

О. Б. Богомолова



ИНФОРМАТИКА

НОВЫЙ
ПОЛНЫЙ

СПРАВОЧНИК

для
ПОДГОТОВКИ

к **ЕГЭ**

! САМЫЙ
ПОПУЛЯРНЫЙ
СПРАВОЧНИК
для подготовки
к ЭКЗАМЕНАМ

100
БАЛЛОВ

О. Б. Богомолова

ИНФОРМАТИКА

НОВЫЙ ПОЛНЫЙ

СПРАВОЧНИК

ДЛЯ ПОДГОТОВКИ

К **ЕГЭ**

Москва
АСТ
2023

УДК 373:002
ББК 32.81я721
Б74

Богомолова, Ольга Борисовна.

Б74 Информатика : Новый полный справочник для подготовки к ЕГЭ / О.Б. Богомолова. — Москва : Издательство АСТ, 2023. — 444, [4] с. — (Самый популярный справочник для подготовки к ЕГЭ).

ISBN 978-5-17-149711-8

Справочник поможет школьнику освежить в памяти основной теоретический материал по всему курсу информатики за 7–11 классы, ознакомиться с принципами решения типовых задач, соответствующих демонстрационному варианту и спецификации ЕГЭ, и подготовиться к экзамену, проводимому в компьютерной форме.

Для школьников, учителей информатики и методистов.

**УДК 373:002
ББК 32.81я721**

ISBN 978-5-17-149711-8

© Богомолова О.Б., 2023
© ООО «Издательство АСТ», 2023

Содержание

Предисловие	7
РАЗДЕЛ 1	
Информация. Измерение информации.	
Кодирование информации	10
Измерение количества информации	10
<i>Конспект</i>	<i>10</i>
<i>Разбор типовых задач</i>	<i>18</i>
Равномерные и неравномерные двоичные коды ..	26
<i>Конспект</i>	<i>26</i>
<i>Разбор типовых задач</i>	<i>29</i>
Передача информации по коммуникационным каналам.	38
<i>Конспект</i>	<i>38</i>
<i>Разбор типовых задач</i>	<i>40</i>
РАЗДЕЛ 2	
Моделирование и компьютерный эксперимент	45
Задачи на графах	45
<i>Конспект</i>	<i>45</i>
<i>Разбор типовых задач</i>	<i>50</i>
РАЗДЕЛ 3	
Системы счисления	67
Двоичная, восьмеричная, шестнадцатеричная системы счисления. Арифметика в указанных системах счисления	67
<i>Конспект</i>	<i>67</i>
<i>Разбор типовых задач</i>	<i>77</i>
Задачи на кодирование, решаемые с применением недесятичных систем счисления	
<i>Конспект</i>	<i>93</i>
<i>Разбор типовых задач</i>	<i>99</i>

РАЗДЕЛ 4

Основы логики 111

Таблицы истинности. Законы алгебры логики.

Задачи, решаемые с использованием

таблиц истинности 111

Конспект 111

Разбор типовых задач 116

РАЗДЕЛ 5

Элементы теории алгоритмов 126

Анализ работы автомата, формирующего число

по заданным правилам 126

Конспект 126

Разбор типовых задач 127

Исполнители:

Робот, Чертёжник, Редактор 143

Конспект 143

Разбор типовых задач 145

Числовые исполнители 165

Конспект 165

Разбор типовых задач 171

РАЗДЕЛ 6

Технология обработки

звуковой и графической информации 193

Определение объёма и скорости передачи

цифровой мультимедиа-информации 193

Конспект 193

Разбор типовых задач 199

РАЗДЕЛ 7

Обработка числовой информации 203

Электронные таблицы 203

Конспект 203

Разбор типовых задач 210

РАЗДЕЛ 8

Технологии поиска

и хранения информации 252

Базы данных. Сортировка данных.

Запросы в базах данных 252

Конспект 252

Разбор типовых задач 261

Поиск информации 276

Конспект 276

Разбор типовых задач 282

РАЗДЕЛ 9

Теория игр 287

Анализ выигрышных ходов 287

Конспект 287

Разбор типовых задач 292

РАЗДЕЛ 10

Программирование 305

Условный оператор. Циклы. 305

Конспект 305

Разбор типовых задач 313

Процедуры и функции 327

Конспект 327

Разбор типовых задач 337

Операции с массивами 343

Конспект 343

Разбор типовых задач. 362

РАЗДЕЛ 11

Обработка текстовых данных.

Операции с файлами 371

Текстовые типы данных. 371

Конспект 371

Операции с файлами	378
<i>Конспект</i>	374
<i>Разбор типовых задач</i>	384
 РАЗДЕЛ 12	
Практическое программирование	399
<i>Конспект</i>	399
<i>Разбор типовых задач</i>	407

Предисловие

Единый государственный экзамен (ЕГЭ) в настоящее время признан в качестве основной формы объективной оценки качества подготовки школьников, освоивших образовательные программы среднего (полного) общего образования. Результаты ЕГЭ являются результатами одновременно государственной (итоговой) аттестации для образовательных организаций среднего (полного) общего образования и вступительных экзаменов по соответствующим общеобразовательным предметам — для образовательных организаций среднего и высшего профессионального образования.

Информатика относится к предметам, сдача ЕГЭ по которым производится на добровольной основе. Однако перечень высших учебных заведений, требующих наличия свидетельства об успешной сдаче ЕГЭ по информатике для поступления на основные специальности, постоянно растёт.

Возрастает год от года и сложность заданий, предлагаемых на ЕГЭ по информатике. Начиная с 2021 года экзамен проводится в компьютерном формате. Этот формат ЕГЭ предполагает:

- увеличена доля задач, связанных с алгоритмикой и программированием, появились сопоставимые по сложности задачи по программированию «олимпиадного» уровня, в которых нужно не только хорошо знать основы программирования на выбранном языке (в том числе для работы с текстовыми данными и с файлами, что ранее не затрагивалось вообще), но и уметь просто «догадаться», найти правильную идею построения алгоритма;

- исключение экспертной проверки: предполагается ввод конкретных ответов в виде чисел или текстовых символов, так что проверка ответов возможна полностью при помощи компьютеров;

- вместе с тем появляется некоторая «вольность» в решении задач: поскольку проверяются только ответы, получить их можно любым способом — расчётами на бумаге, в электронных таблицах, написав программу и пр. (например,

задания, связанные с обработкой чисел в электронных таблицах, можно решить при помощи программы, а задачи на программирование — решить с помощью Excel).

Учитывая всё это, подготовка к ЕГЭ является высоко актуальной задачей как для самих учащихся старшей школы, так и для учителей информатики. Наилучшей стратегией такой подготовки является, конечно же, системное и целенаправленное формирование основных информационных компетенций школьников, отработка решения разнообразных заданий и выработка навыков работы с основными средствами ИКТ по всем без исключения изучаемым темам курса. Ко всему прочему следует принимать в расчёт и недостаточное количество часов, отпущенных на изучение предмета, и практическое отсутствие задачников-практикумов, поддерживающих не фрагментарное ознакомление с отдельными темами, а плотное прохождение всего курса.

Подготовиться к сдаче ЕГЭ на 90–100 баллов — задача достаточно сложная, требующая обоюдной заинтересованности и напряжения сил как учащегося, так и учителя.

Предлагаемый вашему вниманию справочник — результат многолетней педагогической практики автора.

Структура справочника основана на результатах анализа тематики заданий ЕГЭ за последние несколько лет и особенно — компьютерной версии ЕГЭ.

Книгу можно использовать для самостоятельной (в том числе под контролем со стороны учителя) работы школьника при подготовке к ЕГЭ, для повторения ранее изученных основных теоретических сведений и выработки навыков решения задач.

Разделы справочника по каждой изучаемой теме включают конспект теоретического материала и разбор решений ряда типовых заданий ЕГЭ, предлагавшихся за 2–3 последних года. Для выполнения работы необходим компьютер с установленной на нём операционной системой, редакторами электронных таблиц, текстовыми редакторами, средами программирования на языках: Школьный алгоритмический язык, C#, C++, Pascal, Java, Python. Перечень дополнительных устройств и материалов, пользование которыми разре-

шено на ЕГЭ, утверждён приказом Минпросвещения России и Рособрнадзора.

Использование справочника рекомендуется в сочетании с дополнительным решением вариантов заданий ЕГЭ (демо-версией, опубликованной на сайте www.fipi.ru — официального разработчика экзаменационных материалов, пособия издательства «АСТ» Д. М. Ушакова «20 тренировочных вариантов для подготовки к единому государственному экзамену»).

Желаем успешной подготовки к сдаче единого государственного экзамена!

В связи с возможными изменениями в формате и количестве заданий рекомендуем в процессе подготовки к экзамену обращаться к материалам сайта официального разработчика экзаменационных заданий — Федеральному институту педагогических измерений: www.fipi.ru.

РАЗДЕЛ 1

Информация.

Измерение информации.

Кодирование информации

Измерение количества информации



Конспект _____

Вероятностный подход к измерению количества информации. Информация как снятая неопределённость в знаниях

Для определения количества информации, содержащейся в сообщении о каком-либо объекте или событии, используется вероятностный подход. Он основан на следующих соображениях:

- те или иные события имеют некоторую вероятность (возможность произойти или не произойти);
- событие, которое совершается всегда, имеет вероятность, равную 1 (например, восход Солнца); событие, которое не совершается никогда, имеет вероятность, равную 0 (например, восход Солнца на западе); в остальных случаях вероятность совершения события есть дробное число от 0 до 1;
- получая сообщение о совершении (или несвершении) некоторого события, мы получаем некоторое количество информации, которое определяется снятой с её помощью неопределённостью наших знаний об указанном событии:
 - если вероятность совершения события точно равна 1 или 0 (т. е. мы точно знаем, что событие произойдёт (или не произойдёт), то никакой неопределённости в наших знаниях нет и сообщение о таком событии несёт нулевое количество информации;
 - для равновероятных событий чем больше их количество (т. е. шире возможный выбор вариантов и потому меньше вероятность каждого из них), тем

большее количество информации несёт сообщение о совершившемся конкретном событии;

- количество информации в сообщении о совершении (несовершении) нескольких независимых событий равно сумме количеств информации, содержащейся в сообщениях о каждом отдельном таком событии.

Формула Хартли

Для N равновероятных возможных событий количество информации, которое несёт сообщение о выборе (совершении) одного конкретного события, определяется **формулой Хартли**:

$$I = \log_2 N,$$

где $\log$ — *функция логарифма по основанию 2*, обратная возведению значения основания логарифма в степень, равную I , т.е. из формулы Хартли следует зависимость:

$$N = 2^I.$$

Для облегчения вычислений для значений N , представляющих собой степени числа 2, можно составить таблицу:

N	2	4	8	16	32	64	128	256	512	1024
I (бит)	1	2	3	4	5	6	7	8	9	10

Для значений N , не равных степени двойки, при определении количества информации в битах из выше-приведённой таблицы берётся *ближайшее большее значение N* , равное степени 2. Например, для 48 равновозможных событий количество информации, которое содержится в сообщении о совершении конкретного события, принимается равным 6 бит (так как ближайшее большее значение N , равное степени числа 2, равно 64).

«Принцип вилки»

Для приближённого вычисления количества информации при значении N , не равном 2 в некоторой степени, определяются значения количества информации для двух

Раздел 1. Информация. Измерение информации...

соседних значений N , составляющих степени 2, и составляется соответствующее двойное неравенство.

Например, пусть нужно оценить количество информации в сообщении о выпадении на верхней грани игрального кубика шести точек. В этом случае $N = 6$. Ближайшими к нему являются значения — степени двойки: $N = 4 (2 \cdot 2)$ и $N = 8 (2 \cdot 2 \cdot 2)$. Тогда можно составить неравенство:

$$2^2 < 2^I < 2^3.$$

Отсюда искомое количество информации будет больше 2 и меньше 3 битов.

Формула Шеннона.

Связь количества информации с понятием вероятностей

Для N событий с различными вероятностями $p_1, p_2, \dots, p_N$ количество информации определяется **формулой Шеннона**:

$$I = - \sum_{i=1}^N p_i \log_2 \frac{1}{p_i}.$$

Если все эти события равновероятны, т.е. $p_1 = p_2 = \dots = p_N = p$, то очевидно, что формула Шеннона преобразуется в формулу Хартли (которая, таким образом, представляет собой частный случай формулы Шеннона).

Связь между количеством информации и вероятностью события

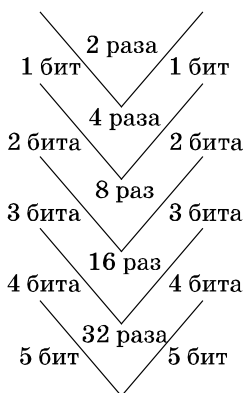
Для N равновероятных событий вероятность одного отдельного события $p = 1/N$. С учётом этого формула Хартли может быть преобразована в соотношение:

$$I = \log_2 \frac{1}{p}.$$

В этом случае вычисление количества информации можно производить по таблице со стр. 8, предварительно вычислив значение N как величину, обратную значению p . Например, для события, вероятность которого (p) составляет 0,018, получается $N = 1/0,018 = 55,56$, тогда берётся ближайшее большее значение N , кратное 2 ($N = 64$), и по таблице определяется, что $I = 6$ битов.

«Принцип ёлочки»

Сколько информации несёт в себе некоторое сообщение? Известно, что количество информации, равное 1 биту, соответствует снятию неопределённости при помощи ответа «да» или «нет» на один элементарный вопрос, т. е. 1 бит соответствует уменьшению неопределённости в 2 раза. А чему соответствует уменьшение неопределённости, например, в 4 раза? В подобном случае можно задать последовательно два вопроса, на которые даются ответы «да» или «нет». В общем, количество информации в n бит позволяет уменьшить неопределённость в 2^n раз.



Бит. Байт. Производные величины

Принято считать, что минимально возможное количество информации соответствует такому сообщению, получение которого уменьшает неопределённость в 2 раза (пример — сообщение о выпадении на подброшенной монете «орла» из двух равновероятных вариантов — «орёл» и «решка»). Это минимальное количество информации получило название «**бит**» (англ. *bit* как сокращение названия *binary digit* — двоичная цифра).

В вычислительной технике бит соответствует одному *двоичному разряду*, который может принимать одно из двух возможных значений: 0 или 1. В качестве более крупной величины принят **байт**, соответствующий двоичному числу из

Раздел 1. Информация. Измерение информации...

8 разрядов (битов). В оперативной памяти компьютера минимальный объём ячейки памяти, выделяемой для хранения какой-либо величины, как правило, равен одному байту. Ячейки большего размера имеют объём, кратный байту с коэффициентом кратности 2: 2 байта (16 битов), 4 байта (32 бита), 8 байтов (64 бита). Такую «порцию» информации принято называть **машинным словом**.



В теории информации количество информации может быть дробной величиной. В вычислительной технике количество информации может составлять только целое число битов (дробное значение при необходимости округляется в **большую** сторону).

В вычислительной технике в большинстве практических задач получаемое количество битов округляется в большую сторону до целого количества байтов, хотя в некоторых случаях возможна «поточковая» запись значений, состоящих из количества битов, не кратного 8.

Для обозначения количеств информации, больших, чем байт, приняты следующие производные величины:

1 **Килобайт** (КБ) = ($2^{10} = 1024$) байт;

1 **Мегабайт** (МБ) = ($2^{10} = 1024$) Килобайт =
= ($2^{20} = 1048576$) байт;

1 **Гигабайт** (ГБ) = ($2^{10} = 1024$) Мегабайт =
= ($2^{20} = 1048576$) килобайт = ($2^{30} = 1073741824$) байт;

1 **Терабайт** (ТБ) = ($2^{10} = 1024$) Гигабайт =
= ($2^{20} = 1048576$) Мегабайт = ($2^{30} = 1073741824$) Килобайт = ($2^{40} = 1099511627776$) байт;

1 **Петабайт** (ПБ) = ($2^{10} = 1024$) Терабайт;

1 **Эксабайт** (ЭБ) = ($2^{10} = 1024$) Петабайт;

1 **Зеттабайт** (ЗБ) = ($2^{10} = 1024$) Эксабайт;

1 **Йоттабайт** (ЙБ) = ($2^{10} = 1024$) Зеттабайт.



Внимание! В отличие от одноименных приставок в кратных величинах в математике изменение величин в вычислительной технике происходит на каждом «шаге» вышеуказанной шкалы на $2^{10} = 1024$, а не на $10^3 = 1000$.

Для избежания этой путаницы были предложены особые, двоичные приставки для производных величин количества информации:

Кибибайт	KiB	2^{10} (1024)
Мебибайт	MiB	2^{20} (1048576)
Гибибайт	GiB	2^{30} (1073741824)
Тебибайт	TiB	2^{40} (1099511627776)
Пебибайт	PiB	2^{50} (1125899906842624)
Эксбибайт	EiB	2^{60} (1152921504606846976)

Алфавитный (алгоритмический) подход к измерению количества информации. Алфавит. Мощность алфавита

В этом случае количество информации в сообщении представляет собой чисто технический параметр (важный с точки зрения хранения или передачи информации) и не зависит от содержания сообщения.

При алфавитном подходе информационное сообщение рассматривается как некоторое количество (K) **знаков (символов, кодов)** из некоторого используемого полного набора, называемого **алфавитом**. Количество (N) знаков в алфавите называется **мощностью** этого алфавита.



В данном конкретном сообщении не обязательно используются **все** знаки алфавита. Мощность алфавита определяется не набором знаков, используемых в конкретном сообщении, а количеством знаков, которые вообще могут быть использованы в сообщениях, кодируемых в соответствии с данным алфавитом.

Алгоритм определения количества информации в сообщении:

- 1) определяется мощность используемого алфавита N ;
 - 2) определяется количество информации, приходящееся в алфавите на один его знак:
- если использование всех знаков равновероятно, то используется формула Хартли (либо её следствие: $N = 2^I$) и табл. 1.1;

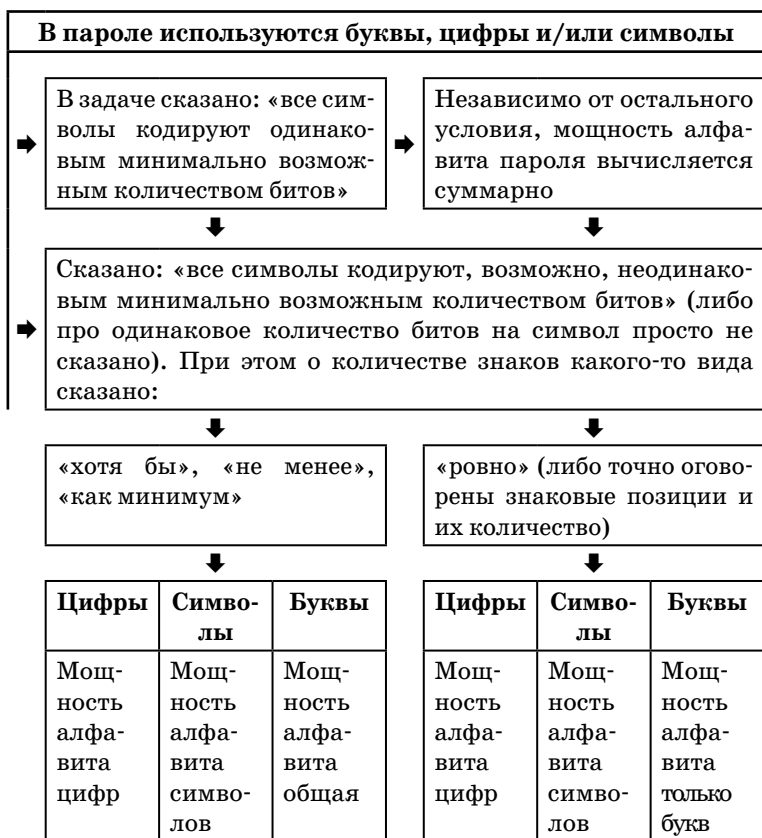
Раздел 1. Информация. Измерение информации...

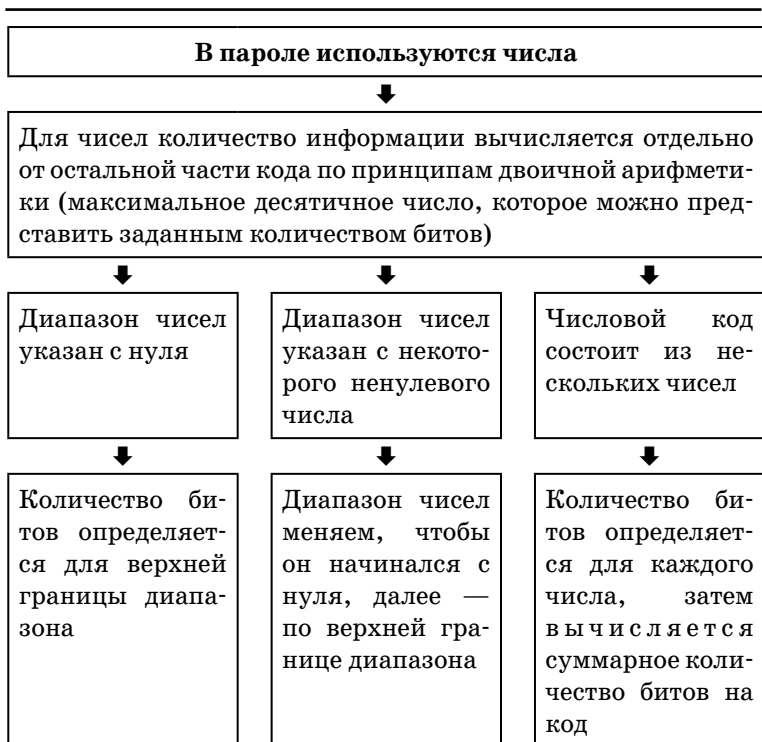
- если известны вероятности использования тех или иных знаков (на основе составленной таблицы частоты встречаемости этих знаков), то используется формула Шеннона;

3) вычисленное количество информации (I), приходящееся на один знак, умножается на количество (K) знаков в данном сообщении:

$$I_{\Sigma} = I \cdot K.$$

Обобщенная схема решения задач на определение количества информации с использованием алфавитного подхода





Количество различных состояний панели, имеющей M элементов, каждый из которых может находиться в N различных состояниях, равно количеству различных M -разрядных чисел (начиная с нуля) в системе счисления с основанием N и вычисляется как N^M .



Количество различных сообщений, включающих в себя M элементов, каждый из которых может иметь N различных состояний, равно количеству различных M -разрядных чисел (начиная с нуля) в системе счисления с основанием N и вычисляется как N^M .



Количество всех M -разрядных чисел в системе счисления с основанием N равно N^M . Максимальное значение M -разрядного числа в системе счисления с основанием N равно $(N^M - 1)$.



Если в условии задачи предлагается определить количество сигналов, подаваемых с помощью разного количества элементов (от X до Y), то нужно отдельно вычислить количества возможных сигналов для каждого возможного числа элементов и просуммировать полученные значения.



Разбор типовых задач _____

Задача 1. Для регистрации на сайте некоторой страны пользователю требуется придумать пароль. Длина пароля — ровно 11 символов. В качестве символов используются десятичные цифры и 12 различных букв местного алфавита, причём все буквы используются в двух начертаниях: как строчные, так и заглавные (регистр буквы имеет значение!).

Под хранение каждого такого пароля на компьютере отводится минимально возможное и одинаковое целое количество байтов, при этом используется посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством битов.

Определите объём памяти в байтах, который занимает хранение 60 паролей.

Решение

Данная задача решается с использованием *алфавитного подхода к измерению количества информации*.

1. Определяется мощность используемого алфавита. Используются десятичные цифры (10 различных знаков) и 12 букв, каждая из которых может иметь два возможных начертания ($12 \cdot 2 = 24$ различных знака). Итого мощность используемого алфавита составляет: $10 + 12 \cdot 2 = 34$ знака.

2. Исходя из известной мощности алфавита определяется количество битов, соответствующее каждому знаку. $N = 34$. Речь идёт о *целом* (не дробном!) количестве битов, минимально достаточном для представления одного знака такого алфавита. Поэтому выбирается ближайшее большее число N , равное степени числа 2: $N = 2^6 = 64$ (значения $2^5 = 32$ недостаточно). Тогда согласно формуле $N = 2^I$ получается: $I = 6$ битов на один знак алфавита.

3. Длина пароля (длина сообщения, кодируемого с использованием рассмотренного алфавита) равна 11 символам ($K = 11$). Тогда согласно формуле: $I_s = I \cdot K$ количество информации в битах, соответствующее всему такому сообщению (паролю), равно $6 \cdot 11 = 66$ битам.

4. По условию задачи *под хранение каждого такого пароля на компьютере отводится минимально возможное и одинаковое целое количество байтов*. Определяется минимальное количество *целых* байтов, достаточное, чтобы уместить 66 битов. 1 байт равен 8 битам. Выполняем деление количества битов (66) на 8 с округлением результата до целого в большую сторону: $66 / 8 = 8,25 \rightarrow 9$ байтов.



Следует не забывать выполнять перевод количества битов в количество байтов!

5. Тогда для хранения 60 паролей потребуется $60 \cdot 9 = 540$ байтов.

Ответ: 540 байтов.

Задача 2. Вес выпускаемых деталей может меняться от 250 до 310 граммов. Компьютер при маркировке деталей записывает данные о каждой из них в базу данных, присваивая деталям порядковые номера в пределах рабочей смены — от 1 до 2000. Для хранения номера детали и веса детали в граммах программа использует минимально возможные количества битов.

Определите информационный объем в битах массива из 1500 записей о промаркированных деталях.

Решение

Эта задача даже проще, чем предыдущие, но содержит небольшую хитрость.

1. Для хранения номера детали (от 1 до 2000) требуется как минимум 11 битов ($2^{11} = 2048$, а 10 битов мало, так как $2^{10} = 1024$).

2. Для хранения веса детали (от 250 до 310 граммов) могло бы потребоваться целых 9 битов (числа до 512). Но мы

Раздел 1. Информация. Измерение информации...

же можем в нашей программе заранее заложить «базовую» константу 250 и в базе данных хранить не сам вес, а разницу между действительным весом и этой «базовой» константой! То есть при занесении информации в базу данных мы вычитаем из веса детали число 250 и записываем в память эту разность, а при последующей выдаче результатов по запросу — наоборот, прибавляем «базовую» константу 250 к хранящемуся значению.

Тогда запоминаемые значения веса будут меняться уже от 0 до 60 ($310 - 250$), а для них достаточно всего 6 битов ($2^6 = 64$).

Итого — целых 3 бита экономии на каждой детали!

3. Тогда одна запись о детали имеет объём:

1 бит номера + 6 битов веса = 17 бит.

4. Для хранения 1500 таких записей потребуется
 $1500 \cdot 17 = 25\,500$ битов.

Ответ: 25 500.

Задача 3. При регистрации на сайте для каждого пользователя формируется учётная запись и выдаётся пароль. Длина пароля составляет 12 символов, при этом в пароле могут использоваться **строчные и заглавные латинские буквы, хотя бы три цифры и хотя бы два символа** из набора: @, #, \$, %, *, & и ~. Для хранения каждого пароля на компьютере отводится минимально возможное и одинаковое целое количество байтов, при этом используется посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством битов. Кроме собственно пароля, учётная запись содержит некоторую дополнительную информацию, объём которой составляет целое число байтов.

Известно, что для хранения всех учётных записей 40 пользователей потребовалось 1200 байтов.

Определите объём памяти в байтах, выделенный для хранения дополнительной информации в учётной записи одного пользователя. (В ответе необходимо записать только число — количество байтов.)

Для справки: латинский алфавит содержит 26 букв.

Указания к решению:

Что изменилось в условии?

Если раньше пароль мог состоять из произвольного набора знаков (букв, цифр, дополнительных символов), то теперь оговорено, что в пароле обязательно должно быть **не менее указанного** количества цифр и **не менее указанного** количества дополнительных символов. Однако точное количество цифр и дополнительных символов мы не знаем, а в условии по-прежнему записано, что каждый символ занимает в памяти **одинаковое** и минимально возможное количество битов.

Поэтому мы по-прежнему должны учитывать, что наш алфавит (набор символов) для пароля включает в себя **все** возможные знаки и имеет мощность 69 знаков. А дальнейший ход решения абсолютно тот же, что и в предыдущих задачах.

Ответ: 19.

Задача 4. При регистрации на сайте для каждого пользователя формируется учётная запись и выдаётся пароль. Длина пароля составляет 12 символов, при этом в пароле могут использоваться строчные и заглавные латинские буквы, **ровно три** цифры и **ровно два** символа из набора: @, #, \$, %, *, & и ~. Для хранения каждого пароля на компьютере отводится минимально возможное и одинаковое целое количество байтов, при этом используется посимвольное кодирование и все символы **кодируются минимально возможным** количеством битов. Кроме собственно пароля, учётная запись содержит некоторую дополнительную информацию, объём которой составляет целое число байтов.

Известно, что для хранения всех учётных записей 40 пользователей потребовалось 1200 байтов.

Определите объём памяти в байтах, выделенный для хранения дополнительной информации в учётной записи одного пользователя. (В ответе необходимо записать только число — количество байтов.)

Для справки: латинский алфавит содержит 26 букв.

Решение

Изменение условия, казалось бы, совсем пустячное. Вместо «хотя бы» записано «**ровно**», и исчезло указание о кодировании символов *одинаковым* количеством битов. Но теперь решать задачу нужно совершенно иначе!

1.1. Мы точно знаем, что три символа пароля из двенадцати — это цифры. Цифр всего 10 (имеется в виду десятичная система счисления), поэтому для хранения одной цифры достаточно **4 битов** ($2^4 = 16$, а $2^3 = 8$ и этого недостаточно).

1.2. Мы также точно знаем, что два символа пароля берутся из 7-символьного набора дополнительных знаков. Для хранения такого символа достаточно **3 битов** ($2^3 = 8$).

1.3. Остальные символы пароля (их $12 - 3 - 2 = 7$) — это буквы. Мощность латинского алфавита, включающего и строчные, и заглавные буквы, составляет $26 + 26 = 52$ символа. Для их хранения требуется **6 битов** ($2^6 = 64$, а $2^5 = 32$, этого мало).

2. Тогда для хранения всего 12-символьного пароля нам понадобится:

$$\underbrace{3 \times 4}_{\text{цифры}} + \underbrace{2 \times 3}_{\text{доп. символы}} + 7 \times \underbrace{6}_{\text{буквы}} = 60 \text{ битов.}$$

Этим 60 битам соответствует **8 байтов** ($8 \times 8 = 64$).

Дальнейший ход решения аналогичен предыдущим задачам.

3. Если для хранения учётных записей 40 пользователей потребовалось 1200 байтов, то для хранения учётной записи одного пользователя требуется $1200 / 40 = 30$ **байтов**.

4. Тогда если пароль занимает 8 байтов, а вся учётная запись — 30 байтов, то на хранение дополнительной информации о пользователе расходуется $30 - 8 = 22$ **байта**.

Ответ: 22.

Задача 5. При регистрации на сайте для каждого пользователя формируется учётная запись и выдаётся пароль. Длина пароля составляет 12 символов, при этом в пароле могут использоваться **строчные и заглавные латинские буквы, хотя бы три цифры и хотя бы два символа** из набора: @, #, \$, %, *, & и ~. Для хранения каждого пароля на компьютере отводится минимально возможное и одинаковое целое количество байтов, при этом используется посимвольное кодирование и все символы кодируются минимально возможным количеством битов (**возможно, неодинаковым**). Кроме собственно пароля, учётная запись содержит некоторую дополнительную информацию, объём которой составляет целое число байтов.

Известно, что для хранения всех учётных записей 40 пользователей потребовалось 1200 байтов.

Определите объём памяти в байтах, выделенный для хранения дополнительной информации в учётной записи одного пользователя. (В ответе необходимо записать только число — количество байтов.)

Для справки: латинский алфавит содержит 26 букв.

Решение

Отличие данного условия от задачи 2 только в явном указании, что для хранения символов может использоваться **неодинаковое** количество битов. Но точное количество цифр и дополнительных знаков (как было в задаче 3) мы не знаем. Как решать?

Главная хитрость здесь в том, что мы знаем о гарантированном наличии в пароле указанных количеств цифр и дополнительных символов. Именно на них можно сэкономить (как делалось в задаче 3).

1.1. Мы знаем, что три символа пароля — это точно цифры. Цифр всего 10 (имеется в виду десятичная система счисления), поэтому для хранения одной цифры достаточно **4 битов** ($2^4 = 16$, а $2^3 = 8$ и этого недостаточно).

1.2. Мы также знаем, что два символа пароля точно берутся из 7-символьного набора дополнительных знаков. Для хранения такого символа достаточно **3 битов** ($2^3 = 8$).

Раздел 1. Информация. Измерение информации...

1.3. Остальные символы пароля (их $12 - 3 - 2 = 7$) могут быть *любыми* — буквами, цифрами или дополнительными символами. Мощность такого объединённого алфавита (как мы вычислили в задачах 1 и 2) составляет 69 символов. Для их хранения требуется **7 битов** ($2^7 = 128$, а $2^6 = 64$, этого мало).

2. Тогда для хранения всего 12-символьного пароля нам понадобится:

$$\underbrace{3 \times 4}_{\text{цифры}} + \underbrace{2 \times 3}_{\text{доп. символы}} + \underbrace{7 \times 7}_{\text{все остальные}} = 67 \text{ битов.}$$

Этим 67 битам соответствует **9 байтов** ($9 \times 8 = 72$).

3. Если для хранения учётных записей 40 пользователей потребовалось 1200 байтов, то для хранения учётной записи одного пользователя требуется $1200 / 40 = 30$ **байтов**.

4. Тогда если пароль занимает 9 байтов, а вся учётная запись — 30 байтов, то на хранение дополнительной информации о пользователе расходуется $30 - 9 = 21$ **байт**.

Ответ: 21.



Итак, нужно очень внимательно читать условия задач.

- Если сказано, что каждый символ кодируется **одинаковым** минимально необходимым количеством битов, то в любом случае надо вычислять **общий** объём используемого алфавита и найденное по нему «универсальное» количество битов на символ умножать на всю длину пароля.

- Если не сказано, что символы кодируются одинаковым числом битов или тем более указано, что количество битов на символ, **возможно, неодинаково**, то надо отдельно просчитать количество битов на символ для цифр, дополнительных знаков и т.д., для которых в условии указано «не менее», «хотя бы» или «ровно». А пароль нужно разделять на соответствующие фрагменты, обсчитывать каждый такой фрагмент по отдельности и затем суммировать полученные количества битов.

• Если количества цифр или дополнительных знаков указаны **точные**, то остальная часть пароля считается **только как буквы**. Если же для цифр и/или дополнительных знаков указано **«хотя бы»** (либо **«как минимум»**), то для оставшейся части пароля вычисляется **общий объём алфавита** по всем возможным символам.

Задача 6. Каждому школьнику выдаётся электронный пропуск, содержащий индивидуальный логин этого школьника, код, обозначающий класс и параллель, а также некоторая дополнительная информация. Логин состоит из 13 символов, каждый из которых может быть одной из 26 заглавных латинских букв. Для записи логина отведено минимально возможное целое число байтов, при этом используют посимвольное кодирование, все символы кодируют одинаковым минимально возможным количеством бит. Данный код — это целое число от 1 до 180, записанное как двоичное число и занимающее минимально возможное целое число байтов. Всего в пропуске хранится 32 байта данных. Сколько байтов выделено для хранения дополнительных сведений?

Решение

1. Рассматриваем отдельно логин.

Алфавит состоит из 26 латинских заглавных букв, следовательно, потребуется 5 бит на один символ.

Всего в логине 13 символов — значит, под них нужно отвести $5 \times 13 = 65$ битов, или 9 байтов (8 байтов — недостаточно).

2. Код класса и параллели — это число от 1 до 180. Для хранения таких чисел в двоичном формате достаточно 1 байта.

3. На всю указанную выше информацию требуется: $9 + 1 = 10$ байтов.

4. Если на всю запись о школьнике отводится 32 байта, то дополнительные сведения занимают:

$$32 - 10 = 22 \text{ байта.}$$

Ответ: 22.

Равномерные и неравномерные двоичные коды



Конспект _____

Равномерные и неравномерные двоичные коды.

Условие Фано

Кодирование символов обычно предполагает, что каждому символу всегда сопоставляется одинаковое количество битов (например, в кодовой таблице ASCII каждому символу сопоставляется один байт, хранящий порядковый номер того или иного символа в этой таблице). Такой способ кодирования прост и удобен, однако очевидно, что он является не самым оптимальным. Для значительной части символов используются не все биты отведённых под них байтов (часть старших битов — нулевые), а при наличии в тексте только части символов, предусмотренных в таблице ASCII (например, если текст содержит только прописные русские буквы), приходится всё равно использовать 8-битный код.

Более компактным является *неравномерный двоичный код* (особенно если при его построении исходить из частоты встречаемости различных символов и присваивать наиболее часто используемым знакам самые короткие коды, как это сделано в *методе Хаффмана*). При этом количество битов, отводимых для кодирования символов, в целом зависит от количества используемых в конкретном случае различных символов (от *мощности алфавита*), а коды, соответствующие разным символам, могут иметь различную длину в битах.

Главное при таком кодировании — обеспечить возможность *однозначного декодирования* записанной с помощью этих кодов строки (поочерёдного, слева направо, выделения и распознавания из сплошной последовательности нулей и единиц кодов отдельных букв). Для этого коды символам необходимо назначать в соответствии с *условиями Фано*.

Прямое условие Фано. Неравномерный код может быть однозначно декодирован, если никакой из кодов не совпадает с началом (префиксом) какого-либо другого, более длинного кода.

A	B	C
10	11	001

D: 00
недопустимо:
↓

C	001
D	00

Код D совпадает с началом кода C

A	B	C
10	11	00

D: 11
недопустимо:
↓

B	11
D	11

Код D совпадает с кодом B

A	B	C
100	110	010

D: 00
допустимо:
↓

Код D не совпадает ни с одним другим кодом и с началом никакого другого кода

Обратное условие Фано. Неравномерный код может быть однозначно декодирован, если никакой из кодов не совпадает с окончанием (постфиксом) какого-либо другого, более длинного кода.

A	B	C
10	11	001

D: 01
недопустимо:
↓

C	001
D	01

Код D совпадает с концом кода C

A	B	C
10	11	00

D: 11
недопустимо:
↓

B	11
D	11

Код D совпадает с кодом B

A	B	C
100	110	010

D: 01
допустимо:
↓

Код D не совпадает ни с одним другим кодом и с началом никакого другого кода

Для однозначности декодирования последовательности кодов достаточно выполнения *хотя бы одного* из двух вышеуказанных условий Фано:

- при выполнении прямого условия Фано последовательность кодов однозначно декодируется с начала;
- при выполнении обратного условия Фано последовательность кодов однозначно декодируется с конца.

Выбрать, какое из двух правил Фано используется при решении конкретной задачи, можно, проанализировав коды в условии задачи (без учёта кода, проверяемого в вариантах ответа): если для исходных кодов выполняется прямое правило Фано, то его и нужно использовать при решении, и наоборот.

Вместе с тем нужно помнить, что правила Фано — это достаточное, но не необходимое условие однозначного декодирования: если не выполняется ни прямое, ни обратное правило Фано, конкретная двоичная последовательность может оказаться такой, что она декодируется однозначно (так как остальные возможные варианты до конца декодирования довести не удаётся). В подобном случае необходимо пытаться строить дерево декодирования в обоих направлениях.



Рекомендуется начинать решение задач такого типа с анализа выполнимости правил Фано для исходных кодов, указанных в условии задачи (т. е. без учёта искомого кода в вариантах ответов). В зависимости от того, какое из двух правил Фано выполняется для исходных кодов, при дальнейшем решении задачи производится сравнение более короткого кода с началом (при выполнении прямого правила Фано) или с концом (при выполнении обратного правила Фано) более длинного кода.



Если для заданной последовательности кодов выполняется прямое правило Фано, то её декодирование необходимо вести с начала (слева направо). Если для заданной последовательности кодов выполняется обратное правило Фано, то её декодирование необходимо вести с конца (справа налево).



При сравнении пары кодов удобно подписывать более короткий код под более длинным, выравнивая эти записи по левому краю — для прямого правила Фано либо по правому краю — для обратного правила Фано.

Дерево Фано

Дерево Фано — это удобный и наглядный способ решения задач, связанных с подбором неравномерных двоичных кодов.

Основные принципы построения дерева Фано

1. Учёт частоты встречаемости символов в тексте — чем чаще встречается какой-либо символ, тем короче для него должен быть код и тем раньше этот символ надо поместить в дерево.

2. Каждый узел дерева Фано порождает ровно две ветви (т. е. дерево Фано является *бинарным*), при этом одной ветви (например, левой) сопоставляется бит 0, а другой ветви — бит 1.

3. На каждом новом этапе ветвления, кроме самого последнего, одна ветвь может быть завершена каким-то символом из тех, для которых генерируются коды, но вторая ветвь обязательно должна служить продолжением дерева (иначе его не удастся построить для остальных символов). Только два последних символа рассматриваемого алфавита можно поместить на концах двух последних ветвей, тем самым «закрыв» и дерево, и генерацию кодов.

4. Для каждого символа код Фано получается последовательной записью всех нулей и единиц по кратчайшему пути от вершины дерева к соответствующему символу.



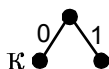
Разбор типовых задач

Задача 1. Для кодирования последовательности символов, состоящей из букв К, И, Н, О, используется неравномерный двоичный код, удовлетворяющий условию Фано. При этом для буквы К использован код 0, а для буквы И — код 11. Требуется определить наименьшую возможную суммарную длину всех кодовых слов указанных букв.

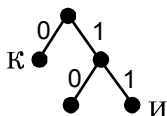
Решение

Решение задач, в которых требуется поиск кратчайших кодов, удовлетворяющих условию Фано, удобно и наглядно выполнять при помощи *дерева Фано*.

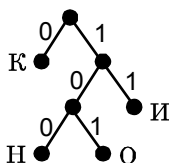
1. Начинаем построение дерева Фано. При этом по условию, код 0 уже занят для буквы К. Поэтому помещаем этот символ на конце соответствующей ветви. Вторая ветвь послужит для продолжения построения дерева.



2. Продолжая построение правой ветви дерева, получаем две ветви — 0 и 1. Но двум ветвям 1 и 1 соответствует код 11, которому, по условию, соответствует буква И. Помещаем её на конце соответствующей ветви.

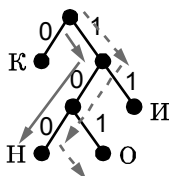


3. Два оставшихся символа мы можем разместить на двух последних ветвях дерева, построенных слева. При этом мы «закрываем» дерево, и больше никакие символы добавить в данный алфавит уже будет нельзя.



Это один из возможных вариантов построения дерева Фано: правильным решением было бы и, наоборот, поместить букву О слева, а букву Н — справа. Но в данной задаче требуется определить суммарную длину всех кодовых слов, поэтому порядок размещения самих добавляемых символов (их конкретные коды) не важен.

4. Пользуясь построенным деревом, записываем коды Фано для добавленных букв: Н — код 100 (сплошные стрелки на рисунке), О — код 101 (пунктирные стрелки).



5. Вычисляем суммарную длину кодов, помня, что код буквы К — 0, а буквы И — 11: $1 + 2 + 3 + 3 = 9$.

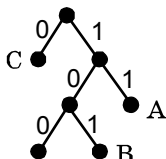
Ответ: 9.

Задача 2. По каналу связи передаются сообщения, содержащие шесть букв: А, В, С, D, Е, F. Для передачи используется неравномерный двоичный код, удовлетворяющий условию Фано. Для букв А, В, С используются кодовые слова 11, 101 и 0 соответственно.

Укажите кодовое слово наименьшей возможной длины, которое можно использовать для буквы F. Если таких слов несколько, то укажите то из них, которое соответствует наименьшему возможному двоичному числу.

Решение

Начинаем с построения дерева Фано. При этом сразу размещаем на его ветвях заданные символы по их известным кодам.



На первом шаге мы поместили в ветвь 0 символ С, так как для него задан код Фано, равный 0. На втором шаге, поскольку код Фано для символа А равен 11, мы этот символ поместили на ветви 1, чтобы путь от вершины к А давал нам две единицы. На третьем шаге определяется место для символа В: его код 101, значит, после ветвления по ветви 1, а потом по ветви 0 нужно В поместить на конце ветви 1.

Теперь строим новое ветвление от левой (оставшейся свободной) ветви:

