

ОСНОВНОЙ ГОСУДАРСТВЕННЫЙ ЭКЗАМЕН

Д.М. Ушаков

ИНФОРМАТИКА

НОВЫЙ ПОЛНЫЙ

СПРАВОЧНИК

ДЛЯ ПОДГОТОВКИ

к ОГЭ

100
БАЛЛОВ

УДК 373:002
ББК 32.81я721
У93

Ушаков, Денис Михайлович.

У93 Информатика : Новый полный справочник для подготовки к ОГЭ / Д.М. Ушаков. — Москва: Издательство АСТ, 2021. — 413, [3] с.: ил.

ISBN 978-5-17-139269-7

(Новый полный справочник для подготовки к ОГЭ)

ISBN 978-5-17-139270-3

(Самый популярный справочник для подготовки к ОГЭ)

В справочнике представлен материал курса информатики в объёме, проверяемом на государственной итоговой аттестации.

Структура справочника соответствует официальным требованиям к контрольным измерительным материалам (КИМ) для проведения ОГЭ. Материал сгруппирован по главам, в каждой из которых изучается определённая тема курса информатики и ИКТ, проверяемая на экзамене. Помимо теоретических сведений главы включают разбор экзаменационных заданий и задачи для самостоятельной тренировки с ответами и критериями оценки в конце пособия.

Книга будет незаменимым помощником при подготовке к экзамену в формате ОГЭ, при изучении нового материала, повторении пройденных тем.

УДК 373:002

ББК 32.81я721

ISBN 978-5-17-139269-7

(Новый полный справочник для подготовки к ОГЭ)

ISBN 978-5-17-139270-3

(Самый популярный справочник для подготовки к ОГЭ)

© Ушаков Д.М., 2021

© ООО «Издательство АСТ», 2021

Содержание

1. Информация. Язык как способ представления и передачи информации: естественные и формальные языки	7
2. Формализация описания реальных объектов и процессов, моделирование объектов и процессов	9
3. Дискретная форма представления информации. Единицы измерения количества информации.	11
4. Процесс передачи информации, источник и приемник информации, сигнал, скорость передачи информации . . .	21
5. Кодирование и декодирование информации	25
5.1. Общие сведения	25
5.2. Префиксный код и дерево декодирования	26
5.3. Постфиксные неравномерные коды	33
5.4. Непрефиксные и непостфиксные неравномерные коды	36
6. Системы счисления	40
6.1. Общие сведения о системах счисления	40
6.2. Перевод из любой системы счисления в десятичную систему счисления	42
6.3. Перевод из десятичной системы счисления в любую другую систему счисления	43
6.4. Другой способ перевода в двоичную систему счисления.	45
6.5. Родственные системы счисления.	46
7. Алгоритм, свойства алгоритмов, способы записи алгоритмов	55
8. Программы. Оператор присваивания. Линейный алгоритм	87
8.1. Вступление	87
8.2. Программа. Линейный алгоритм	88
8.3. Целочисленный тип данных. Переменные.	90
9. Программирование. Логические операции	104
9.1. Вступление	104
9.2. Операции сравнения	104
9.3. Логические операции	107

10. Программирование. Условный оператор if	112
11. Программирование. Оператор цикла for	121
12. Программирование. Оператор цикла while	134
13. Программирование. Обработка последовательностей...	141
13.1. Подсчет элементов последовательности	141
13.2. Поиск максимума (минимума) последовательности	146
14. Программирование. Обработка массивов.....	158
14.1. Введение	158
14.2. Описание массива	159
14.3. Задание начальных значений элементов массива. .	159
14.3.1. Обнуление всех элементов массива	160
14.3.2. Ввод массива с клавиатуры	160
14.3.3. Задание элементов массива определенной последовательностью	160
14.4. Вывод массива на экран	161
14.5. Подсчет количества элементов массива.....	163
14.6. Вычисление суммы положительных элементов массива.....	163
14.7. Нахождение максимального элемента массива ...	164
14.8. Задание значений элементов массива числами, не связанными закономерностью	166
15. Логические значения, операции, выражения	174
15.1. Введение	174
15.2. Логические операции	175
15.2.1. Логическое И	175
15.2.2. Логическое ИЛИ.....	176
15.2.3. Логическое НЕ	176
15.3. Приоритеты логических операций	177
15.4. Вычисление логического выражения по таблице истинности.....	178
16. Разбиение задачи на подзадачи, вспомогательный алгоритм	187
16.1. Основные сведения и удобство использования ...	187
16.2. Удобство исправления ошибок.....	188
16.3. Этапы разработки программ	189
17. Обработываемые объекты: цепочки символов, числа, списки, деревья	192

17.1. Списки	192
17.2. Графы	193
17.3. Деревья	195
17.4. Таблица расстояний	196
17.5. Количество путей на графе.	215
18. Основные компоненты компьютера и их функции.	228
19. Программное обеспечение компьютера	231
20. Файлы и файловая система компьютера	233
21. Оценка количественных параметров информационных объектов.	
Объем памяти, необходимый для хранения объектов . . .	247
21.1. Кодирование текстовой информации	248
21.2. Кодирование графической информации	254
21.2.1. Кодирование цвета	254
21.2.2. Вычисление объема растрового изображения	257
21.3. Кодирование звука	259
22. Работа в текстовом процессоре	262
23. Базы данных.	278
24. Поисковые запросы	284
25. Программирование Робота	303
26. Математические инструменты, динамические (электронные) таблицы	334
27. Создание презентаций	359
28. Сетевые технологии	378
Ответы	385

1. Информация.

Язык как способ представления и передачи информации: естественные и формальные языки

Существует много различных определений термина **информация**. Автору больше всего нравится такое: «Информация — это сведения об окружающем нас мире». Если придирчиво отнестись к этому понятию, получается, что термин «информация» определен через термин «сведения». А если попытаться дать определение термину «сведения», то следует сослаться на термин «данные». В любом случае окажется, что это синонимы, то есть, термин «информация» — это некоторое первичное понятие.

Существует несколько основных действий, которые можно совершать с информацией:

- хранение информации,
- передача информации,
- обработка информации.

Для осуществления этих действий удобнее всего использовать некоторую форму/способ, при помощи которой информация хранится—передается—обрабатывается. Вероятнее всего, первичным действием при этом имеет смысл считать процесс передачи информации. Для выживания люди объединяются в группы, а для совместных действий нужно между собой договариваться. Отсюда и возникает проблема «общего языка» — средства общения между людьми.

За тысячелетия эволюции люди придумали множество различных языков, которые использовались для общения. Эти языки называются **естественными**. Они развились естественным образом в процессе эволюции. Необходимость хранить информацию привела к тому, что люди стали записывать языки разными способами: придумали отдельные буквы, которыми обозначали произносимые звуки. Эти буквы образовали *алфавит*.

Алфавит — набор символов, использующийся для записи слов языка. Со временем образовалось довольно большое количество алфавитов, которыми люди продолжают пользоваться до сих пор (вы наверняка знаете кириллицу, латиницу и греческий алфавиты). Часть народов использовала для записи отдельные слоги или даже целые слова (например, иероглифы). Для удобной обработки информации люди придумали более сложные обозначения, например, цифры для записи чисел.

При дальнейшей эволюции оказалось, что в некоторых случаях использование естественных языков неудобно. Например, через некоторое время после изобретения компьютера оказалось, что человеку общаться с компьютером при помощи естественного языка затруднительно (компьютер был не настолько умен, чтобы понимать все особенности и тонкости какого-нибудь естественного языка). И одним из решений этой проблемы было изобретение формальных языков, которые были бы понятны компьютеру и в то же время которым можно было бы относительно просто научить человека.

Например, к этой категории относятся **языки программирования**. Это формальные языки с весьма ограниченным набором слов, которые можно перевести на язык машинных кодов (в котором работает компьютер), но при этом стараются подобрать такие слова, чтобы общение на них было бы понятно человеку.

Программы-переводчики с языка программирования на язык машинных кодов, называются **трансляторами**. Если человек умеет записывать свои мысли на этом формальном языке (языке программирования), то он имеет возможность общения с компьютером. Такие специалисты называются **программистами**.

2. Формализация описания реальных объектов и процессов, моделирование объектов и процессов

Для обработки данных на компьютере все процессы, объекты, явления, должны быть записаны в виде чисел, так как компьютеры умеют работать только с числами. Все остальные виды информации, которые обрабатывает компьютер, преобразовываются в числовой вид. Существует даже специальный термин — **оцифровка**. Он означает, что информация преобразовывается в числа и может после этого обрабатываться, передаваться и храниться при помощи компьютеров. Люди придумали множество устройств, которые умеют делать оцифровку. Наиболее часто используемые из них — это цифровые фотоаппараты и камеры, сканеры, клавиатуры, мыши, планшеты.

Итак, все данные, которые компьютер обрабатывает, записывают в виде чисел. Иногда это достаточно очевидно. Например, для обработки информации о том, в каком месте экрана вы щелкнули мышью, сохраняется местоположение этой точки на экране (координаты по осям X и Y). Когда вы нажимаете клавишу на клавиатуре, фиксируется номер (код) этой клавиши. По нему определяется, какой символ соответствует клавише, и номер этого символа запоминается. При хранении величин, которые уже имеют числовое значение, это значение просто переводится в двоичную систему счисления (с некоторой точностью, если оно не целое).

Кроме хранения различных величин, при обработке объектов и процессов на компьютере возникает также задача компьютерного описания обрабатываемых объектов и процессов. Основной прием для этого аналогичен тому, который мы только что описали для хранения различных величин. Люди описывают состояние объектов или процессов, перечисляя те состояния или значения характеристик, которыми объекты или процессы могут обладать. После этого составляют таблицу всех возможных значений, нумеруют их и хранят номера в двоичном виде.

Рассмотрим пример, на котором покажем способ компьютерного хранения векторного графического изображения, то есть, изображения, описываемого набором объектов. Например, будем считать, что изображение может состоять только из прямых линий, прямоугольников и эллипсов. Линии, не являющиеся прямыми, будем хранить при помощи последовательности большого количества маленьких прямых отрезков.

Замечание. Для хранения кривых линий придумана специальная технология, называемая **кривые Безье**.

Итак, в нашем примере изображение состоит из объектов трех видов: прямой отрезок, прямоугольник, эллипс. Пронумеруем эти виды объектов: отрезок — 0, прямоугольник — 1, эллипс — 2. Расположение каждого из этих объектов можно описать при помощи двух точек. Местоположение точек будем хранить как их координаты на плоскости (два числа — проекция на ось X и проекция на ось Y). Для отрезка эти точки задаются координатами, из которой и в которую он проведен. Для прямоугольника — левый верхний и правый нижний угол. Для эллипса — координаты левого верхнего и правого нижнего углов того прямоугольника, в который вписан эллипс. Если мы хотим хранить прямоугольники и эллипсы, не параллельные осям координат, то для объектов хранится еще угол поворота объекта относительно своего центра. Получается, что для хранения каждого объекта достаточно хранить 6 чисел:

- целое число от 0 до 2 (вид объекта),
- две пары чисел, хранящих координаты точек на плоскости,
- угол поворота (например, в градусах).

Все эти величины (кроме номера вида объекта) могут быть целыми или вещественными в зависимости от того, с какой точностью требуется их хранение.

Исходя из подобных соображений описывается любой объект или процесс реального мира. Например, для хранения в компьютере модели трехмерного объекта можно разбить внешнюю поверхность объекта на связанные друг с другом треугольники. Каждый треугольник хранить при помощи трех троек чисел (координаты вершин треугольника в трехмерном пространстве). Для хранения правдоподобного изображения объекта для каждого треугольника хранится изображение (**текстура**), которое на треугольник будет наложено. О хранении растровых изображений рассказано в одном из последующих разделов.

Например, при хранении процесса движения лифта достаточно перечислить те действия, которые лифт должен уметь делать: не двигаться, открывать двери, закрывать двери, подниматься, опускаться, — пронумеровав действия (от 0 до 4) и присвоив им соответствующие коды-номера. Также, вероятно, потребуется хранить информацию о том, где в текущий момент находится лифт (на каком этаже или между ними), на какой этаж он должен ехать, закрыты у него двери или открыты. Пронумеровав отдельно все состояния, независимые от других, получим компьютерную модель, которую можно использовать для написания программы управления лифтом.

3. Дискретная форма представления информации.

Единицы измерения количества информации

При обработке различных данных люди столкнулись со следующей проблемой: компьютеры обрабатывают информацию в числовом виде, т.е. в виде чисел. При этом числа хранятся либо в целом виде, либо с некоторой точностью, которая зависит от количества ячеек памяти (**бит**), выделенное для хранения числа. Объекты Окружающего нас мира не обладают «целочисленностью» (большинство величин, которые нас окружают, не являются целыми). Целыми числами мы подсчитываем только количество чего-нибудь. Практически все остальные величины окружающего нас мира — **вещественные**. Например, размеры объектов (длина, ширина, высота) могут считаться целыми только с точки зрения цены деления линейки, которой эти они измерены. То же самое касается и большинства других величин: время, масса, скорость, расстояние, сила тока, напряжение, температура.

Измерительные приборы выдают нам значение с некоторой степенью точности (**погрешности**). При обработке величин размеров на компьютере приходится сталкиваться еще с одной проблемой — потерей точности. Эта проблема возникает при *дискретизации*. **Дискретизация** — запись измеряемой/хранимой величины при помощи определенного, ограниченного количества ячеек памяти (*бит*).

Люди используют три основных способа дискретизации хранимых данных (записи данных при помощи определенного количества двоичных разрядов).

Первый способ. Каждому варианту данных ставится в соответствие своя комбинация двоичных разрядов. Этот способ используется в случае, когда вариантов хранимых данных мало и их все можно выписать и «пронумеровать». Как правило, проще даже не сопоставлять каждому варианту свое отдельное «хитрое» значение двоичных разрядов, а просто выписать весь список вариантов данных в определенном порядке и пронумеровать каждый элемент списка, начиная с нуля. Двоичные значения номеров списка нужно хранить.

Второй способ. Каждому значению сопоставляется определенное целое число. В случае, если хранимые величины при этом не являются целыми числами (например, 6,73 кг), этим способом все равно можно воспользоваться.

Например, пусть мы хотим обрабатывать на компьютере массу продуктов, которые измеряются на весах. Значение массы электронные весы будут хранить в ячейке памяти определенного размера. Размер ячейки памяти не может быть бесконечным. Ничего бесконечного компьютеры обрабатывать не в состоянии. Значит, у ячейки будет определенное ограничение на то количество двоичных разрядов, которое будет для нее выделено. Пусть наибольшая масса, которую мы планируем взвешивать и обрабатывать, составляет 8 кг, то есть от 0 до 8 кг. От того, сколько двоичных разрядов будет выделено внутри электронных весов для хранения массы продукта, зависит, с какой точностью они будут обрабатываться. Пусть, для хранения массы мы выделим всего один бит памяти. Один бит может принимать только два возможных значения — 0 и 1. А мы, напоминая, планируем взвешивать продукты массой от 0 до 8 кг. Значит, любой продукт, который будет иметь массу меньше 4 кг, мы будем кодировать значением 0, а для остальных значений массы — 1.

При обратном преобразовании (из бита в килограммы) возникает вопрос, каким значениям в килограммах соответствуют значения, хранимые в битах. Например, значение 0 — это сколько килограммов? Это зависит от того, хотим ли мы точно измерять граничные значения массы (0 и 8 кг) или более точно измерять массы в среднем. Например, можно считать, что 0 — это 0 кг, а 1 — это 8 кг. Тогда эти (граничные) значения будут храниться точно, а остальные — не очень. В худшем случае (например, при массе 4 кг) погрешность дискретизации будет достигать 4 кг.

Можно уменьшить погрешность вдвое, если принять за точные значения 0 и 1 не крайние значения диапазона, а средние в каждом. То есть, считать, что хранимому нулю соответствует масса 2 кг, а хранимой единице — масса 6 кг. Тогда наибольшая погрешность будет происходить при массе 0, 4 или 8 кг и будет составлять 2 кг.

Очевидно, что хранить значение массы при помощи только одного бита (двоичного разряда) никто не будет из-за возникающей погрешности. Но этот пример позволяет лучше понять проблему потери точности при дискретизации.

Предположим, что для хранения массы мы используем два двоичных разряда. Тогда нам доступно всего 4 различных дискретных значений (0, 1, 2 и 3). Если уменьшать потерю точности, этим двоичным значениям будут соответствовать значения массы 1, 3, 5 и 7 кг соответственно. То есть, если измеряемая масса составляет от 0 до 2 кг, храним это как дискретное число 0 (00 двоичное). При обратном преобразовании все эти значения массы будут считаться как 1 кг (наибольшая погрешность — 1 кг). Аналогич-

но, от 2 до 4 кг храним как 1 (01 двоичное) и считаем как 3 кг, от 4 до 6 кг храним как 2 (10 двоичное) и считаем как 5 кг, от 6 до 8 кг храним как 3 (11 двоичное) и считаем как 7 кг.

Нетрудно заметить, что при добавлении одного разряда для хранения массы, погрешность дискретизации уменьшается в два раза. Можем вычислить, какое наименьшее количество двоичных разрядов требуется использовать в данном примере, чтобы потеря точности при дискретизации была не более одного грамма:

Храним, бит	1	2	3	4	5	6	7	8	9	10	11	12
Точность, г	2000	1000	500	250	125	62,5	31,25	15,625	7,8125	3,90625	1,953	0,977

Получается, что для хранения массы от 0 до 8 кг с точностью до 1 грамма требуется не менее 12 бит.

Обратите внимание, что точность 1 грамм означает, что наибольшая ошибка измерений будет составлять не более 1 грамма. Если же требуется хранить значение массы с шагом 1 грамм, точность следует сделать равной 0,5 грамма, то есть, использовать еще один бит (всего 13). Это значение (13 бит) можно получить другим способом: посчитать, какое количество различных значений должны уметь хранить весы для измерений с шагом 1 грамм. Это 8001 (целое значение массы в граммах — от 0 до 8000). Ячейка памяти, состоящая из k бит, может хранить не более чем 2^k различных значений (от 0 до $2^k - 1$). Найдем наименьшее k такое, что $2^k \geq 8001$. Оно равно 13 ($2^{13} = 8192$), то есть, 13 бит.

Замечание.

Описанный нами только что способ называется записью вещественных чисел с **фиксированной запятой**. В этом способе погрешность дискретизации одинакова как для больших чисел, так и для маленьких. В случае с весами в магазине, где человек платит за вес с точностью до грамма, это актуально.

Третий способ — использование хранения вещественных чисел с **плавающей запятой**. Его принципиальное отличие от хранения чисел с фиксированной запятой состоит в том, что у числа всегда хранится какое-то гарантированное количество **значащих цифр**. Так, для очень маленьких значений массы (например, 0,00356 грамм) можно обеспечить такое же количество значащих цифр, что и для больших значений массы (например, 3 560 000 кг). Это нужно тогда, когда важна относительная погрешность измерений. Например, измеряя массу планеты, не очень важно, сколько

именно десятков или сотен килограммов она составляет, а, измеряя массу молекулы, точности измерения 1 грамм будет явно недостаточно. Изучить описание того, как устроено хранение вещественных чисел в плавающей запятой, мы предлагаем самостоятельно тем, кого это заинтересовало.

Рассмотрим теперь вопрос **измерения количества информации**.

Для измерения количества информации придуманы специальные **единицы измерения информации**.

Основная — *один бит*.

Один бит — это количество информации, уменьшающее неопределенность в два раза.

Что такое неопределенность? Проще всего это понимать как выбор из нескольких вариантов.

Например, Вася должен угадать, в какой из восьми одинаковых коробок, стоящих в ряд, лежит конфета.

Ему говорят, что конфета лежит в одной из левых 4-х коробок.

У Васи был выбор из 8-ми коробок, а остался выбор только их 4-х коробок, то есть, количество вариантов уменьшилось в 2 раза. Неопределенность уменьшилась в 2 раза. Значит, Васе сообщили 1 бит информации. Чтобы прийти к определенности (в какой конкретно коробке лежит конфета), нужно, чтобы остался только один вариант. Значит, нужно оставшееся количество вариантов поделить на 2 (останется 2 варианта, Вася получил еще один бит информации). А затем оставшееся количество вариантов поделить еще раз на 2 (останется 1 коробка, Вася получил еще один бит информации).

Итак, если изначально у нас был выбор из 8-ми коробок, и мы узнали, что конфета лежит в некоей конкретной коробке, мы должны были эту неопределенность (8 коробок) трижды поделить на 2, чтобы осталась только одна коробка. То есть, получить 3 бита информации.

Из этих соображений выведем основную **формулу для вычисления количества информации**.

Пусть у нас есть выбор из N одинаковых объектов. Мы должны выбрать один.

Будем делить количество объектов на 2 (уменьшать неопределенность в 2 раза) столько раз, сколько нужно для получения определенности (чтобы остался только один объект).

Получаем формулу:

$$N/2^i = 1.$$

Заметим, что ровно 1 можно получить только в том случае, если число N является степенью числа 2.

Если же это не так (а это часто не так), нужно понимать, что **бит — минимальная единица измерения информации и она не**

бывает не целой. То есть, если в процессе деления на 2 будут получаться не целые числа, нужно округлять до ближайшего целого (вверх).

Например, при исходных 5 объектах, нужно делить так же 3 раза, как если бы их было 8 ($5/2 = 2,5$; округляем до 3, $3/2 = 1,5$, округляем до 2; $2/2 = 1$. Всего делили 3 раза). Можно это понимать и по-другому: следует считать, что определенность — это когда возможных вариантов остается не больше одного. То есть, нашу формулу правильнее было бы записать так: $N / 2^i \leq 1$.

Если перемножить обе части неравенства на 2^i , получим **формулу Хартли**:

$$2^i \geq N,$$

где N — количество равновероятных событий, i — количество информации (бит) в сообщении об одном таком событии, при этом i — минимальное целое число.

Заметим, что если у нас есть выбор только из одного варианта, то количество информации в сообщении о таком событии равно нулю (сообщение о событии, которое происходит всегда, не несет в себе информации).

Пример экзаменационной задачи

В магазине продается 30 одинаковых упаковок шоколадных шариков. Известно, что в одной из этих упаковок находится приз. Вася покупает одну упаковку. Какое количество информации содержится в сообщении о том, что приз находится именно в упаковке, купленной Васей?

Решение.

Первый способ. Анализируем исходные данные. Так как все 30 упаковок одинаковые и приз находится только в одной из них, то сообщение, что приз находится именно в упаковке, купленной Васей, — это одно из 30 равновероятных событий. То есть, мы применяем формулу Хартли: $2^i \geq N$. Здесь N — количество равновероятных событий (30). Нужно подобрать наименьшее целое i такое, что $2^i \geq 30$. Если Вы не знаете наизусть степени числа 2 (что весьма полезно для подготовки и сдачи ОГЭ по информатике), предлагаем подбирать эти степени последовательно, начиная с первой:

$2^1 = 2$. $2 \geq 30$? Нет. Берем следующую степень (умножаем на 2).

$2^2 = 2 \cdot 2 = 4$. $4 \geq 30$? Нет. Берем следующую степень (умножаем на 2).

$2^3 = 4 \cdot 2 = 8$. $8 \geq 30$? Нет. Берем следующую степень (умножаем на 2).

$2^4 = 8 \cdot 2 = 16$. $16 \geq 30$? Нет. Берем следующую степень (домножаем на 2).

$2^5 = 16 \cdot 2 = 32$. $32 \geq 30$? Да.

Получилось, что наименьшее i , при котором $2^i \geq 30$, — это число 5. Ответ — 5. Не забываем, что в формуле Хартли количества информации измеряется в битах.

Ответ: 5 бит.

Другой вариант нахождения нужного нам числа 5 — делить исходное число 30 на 2 до тех пор, пока не получится число, меньшее или равное 1:

$30/2 = 15$. $15 \leq 1$? Нет. Продолжаем.

$15/2 = 7,5$. Округляем до 8. $8 \leq 1$? Нет. Продолжаем.

$8/2 = 4$. $4 \leq 1$? Нет. Продолжаем.

$4/2 = 2$. $2 \leq 1$? Нет. Продолжаем.

$2/2 = 1$. $1 \leq 1$? Да.

Подсчитываем количество раз, которое мы делили на 2. Получаем 5.

Ответ: 5 бит.

Быстрый вариант: если мы наизусть знаем первые степени числа 2 (рекомендуется знать первые 10 степеней числа 2), то мы можем быстро понять, что $16 \geq 30$? Нет, но $32 \geq 30$? Да. То есть, наименьшая степень числа 2, которая больше или равна исходному числу 30, это число 32. Это 2 в степени 5.

Ответ: 5 бит.

Задачи для тренировки

Задача 3.1. Петя подбросил монетку (она могла упасть вверх орлом или решкой). Выпал орел. Какое количество информации получил Петя?

Задача 3.2. Перед выходом в полуфинал соревновались 8 команд. В финале победила только одна команда. Какое количество информации в сообщении о том, что победила команда под номером 3?

Задача 3.3. Петя бросил 6-гранный кубик. На кубике выпало число 4. Какое количество информации получил Петя?

Задача 3.4. Маша отметила карандашом одну из страниц 100-страничной книги и попросила Васю отгадать номер страницы. Вася может задавать вопросы, на которые можно ответить

только «да» или «нет». Какое минимальное количество вопросов должен задать Вася (и получить на них, конечно же, ответ), чтобы гарантированно узнать, какую страницу отметила карандашом Маша?

Еще одно определение бита: это количество информации в сообщении, которое может принимать только два возможных значения. Например, «да» или «нет».

Для обозначения возможных значений одного бита обычно используют цифры 0 и 1.

Если сообщение состоит из нескольких символов и при этом эти символы равновероятны (т.е. нельзя заранее сказать, что какой-то символ сообщения может появиться чаще какого-нибудь другого), то количество информации в таком сообщении может быть вычислено по формуле: $I = k \cdot i$, где k — количество равновероятных символов в сообщении, i — количество информации (бит) в одном таком символе, I — количество информации (бит) во всем сообщении.

Пример экзаменационной задачи

В племени Мума-Тума в языке используется всего 64 различных слова. Один из членов племени говорит другому фразу, состоящую из 100 слов. Какое количество информации он сообщил?

Решение.

Нас просят найти количество информации в сообщении, то есть, нужно будет использовать формулу $I = k \cdot i$. Но в этой формуле необходимо знать количество информации в одном символе и количество символов в сообщении.

Количество информации в одном символе нам не дано, однако, его можно постараться найти при помощи формулы Хартли: $2^i \geq N$.

Остается понять, какое из чисел — 64 и 100 — является числом равновероятных событий N , а какое — количеством символов в сообщении k . Анализируем условие и понимаем, что сообщение — это фраза, которую один член племени говорит другому. По условию фраза состоит из 100 слов. Но ведь в формуле: $I = k \cdot i$ число k — это количество символов, которое нам известно по условию.

В данном случае нужно понять, что если племя общается между собой при помощи всего 64 слов, то они не разделяют слова на буквы, а используют каждое слово как отдельный, неделимый элемент общения. То есть, их слова — это и есть то, что мы при

анализе сообщений называем символами. Значит, количество символов в сообщении $k = 100$.

Методом исключения получаем, что 64 — это количество равновероятных событий N . Действительно, если люди племени используют в разговоре всего 64 различных слова, то каждое из этих слов и есть одно из равновероятных событий, которые мы подсчитываем в формуле Хартли.

Подставляем подобранные величины в формулы.

Сначала по формуле Хартли найдем количество информации в одном слове (символе): $2^i \geq 64$. Минимальное i , при котором неравенство выполняется, равно 6 ($2^6 = 64$).

Получаем, что в одном слове племени содержится 6 бит информации. Подставляем это в формулу: $I = k \cdot i$.

$$I = 100 \cdot 6 = 600 \text{ бит.}$$

Ответ: 600 бит.

Задачи для тренировки

Задача 3.5. Вася пишет в тетрадь последовательность из 30 букв. При этом буквы в последовательности могут быть только А, Б, В, Г (4 различные символа). Какое количество информации содержит последовательность, которую записал в тетрадь Вася?

Замечание. Для измерения количества информации вовсе не обязательно, чтобы сообщение было хоть каким-нибудь образом осмысленным.

Задача 3.6. Малыши договорились использовать для общения только 16 различных слов. Один из них произносит фразу, состоящую из 5-ти слов. Какое количество информации он сообщает другим малышам?

Возможно, Вы обратили внимание, что 1 бит — достаточно маленькая единица измерения информации. Для большего удобства люди придумали более крупные.

$$1 \text{ байт} = 8 \text{ бит.}$$

$$1 \text{ Кбайт} = 1024 \text{ байт} (= 2^{10} \text{ байт}).$$

$$1 \text{ Мбайт} = 1024 \text{ Кбайт} (= 2^{20} \text{ байт}).$$

$$1 \text{ Кбит} = 1024 \text{ бит.}$$

$$1 \text{ Мбит} = 1024 \text{ Кбит.}$$

Приставки К и М, которые мы используем применительно к терминам бит и байт традиционно считаются как $1024 (2^{10})$ и $1024^2 (2^{20})$, а не как $1000 (10^3)$ и $1\,000\,000 (10^6)$, как это принято для дру-

гих единиц измерения. Исторически сложилось, что эти приставки принято читать как «кило» и «мега». Но при этом возникала путаница, потому что они обозначают 10^3 и 10^6 соответственно.

Чтобы избежать этой путаницы, приняли решение: для обозначения множителей 2^{10} и 2^{20} использовать термины «киби» и «миби». То есть, 1 килобайт = 1000 байт, а 1 кибибайт = 1024 байт.

Однако для приставок К и М такого правила формально введено не было. Поэтому в учебниках по информатике, на экзаменах по информатике и вообще в ИТ-среде считается, что К — это 2^{10} , а М — это 2^{20} (применительно к битам и байтам).

Пример экзаменационной задачи

Какое количество байт в одном Кбите?

Решение.

У нас имеется 1 Кбит. Нужно перевести это в байты.

Составим дробь. В числителе запишем исходное количество информации. В знаменателе — то количество информации, которое нам нужно получить: 1 Кбит/1 байт.

Чтобы можно было делать действия с этими величинами, нужно привести их к единой размерности. Проще всего все сводить к самой маленькой величине — к битам.

Достаточно только выучить, сколько бит в каждой из 5-ти единиц измерения информации, используемых в учебных задачах:

$$1 \text{ Кбит} = 1024 \text{ бит} = 2^{10} \text{ бит}$$

$$1 \text{ Мбит} = 1024 \text{ Кбит} = 2^{20} \text{ бит}$$

$$1 \text{ байт} = 8 \text{ бит} = 2^3 \text{ бит}$$

$$1 \text{ Кбайт} = 1024 \text{ байт} = 1024 \cdot 8 \text{ бит} = 2^{13} \text{ бит}$$

$$1 \text{ Мбайт} = 1024 \text{ Кбайт} = 2^{23} \text{ бит}$$

Подставим эти величины в нашу дробь и сократим степени двойки: $2^{10}/2^3 = 2^7 = 128$.

Ответ: 128 байт.

Задачи для тренировки

Задача 3.7. Какое количество Кбайт в одном Мбите?

Задача 3.8. Какое количество Мбит в 256 Кбайт?

В использованных ранее определениях часто используется термин «**равновероятный**» (*равновероятное событие, равновероятный символ*). Это можно понимать так, что ни про одно событие (ни про один символ) нельзя заранее сказать, что частота его появления больше, чем у какого-нибудь другого события (символа).

В случае, когда это свойство не выполняется, применяют другие способы вычисления количества информации. Они, как правило, связаны с понятием *энтропии* и знанием термина «*логарифм*» в математике.

В рассматриваемом нами курсе информатики до 9-го класса включительно эти термины не считаются изученными и поэтому не используются. То есть, на ОГЭ Вам не встретятся задачи, в которых события (символы) не будут равновероятными.

Наиболее известный пример такого (неравновероятного) события — встретить на улице динозавра. Так как это событие очень и очень маловероятно, то неверным будет считать, что количество информации в сообщении «Я встретил сегодня на улице динозавра» равно одному биту! Действительно, можно предполагать, что ответ на вопрос «Встретил ли ты сегодня на улице динозавра?» имеет только два возможных варианта ответа — «да» и «нет» — и поэтому несет в себе только один бит информации. Но это верно только для равновероятных вариантов ответа. В данном же случае ответ «нет» встречается гораздо чаще ответа «да», и поэтому к нему не может применяться это определение бита.

Задача 3.9. В корзине лежит 16 яблок. Известно, что только одно из них червивое. Какое количество информации содержит сообщение, что Петя взял из корзины червивое яблоко?

Задача 3.10. На автобусной остановке останавливаются автобусы 6-ти различных маршрутов (10, 28, 29, 107, 458, 502). Какое количество информации содержит в себе сообщение, что только что подъехавший к остановке автобус — номер 28? (Считать, что автобусы приходят к остановке с одинаковым интервалом и мы ничего не знаем относительно того, какие автобусы уже приходили ранее).

4. Процесс передачи информации, источник и приемник информации, сигнал, скорость передачи информации

Существует три основных **информационных процесса** — *хранение, передача и обработка информации*.

При **передаче** информации принято различать следующие понятия. **Источник информации** — тот объект, который информацию передает (из которого информация исходит). **Приемник информации** — тот объект, который информацию получает (в который информация входит).

Чтобы передать информацию от одного объекта к другому, необходимо наличие **канала связи**. Этим каналом может быть, например, воздух (именно так люди обмениваются информацией при разговоре), электрические провода (например, телефонный разговор), просто любое пространство (если это беспроводная передача данных посредством радиоволны, например). По каналу связи информация передается в виде **сигналов**. Ими могут быть, например, звуки (при обычном разговоре), электрические импульсы (по телефонной линии), электромагнитная волна (при сотовой связи, например) и многое другое. Соответственно, возникает вопрос о *скорости передачи информации*.

Скорость передачи информации — это количество информации, переданное за единицу времени. Принято измерять количество информации в *битах в секунду*. То есть, чтобы посчитать скорость передачи информации, нужно поделить количество переданной информации (в битах) на время, которое эта информация передавалась (в секундах). Получаем формулу:

$$v = I / t,$$

где v — скорость передачи информации,

I — количество информации в сообщении (в битах),

t — время передачи сообщения (в секундах).

Пример экзаменационной задачи 1

Вася передает Пете сообщение, состоящее из 200 символов в течение 10 секунд. При этом каждый символ несет в себе 5 бит. Определите скорость передачи информации.

Решение.

Для вычисления скорости передачи информации нам нужно знать количество передаваемой информации и время передачи

информации. Время передачи мы знаем — 10 секунд. Посчитаем объем передаваемой информации.

Нам дано количество передаваемых символов и количество информации в каждом символе. Воспользуемся формулой:

$$I = k \cdot i.$$

Количество символов равно 200. Это k . Количество информации в одном символе равно 5 бит. Это i . Подставим это в формулу: $I = 200 \cdot 5 = 1000$ бит. Теперь можно посчитать скорость передачи: $v = 1000 / 10 = 100$ бит в секунду.

Ответ: 100 бит в секунду.

Пример экзаменационной задачи 2

Маша передает Даше сообщение, состоящее из 60 вывешенных в ряд флажков в течение полминуты. Известно, что в сообщении используется только 16 различных флажков. Какова скорость передачи информации?

Решение.

Для вычисления скорости передачи информации нужно знать количество информации в сообщении и время передачи сообщения ($v = I/t$). Время передачи нам дано — это полминуты, т.е. 30 секунд. Будем искать количество информации в сообщении. Для этого нужно знать количество символов в сообщении и количество информации в одном символе ($I = k \cdot i$). Попробуем понять, что в данном случае является символами сообщения: что последовательно передается в качестве сообщения. В данном случае — это флажки. То есть, один флажок — один символ. Про них нам известно количество 60 и количество 16.

Что из этого k — количество символов в сообщении? Это то, сколько символов передается в ряд. Так как в ряд висит 60 флажков, то $k = 60$.

Еще нужно знать количество информации в одном символе. Его можно найти по формуле Хартли ($2^i \geq N$). Так как N — количество равновероятных событий, то это и есть 16 — количество различных флажков. Подставляем: $2^i \geq 16 \Rightarrow i = 4$ (количество информации в одном символе-флажке). Подставляем это в формулу $I = k \cdot i = 60 \cdot 4 = 240$ бит (количество информации в сообщении). Подставляем это в формулу $v = I/t = 240 / 30 = 8$.

Ответ: 8 бит в секунду.

Задачи для тренировки

Задача 4.1. От одного компьютера к другому передается 1000 бит в течение 25 секунд. Какова скорость передачи информации?

Задача 4.2. По каналу связи передается 300 символов в течение 40 секунд. Известно, что в одном символе содержится 8 бит. Какова скорость передачи информации?

Задача 4.3. Из одного города в другой передают письмо на бурбарском языке, содержащее 600 слов. Письмо передается за 20 минут. Известно, что в бурбарском языке имеется всего 256 различных слов. Какова скорость передачи информации?

Пример экзаменационной задачи 3

Файл размером 2000 Кбайт передается через некоторое соединение в течение 30 секунд. Определите размер файла (в Кбайт), который можно передать через это соединение за 12 секунд. В ответе укажите одно число — размер файла в Кбайт. Единицы измерения писать не нужно.

Решение.

Формула для вычисления скорости передачи информации: $v = I/t$. Определим скорость передачи через указанное соединение: $v = 2000 \text{ Кбайт} / 30 \text{ секунд} = 200/3 \text{ Кбайт/сек.}$

Нас спрашивают, какого размера файл можно передать через это соединение (т.е., с этой же скоростью) за 12 секунд. Из формулы $v = I/t$ выразим I : $I = v \cdot t$. Подставим в эту формулу имеющиеся у нас величины: $I = (200/3) \cdot 12 = 800 \text{ Кбайт.}$

Ответ: 800.

Другое решение задачи.

Из формулы скорости передачи информации $v = I/t$ видно, что величины v и I прямо пропорциональны друг другу (увеличение в n раз одной величины приведет к увеличению в такое же количество раз другой величины). Составим пропорцию:

2000 Кбайт — 30 секунд

X Кбайт — 12 секунд

Отсюда выразим X :

$X = 2000 \text{ Кбайт} \cdot 12 \text{ секунд} / 30 \text{ секунд} = 800 \text{ Кбайт.}$

Ответ: 800.

Задачи для тренировки

Задача 4.4. Файл размером 600 Кбайт передается через некоторое соединение в течение 30 секунд. Определите размер файла (в Кбайт), который можно передать через это соединение за 50 секунд.

В ответе укажите одно число — размер файла в Кбайт. Единицы измерения писать не нужно.

Задача 4.5. Файл размером 256 Кбайт передается через некоторое соединение со скоростью 2048 бит в секунду. Определите размер файла (в Кбайт), который можно передать за то же время через другое соединение со скоростью 128 бит в секунду.

В ответе укажите одно число — размер файла в Кбайт. Единицы измерения писать не нужно.

Задача 4.6. Через некоторое соединение в течение 40 секунд со скоростью 256 бит в секунду передается некоторый файл. Определите, за какое время можно передать этот же файл через другое соединение со скоростью 1024 бит в секунду.

В ответе укажите одно число — время передачи файла в секундах. Единицы измерения писать не нужно.

Задача 4.7. Через некоторое соединение в течение 150 секунд со скоростью 512 бит в секунду передается некоторый файл. Определите, с какой скоростью должно работать соединение, чтобы этот же файл можно было передать за 60 секунд.

В ответе укажите одно число — скорость передачи файла в битах в секунду. Единицы измерения писать не нужно.

5. Кодирование и декодирование информации

5.1. Общие сведения

При передаче информации часто возникает необходимость в особой форме кодирования информации. Как правило, это делается для ускорения передачи информации или для исправления ошибок, возникающих в процессе передачи.

Основной способ кодирования при передаче информации — такой же, как и при представлении информации в компьютерном виде. То есть, каждому передаваемому символу ставится в соответствие определенный код. Этот код известен передатчику (его принято называть **источник**) и **приемнику**.

Источник преобразует передаваемые символы в соответствующие коды (кодирует информацию), коды передаются по каналу связи в виде сигналов, после чего приемник преобразует полученные коды в символы (декодирует информацию). Последовательность сигналов, которая соответствует передаваемому символу, называется **кодовым словом**. То есть, код, используемый при передаче для кодирования и декодирования информации, состоит из кодовых слов.

Коды, которые присваиваются символам для передачи, могут быть *равномерными* или *неравномерными*.

Равномерным называется код, у которого длина каждого кодового слова одинаковая. **Неравномерным** называется код, у которого хотя бы у двух кодовых слов длины различаются.

В обоих случаях **процесс кодирования** передаваемого сообщения одинаков — каждому передаваемому символу по очереди из кодовой таблицы выбирается его кодовое слово, и это кодовое слово передается по каналу связи.

Пример равномерного кода:

Символ	А	Б	В	Г	Д
Код символа	000	001	010	011	100

Пример кодирования равномерным кодом

Закодируем приведенным выше равномерным кодом последовательность символов АВГАБ. Будем последовательно записывать коды символов в том порядке, в котором они даны в кодируемой последовательности: 000 010 011 000 001. Пробелы мы

поставили только для понятности того, что мы делаем. В действительности в результате кодирования получается слитная последовательность. То есть, 0000100110000001.

Если используется *равномерный* код, процесс декодирования происходит так же просто — приемник знает длины кодовых слов, отсчитывает из канала связи количество сигналов, равное длине кодового слова, «заглядывает» в таблицу соответствия передаваемых символов и кодовых слов и извлекает оттуда соответствующий символ.

То есть, получив на вход последовательность 0000100110000001, приемник однозначно разбивает ее на кодовые слова одинаковой длины (3) 000 010 011 000 001 и записывает вместо каждого кодового слова соответствующий этому кодовому слову символ: АВГАБ.

Пример неравномерного кода:

Символ	А	Б	В	Г	Д
Код символа	000	001	00	1	10

Пример кодирования неравномерным кодом.

Закодируем приведенным выше неравномерным кодом последовательность символов АВГАБ.

Принцип кодирования точно такой же — последовательно для каждого кодируемого символа из таблицы извлекается кодовое слово и записывается в том же порядке. В данном случае получается: 000 00 1 000 001. Как и в прошлый раз, мы поставили пробелы для лучшего понимания, как происходил процесс. В действительности получается сплошная последовательность: 000001000001.

Основная проблема, возникающая при *декодировании неравномерного* кода, — это как разбить входящую последовательность на кодовые слова. Ведь теперь нам неизвестно, на последовательности какой длины нужно разбить последовательность (длины кодовых слов у разных символов разные).

5.2. Префиксный код и дерево декодирования

Самый простой способ, при котором эта задача решается просто, — использовать для передачи *префиксный код*. Для **префиксного кода** должно выполняться следующее правило: **никакое кодовое слово не должно быть началом никакого другого кодового слова**.