# 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)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **\_** | 6 |  | | | |  |
| **E** | 5 | 10 |  | 18 |  | 32 |
| **N** | 5 |  |  |
| **A** | 4 |  | 8 |  |
| **P** | 2 | 4 |  |
| **Y** | 2 |  |
| **S** | 2 | 4 |  | 8 | 14 |  |
| **D** | 2 |  |
| **V** | 1 | 2 | 4 |  |
| **I** | 1 |  |
| **R** | 1 | 2 |  |
| **.** | 1 |  |

1

0

* Объем алфавита |X| = 256
* Кодовое слово c(x) = c1(x) + c2(x), len(c1(x)) = l1, len(c2(x)) = l2

l2 = 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 |

l1 = 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 ni | Диапазон значений ni | 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:

* Количество бит для передачи сообщения, используя регулярное дерево Хаффмана

l = 106 + 15 + 86 = **207 бит**

1. **Адаптивное кодирование с применением арифметического кодирования**

Воспользуемся алгоритмом D

Рассмотрим оценку

Объем алфавита M = 256

Mn -- число различных букв в последовательности длины n.

tn(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

\*10-61

1. **Нумерационное кодирование**

Кодовое слово = номер композиции tn(**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
* Длина первой части кодового слова
* Общая длина

l(x) = l1(x) + l2(x) = 105 + 86 = **191 бит**

1. **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** |

1. **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** |

1. **Алгоритмы PPM**

**Логика не ясна☹**

1. **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:

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

1. 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:

|  |  |  |
| --- | --- | --- |
| i | Последовательность | yi(yi, 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

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.