Домашнее задание 3

Дарья Яковлева, M3439

Вариант 32

Задание: Изучить универсальные алгоритмы сжатия данных на заданном тесте и сопоставить их результаты с характеристиками стандартных архиваторов.

Текст: A_PENNY_SAVED_IS_A_PENNY_EARNED. (1)

Длина: 32

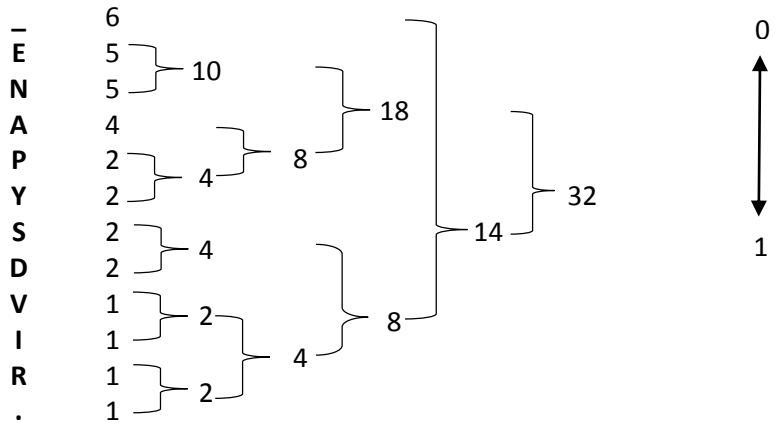
Алгоритмы сжатия

1) **Двухпроходное кодирование с использованием кода Хаффмена**

Код Хаффмена (в том числе регулярный)

Буква	Число появлений	Длина кодового слова	Кодовое слово
_	6	2	10
E	5	3	000
N	5	3	001
A	4	3	010
P	2	4	0110
Y	2	4	0111
S	2	4	1100
D	2	4	1101
V	1	5	11100
I	1	5	11101
R	1	5	11110
.	1	5	11111

Кодовое дерево кода Хаффмена (в том числе регулярное)



- Объем алфавита |X| = 256
- Кодовое слово $c(x) = c_1(x) + c_2(x)$, $len(c_1(x)) = l_1$, $len(c_2(x)) = l_2$

$l_2 = 6 * 2 + 5 * 3 * 2 + 4 * 3 + 2 * 4 * 4 + 1 * 5 * 4 = 106$ бит

- Двоичная последовательность кодового дерева

Построение: по ярусам, 0 – промежуточная вершина, 1 – концевая вершина

вершины	0	00	0010	111000	111100	1111
Ярус	0	1	2	3	4	5

l_1 = 23 + 8 * 12 = 119 бит

- Передача всего сообщения потребует 119 + 106 = **225 бит**

  Без кодирования = 8 * 32 = 256 бит

- Подсчет количества битов на передачу дерева регулярного кода

Ярус	Число вершин	Число концевых вершин n_i	Диапазон значений n_i	Затраты в битах
0	1	0	0..1	1
1	2	0	0..2	2
2	4	1	0..4	3
3	6	3	0..6	3
4	6	4	0..6	3
5	4	4	0..4	3
Всего на передачу регулярного кода				**15**

На передачу информации о буквах потребуется:

$$\left\lceil \log\binom{256}{1} \right\rceil + \left\lceil \log\binom{255}{3} \right\rceil + \left\lceil \log\binom{252}{4} \right\rceil + \left\lceil \log\binom{248}{4} \right\rceil = \left\lceil \log\left(\frac{256!}{1!255!}\right) \right\rceil + \left\lceil \log\left(\frac{255!}{3!252!}\right) \right\rceil + \left\lceil \log\left(\frac{252!}{4!248!}\right) \right\rceil + \left\lceil \log\left(\frac{248!}{4!244!}\right) \right\rceil = 8 + 22 + 28 + 28 = 86 \text{ бит}$$

- Количество бит для передачи сообщения, используя регулярное дерево Хаффмана

  l = 106 + 15 + 86 = **207 бит**

2) Адаптивное кодирование с применением арифметического кодирования

Воспользуемся алгоритмом D

Рассмотрим оценку

$$\mathrm{p_n}(a) = \begin{cases} \dfrac{t_n(a) - \frac{1}{2}}{n}, t_n(a) > 0 \\ \dfrac{M_n}{2n} * \dfrac{1}{M - M_n}, t_n(a) = 0 \end{cases}$$

Объем алфавита M = 256

M_n -- число различных букв в последовательности длины n.

$t_n(a)$ – число появлений буквы a в последовательности длины n.

- Отсортированная композиция:

  t = (6, 5, 5, 4, 2, 2, 2, 2, 1, 1, 1, 1)

- Рассчитаем оценку вероятности последовательности G

$$G = \frac{(6*5^2*4*2^4)}{32!} * \frac{(256-12)!}{256!} = 6*10^{-61}$$

$$l = -[\log(G)] + 1 = \mathbf{202} \text{ бита}$$

3) Нумерационное кодирование

Кодовое слово = номер композиции t_n(**x**) в списке всех возможных; номер данной последовательности в лексикографически упорядоченном списке последовательностей с одинаковой композицией.

Рассмотрим нашу отсортированную композицию:

t = (6, 5, 5, 4, 2, 2, 2, 2, 1, 1, 1, 1)

Композиция сортированной композиции:

t' = (1, 2, 1, 4, 4, 244)

- Длина второй части кодового слова

$$l_2(x) = [\log(N(t)] = \left[\log\left(\frac{n!}{t_0!..t_M!}\right)\right] = \left[\log\left(\frac{32!}{6!\,5!^2\,4!\,2!^4}\right)\right] = 86 \text{ бит}$$

- Длина первой части кодового слова

$$l_1(x) = [\log\left(n * \prod\prod t_j)\right] + \left[\log\left(\frac{M!}{\prod t_j'!}\right)\right] = [\log(32 * \ 6 * 5^2 \ * 4 * 2^4)] + \left[\log\left(\frac{256!}{4!^2\,2!\,244!}\right)\right]$$
$$= 19 + 86 = 105 \text{ бит}$$

- Общая длина
  $l(x) = l_1(x) + l_2(x) = 105 + 86 =$ **191 бит**

4) Алгоритм Зива-Лемпела-77 (метод скользящего словаря)

Флаг -- найдено ли подходящее слово в словаре

d – расстояние до слова в словаре, передается равномерным кодом, в скобках указывается текущая длина окна

l – длина слова (совпадения), кодируется монотонным кодом

bin(x) – стандартный 8-битный код буквы x

Шаг	**Флаг**	**Последовательность букв**	***d***	***l***	**Кодовая последовательность**	**Биты**
0	0	A	-	0	0 bin(A)	9
1	0	_	-	0	0 bin(_)	9
2	0	P	-	0	0 bin(P)	9
3	0	E	-	0	0 bin(E)	9
4	0	N	-	0	0 bin(N)	9
5	1	N	0(5)	1	1 000 0	5
6	0	Y	-	0	0 bin(Y)	9
7	1	_	5(7)	1	1 0101 0	6
8	0	S	-	0	0 bin(S)	9
9	1	A	8(9)	1	1 01000 0	7
10	0	V	-	0	0 bin(V)	9
11	1	E	7(11)	1	1 00111 0	7
12	0	D	-	0	0 bin(D)	9

13	1	_	5(13)	1	1 00101 0	7
14	0	I	-	0	0 bin(I)	9
15	1	S	6(15)	1	1 00110 0	7
16	1	_	2(16)	1	1 00010 0	7
17	1	A_PENNY_	16(17)	8	1 010000 1110000	14
18	1	E	4(25)	1	1 000100 0	8
19	1	A	8(26)	1	1 001000 0	8
20	0	R	-	0	0 bin(R)	9
21	1	N	5(28)	1	1 000101 0	8
22	1	ED	17(29)	2	1 010001 100	10
23	0	.	-	0	0 bin(.)	9
					Всего:	**202 бит**

5) Алгоритм Зива-Лемпела-78 (Зива-Лемпела-Велча)

Шаг	Словарь	Номер слова	Кодовые символы	Затраты (бит)
0	esc	-	-	-
1	A	0	bin(A)	0+8
2	_	0	bin(_)	0+8
3	P	0	0bin(P)	1+8=9
4	E	0	00bin(E)	2+8=10
5	N	0	00bin(N)	2+8=10
6	NY	5	101	3
7	Y	0	000bin(Y)	3+8=11
8	_S	2	010	3
9	S	0	000bin(S)	3+8=11
10	AV	1	0001	4
11	V	0	0000bin(V)	12
12	ED	4	0100	4
13	D	0	0000bin(D)	12
14	_I	2	0010	4
15	I	0	0000bin(I)	12
16	S_	9	1001	4
17	_A	2	0010	4
18	A_	1	00001	5
19	_P	2	00010	5
20	PE	3	00011	5
21	EN	4	00100	5
22	NN	5	00101	5
23	NY_	6	00110	5
24	_E	2	00010	5
25	EA	4	00100	5
26	AR	1	00001	5
27	R	0	00000bin(R)	13
28	NE	5	00101	5
29	ED.	12	01100	5
30	.	0	00000bin(.)	13
			Всего	**210 бит**

6) Алгоритмы PPM

Логика не ясна☹

7) Алгоритмы на основе преобразования Барроуза-Уиллера

Рассмотрим все циклические сдвиги:

1	.A_PENNY_SAVED_IS_A_PENNY_EARNED
2	ARNED.A_PENNY_SAVED_IS_A_PENNY_E
3	AVED_IS_A_PENNY_EARNED.A_PENNY_S
4	A_PENNY_EARNED.A_PENNY_SAVED_IS_
5	A_PENNY_SAVED_IS_A_PENNY_EARNED.
6	D.A_PENNY_SAVED_IS_A_PENNY_EARNE
7	D_IS_A_PENNY_EARNED.A_PENNY_SAVE
8	EARNED.A_PENNY_SAVED_IS_A_PENNY_
9	ED.A_PENNY_SAVED_IS_A_PENNY_EARN
10	ED_IS_A_PENNY_EARNED.A_PENNY_SAV
11	ENNY_EARNED.A_PENNY_SAVED_IS_A_P
12	ENNY_SAVED_IS_A_PENNY_EARNED.A_P
13	IS_A_PENNY_EARNED.A_PENNY_SAVED_
14	NED.A_PENNY_SAVED_IS_A_PENNY_EAR
15	NNY_EARNED.A_PENNY_SAVED_IS_A_PE
16	NNY_SAVED_IS_A_PENNY_EARNED.A_PE
17	NY_EARNED.A_PENNY_SAVED_IS_A_PEN
18	NY_SAVED_IS_A_PENNY_EARNED.A_PEN
19	PENNY_EARNED.A_PENNY_SAVED_IS_A_
20	PENNY_SAVED_IS_A_PENNY_EARNED.A_
21	RNED.A_PENNY_SAVED_IS_A_PENNY_EA
22	SAVED_IS_A_PENNY_EARNED.A_PENNY_
23	S_A_PENNY_EARNED.A_PENNY_SAVED_I

24	VED_IS_A_PENNY_EARNED.A_PENNY_SA
25	Y_EARNED.A_PENNY_SAVED_IS_A_PENN
26	Y_SAVED_IS_A_PENNY_EARNED.A_PENN
27	_A_PENNY_EARNED.A_PENNY_SAVED_IS
28	_EARNED.A_PENNY_SAVED_IS_A_PENNY
29	_IS_A_PENNY_EARNED.A_PENNY_SAVED
30	_PENNY_EARNED.A_PENNY_SAVED_IS_A
31	_PENNY_SAVED_IS_A_PENNY_EARNED.A
32	_SAVED_IS_A_PENNY_EARNED.A_PENNY

- Последний столбец таблицы:

  DES_.EE_NVPP_REENN__A_IANNSYDAAY

- Номер исходной последовательности:
  5

1) Метод «Стопка книг»

Получаем последовательность:

esc esc esc esc esc 3 0 2 esc esc esc 0 3 esc 5 0 5 0 3 0 esc 1 esc 2 3 0 9 esc 11 4 0 2

Воспользуемся нумерационным кодированием:

Буква	Количество	Биты
Esc	12	Log(32)
0	7	Log(21)
1	1	Log(14)
2	3	Log(13)
3	4	Log(10)
4	1	Log(6)
5	2	Log(4)
6	0	Log(4)
7	0	Log(4)
8	0	Log(4)
9	1	Log(4)
10	0	Log(3)
11	1	Log(2)
	Всего:	**36 бит**

Для передачи букв 8 * 12 = 96 бит

Для передачи номера: $\left\lceil \log\left(\frac{32!}{12!7!3!4!2!}\right)\right\rceil = 69$ бит

Всего: l = 36 + 96 + 69 + 6 (передача индекса) = **207 бит**

2) Метод кодирования расстояний
Последовательность для передачи:

DES_.EE_NVPP_REENN__A_IANNSYDAAY

На передачу числа различных символов: [log(32)] = 5 бит
На значения символов: 12 * 8 = 96 бит
На позиции символов: $log\left[\binom{32}{12}\right] = 28$ бит

i	Последовательность	$y_i(y_{i,\,max})$
0	|DES_.???NVP??R??????A?I????Y?|???	17 (20)
1	D|ES_.?|??NVP??R??????A?I????YD???	1 (19)
2	DE|S_.E??NVP??R??????A?I????|YD???	15 (18)
3	DES|_.E??|NVP??R??????A?I???SYD???	2 (17)
4	DES_|.E?_NVP??R??????A?I???SYD???	0
5	DES_.|E?_NVP??R?|?????A?I???SYD???	3 (15)
6	DES_.EE|_NVP??|RE?????A?I???SYD???	2 (14)
7	DES_.EE_|NVP?_RE??|???A?I???SYD???	3 (13)
8	DES_.EE_N|VP?_RE?N???A?I???SYD???	0
9	DES_.EE_NV|P?_RE?N???A?I???SYD???	0
10	DES_.EE_NVPP|_RE?N??|?A?I???SYD???	3 (11)
11	DES_.EE_NVPP_|RE?N?_?A?I???SYD???	0
12	DES_.EE_NVPP_R|E?N?_?A?I???SYD???	0
13	DES_.EE_NVPP_REE|N?_?A?I??|?SYD???	4 (8)
14	DES_.EE_NVPP_REENN|_?A?|I?N?SYD???	1 (6)
15	DES_.EE_NVPP_REENN__|A_I?|N?SYD???	1 (5)
16	DES_.EE_NVPP_REENN__A|_IAN?SYD???	0
17	DES_.EE_NVPP_REENN__A_|IAN?SYD???	0
18	DES_.EE_NVPP_REENN__A_I|AN?SYD?|??	2 (4)
19	DES_.EE_NVPP_REENN__A_IA|N?SYDA??	0
20	DES_.EE_NVPP_REENN__A_IANN|SYDA??	0
21	DES_.EE_NVPP_REENN__A_IANNS|YDA??|	2 (2)
	DES_.EE_NVPP_REENN__A_IANNSYDAAY	

Кодер расстояний сформировал последовательность:
y = (17, 1, 15, 2, 0, 3, 2, 3, 0, 0, 3, 0, 0, 4, 1, 1, 0, 0, 2, 0, 0, 2)
Для передачи данной последовательности будем использовать арифметическое кодирование, приписав буквам на шаге i вероятности

$$\mathrm{p}_i(a) = \begin{cases} \frac{1}{2}, a = 0 \\ \frac{1}{2} * \frac{1}{y_{i,max}}, a \neq 0 \end{cases}$$

$$21 + \left[-\log\left(\frac{1}{20 * 19 * 18 * 17 * 15 * 14 * 13 * 11 * 8 * 6 * 5 * 4 * 2}\right)\right] = 64 \text{ бита}$$

Всего будет затрачено l = 5 + 96 + 28 + 64 + 6 (передача индекса в таблице сдвигов) = **199 бит**

Сравнение алгоритмов

Алгоритм	**Затраты, бит**
Двухпроходное кодирование с использованием кода Хаффмена	225
Двухпроходное кодирование с использованием регулярного кода Хаффмена	207
Адаптивное кодирование с применением арифметического кодирования	**202**
Нумерационное кодирование	**191**
Алгоритм Зива-Лемпела-77 (метод скользящего словаря)	**202**
Алгоритм Зива-Лемпела-78 (Зива-Лемпела-Велча)	210
Алгоритмы PPM	☹
Барроуз-Уиллер + «стопка книг»	207
Барроуз-Уиллер + кодирование расстояний	**199**
Стандартный архиватор (zip)	168*8=1344

Вывод:

В результате работы оказалось, что наилучшими алгоритмами кодирования для данной последовательности являются нумерационное кодирование (191 бит), сжатие с использованием преобразования Барроуза-Уиллера с использованием метода кодирование расстояний, также видно, что стандартный архиватор сравнительно хуже наших алгоритмов.