

ЛАБОРАТОРНАЯ РАБОТА № 4

КОДИРОВАНИЕ ИНФОРМАЦИИ

БЛОК ОСНОВНЫХ ЗАДАНИЙ (выполняется в среде ILIAS)

Задание 1. Алгоритм Шеннона-Фано

Задание 2. Кодирование и декодирование информации

В ПРОЦЕССЕ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ ФОРМИРУЮТСЯ СЛЕДУЮЩИЕ УМЕНИЯ:

1. Применять алгоритм кодирования/декодирования сообщений прямым равномерным кодом.
2. Применять алгоритм кодирования/декодирования сообщений ASCII-кодом.
3. Применять алгоритм кодирования/декодирования сообщений кодом Грея.
4. Составлять проверочные выражения в коде Хэмминга.
5. Обнаруживать с помощью синдрома кода Хэмминга одиночную ошибку в переданном сообщении.

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Какие коды называются избыточными?
2. Какие коды называются равномерными?
3. В чем заключается алгоритм кодирования данных кодом Грея?
4. В каком коде большие и малые буквы кодируются одинаково?
5. Для чего нужен синдром в коде Хэмминга?
6. Как по синдрому определяется, в каком разряде была ошибка?
7. Как выполняется сложение по модулю?
8. Как найти вес кодовой комбинации?
9. Как найти минимальное кодовое расстояние?
10. При каком условии код может обнаружить и исправить ошибку?

БЛОК ОСНОВНЫХ ЗАДАНИЙ

ЗАДАНИЕ 1. АЛГОРИТМ ШЕННОНА-ФАНО

Дана последовательность символов: AAABCCCCDEEEFG

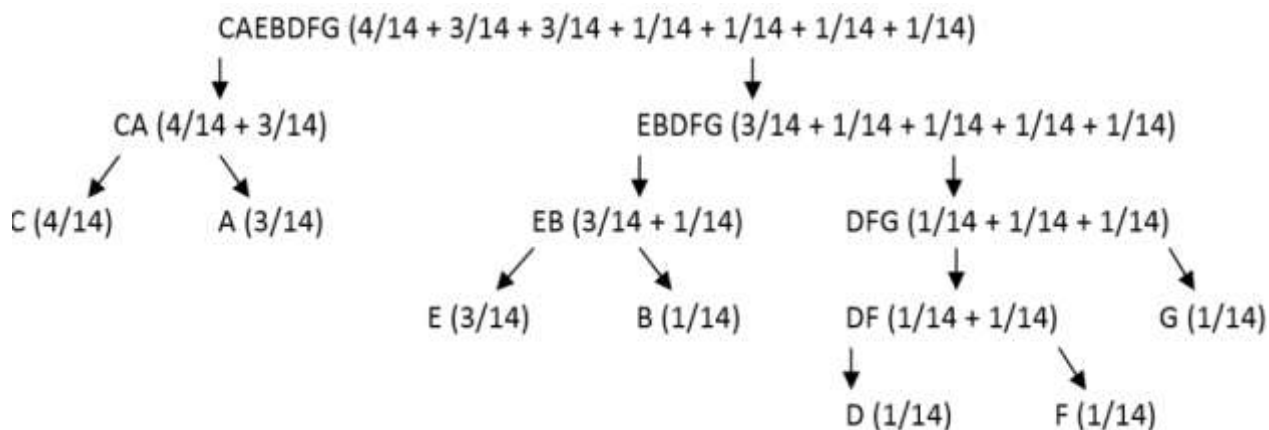
Встречаемость символов: A = 3, B = 1, C = 4, D = 1, E = 3, F = 1, G = 1.

Таблица с вероятностями, отсортированная в порядке уменьшения вероятностей.

Символ	Вероятность		Символ	Вероятность
A	3/14		C	4/14
B	1/14		A	3/14
C	4/14		E	3/14
D	1/14		B	1/14
E	3/14		D	1/14
F	1/14		F	1/14
G	1/14		G	1/14

Для построения кодового дерева по методу Шеннона-Фано (от корня к листьям) необходимо на каждом уровне дерева выполнять ветвление на 2 ветки с максимально близкой суммарной вероятностью.

Ветвление осуществлять, используя последовательные символы, полученные в результате сортировки вероятностей, например, CA + EBDFG, а не CBDF + AEG. Суммарный вес левой ветви больше или равен суммарному весу правой ветви.



Присвоить значение 1 левому символу, 0 — правому символу.

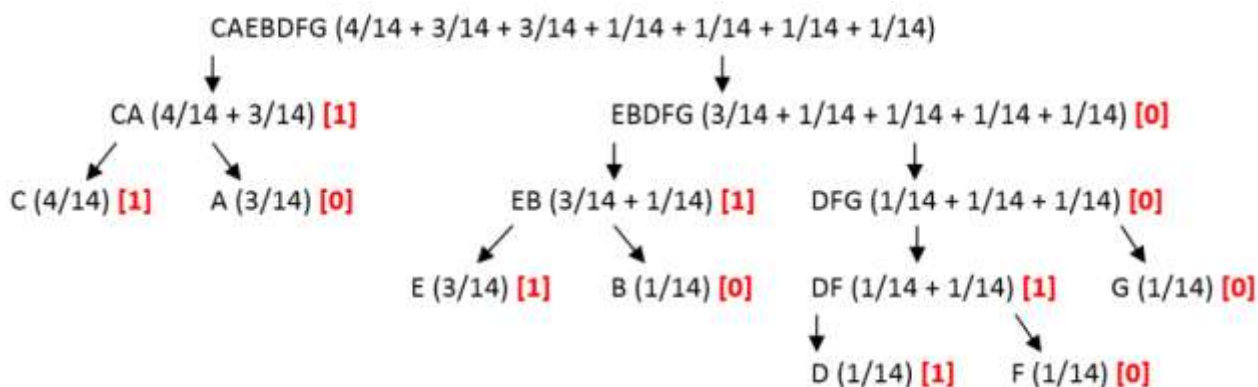


Таблица кодировки

Символ	1	2	3	4	Код
C	1		11		11
A			10		10
E		01	011		011
B			010		010
D	0	00	001	0011	0011
F				0010	0010
G			000		000

По кодировочной таблице можно определить количество бит в последовательности символов: AAABCCCCDEEEFG

10.10.10.010.11.11.11.11.0011.011.011.011.0010.000 — 37 бит

Коэффициент сжатия = $(14 * 16 \text{ бит}) / 37 \text{ бит} \approx 6,054$.

Средняя длина кодового слова = $(C) 4/14 * 2 + (A) 3/14$

$* 2 + (E) 3/14 * 3 + (B) 1/14 * 3 + (D) 1/14 * 4 + (F)$

$1/14 * 4 + (G) 1/14 * 3 = 8/14 + 6/14 + 9/14 + 3/14 +$

$4/14 + 4/14 + 3/14 = 37 / 14 \approx 2,643 \text{ бит/символ}$

Задание 2. Кодирование и декодирование информации

Кодирование прямым равномерным кодом (по образцу) применяется для представления дискретного сигнала на машинном носителе. При кодировании используются кодовые слова одинаковой длины, представленные в двоичном или шестнадцатеричном форматах.

Пример. Прямой код для символов a, b, c, d.

Символ	Номер	Двоичный код	Кодовая последовательность для слова bad — 010011
a	0	00	
b	1	01	
c	2	10	
d	3	11	

В коде Грэя двоичные коды упорядоченных символов различаются одним разрядом.

Пример. Код Грэя для 16 натуральных чисел.

Шаг 1. Построение таблицы 2x2 для получения кодов первых четырех чисел. 0_2 и 1_2 используются в качестве номеров строк и столбцов. Стрелки показывают порядок заполнения ячеек таблицы упорядоченными данными.

	0	1
0	0→	1↓
1	3	←2

Число	Код	Код числа = <номер строки><номер столбца>
0	00	
1	01	
2	11	
3	10	

Шаг 2. Построение таблицы 4x4 для получения кодов первых четырех чисел. В качестве номеров строк и столбцов используются коды, полученные на первом шаге.

	00	01	11	10
00	0→	1→	2→	3↓
01	7↓	←6	←5	←4
11	8→	9→	10→	11↓
10	15	←14	←13	←12

В результате получим коды всех 16 натуральных чисел.

Число	Код
0	0000
1	0001
2	0011
3	0010
...	...
15	1000

Кодировать можно любой набор упорядоченных символов без различия регистра.