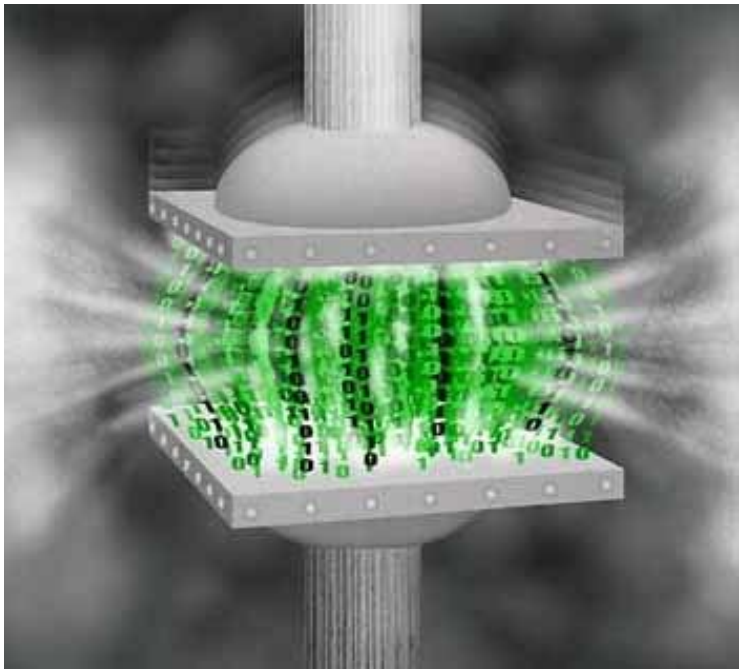
Сжатие информации

Алгоритм Хаффмана

Для того чтобы сэкономить место на внешних носителях (винчестерах, флэш-дисках) и ускорить передачу информации по компьютерным сетям, нужно ее сжать – уменьшить информационный объем, сократить длину двоичного кода. Возможны две ситуации при сжатии:

1) Потеря информации в результате сжатия недопустима;
2) Допустима частичная потеря информации в результате сжатия.

Сжатие информации



Сжатие данных – сокращение объема данных при сохранении закодированного в них содержания.

При **упаковке данных в файловые архивы** производится их *сжатие без потери информации*.

Сжатие с частичной потерей информации производится при сжатии кода изображения (графики, видео) и звука.

Сжатие без потери информации:

- использование неравномерного кода;
- выявление повторяющихся фрагментов кода.

Сжатие информации

Сжатие происходит за счет устранения избыточности кода, например, за счет упрощения кодов, исключения из них постоянных битов или представления повторяющихся символов в виде коэффициента повторения.

Важнейшая характеристика процесса сжатия – коэффициент сжатия.

Коэффициент сжатия – отношение объема исходного сообщения к объему сжатого.

Алгоритмы сжатия, использование неравномерного кода

В основе первого подхода лежит использование неравномерного кода.

В восьмиразрядной таблице символьной кодировки (ASCII), каждый символ кодируется восемью битами и, следовательно, занимает в памяти 1 байт. Информационный вес символа тем больше, чем меньше его частота встречаемости. С этим обстоятельством и связана идея сжатия текста в компьютерной памяти: отказаться от одинаковой длины кодов символов.

Сжатие с использованием кодов переменной длины

Одним из простейших, но весьма эффективных способов построения двоичного неравномерного кода, не требующего специального разделителя является алгоритм Д.Хаффмана.

Алгоритм Хаффмана

Алгоритм Хаффмана — адаптивный алгоритм оптимального префиксного кодирования алфавита с минимальной избыточностью.

Был разработан 1952 году аспирантом Массачусетского технологического института Дэвидом Хаффманом при написании им курсовой работы. В настоящее время используется во многих программах сжатия данных.

Таблица Хаффмана

Таблица 1.8. **Вариант кодовой таблицы Хаффмана**

Буква	Код Хаффмана	Буква	Код Хаффмана	Буква	Код Хаффмана	Буква	Код Хаффмана
E	100	S	0110	U	00010	K	110100011
T	001	H	0101	G	00001	X	110100001
A	1111	D	11011	Y	00000	J	110100000
O	1110	L	01111	P	110101	Q	1101000101
N	1100	F	01001	W	011101	Z	1101000100
R	1011	C	01000	B	011100		
I	1010	M	00011	V	1101001		

Особенностью данного кода является его **префиксная структура**. Это значит, что код любого символа не совпадает с началом кода всех остальных символов.

Префиксные коды

Чтобы понять, как строятся префиксные коды, рассмотрим, как построить ориентированный граф, определяющий этот код.

Например, кодовые слова 00, 01, 10, 011, 100, 101, 1001, 1010, 1111, кодируют соответственно буквы: a, b, c, d, e, f, g, h, i.

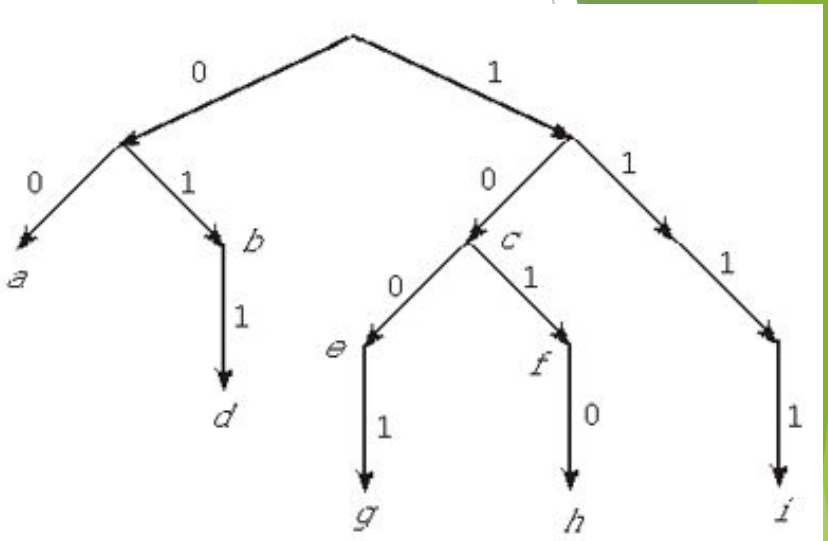
Префиксные коды

Построим граф этого кода.

Из начальной вершины выходят две дуги, помеченные 0 и 1. Затем из конца каждой такой дуги входят новые дуги, помеченные 0 и 1 так, чтобы, идя по этим дугам от корня, читалось начало какого-либо кодового слова.

Префиксные коды

Если при этом какое-то последовательность оказывается прочитанным полностью, то у конца последней дуги пишется кодируемый символ.



Из получившихся вершин снова проводятся дуги — и так далее, до тех пор, пока не будут исчерпаны все коды.

Пример 1. Используя код Хаффмана, закодировать следующий текст, состоящий из 29 знаков:

WENEEDMORESNOWFORBETTERSKIING

Используя табл. 1.8, закодируем строку:

011101 100 1100 100 100 11011 00011 1110 1011 100 0110 1100 1110 011101 01001 1110 1011 011100 100 001 001 100 1011 0110 110100011 1010 1010 1100 00001

После размещения этого кода в памяти побайтно, он примет вид:

01110110 01100100 10011011 00011111 01011100 01101100 11100111 01010011 11010110 11100100 00100110 01011011 01101000 11101010 10110000 001

В шестнадцатеричной форме он запишется так:

76 64 9B 1F 5C 6C E7 53 D6 E4 26 5B 68 EA B0 20.

Таким образом, текст, занимающий в кодировке ASCII 29 байтов, в кодировке Хаффмана займёт 16 байтов.

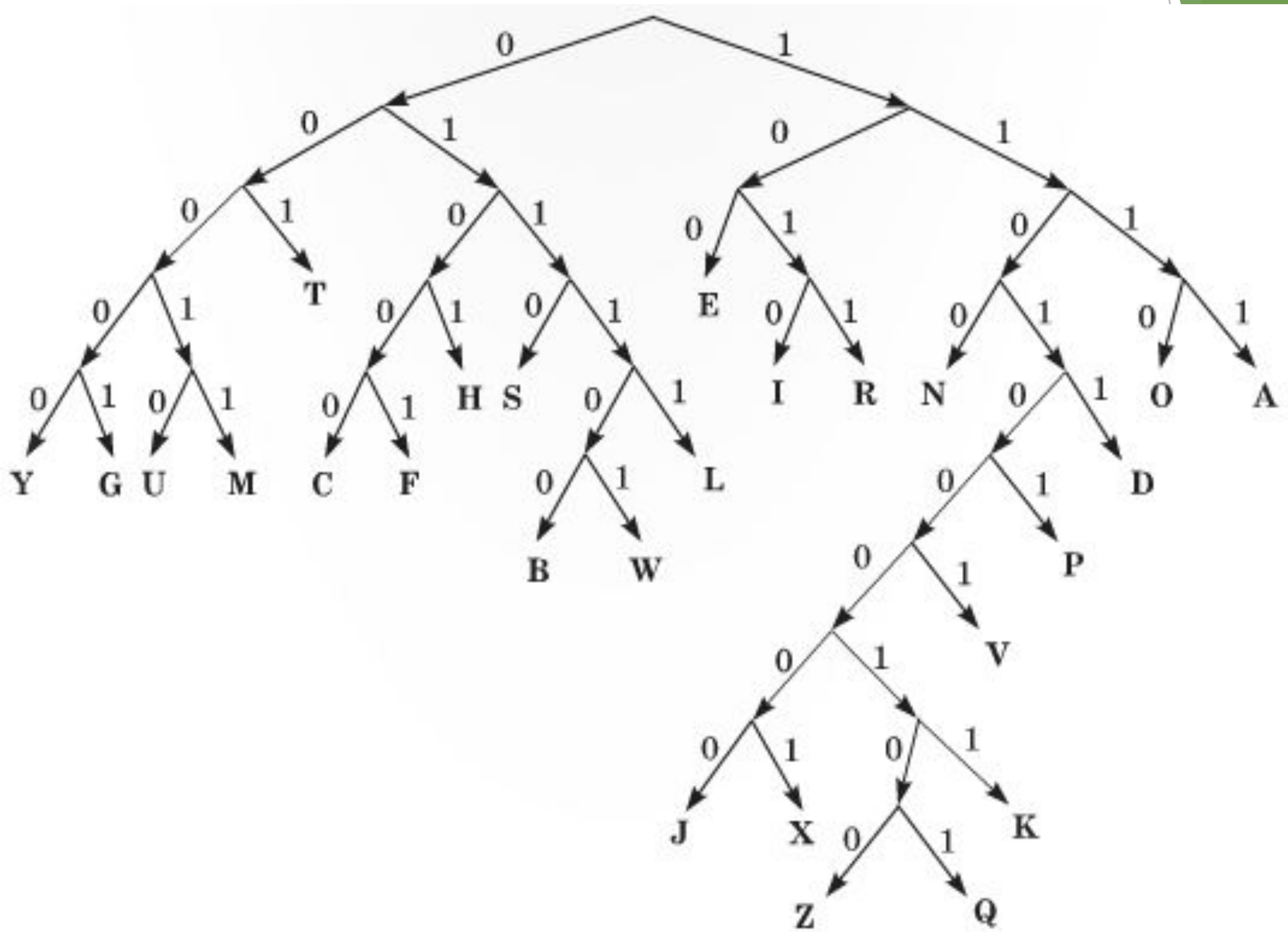
Коэффициентом сжатия *называют отношение длины кода в байтах после сжатия к его длине до сжатия.*

Деревом называется графическое представление (граф) структуры связей между элементами некоторой системы.

Двоичным деревом называется дерево, в котором любая вершина, имеет не более двух потомков.

Корнем дерева называется единственная вершина, не имеющая родительской вершины.

Листьями дерева называются вершины, не имеющие потомков.


T
E
H
S
I
R
N
O
A
Y
G
U
M
C
F
D
L
B
W
P
V
J
X
K
Z
Q

Пример 2. Раскодировать следующий двоичный код, полученный по алгоритму Хаффмана (пробелами код разделен на байты) :

01010001 00100101 00100011 11111100

Двигаясь по дереву Хаффмана, начиная от первого слева разряда, получим следующую расшифровку:

0101 → H; 00010 → U; 01001 → F; 01001 → F;
00011 → M; 1111 → A; 1100 → N.

Получилось слово HUFFMAN. Упакованный код занимал 4 байта, исходный код — 7 байтов. Следовательно, коэффициент сжатия был равен $4/7 \approx 0{,}57$.

Пример:

Предположим, что необходимо выполнить компрессию текстового документа с фразой “мама_мыла_раму”. Наш исходный текст “весит” 112 бит, так как каждый символ занимает 8 бит в кодовой таблице, а таких символов у нас 14 штук.

1. Составляем таблицу частот, то есть, подсчитываем количество вхождений каждой буквы во фразу, в результате чего получим вес каждой буквы:

у	р	л	ы	-	а	М
1	1	1	1	2	4	4

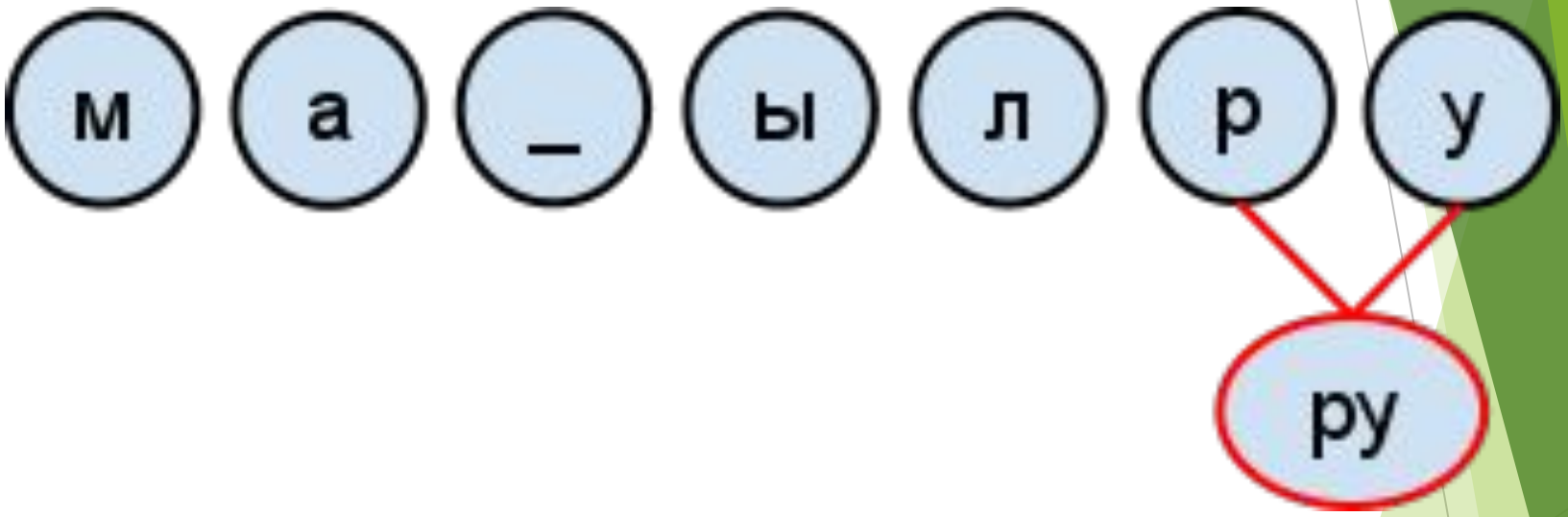
2. Сортируем значения в таблице по весам, в порядке спадания:

м	а	-	ы	л	р	у
4	4	2	1	1	1	1

3. Выбираем 2 значения с минимальными весами (“р” и “у”), суммируем их веса и заменяем эти значения в таблице одним объединенным значением:

м	а	-	ы	л	ру
4	4	2	1	1	2

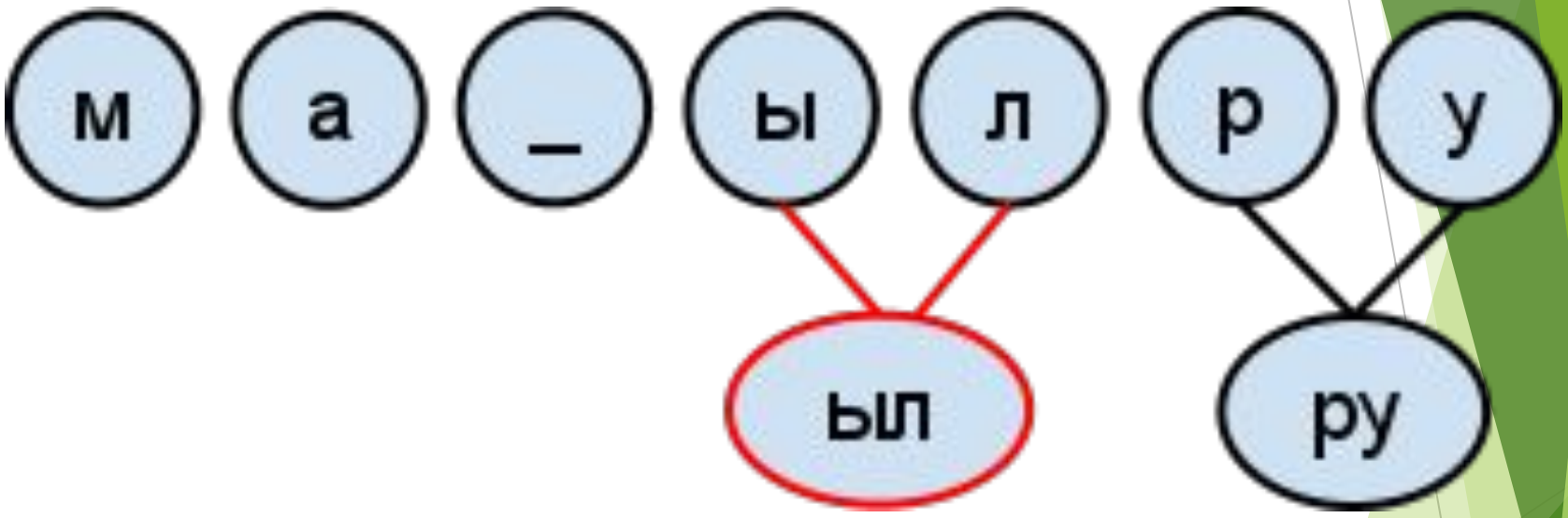
Формируем дерево



4. Снова выбираем 2 значения с минимальными весами (“ы” и “л”), делаем с ними то же, что и на предыдущем шаге:

М	А	-	ЫЛ	РУ
4	4	2	2	2

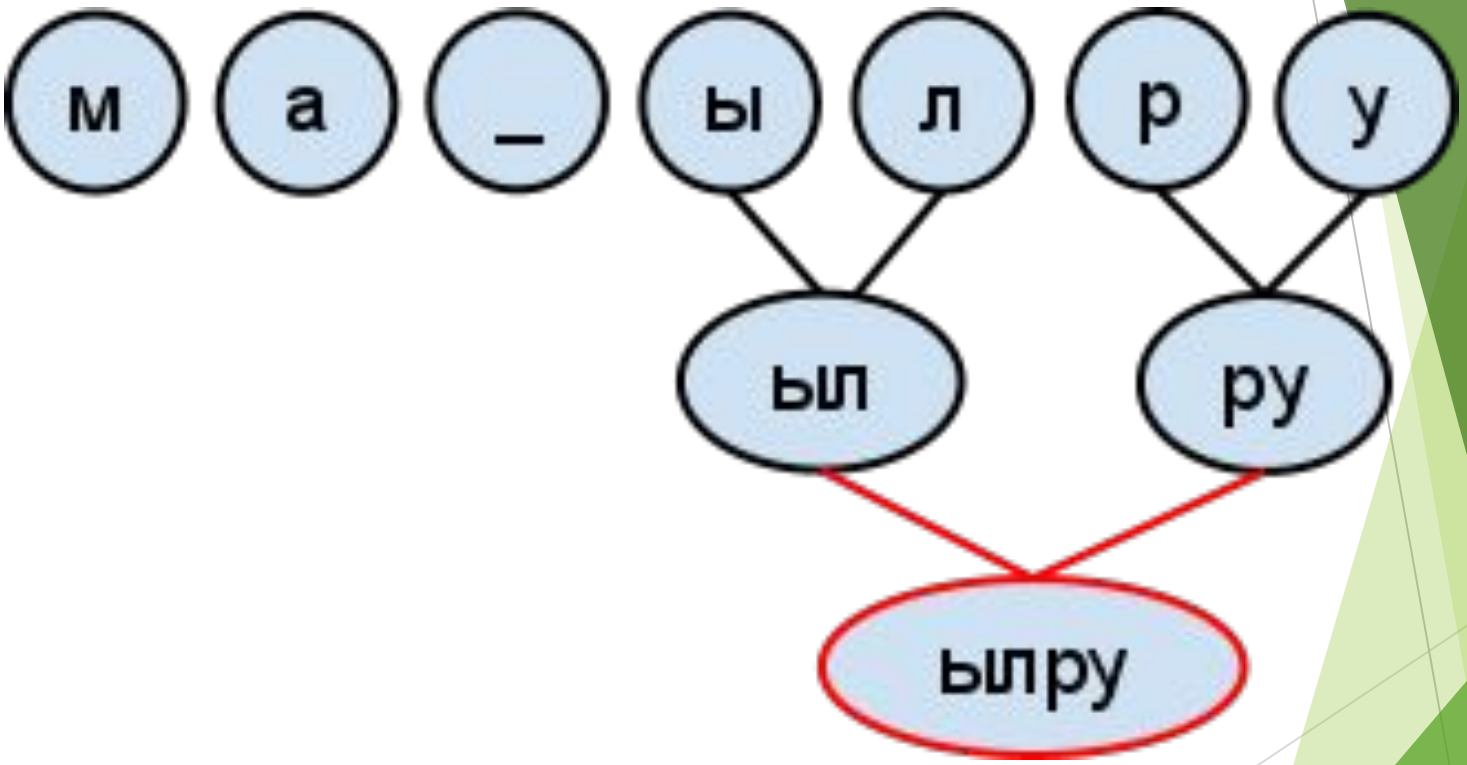
Дерево стало таким:



5. Снова выбираем 2 значения с минимальными весами (“ыл” и “ру”), делаем с ними то же, что и на предыдущем шаге:

М	Ф	-	ЫЛРУ
4	4	2	4

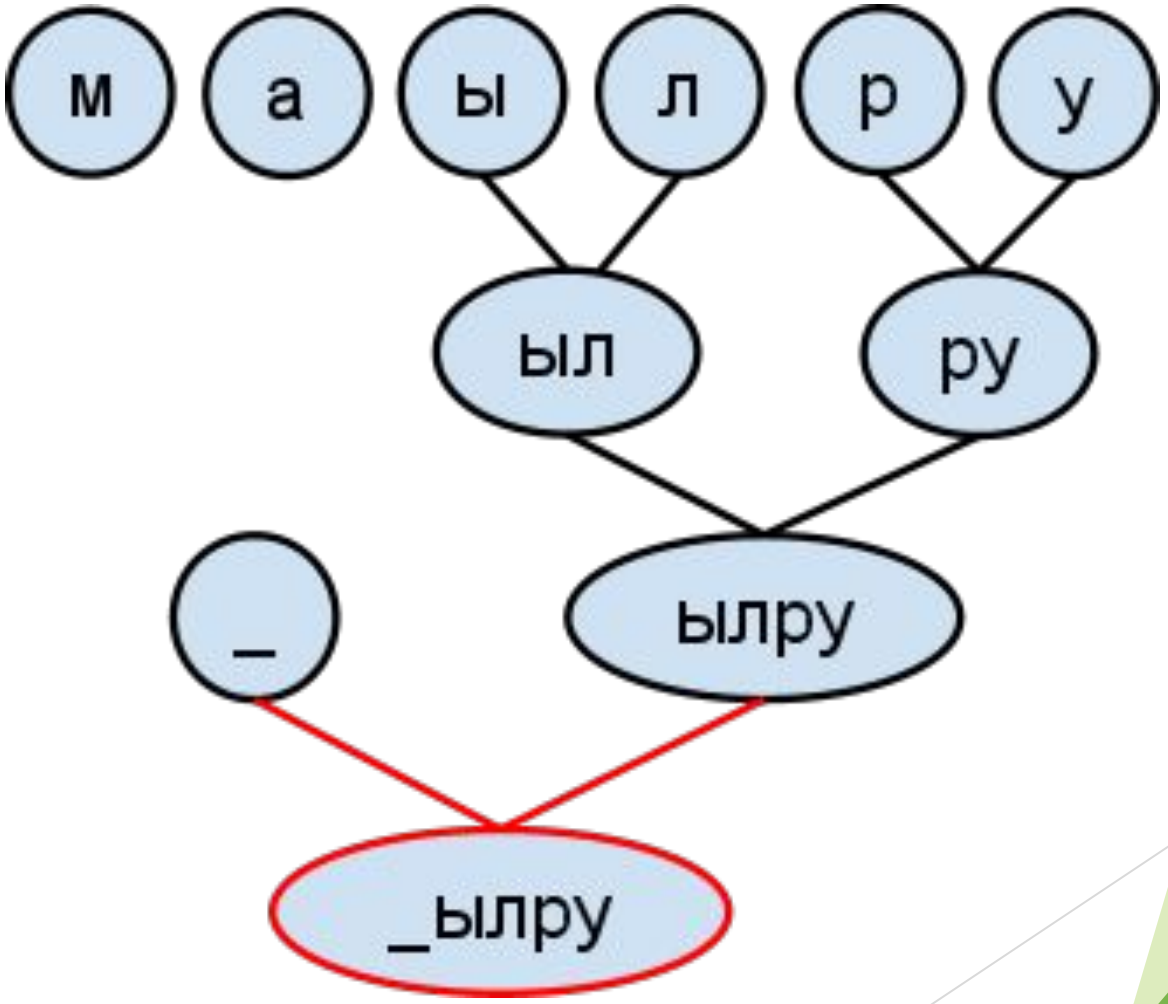
Дерево стало таким:



6. Снова выбираем 2 значения с минимальными весами (“_” и “ылру”), делаем с ними то же, что и на предыдущем шаге

М	А	-ЫЛРУ
4	4	6

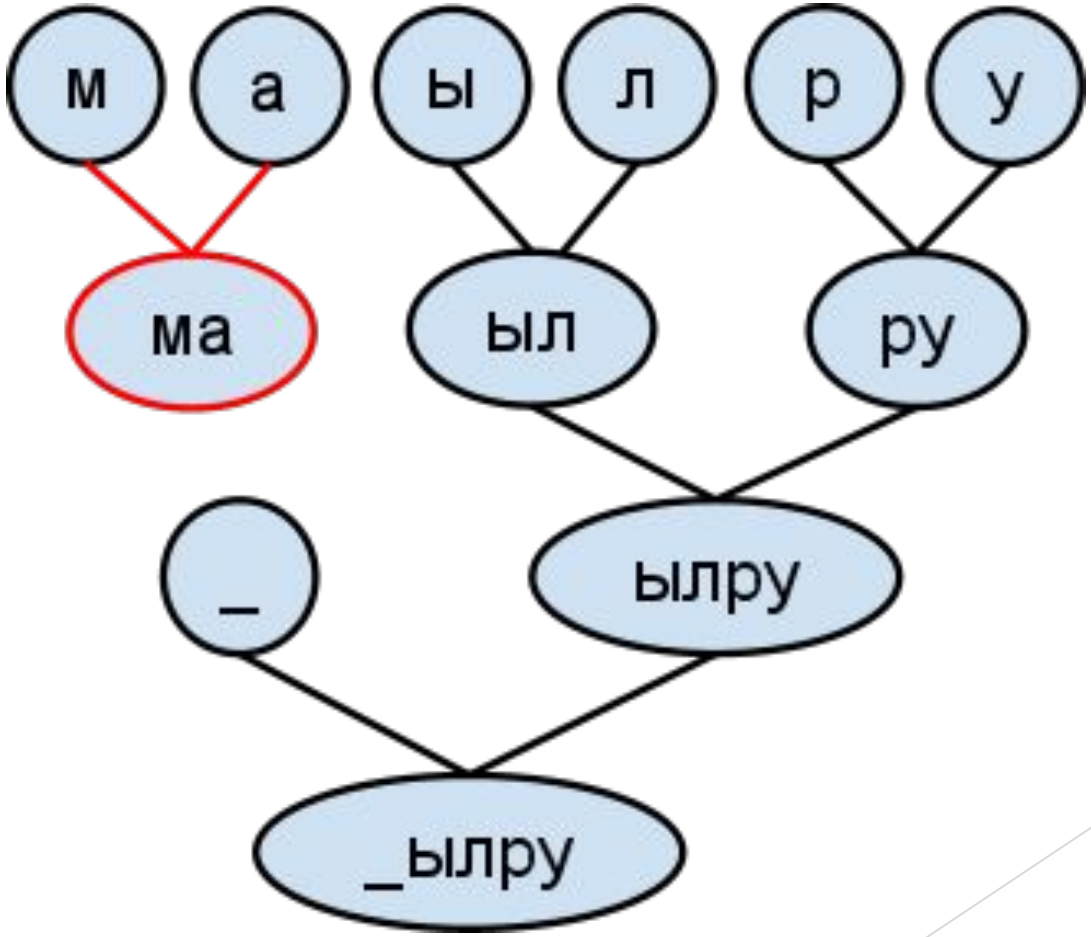
Дерево стало таким:



7. Снова выбираем 2 значения с минимальными весами (“м” и “а”), делаем с ними то же, что и на предыдущем шаге:

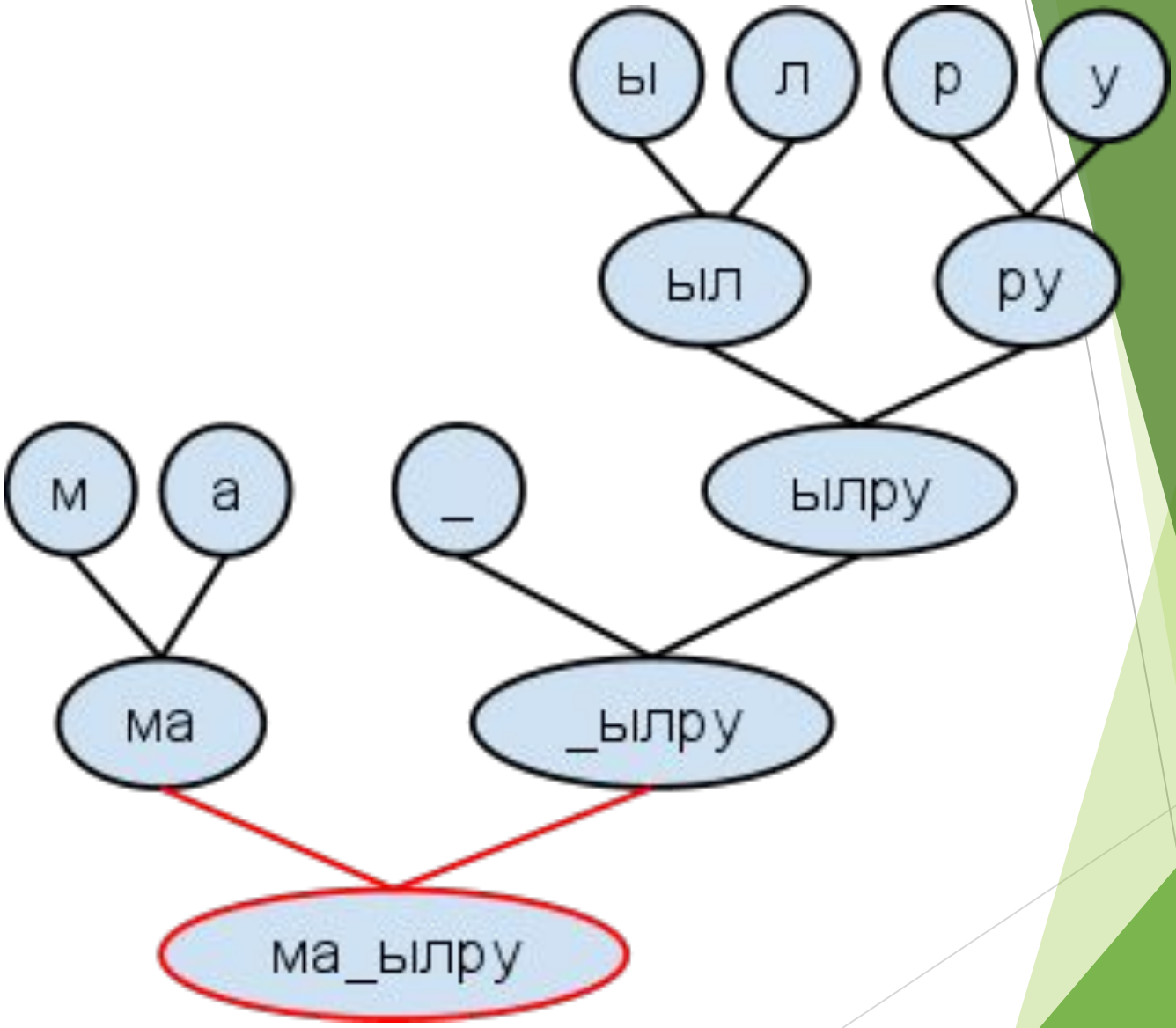
МА	-ЫЛРУ
8	6

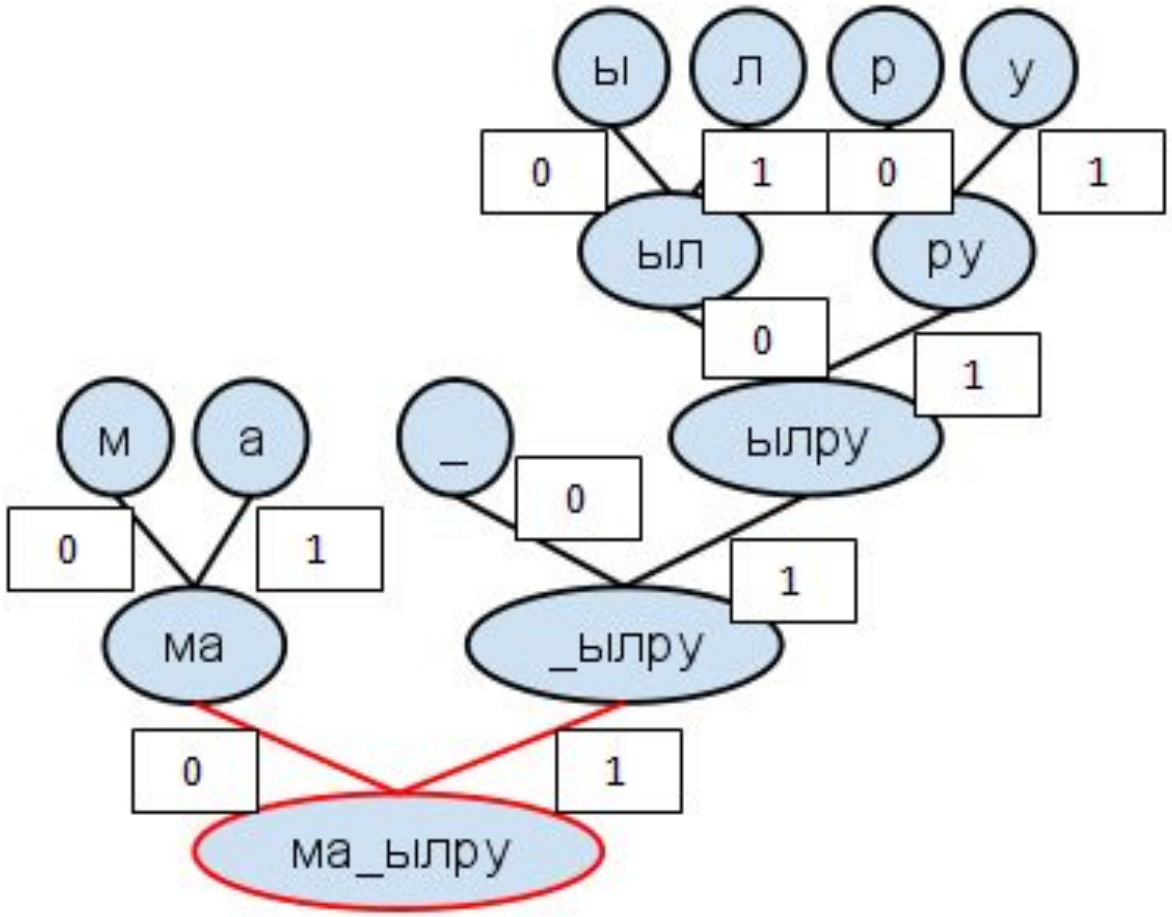
Дерево стало таким:



Последний шаг:

МА_ЫЛРУ
14


ы
л
р
у
ыл
ру
м
а
_
ылру
ма
_ылру
ма_ылру


ы
л
р
у
0
1
0
1
ыл
ру
0
1
ылру
м
а
_
0
0
1
1
ма
_ылру
0
1
ма_ылру

РЕЗУЛЬТАТ

м	а	-	ы	л	р	у
00	01	10	1100	1101	1110	1111
4	4	2	1	1	1	1

КОЭФФИЦИЕНТ СЖАТИЯ: **112/40=2,8**

Алгоритм кода Хаффмана:

1. Символы исходного алфавита образуют вершины. Вес каждой вершины вес равен количеству вхождений данного символа в сжимаемое сообщение.
2. Среди вершин выбираются две с наименьшими весами (если таких пар несколько, выбирается любая из них).
3. Создается следующая вершина графа, из которой выходят две дуги к выбранным вершинам; одна дуга помечается цифрой 0, другая — символом 1.

Вес созданной вершины равен сумме весов, выбранных на втором шаге вершин.

4. К новым вершинам применяются шаги 2 и 3 до тех пор, пока не останется одна вершина с весом, равным сумме весов исходных символов.

Математики доказали, что среди алгоритмов, кодирующих каждый символ по отдельности и целым количеством бит, алгоритм Хаффмана обеспечивает наилучшее сжатие.

Ко второму подходу к сжатию без потерь относится подход, основанный на идее выявления повторяющихся фрагментов кода.

Учет повторений фрагментов кода:	*Алгоритм RLE: коэффициент повторения + повторяющийся байт.* *Алгоритмы Лемпеля–Зива: словарь повторяющихся слов, ссылки на слова*	Сжатие кода звука:	*удаление неслышимых гармоник;* *использование нелинейной зависимости громкости от амплитуды*

Решить самостоятельно:

1. Постройте код Хаффмана для фраз и определить коэффициент сжатия.

- **КАРЛ_ У_КЛАРЫ_УКРАЛ_ КОРАЛЛЫ, А_КЛАРА_У_КАРЛА_УКРАЛА_КЛАР НЕТ**
- **НА_ ДВОРЕ_ ТРАВА,_ НА_ ТРАВЕ_ ДРОВА**

Решить самостоятельно:

2. Закодируйте с помощью кода Хаффмана следующий текст:

HAPPYNEWYEAR

3. Расшифруйте с помощью двоичного дерева Хаффмана следующий код:

11110111 10111100 00011100 00101100

10010011 01110100 11001111 11101101

001100

Задача А9

Для кодирования сообщения, состоящего из букв А, Б, В, Г и Д, используется неравномерный двоичный код, позволяющий однозначно декодировать полученную двоичную последовательность.

А–00, Б–010, В–011, Г–101, Д–111.

Можно ли сократить для одной из букв длину кодового слова так, чтобы код по-прежнему можно было декодировать однозначно?

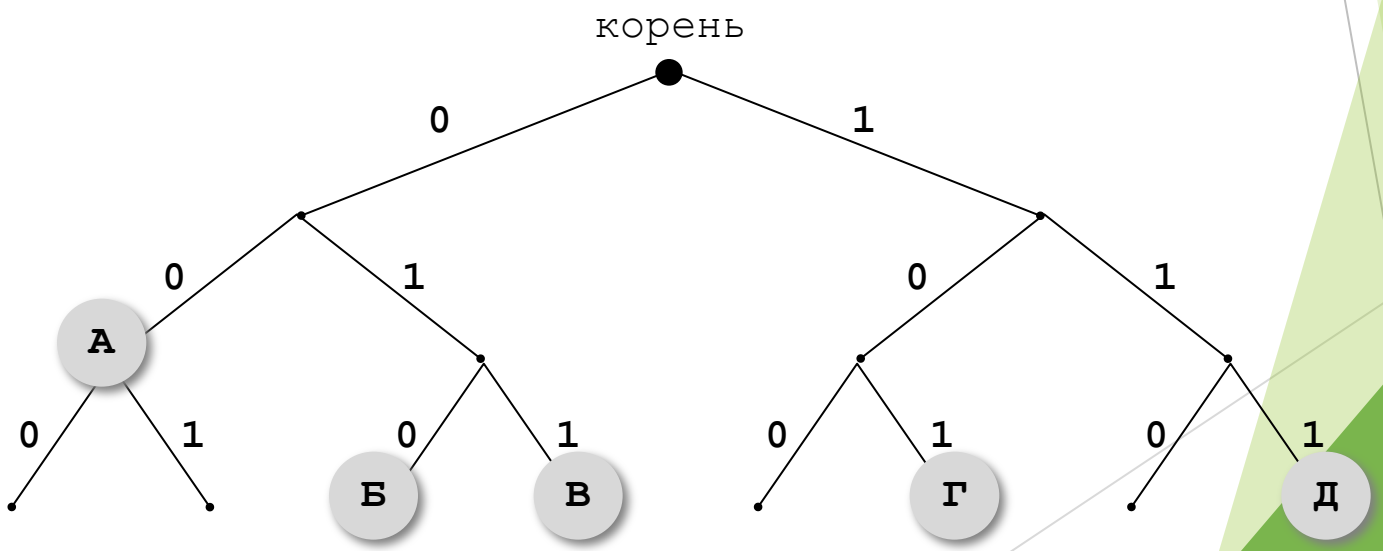
Выберите правильный вариант ответа.

1) для буквы Б – 01 2) это невозможно

3) для буквы В – 01 4) для буквы Г – 01

Задача А9. Решение.

Построим двоичное дерево, в котором от каждого узла отходит две ветки: 0 или 1. Разместим на дереве буквы А, Б, В, Г и Д так, чтобы их код получался как последовательность чисел на рёбрах:



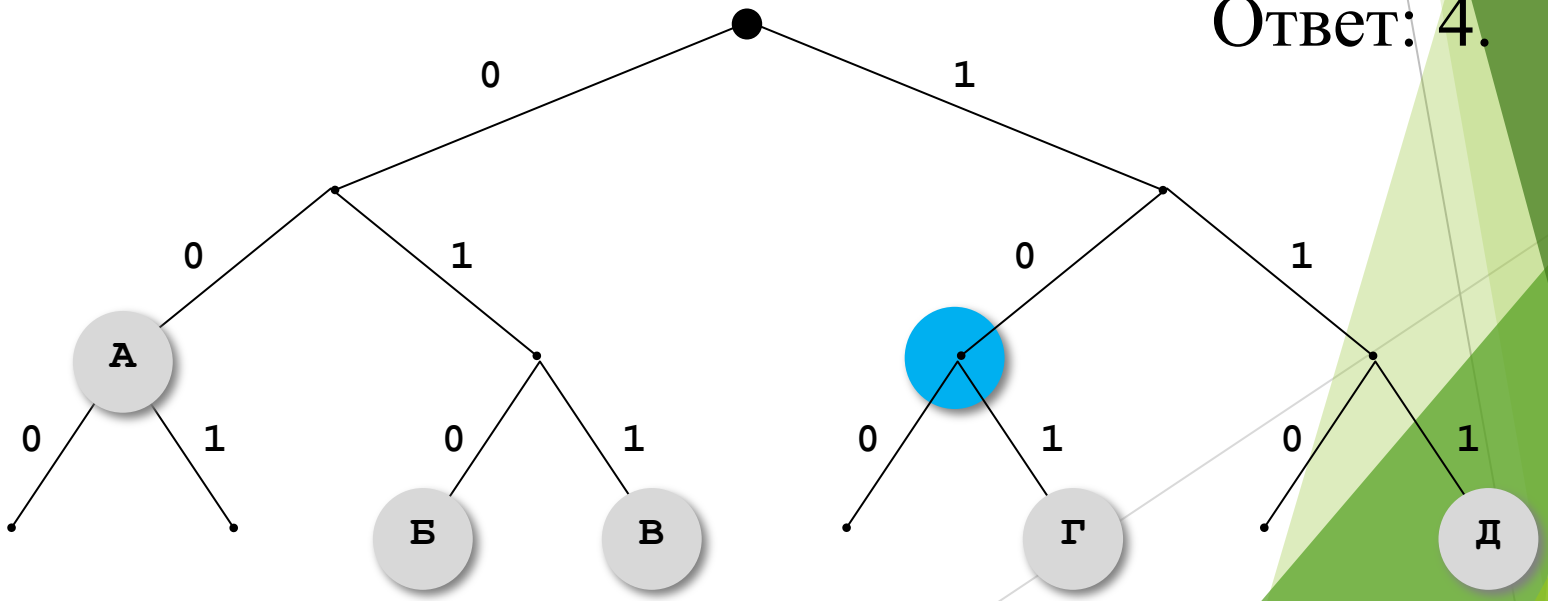
Задача А9. Решение.

По дереву определим, что для букв Г и Д код можно сократить. Выберем ответ из предложенных вариантов:

1) для буквы Б – 01 2) это невозможно

3) для буквы В – 01 4) для буквы Г – 01

Ответ: 4.



Для самостоятельной работы

Для передачи по каналу связи сообщения, состоящего только из букв А, Б, В, Г, решили использовать неравномерный по длине код:
А=0, Б=10, В=110.
Как нужно закодировать букву Г, чтобы длина кода была минимальной и допускалось однозначное разбиение кодированного сообщения на буквы?
1) 1 2) 1110 3) 111 4) 11

Задача А9

Для 5 букв латинского алфавита заданы их двоичные коды.
Эти коды представлены в таблице:

A	B	C	D	E
000	01	100	10	011

Определить, какой набор букв закодирован двоичной строкой 0110100011000

Задача А9

Для передачи по каналу связи сообщения, состоящего только из букв А, Б, В, Г, решили использовать неравномерный по длине код:

А=0, Б=10, В=110.

Как нужно закодировать букву Г, чтобы длина кода была минимальной и допускалось однозначное разбиение кодированного сообщения на буквы?

Д/З

1. Постройте код Хаффмана для фраз и определить коэффициент сжатия.

- **ОТ_ТОПОТА_КОПЫТ_ПЫЛЬ_ПО_ПОЛЮ_ЛЕТИТ**
- **ШЛА_САША_ПО_ШОССЕ_И СОСАЛА_СУШКУ**

Список используемой литературы:

- http://crazycode.net/blog/10-algorithms-and-data-structures/31-huffman
- http://edu.1september.ru/courses/07/008/01.pdf
- http://www.lukomor.ru/attach/pages/ege13/A09.pdf?PHPSESSID=9fa7039ee3de232e76b2a13614accbf5
- Педагогический университет «Первое сентября», 2008г.
- И.Г. Семакин «Информатика и ИКТ» 10 класс профильный уровень,2012