Код Хаффмана

Гусев Илья, Булгаков Илья

Московский физико-технический институт

Москва, 2018

Содержание

Коды

Задача

- Кодируем последовательность символов битам:
  - Без потерь
  - Однозначно:
    - Блочный код (фиксированная длина)
    - Префиксный код
    - Постфиксный код
  - Максимально коротко

ASCII TABLE

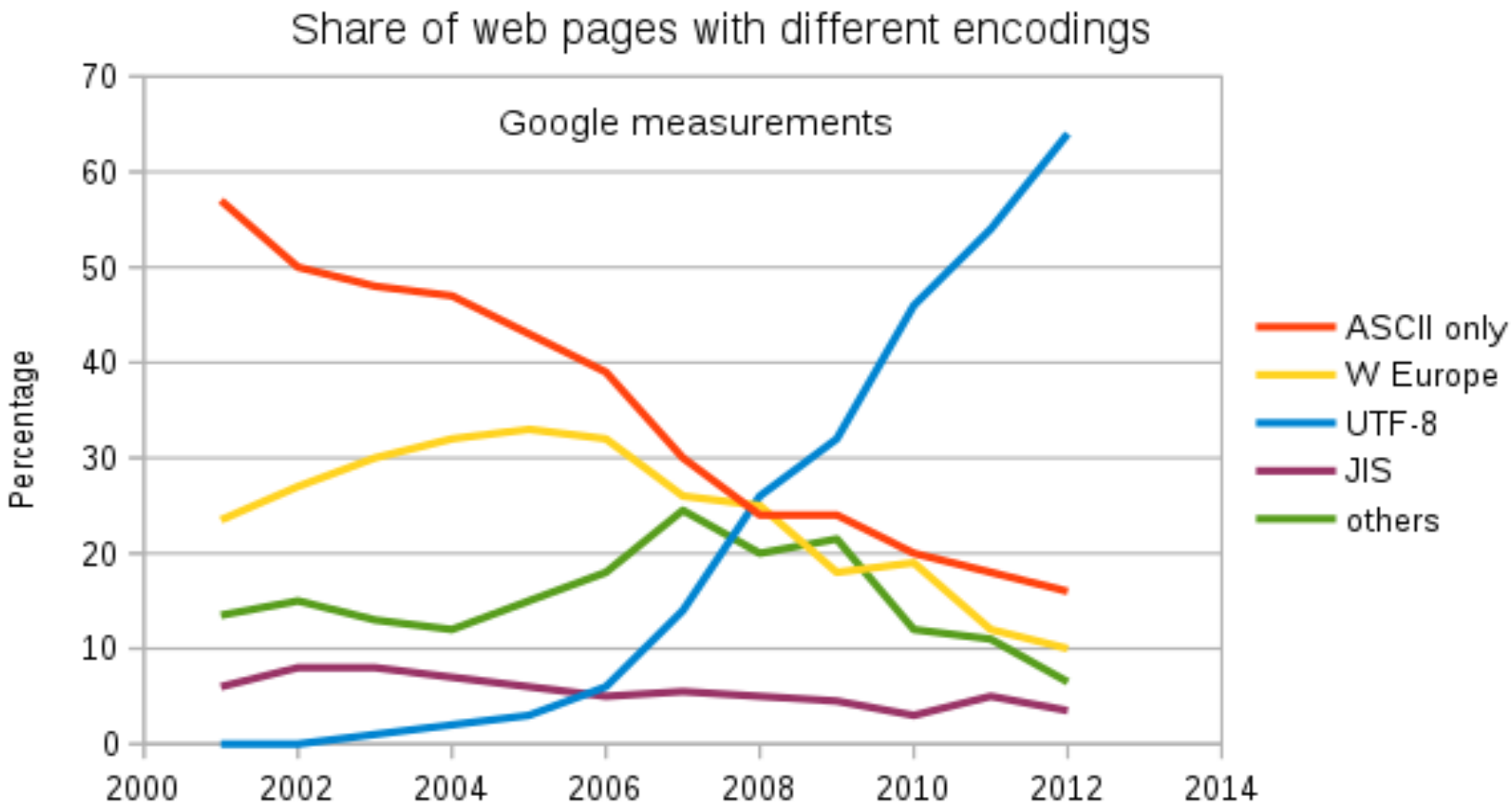
Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	*[NULL]*
1	1	1	1	*[START OF HEADING]*
2	2	10	2	*[START OF TEXT]*
3	3	11	3	*[END OF TEXT]*
4	4	100	4	*[END OF TRANSMISSION]*
5	5	101	5	*[ENQUIRY]*
6	6	110	6	*[ACKNOWLEDGE]*
7	7	111	7	*[BELL]*
8	8	1000	10	*[BACKSPACE]*
9	9	1001	11	*[HORIZONTAL TAB]*
10	A	1010	12	*[LINE FEED]*
11	B	1011	13	*[VERTICAL TAB]*
12	C	1100	14	*[FORM FEED]*
13	D	1101	15	*[CARRIAGE RETURN]*
14	E	1110	16	*[SHIFT OUT]*
15	F	1111	17	*[SHIFT IN]*
16	10	10000	20	*[DATA LINK ESCAPE]*
17	11	10001	21	*[DEVICE CONTROL 1]*
18	12	10010	22	*[DEVICE CONTROL 2]*
19	13	10011	23	*[DEVICE CONTROL 3]*
20	14	10100	24	*[DEVICE CONTROL 4]*
21	15	10101	25	*[NEGATIVE ACKNOWLEDGE]*
22	16	10110	26	*[SYNCHRONOUS IDLE]*
23	17	10111	27	*[ENG OF TRANS. BLOCK]*
24	18	11000	30	*[CANCEL]*
25	19	11001	31	*[END OF MEDIUM]*
26	1A	11010	32	*[SUBSTITUTE]*
27	1B	11011	33	*[ESCAPE]*
28	1C	11100	34	*[FILE SEPARATOR]*
29	1D	11101	35	*[GROUP SEPARATOR]*
30	1E	11110	36	*[RECORD SEPARATOR]*
31	1F	11111	37	*[UNIT SEPARATOR]*
32	20	100000	40	*[SPACE]*
33	21	100001	41	!
34	22	100010	42	"
35	23	100011	43	#
36	24	100100	44	$
37	25	100101	45	%
38	26	100110	46	&
39	27	100111	47	'
40	28	101000	50	(
41	29	101001	51	)
42	2A	101010	52	*
43	2B	101011	53	+
44	2C	101100	54	,
45	2D	101101	55	-
46	2E	101110	56	.
47	2F	101111	57	/
48	30	110000	60	**0**
49	31	110001	61	**1**
50	32	110010	62	**2**
51	33	110011	63	**3**
52	34	110100	64	**4**
53	35	110101	65	**5**
54	36	110110	66	**6**
55	37	110111	67	**7**
56	38	111000	70	**8**
57	39	111001	71	**9**
58	3A	111010	72	:
59	3B	111011	73	;
60	3C	111100	74	<
61	3D	111101	75	=
62	3E	111110	76	>
63	3F	111111	77	?
64	40	1000000	100	@
65	41	1000001	101	**A**
66	42	1000010	102	**B**
67	43	1000011	103	**C**
68	44	1000100	104	**D**
69	45	1000101	105	**E**
70	46	1000110	106	**F**
71	47	1000111	107	**G**
72	48	1001000	110	**H**
73	49	1001001	111	**I**
74	4A	1001010	112	**J**
75	4B	1001011	113	**K**
76	4C	1001100	114	**L**
77	4D	1001101	115	**M**
78	4E	1001110	116	**N**
79	4F	1001111	117	**O**
80	50	1010000	120	**P**
81	51	1010001	121	**Q**
82	52	1010010	122	**R**
83	53	1010011	123	**S**
84	54	1010100	124	**T**
85	55	1010101	125	**U**
86	56	1010110	126	**V**
87	57	1010111	127	**W**
88	58	1011000	130	**X**
89	59	1011001	131	**Y**
90	5A	1011010	132	**Z**
91	5B	1011011	133	[
92	5C	1011100	134	\
93	5D	1011101	135	]
94	5E	1011110	136	^
95	5F	1011111	137	_
96	60	1100000	140	`
97	61	1100001	141	**a**
98	62	1100010	142	**b**
99	63	1100011	143	**c**
100	64	1100100	144	**d**
101	65	1100101	145	**e**
102	66	1100110	146	**f**
103	67	1100111	147	**g**
104	68	1101000	150	**h**
105	69	1101001	151	**i**
106	6A	1101010	152	**j**
107	6B	1101011	153	**k**
108	6C	1101100	154	**l**
109	6D	1101101	155	**m**
110	6E	1101110	156	**n**
111	6F	1101111	157	**o**
112	70	1110000	160	**p**
113	71	1110001	161	**q**
114	72	1110010	162	**r**
115	73	1110011	163	**s**
116	74	1110100	164	**t**
117	75	1110101	165	**u**
118	76	1110110	166	**v**
119	77	1110111	167	**w**
120	78	1111000	170	**x**
121	79	1111001	171	**y**
122	7A	1111010	172	**z**
123	7B	1111011	173	{
124	7C	1111100	174	|
125	7D	1111101	175	}
126	7E	1111110	176	~
127	7F	1111111	177	*[DEL]*

UTF-8

- Фиксированный код переменной длины
- Все символы всех живых и мёртвых языков, спец. символы, эмодзи
- ASCII-совместима

Bytes	Bits	First cp	Last cp	Byte 1	Byte 2	Byte 3	Byte 4
1	7	U+0000	U+007F	0xxxxxxx			
2	11	U+0080	U+07FF	110xxxxx	10xxxxxx		
3	16	U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx	
4	21	U+10000	U+10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

UTF-8


Share of web pages with different encodings
Google measurements
Percentage
70
60
50
40
30
20
10
0
2000
2002
2004
2006
2008
2010
2012
2014
ASCII only
W Europe
UTF-8
JIS
others

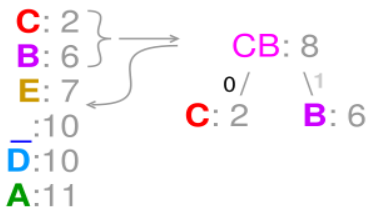
Статический код Хаффмана

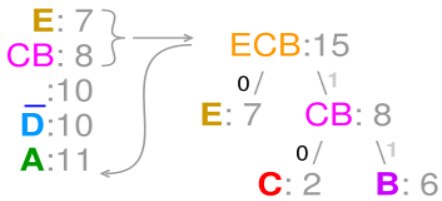
- Переменная длина, префиксный код
- Статичность: заранее считаем частоты символов
- Реже встречается символ $\Rightarrow$ больше тратим бит
- Используем бинарное дерево
- Шаг построения:
  - Две наименее частотных вершины подсоединяем к общему родителю
  - На ребре к меньшей пишем 0, к большей пишем 1
- Путь до символа из корня - его код

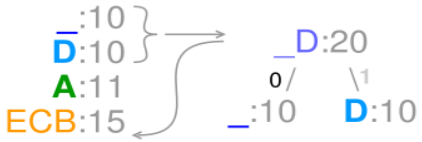
Статический код Хаффмана

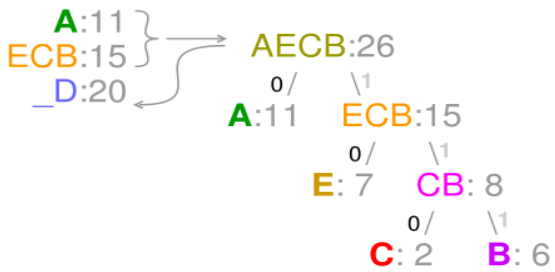
Иллюстрация

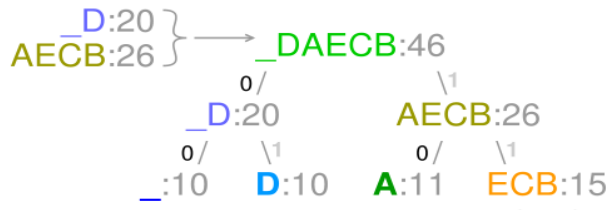
1. "A_DEAD_DAD_CEDED_A_BAD_BABE_A_BEADED_ABACA_BED"

2. 

3. 

4. 

5. 

6. 

7. _: 00
D: 01
A: 10
E: 110
C: 1110
B: 1111

8. "1000011101001000110010011101100111001001000111110010011111101111100100011111101001110010010111111011101000111111001"

Статический код Хаффмана

Особенности

- Код любого символа не является префиксом любого другого кода
- Нужно дописывать таблицу частот в сжатый файл
- 2 прохода по сообщению

Адаптивный код Хаффмана

Адаптивность

- Кодируем на ходу
- Один проход по сообщению
- Не нужно запоминать таблицу символов

Адаптивный код Хаффмана

Реализация

- Заведём специальный 0-символ с всегда нулевой частотой
- Встречаем новый символ $\rightarrow$ пишем 0-символ и сам новый символ
- Встречаем старый символ $\rightarrow$ пишем код символа по текущему дереву
- Перестраиваем дерево

Адаптивный код Хаффмана

Варианты перестроения дерева

- Алгоритм Фоллера, Галлагера, Кнута (ФГК)
- Алгоритм Виттера

Запись битов

```
std::vector<char> data;
int lastBitsCount = 0;

void WriteBit(bool bit)
{
    if(lastBitsCount == 0) {
        data.push_back(0);
    }
    if(bit) {
        data.back() |= 1 << lastBitsCount;
    }
    lastBitsCount = (lastBitsCount + 1) % 8;
}

void WriteByte(char value)
{
    for( int i = 0; i < 8; ++i ) {
        WriteBit((value & (1 << i)) != 0);
    }
}
```

Полезные ссылки I


Вики: Юникод
https://ru.wikipedia.org/wiki/Юникод


Wiki: Huffman coding
https://en.wikipedia.org/wiki/Huffman_coding


Вики: Адаптивный алгоритм Хаффмана
https://bit.ly/2DBDFoA


Викиконспекты: Алгоритм Хаффмана за O(n)
https://bit.ly/2qFhK7B


Compression.ru: Динамическое сжатие методом Хаффмана
https://bit.ly/2Dy4Dxp