

ОТЧЕТ

по лабораторной работе №1

Тема: Кодирование и декодирование кодом Хаффмана (адаптивное)

Выполнила: Белокон Юлия, ст. гр. М4106

Описание алгоритма кодирования и декодирования

Адаптивный алгоритм кодирования-декодирования Хаффмана является модификацией обычного. Он позволяет не передавать таблицу кодов и ограничиться одним проходом по сообщению (и при кодировании, и при декодировании).

Суть такого алгоритма состоит в том, что при каждом сопоставлении символу кода изменяется внутренний ход вычислений так, что в следующий раз этому же символу может быть сопоставлен другой код, то есть происходит адаптация алгоритма к поступающим для кодирования символам.

В адаптивном алгоритме Хаффмана используется упорядоченное бинарное дерево. В начале работы оно содержит только один специальный символ, всегда имеющий частоту 0 (он необходим для занесения новых символов).

Элементы входного сообщения считываются побайтно.

Если входной символ присутствует в дереве, в выходной поток записывается код (последовательность нулей и единиц согласно веткам дерева), вес узла увеличивается на 1. Веса узлов-предков корректируются. Если дерево становится неупорядоченным, происходит его упорядочение.

Если же входной символ в дереве отсутствует, в выходной поток записывается последовательность нулей и единиц, которыми помечены ветки бинарного дерева при движении от корня к символу, а затем 8 бит кода нового символа. В дерево добавляется ветка (родитель и два потомка, левый из которых становится символом, а правый – новым добавленным в дерево символом).

При декодировании происходит аналогичный процесс. Элементы входного сообщения считываются побитно, и каждый раз при считывании 0 или 1 происходит перемещение от корня вниз по соответствующей ветке бинарного дерева, пока не будет достигнут лист.

Если достигнут лист, соответствующий символу, в выходное сообщение записывается код данного символа. Вес листа увеличивается на 1, веса узлов-предков корректируются, дерево при необходимости упорядочивается.

Если же достигнут символ, из входного сообщения считываются 8 следующих бит, соответствующих коду нового символа. В выходное сообщение записывается код данного символа. В дерево добавляется новый символ, веса узлов-предков корректируются, при необходимости дерево упорядочивается.

Оценка кодирования и декодирования

Имя файла	$H(X)$	$H(X X)$	$H(X XX)$	Ср. затраты на символ (бит)	Размер сжатого файла (байт)
bib	5,20	3,36	2,31	5,24	72 889
book1	4,53	3,59	2,81	4,56	438 513
book2	4,79	3,75	2,74	4,83	368 478
geo	5,65	4,26	3,46	5,70	72 939
news	5,19	4,09	2,92	5,23	246 583
obj1	5,95	3,46	1,40	6,11	16 411
obj2	6,26	3,87	2,27	6,31	194 540
paper1	4,98	3,65	2,33	5,04	33 484
paper2	4,60	3,52	2,51	4,65	47 760
paper3	4,67	3,56	2,56	4,71	27 403
paper4	4,70	3,48	2,21	4,80	7 977
paper5	4,94	3,53	2,04	5,06	7 562
paper6	5,01	3,61	2,25	5,07	24 170
pic	1,21	0,82	0,71	1,66	106 762
progс	5,20	3,60	2,13	5,26	26 061
progl	4,77	3,21	2,04	4,81	43 117
progp	4,87	3,19	1,76	4,92	30 355
trans	5,53	3,36	1,93	5,58	65 386
Суммарный размер сжатых файлов (байт)					1830390

Использованные модули

Модуль **os** – множество функций для работы с операционной системой; здесь использован для работы с файлами, а именно для обращения к путям до этих файлов и проверки их существования.

Модуль **sys.argv** – модуль, позволяющий получать аргументы командной строки; здесь использован для считывания имени входного файла.

Модуль **queue** – модуль для работы с очередями; здесь используется при работе с деревом.

Остальные конструкции «from ... import ...» прописаны для подключения классов, разработанных в рамках данной работы.

Запуск программы на исполнение

1. Переместите файлы, с которыми планируется работать, в папку с программой;
2. Откройте командную строку;
3. В командной строке перейдите в папку, где лежит код и необходимые файлы;
4. Введите команду для начала исполнения, где первым аргументом является файл запускаемой программы, а вторым – файл, с которым эта программа будет работать.

Когда программа отработает, в командную строку выведется слово «Completed», а рядом с исходным файлом появится файл с припиской «_encoded» или «_decoded» в конце имени в зависимости от того, выполнялось кодирование или декодирование.

Примеры:

```
C:\Users\julia> cd ./desktop/try  
  
C:\Users\julia\Desktop\try>python encode.py bib  
Completed
```

```
C:\Users\julia\Desktop\try>python decode.py bib_encoded  
Completed
```

Использованные источники

1. Кудряшов Б.Д. – Теория информации, 2009
2. Кудрина М.А., Кудрин К.А., Дегтярева О.А., Сопченко Е.В. – Адаптивный алгоритм Хаффмана сжатия информации» - Труды Международного симпозиума Надежность и качество, 2015, том 1