

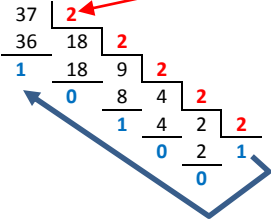
I. Системы счисления

1. Перевод из десятичной в n-ичную

1.1. Перевод целых десятичных чисел в 2-ичные

$$37_{10} \rightarrow ?_2$$

$$37_{10} = 100101_2$$



ПОСЛЕДНЯЯ ЦИФРА ДВОИЧНОГО ЧИСЛА – ЕСТЬ ПЕРВЫЙ ОСТАТОК ОТ ДЕЛЕНИЯ ДЕСЯТИЧНОГО ЧИСЛА НА ОСНОВАНИЕ СИСТЕМЫ!

1.2. Перевод десятичных дробей в 2-ную систему

$$0,62_{10} \rightarrow ?_2$$

$$0,62_{10} = 0,1001_2$$

0,	62	×2
1,	24	×2
0,	48	×2
0,	96	×2
1,	92	...

Итого: $37,62_{10} = 100101,1001_2$

1.3. Способ перевода десятичного числа в двоичное без деления

Число **49**. Ближайшая большая либо равная степень **двойки** – **6** ($2^6 = 64 \geq 49$). Следовательно, будет **6** разрядов двоичного числа.

$$49 = 32 + 16 + 1.$$

$$49_{10} = 110001_2$$

1	2	3	4	5	6					
1	1	0	0	0	1					
×	×	×	×	×	×					
32	+	16	+	8	+	4	+	2	+	1
2^5		2^4		2^3		2^2		2^1		2^0

2. Перевод из n-ичной системы в десятичную

$$426,22_8 \rightarrow ?_{10}$$

$$2 = 3 - 1$$

$$4 \times 8^2 + 2 \times 8^1 + 6 \times 8^0 + 2 \times 8^{-1} + 2 \times 8^{-2} = 278,28125_{10}$$

3. Степени двойки

0	1	2	3	4	5	6	7	8	9	10
1	2	4	8	16	32	64	128	256	512	1024

Десятичное число 2 в любой степени в двоичном представлении дает единицу с последующим за ним таким количеством нулей, какова степень двойки.

$$2^{12}_{10} = 1 \underbrace{0000\ 0000\ 0000}_2$$

12

4. Перевод между 2-ичной, 8-ричной и 16-ричной системами

4.1. Триады

Чтобы записать цифры 8-ричной системы в двоичном виде понадобится 3 разряда ($8 = 2^3$).

0	1	2	3	4	5	6	7
000	001	010	011	100	101	110	111

$$1011101,11_2 \rightarrow ?_8$$



001	011	101	,	110
1	3	5		6

Обязательно добить числа до и после запятой незначащими нулями так, чтобы количество цифр было кратным трем! До запятой нули ставятся перед числом, после запятой – за числом.

ОТСЧЕТ ВЕДЕТСЯ ОТ ЗАПЯТОЙ!

$$1011101,11_2 = 135,6_8$$

Обратное преобразование

$$63,2_8 \rightarrow ?_2$$

$$6_8 = 110_2; 3_8 = 011_2; 2_8 = 010_2 \Rightarrow 63,2_8 = 110011,010_2$$

4.2. Квадриады

Чтобы записать цифры 16-ричной системы в двоичном виде, понадобится 4 разряда ($16 = 2^4$)

0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111

8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

5. Арифметические операции в двоичной системе

5.1. Сложение

1	1	10
1	0	1
0	1	1
0	0	0

$$\begin{array}{r}
 111 \\
 1011 \\
 + \\
 \hline
 0110 \\
 10001
 \end{array}$$

На ум
пошло

5.2. Умножение

1	1	1
1	0	0
0	1	0
0	0	0

Любое число, умноженное на 1, дает само себя.
Любое число, умноженное на 0, дает 0.

5.3. Вычитание

1	1	0
1	0	1
0	1	1
0	0	0

$$\begin{array}{r}
 \bullet 1 \\
 1001 \\
 - \\
 \hline
 010 \\
 0111
 \end{array}$$

Занимаем

Т.к. $1+1=10 \Rightarrow 10-1=1$

5.4. Деление

1	1	1
0	1	0

$$\begin{array}{r}
 110101 \quad | \quad 1011 \\
 - \quad 100,1... \\
 \hline
 1011 \\
 \underline{10011} \\
 - \\
 \underline{1011} \\
 1000
 \end{array}$$

Принцип тот же, что и при обычном делении столбиком. Делим до заданной точности.

II. Логика

1. Таблицы истинности

1.1. Дизъюнкция

A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0



Правило единицы. Если есть хоть одна 1, ставим 1

1.2. Конъюнкция

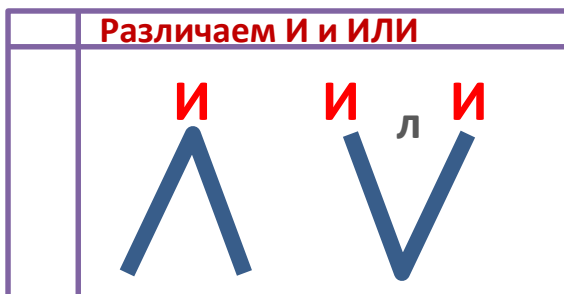
A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0



Правило нуля. Если есть хоть один 0, ставим 0

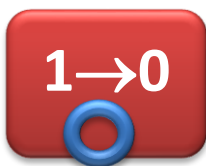
1.3. Отрицание

A	$\neg A$
1	0
0	1



1.4. Импликация

A	B	$A \rightarrow B$
1	1	1
1	0	0
0	1	1
0	0	1



Ложь только тогда, когда из правды следует ложь.
Из лжи следует все, что угодно

1.5. Эквиваленция

A	B	$A \leftrightarrow B$
1	1	1
1	0	0
0	1	0
0	0	1



Истина только тогда, когда оба высказывания одинаковы (оба истинны, или оба ложны)

2. Законы эквивалентности

1. $A \vee B \equiv B \vee A$	1'. $A \wedge B \equiv B \wedge A$
2. $A \vee (B \vee C) \equiv A \vee (B \vee C)$	2'. $A \wedge (B \wedge C) \equiv A \wedge (B \wedge C)$
3. $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$	3'. $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$
4. $A \vee A \equiv A$	4'. $A \wedge A \equiv A$
5. $A \vee \bar{A} \equiv 1$	5'. $A \wedge \bar{A} \equiv 0$
6. $A \vee 1 \equiv 1$	6'. $A \wedge 1 \equiv A$
7. $A \vee 0 \equiv A$	6'. $A \wedge 0 \equiv 0$
8. $A \vee (A \wedge B) \equiv A$	8'. $A \wedge (A \vee B) \equiv A$
9. $(A \wedge B) \vee (\bar{A} \wedge B) \equiv B$	9'. $(A \vee B) \wedge (\bar{A} \vee B) \equiv B$
10. $A \vee (\bar{A} \wedge B) \equiv A \vee B$	10'. $\bar{A} \wedge (A \vee B) \equiv \bar{A} \wedge B$
11. $\overline{A \vee B} \equiv \bar{A} \wedge \bar{B}$	11'. $\overline{A \wedge B} \equiv \bar{A} \vee \bar{B}$
12. $\overline{\bar{A}} \equiv A$	
13. $A \rightarrow B \equiv \bar{A} \vee B$	
14. $A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A) \equiv (\bar{A} \vee B) \wedge (\bar{B} \vee A)$	

III. Измерение количества информации

1. Единицы измерения информации

1 бит

1 байт = 8 бит = 2^3 бит

1 КБ = 1024 байта = 2^{10} байт = 2^{13} бит

1 МБ = 1024 КБ = 1024^2 байт = 2^{20} байт = 2^{23} бит

1 ГБ = 1024 МБ = 1024^2 КБ = 1024^3 байт = 2^{30} байт = 2^{33} бит

1.1. Перевод из числового вида в степени двойки

Перевести 128 МБ в вид 2^x бит

1. Обязательно проверяем – во что нам надо перевести (в биты или байты)!!!

2. Переводим МБ в биты. Это будет 2^{23} бит.

3. Переводим 128 в степень двойки. Это будет 2^7 .

4. Следовательно, $128 \text{ МБ} = 2^7 \times 2^{23} \text{ бит} = 2^{30} \text{ бит}$.

1.2. Перевод из степеней двойки в числовой вид

Перевести 2^{19} бит в числовой вид

1. Обязательно проверяем – данное нам число измеряется в битах или байтах!!!

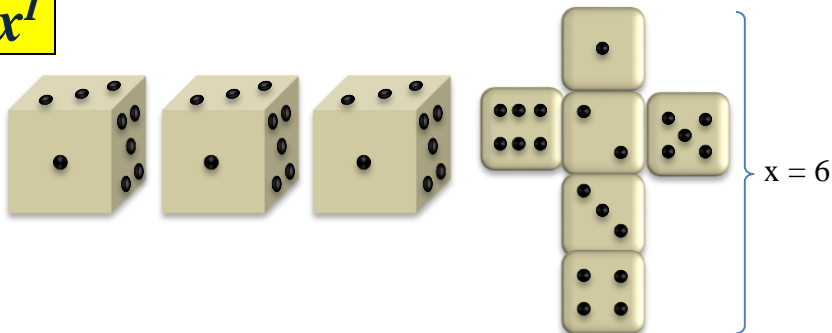
2. Переводим биты в байты (если это необходимо). $2^{19} : 2^3 = 2^{16}$ байт.

3. Определяем приставку по разряду десятков степени. Получается $2^{10} \times 2^6$ байт или 2^6 КБ.

4. Преобразуем 2^6 в число. Следовательно, 2^{19} бит = 64 КБ.

2. Определение количества возможных вариантов

$$N = x^I$$

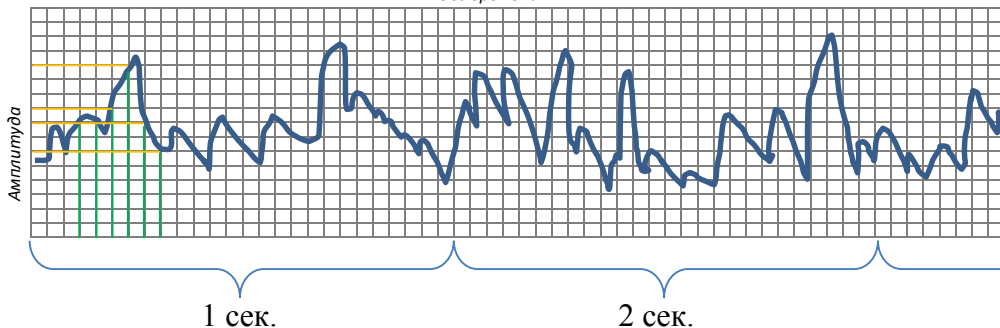


$$N = 6^3 = 216 \text{ вариантов}$$

3. Измерение количества звуковой информации

Принцип оцифровки одноканального звукового сигнала

Ось времени



Частота дискретизации – количество **фрагментов** за единицу времени

Глубина звука (разрешение) – количество **бит**, которыми может измеряться амплитуда

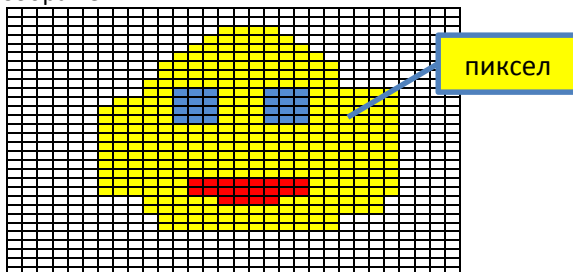
Чтобы найти объем информации 1 секунды файла, нужно умножить глубину звука на частоту дискретизации. Чтобы найти полный объем файла, нужно умножить полученное значение на продолжительность звука. Если звукозапись одноканальная (моно), это и будет ответом; если запись двухканальная (стерео), то значение нужно умножить на 2.

$$\text{Объем} = \text{ЧД} \times \text{ГЗ} \times \text{Время} \times \text{КК}$$

ЧД – частота дискретизации; ГЗ – глубина звука, КК – количество каналов (1 или 2).

4. Измерение количества графической информации

Принцип оцифровки изображения

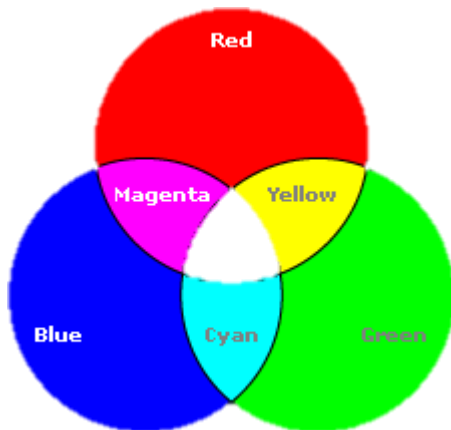


Изображение разбивается на отдельные точки (пиксели), каждый из которых имеет цветовую характеристику (**глубина цвета**), измеряющуюся в битах. Например, глубина цвета 16 бит соответствует $2^{16} = 65536$ различных цветов. Чтобы найти объем изображения, нужно количество пикселей (**разрешение изображения**) умножить на глубину цвета. *Если глубина цвета не дана, ее можно вычислить, исходя из известного количества цветов.*

$$\text{Объем} = \text{Кх} \times \text{Ку} \times \text{ГЦ}$$

Кх – количество пикселей по ширине, Ку – количество пикселей по высоте, ГЦ – глубина цвета

4.1. Цветовые модели



Глубина цвета записывается в виде $\#RRGGBB$ (красный, зеленый, синий), где R , G , B – шестнадцатеричные числа, изменяющиеся от 0 до F . Соответственно, 00 – полное отсутствие составляющей, FF – максимальное присутствие. Таким образом, цвет $\#FF0000$ – красный, т.к. присутствует максимальное количество красного цвета, и отсутствуют все другие. А вот $\#00FFFF$ – это голубой цвет (Cyan), т.к. здесь отсутствует красный, но по максимуму присутствуют синий и зеленый. Очевидно, что $\#FFFFFF$ дает нам белый цвет, а $\#000000$ – черный. $\#BBBBBB$ или $\#121212$ дадут нам оттенки серого цвета (первое значение – более светлое, второе – более темное).

5. Измерение скорости передачи информации

Чтобы найти объем передаваемого по сети файла, нужно пропускную способность канала связи умножить на время передачи данных.

$$\text{Объем} = \text{ПС} \times \text{время}$$

ПС – пропускная способность канала связи (измеряется в бит/сек).

Внимание! Если одна величина дана в числовом виде (например, объем равен 12 МБ), а другая – в виде степени двойки (например, пропускная способность 2^{30} бит), то целесообразнее **все привести к степеням двойки** – удастся избежать долгих и сложных вычислений.

IV. Сети и файловая система

1. Вычисление IP-адреса сети

Каждый клиент сети (а так же каждая локальная сеть, как клиент более крупной сети) имеют собственный IP-адрес вида XXX.XXX.XXX.XXX, где каждое число принимает значение от 0 до 255.

Локальная сеть имеет собственный адрес, и каждый клиент этой сети также имеет собственный адрес. Степень совпадения этих адресов характеризует маска сети. Она показывает, какие компоненты адресов сети и узла совпадают, а какие – нет.

Вычисление IP-адреса сети.

1. Если в маске сети присутствует значение 255, то соответствующее значение у адресов сети и узла одинаковы.
2. Если в маске сети присутствует значение 0, то соответствующее значение у адреса сети также будет равно 0.
3. Переводим значения IP-адреса узла и маски сети (те, которые не равны указанным выше значениям) в двоичный вид (обязательно восьмимбитное; если разрядов меньше – добить слева незначащими нулями).
4. Поразрядно умножить полученные значения.
5. Получившееся число вновь перевести в десятичный вид.

Пример:

IP-адрес узла: 138.92.54.12

Маска сети: 255.255.100.0

Очевидно, что IP-адрес сети будет иметь вид 138.92.XXX.0. Найдем недостающее значение.

$$54_{10} = 00110110_2$$

$$100_{10} = 01100100_2$$

Перемножим поразрядно, исходя из правила умножения двоичных чисел:

0	0	1	1	0	1	1	0
x	x	x	x	x	x	x	x
0	1	1	0	0	1	0	0

$$00110110_2 \times 01100100_2 = 0001000100_2$$

$100100_2 = 36_{10}$. Следовательно, IP-адрес сети будет равен **138.92.36.0**

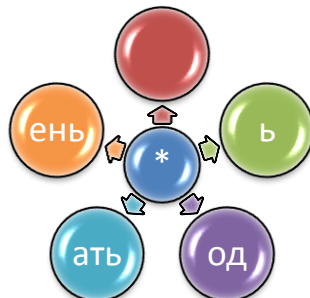
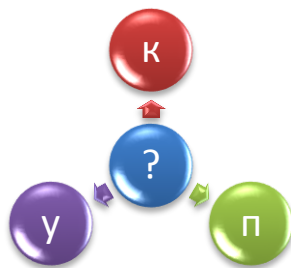
2. Маски ввода в файловой системе Windows

? – заменяет один любой символ

* – заменяет последовательность символов длиной от 0 до 255 знаков

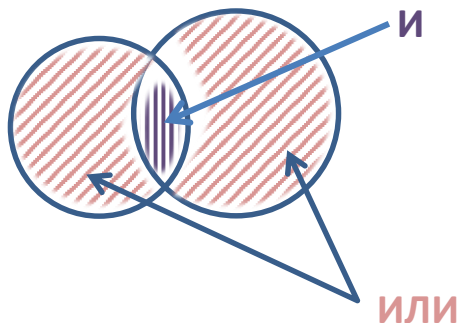
?ров*.do*

кров.doc
кровь.docx
провод.do
кровать.doc
уровень.docx



3. Поиск информации в сети

Круги Эйлера



V. Алгоритмы

1. Цепочки символов.

1.1. Найти количество некоторых символов в некоторой строке

1. Находим зависимость длины строки от ее номера. Практически на 100% эта зависимость будет $2^n - 1$.

Например,

(1) А	1	= $2^1 - 1$
(2) ВАА	3	= $2^2 - 1$
(3) СВААВАА	7	= $2^3 - 1$
(4) DCBAABAACBAABAА	15	= $2^4 - 1$

2. Находим зависимость количества букв от строки. Например, если ищем букву В, тогда:

2 – 1

3 – 2

4 – 4

Получится зависимость 2^{n-2} . Вообще, как правило, всегда получаются зависимости, где номер строки так или иначе фигурирует в степени двойки, главное, найти как. Тогда можно найти количество букв В в любой строке.

1.2. Найти символ (последовательность символов), стоящих на некотором месте в некой строке

1. Данный пункт абсолютно аналогичен.

2. Для решения данного типа задач нужно знать, как правило, не только длину искомой строки, но и предыдущей, т.к. каждая последующая строка формируется путем сложения предыдущих строк.

3. Искомые символы обычно находятся где-нибудь на стыке двух предыдущих строк, формирующих новую строку. Т.е. чаще всего достаточно выяснить, чем начинается/заканчивается предыдущая строка.

2. Исполнитель Калькулятор

1. прибавь 2,
2. умножь на 3.

Запишите порядок команд в программе преобразования числа 3 в число 69, содержащей не более 5 команд.

Тут лучше использовать обратный алгоритм – преобразование 69 в 3 при помощи обратных команд: 1 – вычти 2, 2 – подели на 3. **Правило простое: делится – дели, не делится – вычитай!**

69	23	21	7	5	3
2	1	2	1	1	

Главное не забыть, что команды надо читать справа налево! Т.е. ответ будет 11212.

3. Шифрование.

Можно ли заменить символ, чтобы код можно было декодировать однозначно?

Здесь правило простое. Если данной последовательностью начинается или заканчивается код другого символа, значит, эту последовательность использовать нельзя. Если же таких накладок нет – значит, можно.

А – 00, Б – 01, В – 100, Г – 101, Д – 110.

- 1) для буквы Д – 11
- 2) это невозможно
- 3) для буквы Г – 10
- 4) для буквы Д – 10

Очевидно, отмечаются 3 (В начинается на 10, а Д - заканчивается) и 4 (по той же причине – В и Г). А вот 1 ничему не противоречит, значит, это правильный ответ. ☺

Оглавление

I. Системы счисления	1
1. Перевод из десятичной в n-ичную	1
1.1. Перевод целых десятичных чисел в 2-ичные	1
1.2. Перевод десятичных дробей в 2-ую систему	1
1.3. Способ перевода десятичного числа в двоичное без деления	1
2. Перевод из n-ичной системы в десятичную	1
3. Степени двойки	2
4. Перевод между 2-ичной, 8-ричной и 16-ричной системами	2
4.1. Триады	2
4.2. Квадриады	2
5. Арифметические операции в двоичной системе	3
5.1. Сложение	3
5.2. Умножение	3
5.3. Вычитание	3
5.4. Деление	3
II. Логика	4
1. Таблицы истинности	4
1.1. Дизъюнкция	4
1.2. Конъюнкция	4
1.3. Отрицание	4
1.4. Импликация	4
1.5. Эквиваленция	4
2. Законы эквивалентности	5
III. Измерение количества информации	6
1. Единицы измерения информации	6
1.1. Перевод из числового вида в степени двойки	6
1.2. Перевод из степеней двойки в числовой вид	6
2. Определение количества возможных вариантов	6
3. Измерение количества звуковой информации	7
4. Измерение количества графической информации	7
4.1. Цветовые модели	8
5. Измерение скорости передачи информации	8
IV. Сети и файловая система	9
1. Вычисление IP-адреса сети	9
2. Маски ввода в файловой системе Windows	9
3. Поиск информации в сети	10
V. Алгоритмы	10
1. Цепочки символов	10
1.1. Найти количество некоторых символов в некоторой строке	10
1.2. Найти символ (последовательность символов), стоящих на некотором месте в некой строке	10
2. Исполнитель Калькулятор	11
3. Шифрование	11