**Информация и энтропия**

|  |
| --- |
| Клоду Шеннону удалось придумать удивительно простую и глубокую модель передачи информации, без которой теперь не обходится ни один учебник. Он ввел понятия: источник сообщения, передатчик, канал связи, приемник, получатель сообщения и источник шума, который может исказить сигнал. Практически любой, даже очень сложный, обмен сообщениями можно успешно описать в этих терминах. |
| рис. 1 |

**Дискретное сообщение** — это любой набор символов, который формируется Источником (им может быть, например, человек). От Источника сообщение переходит к Передатчику, который преобразует его к виду, который уже можно передавать по Каналу связи. Например, Передатчик может кодировать сообщение. Преобразованное сообщение называется Сигналом. Канал связи — это технический комплекс аппаратуры, который позволяет передать Сигнал. В общем случае в процессе передачи сигнал в канале искажается шумом, который исходит от Источника шума. Приемник обычно выполняет операцию, обратную по отношению к той, что производится передатчиком, — т. е. восстанавливает сообщение по сигналам. Процесс преобразования сигнала в сообщения, осуществляемый в Приемнике, называют декодированием. Получатель — это человек или аппарат, для которого предназначено сообщение.

**Энтропия**

В 1946 г. американский ученый-статистик Джон Тьюки предложил название БИТ (BIT — аббревиатура от BInary digiT), одно из главных понятий XX века. Тьюки избрал бит для обозначения одного двоичного разряда, способного принимать значение 0 или 1. Шеннон использовал бит как единицу измерения информации. Мерой количества информации Шеннон предложил считать функцию, названную им энтропией.

Пусть сообщение — осмысленное предложение на русском языке. Шеннон заметил, что при передаче различных букв мы передаем разное количество информации. Если мы передаем часто встречающиеся буквы, то информация меньше; при передаче редких букв — больше. Это видно при кодировании букв алфавита азбукой Морзе. Наиболее частые буквы передаются коротко, а для редких используют более длинные цепочки. Так, буква «Е» кодируется одной точкой «.», а редкая «Ш» — четырьмя тире «––––» (это самая длинная последовательность на букву в азбуке Морзе).

Количество информации на букву связано с частотой употреблений этой буквы во всех сообщениях, формируемых на языке. Чем более редкую букву мы передаем, тем больше в ней информации.  
Энтропия — мера непредсказуемости. Это понятие Шеннон взял из статистической термодинамики. Пусть вероятность i-того символа алфавита, состоящего из n символов (мера частоты, с которой встречается символ во всех сообщениях языка), равна pi. Тогда:

http://dssp.petrsu.ru/p/tutorial/informatics/chapter1/3/59214.jpg

(здесь log — логарифм по основанию 2).  
Шеннон пишет: «Величина H играет центральную роль в теории информации в качестве меры количества информации, возможности выбора и неопределенности». Количество информации, передаваемое в сообщении, тесно связано с мерой неопределенности, или непредсказуемости передаваемых символов.

Каков же содержательный смысл энтропии? Ожидая очередное сообщение от источника, получатель находится в состоянии неопределенности. После получения этого очередного

(одного) символа данная неопределенность снимается. Энтропия — это 1) количество неопределенности относительно того, что послал источник, которое снялось после получения этого одного символа, или, что то же самое, 2) количество информации

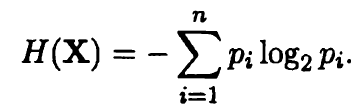
относительно того, что послал источник, которое мы получили с этим одним символом.

Энтропия ограничивает максимально возможное сжатие без потерь (или почти без потерь), которое может быть реализовано при использовании теоретически — [типичного набора](https://ru.wikipedia.org/w/index.php?title=%D0%A2%D0%B8%D0%BF%D0%B8%D1%87%D0%BD%D1%8B%D0%B9_%D0%BD%D0%B0%D0%B1%D0%BE%D1%80&action=edit&redlink=1) или, на практике, — [кодирования Хаффмана](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%A5%D0%B0%D1%84%D1%84%D0%BC%D0%B0%D0%BD%D0%B0), [кодирования Лемпеля — Зива — Велча](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%9B%D0%B5%D0%BC%D0%BF%D0%B5%D0%BB%D1%8F_%E2%80%94_%D0%97%D0%B8%D0%B2%D0%B0_%E2%80%94_%D0%92%D0%B5%D0%BB%D1%87%D0%B0) или [арифметического кодирования](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BA%D0%BE%D0%B4%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5).

Рассмотрим пример.

Пусть источник может равновероятно посылать одно из двух значений, т. е. таблица частот имеет вид:



Согласно 

имеем

,

т. е. был получен 1 бит информации.

Пусть теперь источник может равновероятно посылать

одно из тридцати двух значений, т. е. таблица частот имеет вид:



Согласно формуле имеем

т. е. было получено 5 бит информации.

Обсудим полученные результаты. И в первом случае, и во втором получатель, приняв сигнал от источника, точно знает, что тот посылал. Однако во втором случае информации доставляется в пять раз больше (и, соответственно, в пять раз больше снимается неопределенности). Так происходит потому, что предугадать значение, которое пошлет источник во втором случае, труднее: 25 вариантов во втором случае против 21 в первом.

Рассмотрим случай с неравномерным распределением частот.



Очевидно, что гораздо проще предугадывать значение случайной величины Хз (так как она почти всегда равна нулю), чем Х1. Значит, неопределенности при получении очередного символа снимается меньше, и информационная ценность получаемого символа меньше. Действительно, согласно формуле имеем

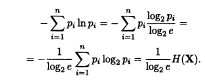
,т. е. было

получено 0.4689956 бит информации.

Приведенные примеры показывают, что возможна и третья трактовка энтропии. Энтропия— это минимум среднего количества бит, которое нужно передавать по каналу связи о текущем состоянии источника.

В приведенных определениях энтропии логарифмы вычислялись по основанию 2 . Однако

ничто не мешает определить эти величины, вычисляя логарифмы по другому основанию, например, е или 10 ; все математические свойства при этом останутся. Действительно, согласно формуле перехода к другому основанию , имеем



Таким образом, переход к другому основанию эквивалентен домножению на нормировочный коэффициент.

Некоторые авторы предлагают в качестве основания логарифма брать *п* — число возможных исходов случайной величины; или, в терминах источник—получатель, количество различных сигналов, которые может послать источник:

**

Такое разнообразие подходов к измерению энтропии не должно смущать — вспомните, сколько существует единиц измерения длины: метр, фут, морская миля, световой год и

т. д. Все эти меры приводимы друг к другу.

**Постановка задачи**

На вход подается строка — последовательность букв, ввод с консоли.

Требуется:

1) построить алфавит (т. е. множество всех различных символов

исходной строки) и найти частоту для всех символов алфавита, показать на экране: символ – количество появлений символа в строке;

2) для полученной таблицы рассчитать энтропию согласно

**

Где n - количество символов в алфавите (т. е. количество всех различных символов

исходной строки).

Расчет pi: N – длина строки, ni - количество появлений символа в строке, тогда pi= ni/N.

Пример 1. На вход поступает строка abed.



энтропия равна 1 .

Пример 2. На вход поступает строка abrakadabra.



энтропия приблизительно равна 0.87874099**.**

Пример 3. На вход поступает строка ааааа.



энтропия равна 0 .

Указания к решению

CScript:

str=WScript.StdIn.ReadLine() - чтение с консоли

Node.js

Ввод с консоли

let readline = require('readline');

const rl = readline.createInterface({

input: process.stdin,

output: process.stdout

});

rl.question(' Ввод', (answer) => {

rl.close();

let output = answer[1]; действия с введенным выражением

console.log(output);

});

Для построения алфавита и таблицы частот рекомендуем воспользоваться следующим фрагментом кода:

alphch=new Array();

for(i=0;i<str.length;i++)

if (alphch[str.charAt(i)])

alphch[str.charAt(i)]++

else

alphch[str.charAt(i)]=1

Пройдем по заполненным ячейка массива

for(a in alphch){}

<https://learn.javascript.ru/array>