Одно из применений сбалансированных бинарных деревьев (как строили на рк ака деревья сжатия) - поиск
Как строить деревья поиска? 2 подхода:

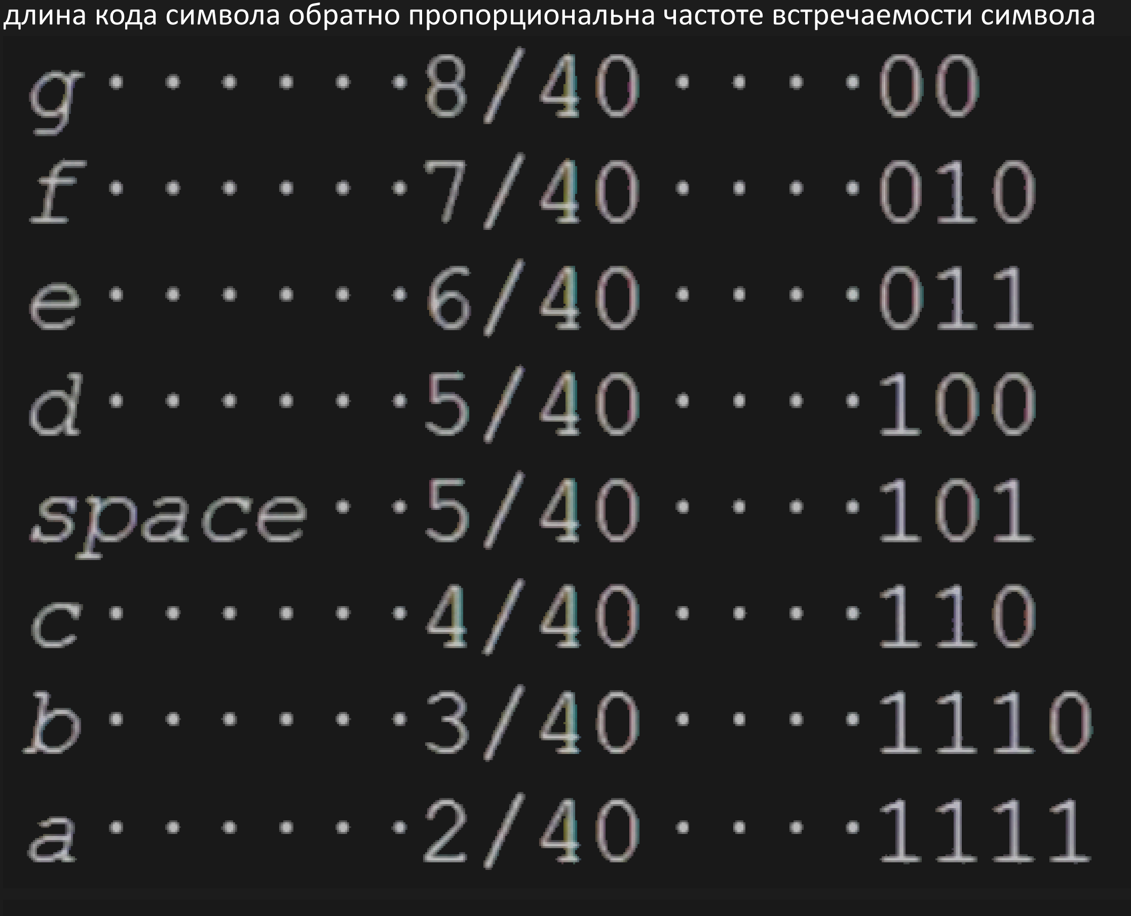
- Самые частые данные - выше. Данные встречаются чаще -> их чаще будут искать
- Более часто искомые данные - выше. По очевидным причинам: чаще ищут - быстрее выдавать

Статистические методы сжатия данных

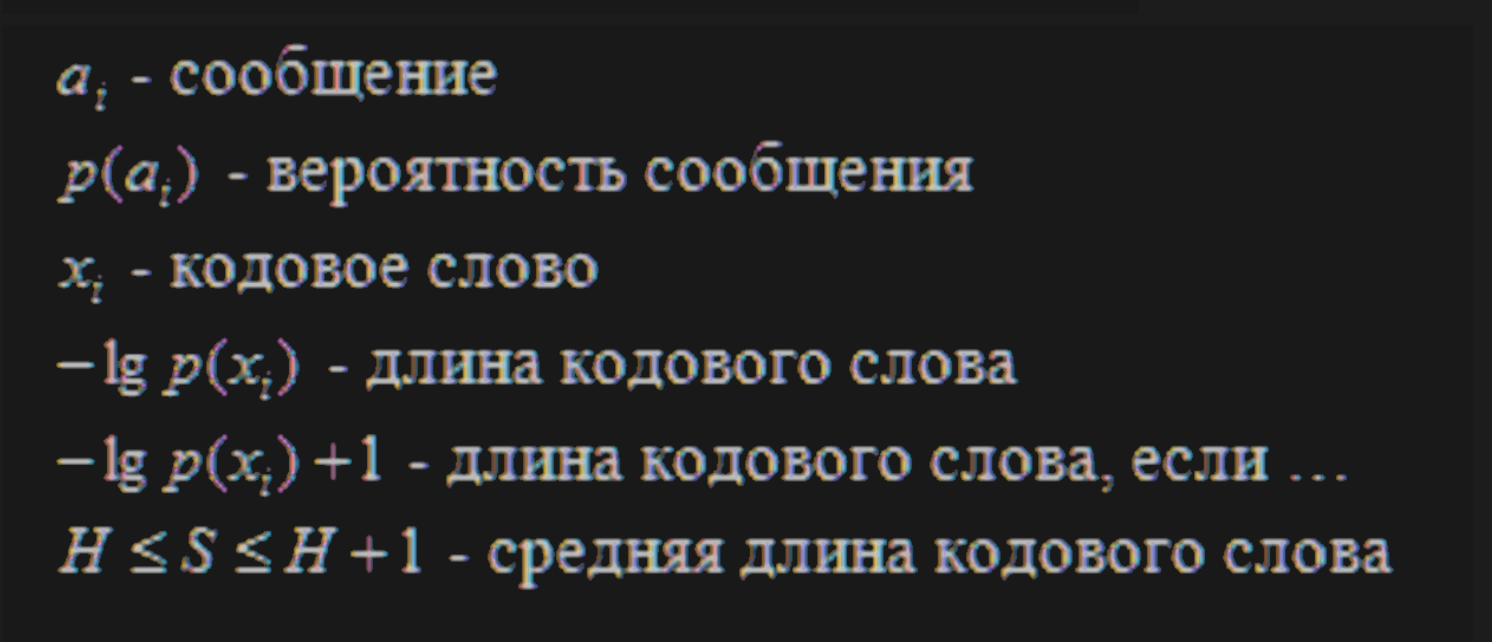
Код	Условия оптимальности	Схема
Шеннона-Фано, 1948	Асимптотическое достиженеие оптимальности при увеличении числа сообщений (больше исходник - лучше сжатие)	var-var, block-var
Хаффмана, 1952	1) Каждое сообщение - уникальное кодовое слово 2) Сжатый текст = конкатенация кодовых слов	var-var, block-var
Универсальные коды, 1975		
Арифметические коды, 1963		

Коды Шеннона-Фано

- длина кода символа обратно пропорциональна частоте встречаемости символа



```
g······8/40····00
f······7/40····010
e······6/40····011
d······5/40····100
space··5/40····101
c······4/40····110
b······3/40····1110
a······2/40····1111
```



a_i - сообщение

$p(a_i)$ - вероятность сообщения

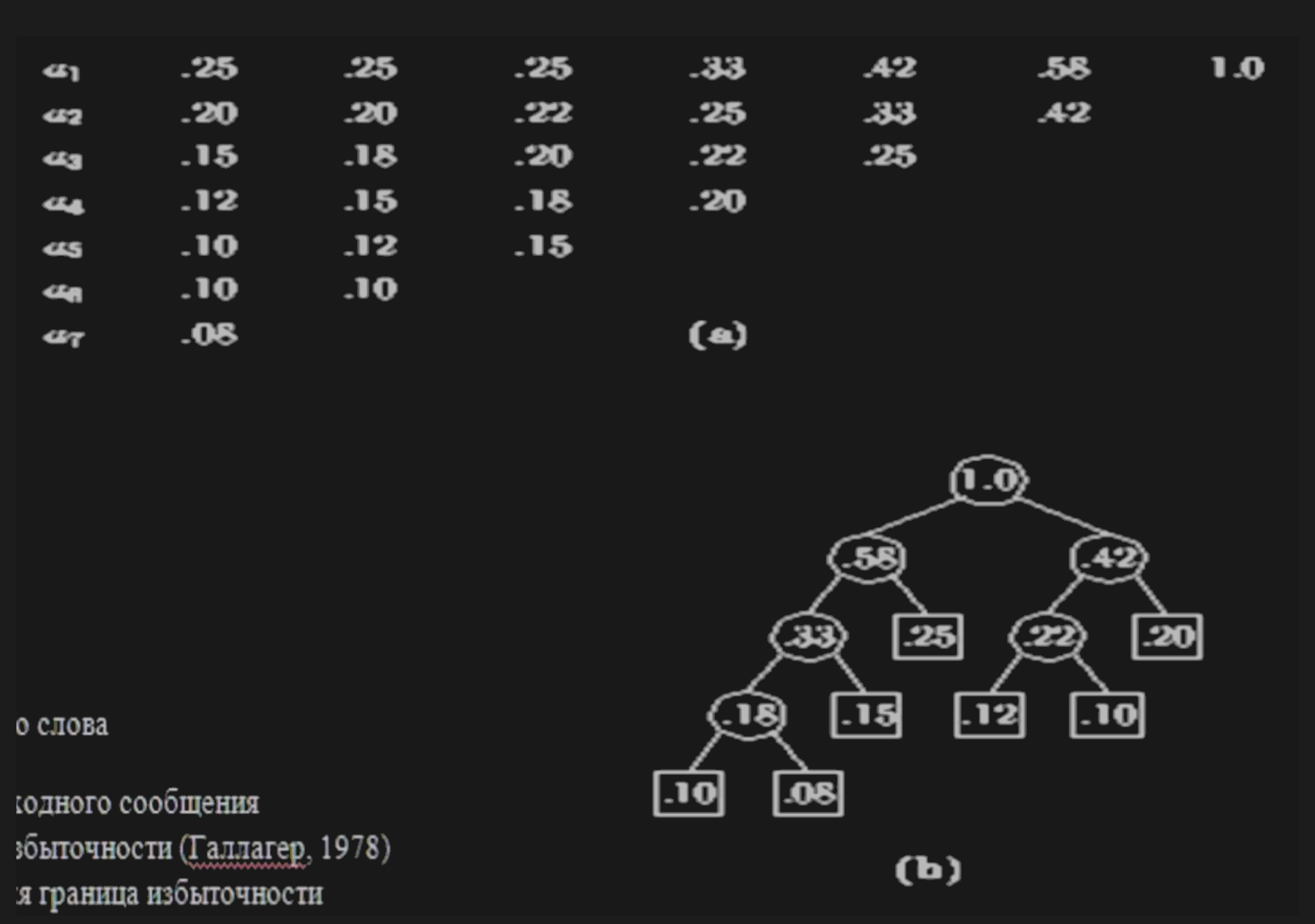
x_i - кодовое слово

$-\lg p(x_i)$ - длина кодового слова

$-\lg p(x_i)+1$ - длина кодового слова, если …

$H \le S \le H+1$ - средняя длина кодового слова

Статические коды Хаффмана



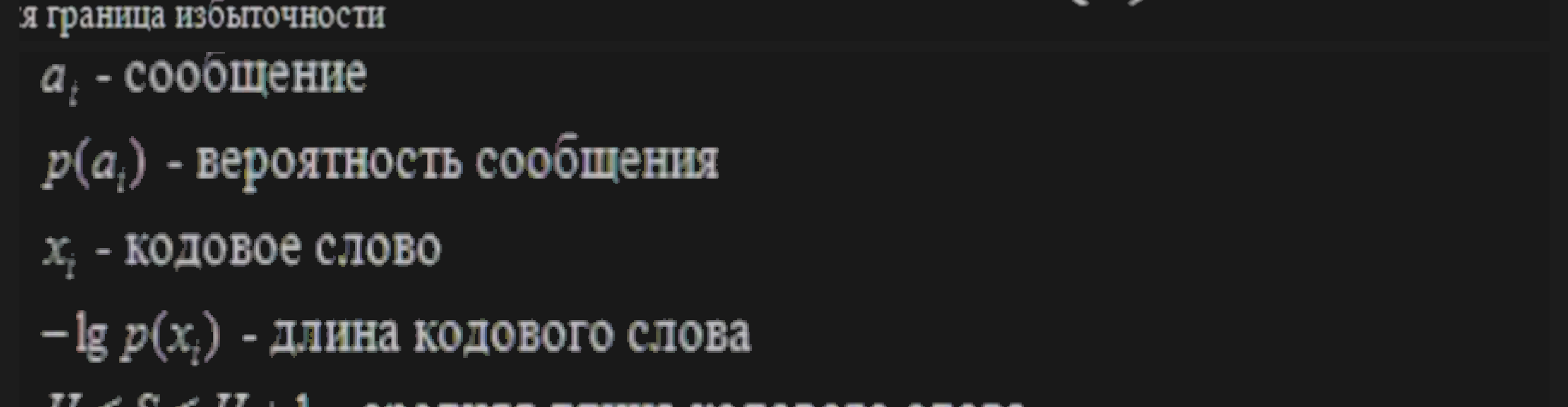
a_1	.25	.25	.25	.33	.42	.58	1.0
a_2	.20	.20	.22	.25	.33	.42	
a_3	.15	.18	.20	.22	.25		
a_4	.12	.15	.18	.20			
a_5	.10	.12	.15				
a_6	.10	.10					
a_7	.08						

(a)

(b)

о слова

кодного сообщения
збыточности (Галлагер, 1978)
я граница избыточности



a_i - сообщение

$p(a_i)$ - вероятность сообщения

x_i - кодовое слово

$-\lg p(x_i)$ - длина кодового слова

$H \le S \le H+1$ - средняя длина кодового слова

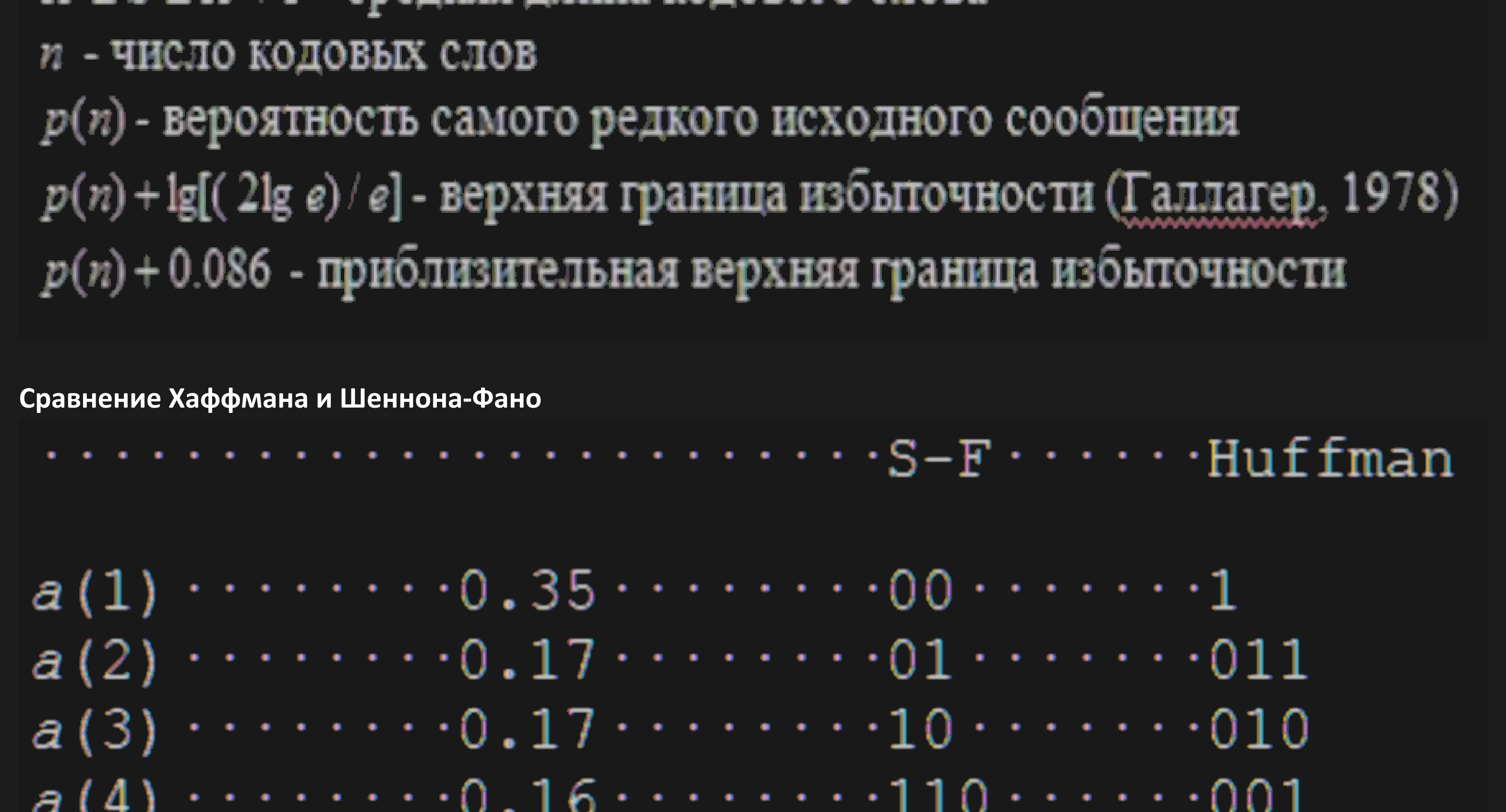
n - число кодовых слов

$p(n)$ - вероятность самого редкого исходного сообщения

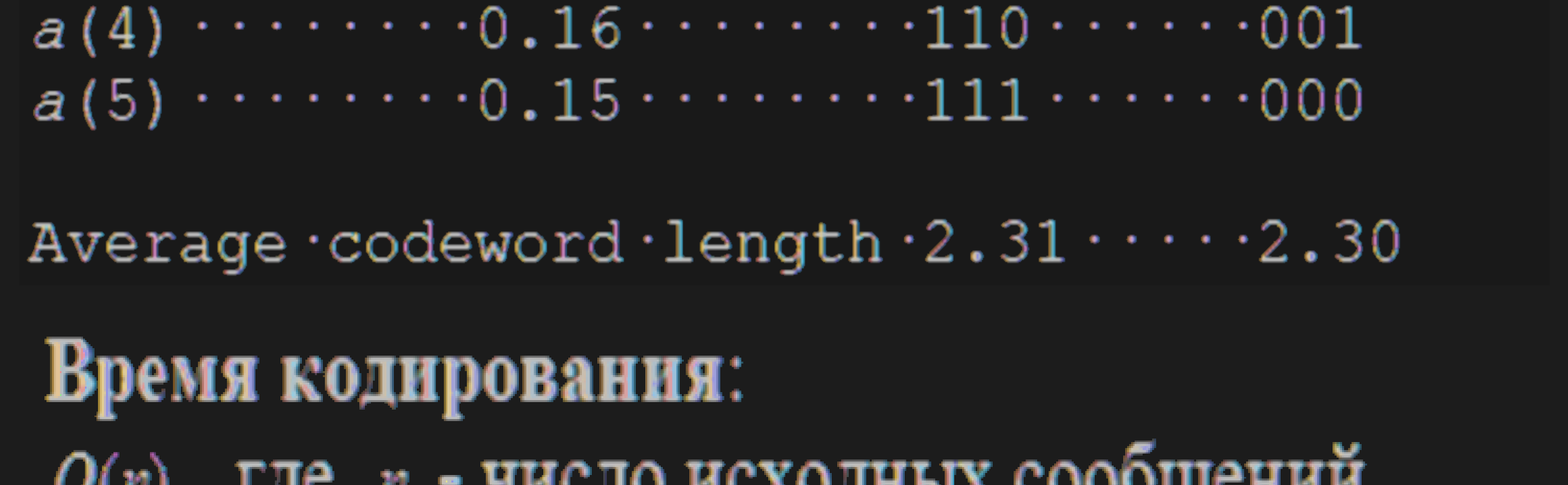
$p(n)+\lg[(2\lg e)/e]$ - верхняя граница избыточности (Галлагер, 1978)

$p(n)+0.086$ - приблизительная верхняя граница избыточности

Сравнение Хаффмана и Шеннона-Фано



```
························S-F······Huffman

a(1)·······0.35·······00·······1
a(2)·······0.17·······01·······011
a(3)·······0.17·······10·······010
a(4)·······0.16·······110······001
a(5)·······0.15·······111······000

Average·codeword·length·2.31····2.30
```



Время кодирования:

$O(n)$, где n - число исходных сообщений

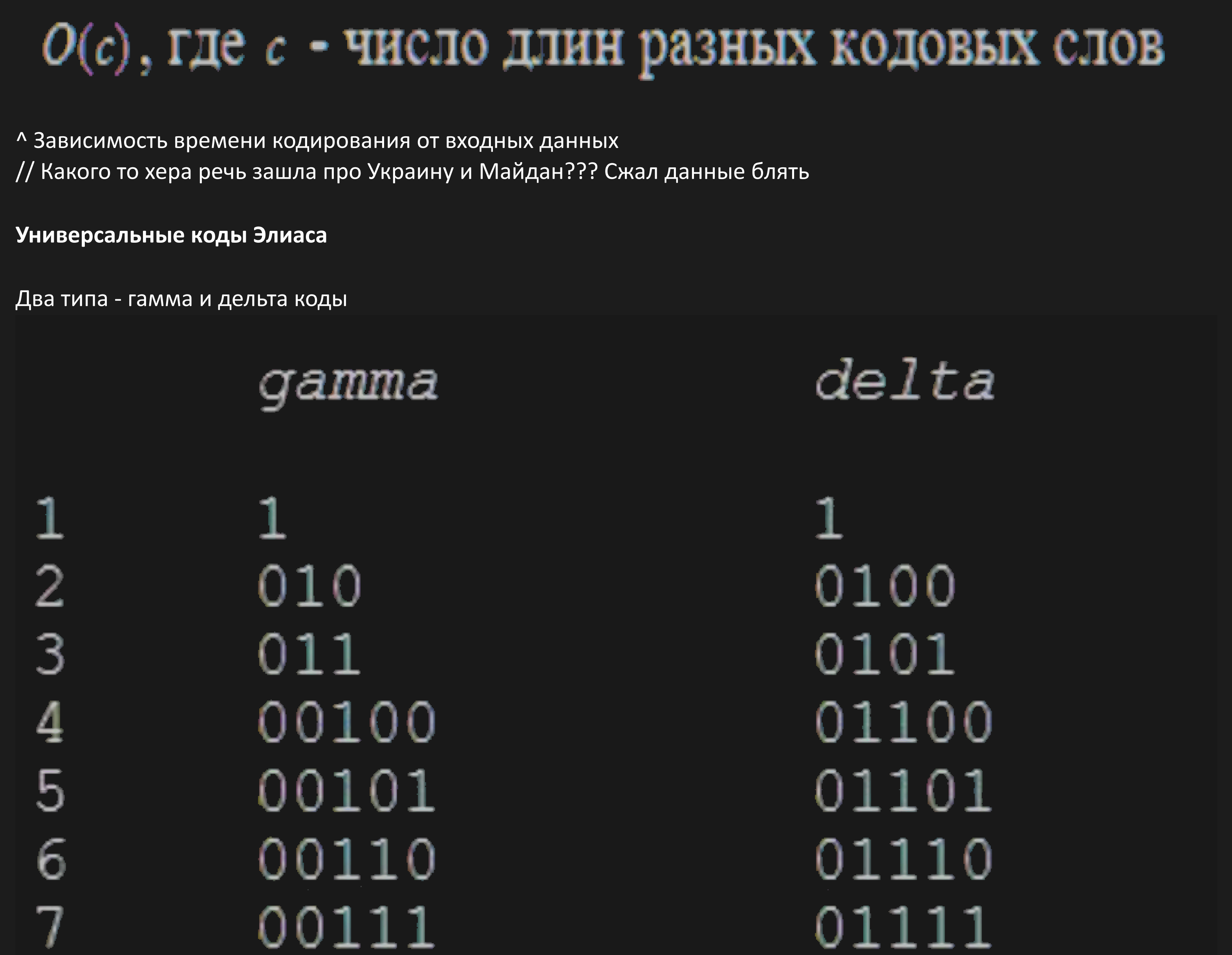
$O(l)$, где l - длина пути в кодовом дереве

$O(c)$, где c - число длин разных кодовых слов

^ Зависимость времени кодирования от входных данных
// Какого то хера речь зашла про Украину и Майдан??? Сжал данные блять

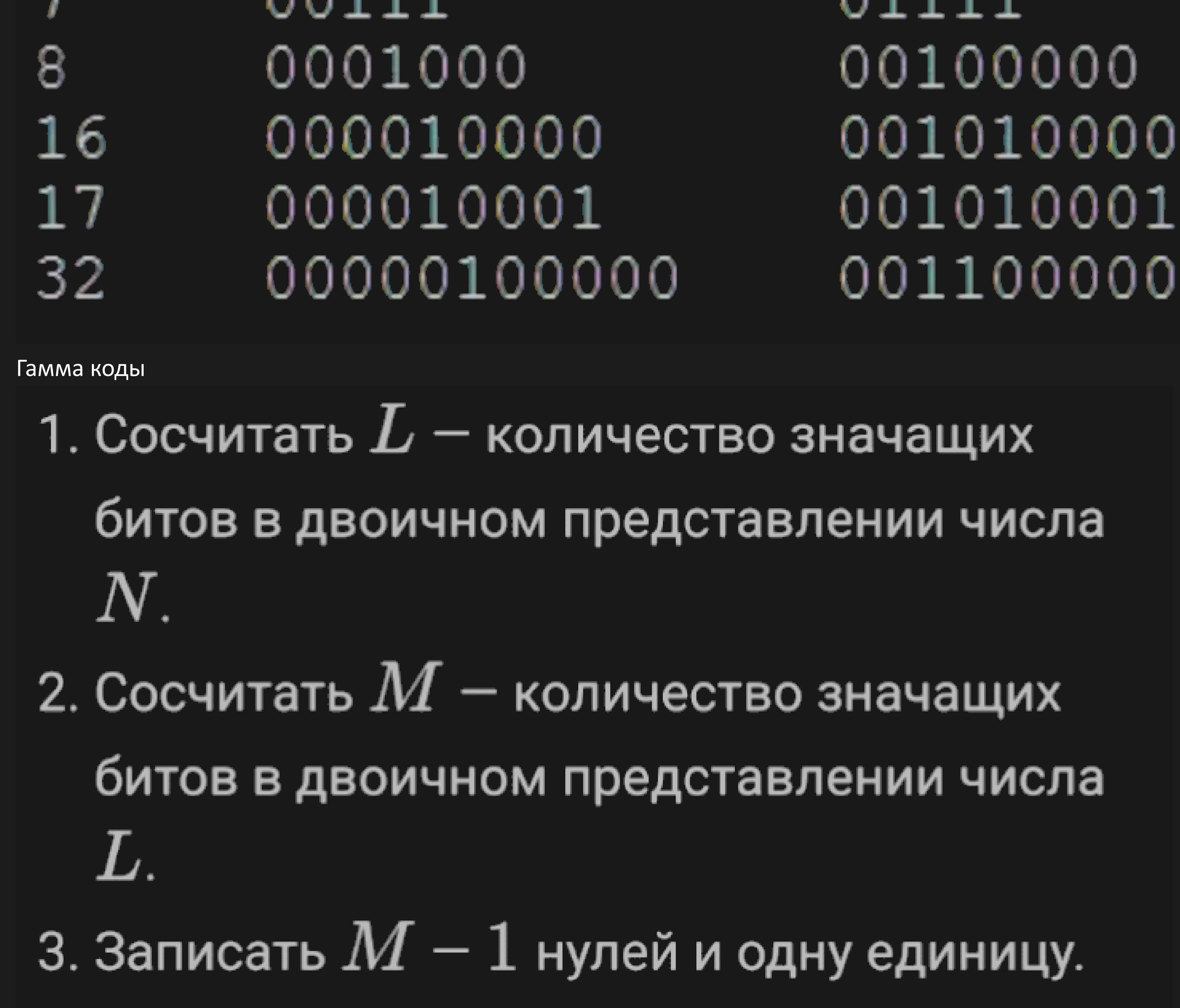
Универсальные коды Элиаса

Два типа - гамма и дельта коды



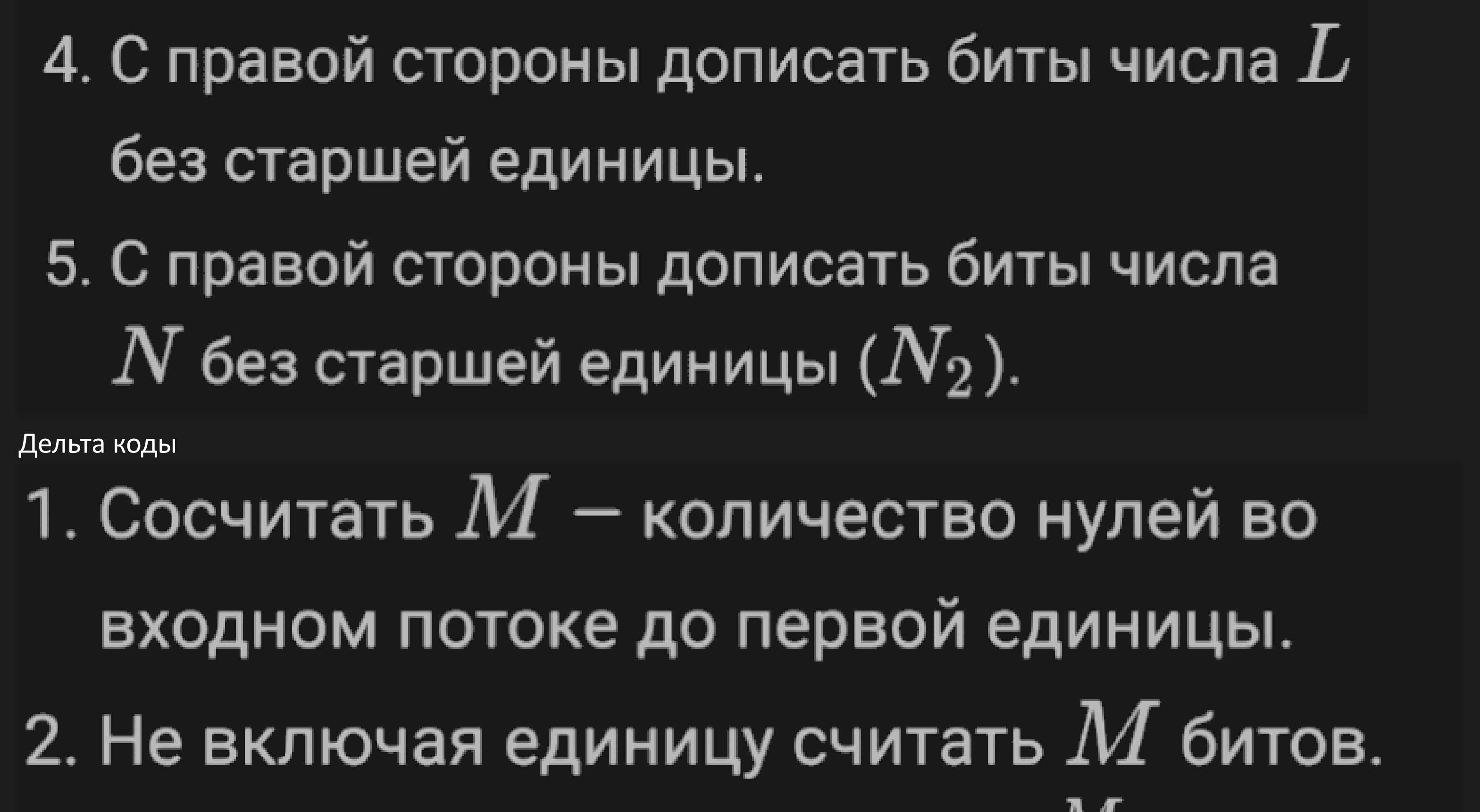
	gamma	delta
1	1	1
2	010	0100
3	011	0101
4	00100	01100
5	00101	01101
6	00110	01110
7	00111	01111
8	0001000	00100000
16	000010000	001010000
17	000010001	001010001
32	00000100000	0011000000

Гамма коды



1. Сосчитать L — количество значащих битов в двоичном представлении числа N.
2. Сосчитать M — количество значащих битов в двоичном представлении числа L.
3. Записать $M-1$ нулей и одну единицу.
4. С правой стороны дописать биты числа L без старшей единицы.
5. С правой стороны дописать биты числа N без старшей единицы (N_2).

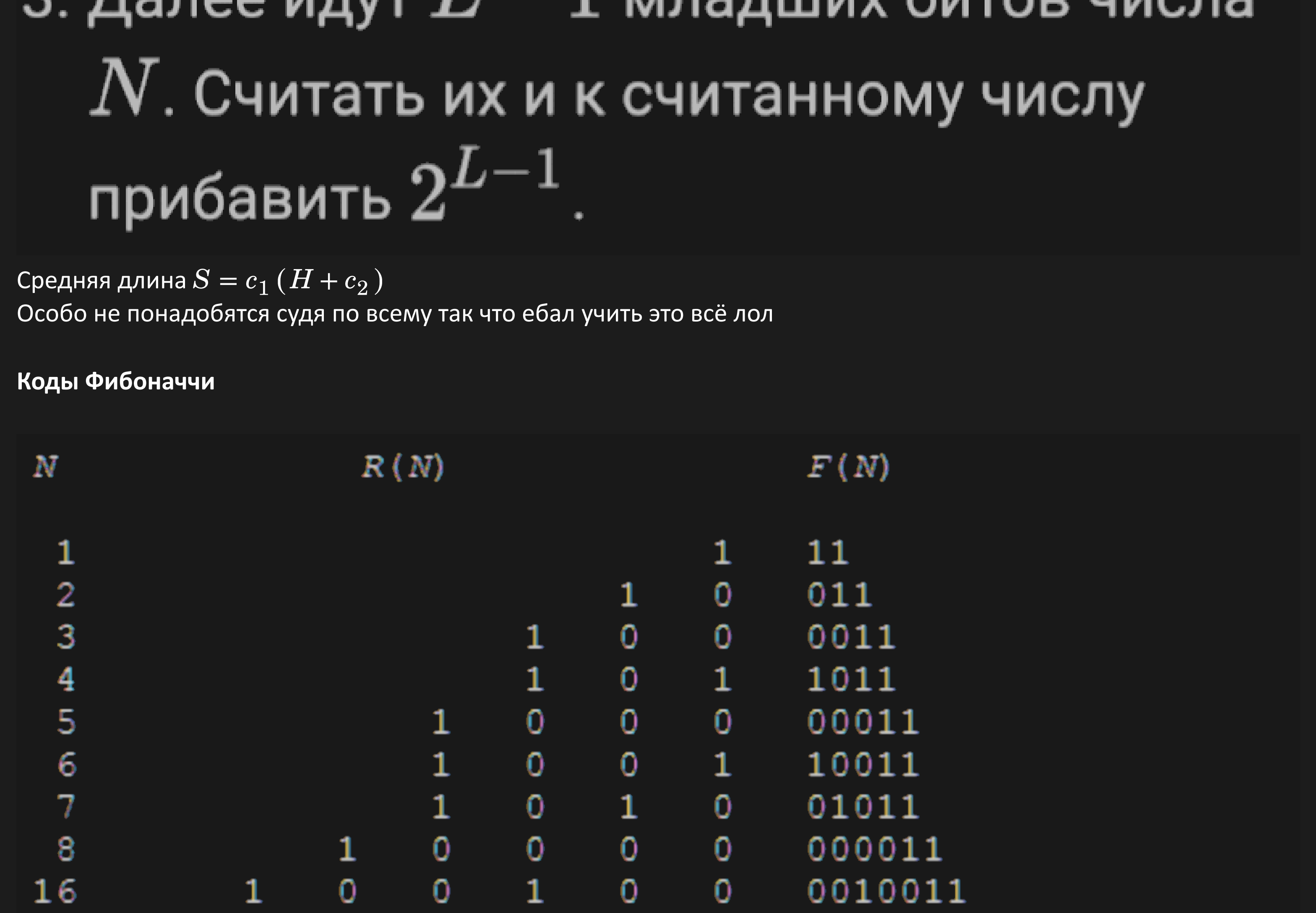
Дельта коды



1. Сосчитать M — количество нулей во входном потоке до первой единицы.
2. Не включая единицу считать M битов. Считанное число в сумме с 2^M дает L.
3. Далее идут $L-1$ младших битов числа N. Считать их и к считанному числу прибавить 2^{L-1}.

Средняя длина $S = c_1\,(H + c_2\,)$
Особо не понадобятся судя по всему так что ебал учить это всё лол

Коды Фибоначчи

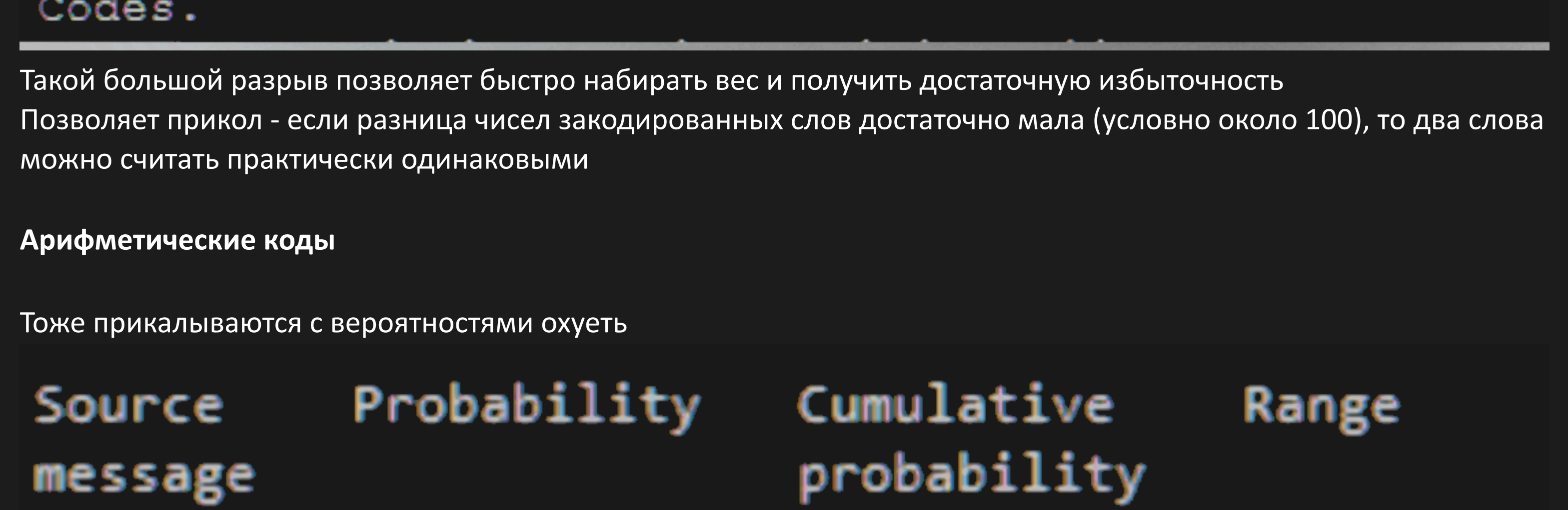


```
N                 R(N)                      F(N)

  1                                   1     11
  2                              1    0     011
  3                         1    0    0     0011
  4                         1    0    1     1011
  5                    1    0    0    0     00011
  6                    1    0    0    1     10011
  7                    1    0    1    0     01011
  8               1    0    0    0    0     000011
 16          1    0    0    1    0    0     0010011
 32     1    0    1    0    1    0    0     00101011

       21   13    8    5    3    2    1

Figure 3.7 -- Fibonacci Representations and Fibonacci
Codes.
```

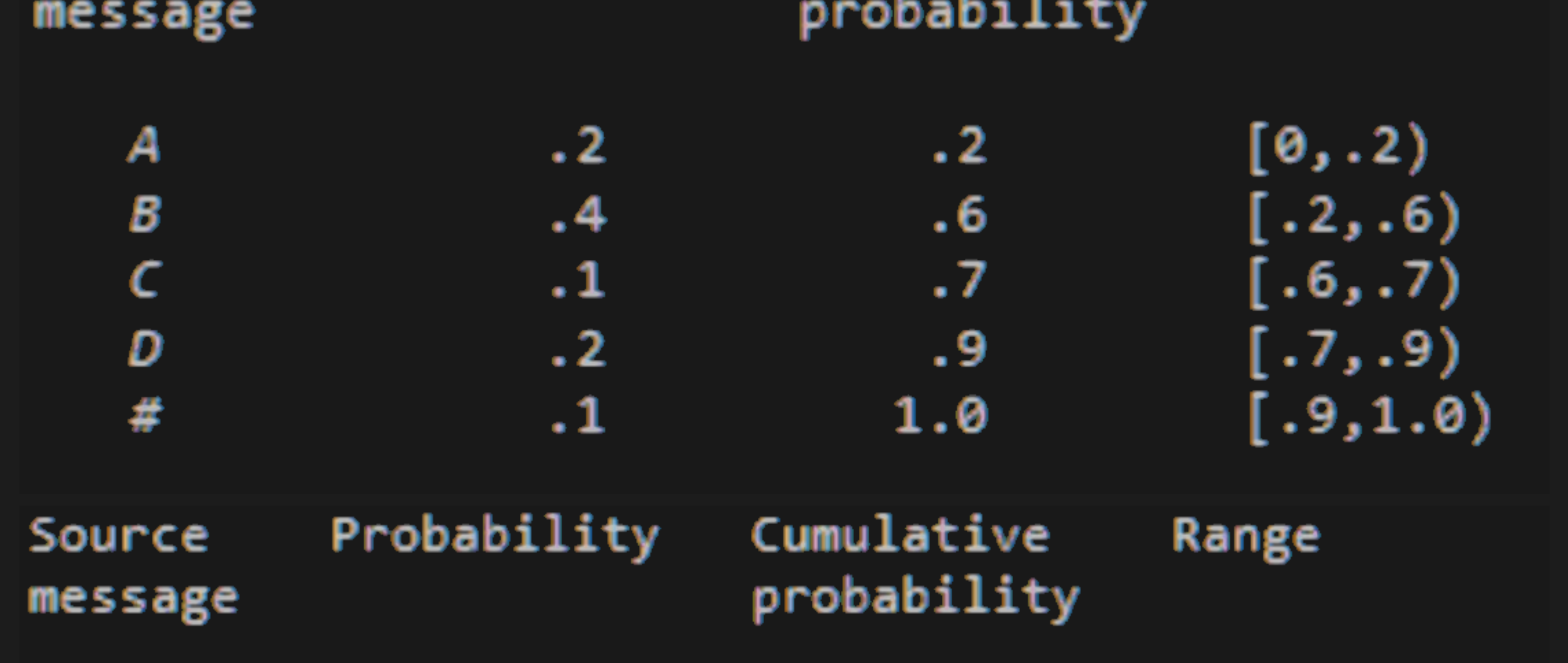
Такой большой разрыв позволяет быстро набирать вес и получить достаточную избыточность
Позволяет прикол - если разница чисел закодированных слов достаточно мала (условно около 100), то два слова можно считать практически одинаковыми

Арифметические коды

Тоже прикалываются с вероятностями охуеть



Source message	Probability	Cumulative probability	Range
A	.2	.2	[0,.2)
B	.4	.6	[.2,.6)
C	.1	.7	[.6,.7)
D	.2	.9	[.7,.9)
#	.1	1.0	[.9,1.0)



Source message	Probability	Cumulative probability	Range
a	.05	.05	[0,.05)
b	.075	.125	[.05,.125)
c	.1	.225	[.125,.225)
d	.125	.35	[.225,.35)
e	.15	.5	[.35,.5)
f	.175	.675	[.5,.675)
g	.2	.875	[.675,.875)
space	.125	1.0	[.875,1.0)

Прикол судя по всему в том, что наиболее вероятные символы вероятнее попадутся. Весь метод - проекция встречаемости на отрезок [0, 1]

Применение всей этой ебалы

Коды Элиаса - позволяют быстро кодировать/декодировать на ходу благодаря отсутствию особой привязки к вероятностям, пусть они и сравнительно неэффективны
В кодах Элиаса таблица соответствия составляется до кодирования, то есть её не надо передавать. Они интересны когда:

- Заранее неизвестна последовательность
- Имеется возможность составить таблицу кодирования до прохода

Арифметические коды - каждому символу можно подобрать несколько способов кодирования. Интересны когда:

- Символы в сообщениях появляются очень неравномерно
  При реализации сжатия на ПЛИСах/МК лучше выбирать более простые алгоритмы