

Богомолова Ольга Борисовна,
Усенков Дмитрий Юрьевич

ФАНО И ЕГО «КОЛЛЕГИ»: РАСТИМ ДЕРЕВО

Задания, в которых нужно кодировать и декодировать сообщения при помощи неравномерного двоичного кода либо подобрать такой код, похоже, прочно «прописались» на ЕГЭ. И вызывают у учащихся заметные трудности, поскольку эта тема в учебниках информатики практически не затрагивается. А задачи становятся всё сложнее... Ну что же, – попробуем в них разобраться? Но сначала – немного теории.

НЕРАВНОМЕРНЫЙ ДВОИЧНЫЙ КОД. УСЛОВИЕ ФАНО

Кодирование символов при обработке текстовой информации в компьютере обычно предполагает, что каждому символу всегда сопоставляется одинаковое количество

битов, – например, в кодовой таблице ASCII каждому символу сопоставляется один байт, хранящий порядковые номера символов в этой таблице. Такой способ кодирования прост и удобен, но он является не самым оптимальным. Ведь для значительной части символов используются не все биты отведенных под них байтов (часть старших битов – нулевые), а при наличии в тексте не всех символов, предусмотренных в кодовой таблице (например, если текст содержит только прописные русские буквы), приходится все равно использовать 8-битный код.

Более компактным является **неравномерный двоичный код**. В нем количества битов, отводимых для кодирования символов, зависят от количества используемых в конкретном случае различных символов (от *мощности алфавита*), а коды, соответствующие разным символам, могут иметь различную длину в битах. Такой код можно оптимизировать, если при его построении исходить из частоты встречаемости различных символов и присваивать наиболее часто используемым знакам самые короткие коды (как это делается в *методе Хаффмана* и при построении *дерева Фано*).

Главное при кодировании неравномерных кодов – обеспечить возможность **однозначного декодирования** записанной с их помощью строки. Такое декодирование обычно производится поочередным выделением и распознаванием из сплошной после-



довательности нулей и единиц кодов отдельных букв.

Чтобы обеспечить однозначное декодирование текста, закодированного при помощи неравномерных кодов, сами эти коды необходимо назначать символам 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. Такой код называют «префиксным».

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

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.

При решении конкретной задачи на неравномерное кодирование и декодирование прежде всего нужно проанализировать коды, заданные в условии. Если для исходных кодов выполняется прямое условие Фано, то его и надо использовать при решении. Если же прямое условие Фано не выполняется, то нужно проверить выполнение обратного условия Фано и, если оно выполнено, использовать его.

Далее, в зависимости от выполнения для имеющегося набора кодов прямого или обратного условия Фано, выбирается направление декодирования имеющегося кода:

- если для заданной последовательности кодов выполняется **прямое условие Фано**, то декодирование необходимо вести **с начала (слева направо)**;
- если для заданной последовательности кодов выполняется **обратное условие**

Фано, то декодирование необходимо вести **с конца (справа налево)**;

- если выполнены оба условия Фано, то декодирование можно выполнять в любом из двух указанных направлений – результат будет одним и тем же.

ПРОВЕРКА ВЫПОЛНЕНИЯ УСЛОВИЙ ФАНО

Чтобы проверить, выполняется ли прямое условие Фано, нужно по очереди сравнить друг с другом каждую пару кодов по следующим правилам:

- если оба сравниваемых кода имеют одинаковую длину в битах, то достаточно проверить, не совпадают ли они;
- если длина сравниваемых кодов различна, то проверяется, совпадает ли более короткий код с началом более длинного кода (проверка выполнения прямого условия Фано) или с концом более длинного кода (проверка выполнения обратного условия Фано).

В последнем случае проверку кодов можно выполнить следующим способом. Сначала нужно записать более длинный код. Затем надо подписать под ним более короткий код, выровняв его относительно верхнего кода: для проверки **прямого условия Фано** – по левой знаковой позиции, а для проверки **обратного условия Фано** – по правой знаковой позиции.

Пример

Пусть заданы следующие коды букв: А – 10, В – 11, С – 011. Требуется определить, выполняются ли для этого набора кодов условия Фано.

Решение

1. Сравниваем коды букв А и В. Длины этих кодов равны (оба имеют длину 2 бита). Сами коды (10 и 11) не совпадают. Следовательно, для этой пары кодов выполняются и прямое, и обратное условия Фано.
2. Сравниваем коды букв А и С. Длины этих кодов различны.
 - 2.1. Проверка выполнения прямого условия Фано:

С: $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} 1$
 А: $\begin{bmatrix} 1 & 0 \end{bmatrix}$

Код буквы А (более короткий) не совпадает с началом кода буквы С (более длинного). Следовательно, для этой пары кодов выполняется прямое условие Фано.

2.2. Проверка выполнения обратного условия Фано:

С: $0 \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$
 А: $\begin{bmatrix} 1 & 0 \end{bmatrix}$

Код буквы А (более короткий) не совпадает с концом кода буквы С (более длинного). Следовательно, для этой пары кодов выполняется обратное условие Фано.

3. Сравниваем коды букв В и С. Длины этих кодов тоже различны.

3.1. Проверка выполнения прямого условия Фано:

С: $\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} 1$
 В: $\begin{bmatrix} 1 & 1 \end{bmatrix}$

Код буквы В (более короткий) не совпадает с началом кода буквы С (более длинного). Следовательно, для этой пары кодов выполняется прямое условие Фано.

3.2. Проверка выполнения обратного условия Фано:

С: $0 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$
 В: $\begin{bmatrix} 1 & 1 \end{bmatrix}$

Код буквы В (более короткий) **совпал** с концом кода буквы С (более длинного). Следовательно, для этой пары кодов обратное условие Фано **не выполняется**.

4. Делаем выводы.

Прямое условие Фано выполнено для всех возможных в данном наборе пар кодов. Следовательно, **прямое условие Фано выполнено для всего данного набора кодов**.

Обратное условие Фано не выполнено для одной из возможных в данном наборе пар кодов. Следовательно, **обратное условие Фано не выполнено и для всего данного набора кодов**.

Ответ: для данного набора кодов выполняется прямое условие Фано, но не выполняется обратное условие Фано.

ПОСТРОЕНИЕ ДЕРЕВА ФАНО

Дерево Фано – это удобный и наглядный способ решения задач, связанных с подбо-

ром неравномерных двоичных кодов. Он связан с алгоритмом Шеннона-Фано и во многом – с алгоритмом Хаффмана, который используется в архиваторах (например, в ZIP).

Основные принципы построения дерева Фано:

1) учет частоты встречаемости символов в тексте – чем чаще встречается какой-либо символ, тем короче для него имеет смысл назначить код и тем раньше этот символ надо поместить в дерево;

2) каждый узел дерева Фано порождает ровно две ветви (то есть дерево Фано является двоичным, *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, а вторая 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;

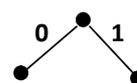
4) для каждого символа код Фано получается последовательной записью всех нулей и единиц по кратчайшему пути от вершины дерева к соответствующему символу.

Пример

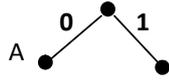
Пусть в тексте используются буквы А, Б, В, при этом чаще всего встречается буква А, а реже всего – буква В. Требуется сгенерировать для них наиболее оптимальные (кратчайшие) коды Фано.

Решение

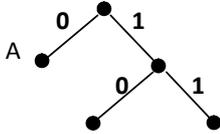
Сразу начинаем строить дерево, начиная с исходного узла:



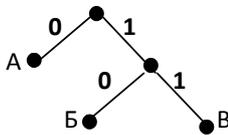
На одну из ветвей мы должны «повесить» первый, наиболее часто встречающийся символ А:



От второй ветви продолжаем построение дерева:

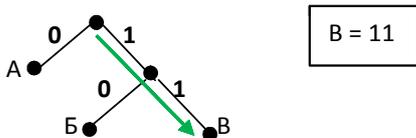
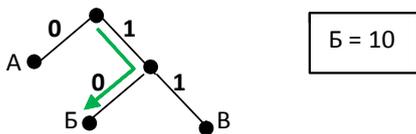
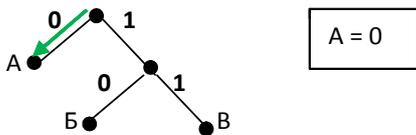


Так как наш алфавит состоит всего из трех символов, и у нас осталось только два последних, мы можем «закрыть» дерево, поместив эти два символа на концах только что построенных ветвей:



Это, конечно, один из нескольких допустимых вариантов решения – вполне можно было бы на первом шаге символ А «прикрепить» к правой ветви, продолжая построение через левую, и/или поменять местами Б и В. Кроме того, когда символов много, можно и обе ветви дерева отвести на его продолжение. Но задания ЕГЭ обычно формулируются так, что эти вариации на получаемый ответ не влияют, – например, спрашивается суммарная длина получаемых кодов.

А теперь можно генерировать для символов коды Фано, «пробегая» по дереву от его начала к нужному символу:



Ответ: оптимальные коды Фано: А = 0, Б = 10, В = 11. (Еще раз отметим, что это – один из возможных правильных ответов, могут быть и другие варианты.)

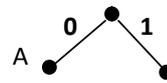
В реальных заданиях ЕГЭ обычно бывают изначально заданы коды Фано для части символов. В этом случае нужно при построении дерева Фано учитывать заданные места размещения этих символов, «развешивая» остальные символы на ветвях дерева уже по своему усмотрению.

А теперь перейдем к примерам задач.

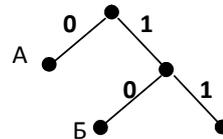
Задача 1. Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г, Д, Е, используется неравномерный двоичный код, удовлетворяющий условию Фано. Для буквы А использовано кодовое слово 0, для буквы Б – кодовое слово 10. Какова наименьшая возможная сумма длин всех шести кодовых слов?

Решение

Начинаем построение дерева Фано. При этом, так как для буквы А задан код 0, левая ветвь дерева (0) может вести от корня только к символу А, тогда как правая (1) послужит продолжению построения дерева:

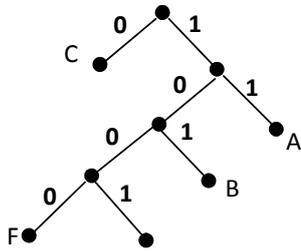


Для буквы Б также уже задан код 10. Это означает, что в правой части дерева новая ветвь 0 должна приводить к символу Б. Вторую ветвь (1) мы будем продолжать.

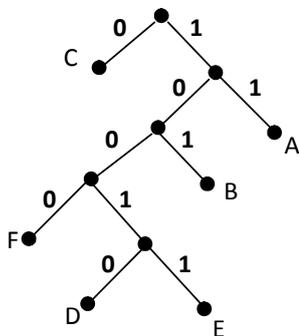


Теперь заранее заданные коды закончились, и мы можем свободно выстраивать оставшуюся часть дерева по своему усмотрению. При этом мы будем, прежде всего, «ветвить» дерево, закрывая символами его конечные вершины (можно проверить, что такой способ наиболее оптимален: если на каждом ветвлении «вешать» на одну из ветвей один из символов, то не удастся «закрыть» дерево последней парой):

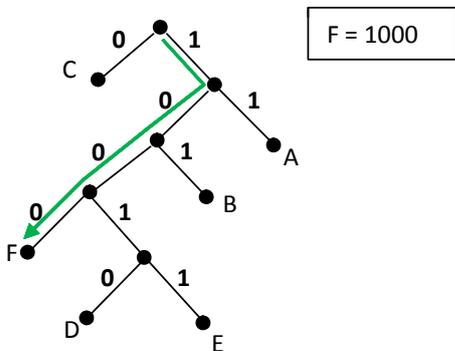
Вот здесь-то и становится важным второе требование: код буквы F должен быть наименьшим двоичным числом. А раз так, мы должны использовать ветвь 0, а не ветвь 1:



Остальные коды символов нас в условии задания не интересуют, но можно и достроить дерево, включив в него эти символы (D и E), – например, так:



А теперь генерируем по построенному дереву искомый код Фано для символа F:



Ответ: 1000.

Задача 3. В сообщении встречается 50 букв А, 30 букв Б, 20 букв В и 5 букв Г. При его передаче использован неравномерный

Богомолова Ольга Борисовна,
доктор педагогических наук, учитель
математики и информатики
ГБОУ СОШ № 1360, г. Москва.

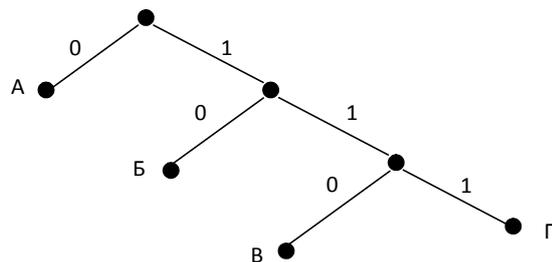
двоичный префиксный код, который позволил получить минимальную длину закодированного сообщения. Какова она в битах?

В этой задаче нет изначально заданных кодов, но зато учитывается частота встречаемости букв. Напомним, что термин «префиксный» означает, что речь идет о прямом условии Фано. Спрашивается же общая длина в битах получаемого сообщения, закодированного с использованием полученных кодов Фано.

Решение

Сначала нам не важны конкретные количества букв – важна только общая статистика, чтобы выстроить буквы по порядку уменьшения частоты их встречаемости в тексте. В нашем случае этот порядок уже задан: А, Б, В, Г, но, вообще говоря, могут встретиться и задания, где правильный порядок букв придется выстроить самому.

Теперь «выращиваем» дерево Фано, начиная с самых частых букв к менее частым:



Теперь определились коды букв: А – 0, Б – 10, В – 110, Г – 111. (Коды могут быть и другими – важно, что для более частых букв коды должны быть короче.)

Наконец, определяем общую длину сообщения в битах:

50 букв А с кодом из 1 бита + 30 букв Б с кодом из 2 бит + 20 букв В с кодом из 3 бит + 5 букв Г с кодом также из 3 бит = $50 \times 1 + 30 \times 2 + 20 \times 3 + 5 \times 3 = 50 + 60 + 60 + 15 = 185$ бит.

Ответ: 185 бит.

Усенков Дмитрий Юрьевич,
Московский государственный
институт индустрии туризма
имени Ю.А. Сенкевича, г. Москва.