован динамический алгоритм Хаффмана методом ФГК.