Практическая Работа № 5 Практическое применение различных алгоритмов сжатия

Цель работы: научиться сжимать информацию с помощью метода Хаффмана и метода RLE.

Методические указания:

Код Хаффмана

Определение 1: Пусть $A=\{a_1,a_2,...,a_n\}$ - алфавит из n различных символов, $W=\{w_1,w_2,...,w_n\}$ - соответствующий ему набор положительных целых весов. Тогда набор бинарных кодов $C=\{c_1,c_2,...,c_n\}$, такой что:

1) c_i не является префиксом для c_j, при i!=j
2) $\sum_{i=1}^{n} w_i|c_i|$ минимальна ($|c_i|$ длина кода c_i)

называется *минимально-избыточным префиксным кодом* или иначе *кодом Хаффмана*.

Замечания:

1. Свойство (1) называется *свойством префиксности*. Оно позволяет однозначно декодировать коды переменной длины.
2. Сумму в свойстве (2) можно трактовать как размер закодированных данных в битах. На практике это очень удобно, т.к. позволяет оценить степень сжатия не прибегая непосредственно к кодированию.
3. В дальнейшем, чтобы избежать недоразумений, под кодом будем понимать битовую строку определенной длины, а под минимально-избыточным кодом или кодом Хаффмана - множество кодов (битовых строк), соответствующих определенным символам и обладающих определенными свойствами.

Известно, что любому бинарному префиксному коду соответствует определенное бинарное дерево.

Определение 2: Бинарное дерево, соответствующее коду Хаффмана, будем называть *деревом Хаффмана*.

Задача построения кода Хаффмана равносильна задаче построения соответствующего ему дерева. Приведем общую схему построения дерева Хаффмана:

1. Составим список кодируемых символов (при этом будем рассматривать каждый символ как одноэлементное бинарное дерево, вес которого равен весу символа).
2. Из списка выберем 2 узла с наименьшим весом.
3. Сформируем новый узел и присоединим к нему, в качестве дочерних, два узла выбранных из списка. При этом вес сформированного узла положим равным сумме весов дочерних узлов.
4. Добавим сформированный узел к списку.
5. Если в списке больше одного узла, то повторить 2-5.

Приведем пример: построим дерево Хаффмана для сообщения S="A H F B H C E H E H C E A H D C E E H H H C H H H D E G H G G E H C H H".

Для начала введем несколько обозначений:

1. Символы кодируемого алфавита будем выделять жирным шрифтом: **A**, **B**, **C**.
2. Веса узлов будем обозначать нижними индексами: $\mathbf{A}_5$, $\mathbf{B}_3$, $\mathbf{C}_7$.
3. Составные узлы будем заключать в скобки: $((\mathbf{A}_5+\mathbf{B}_3)_8+\mathbf{C}_7)_{15}$.

Итак, в нашем случае A={**A**, **B**, **C**, **D**, **E**, **F**, **G**, **H**}, W={2, 1, 5, 2, 7, 1, 3, 15}.

1. $\mathbf{A}_2\ \mathbf{B}_1\ \mathbf{C}_5\ \mathbf{D}_2\ \mathbf{E}_7\ \mathbf{F}_1\ \mathbf{G}_3\ \mathbf{H}_{15}$
2. $\mathbf{A}_2\ \mathbf{C}_5\ \mathbf{D}_2\ \mathbf{E}_7\ \mathbf{G}_3\ \mathbf{H}_{15}\ (\mathbf{F}_1+\mathbf{B}_1)_2$
3. $\mathbf{C}_5\ \mathbf{E}_7\ \mathbf{G}_3\ \mathbf{H}_{15}\ (\mathbf{F}_1+\mathbf{B}_1)_2\ (\mathbf{A}_2+\mathbf{D}_2)_4$
4. $\mathbf{C}_5\ \mathbf{E}_7\ \mathbf{H}_{15}\ (\mathbf{A}_2+\mathbf{D}_2)_4\ ((\mathbf{F}_1+\mathbf{B}_1)_2+\mathbf{G}_3)_5$
5. $\mathbf{E}_7\ \mathbf{H}_{15}\ ((\mathbf{F}_1+\mathbf{B}_1)_2+\mathbf{G}_3)_5\ (\mathbf{C}_5+(\mathbf{A}_2+\mathbf{D}_2)_4)_9$
6. $\mathbf{H}_{15}\ (\mathbf{C}_5+(\mathbf{A}_2+\mathbf{D}_2)_4)_9\ (((\mathbf{F}_1+\mathbf{B}_1)_2+\mathbf{G}_3)_5+\mathbf{E}_7)_{12}$
7. $\mathbf{H}_{15}\ ((\mathbf{C}_5+(\mathbf{A}_2+\mathbf{D}_2)_4)_9+(((\mathbf{F}_1+\mathbf{B}_1)_2+\mathbf{G}_3)_5+\mathbf{E}_7)_{12})_{21}$
8. $(((\mathbf{C}_5+(\mathbf{A}_2+\mathbf{D}_2)_4)_9+(((\mathbf{F}_1+\mathbf{B}_1)_2+\mathbf{G}_3)_5+\mathbf{E}_7)_{12})_{21}+\mathbf{H}_{15})_{36}$

В списке, как и требовалось, остался всего один узел. Дерево Хаффмана построено. Теперь запишем его в более привычном для нас виде.

```
                ROOT
                 /\
                0  1
               /    \
              /\     H
               / \
              /   \
             /     \
            0       1
           /         \
          /           \
         /             \
        /               \
       /\               /\
      0  1             0  1
     /    \           /    \
    C     /\         /\     E
         0  1       0  1
        /    \     /    \
       A      D   /\     G
                 0  1
                /    \
               F      B
```

Листовые узлы дерева Хаффмана соответствуют символам кодируемого алфавита. Глубина листовых узлов равна длине кода соответствующих символов.

Путь от корня дерева к листовому узлу можно представить в виде битовой строки, в которой "0" соответствует выбору левого поддерева, а "1" - правого. Используя этот механизм, мы без труда можем присвоить коды всем символам кодируемого алфавита. Выпишем, к примеру, коды для всех символов в нашем примере:

A=0010_{bin}	**C**=000_{bin}	**E**=011_{bin}	**G**=0101_{bin}
B=01001_{bin}	**D**=0011_{bin}	**F**=01000_{bin}	**H**=1_{bin}

Теперь у нас есть все необходимое для того чтобы закодировать сообщение S. Достаточно просто заменить каждый символ соответствующим ему кодом:

$S^/$="0010 1 01000 01001 1 000 011 1 011 1 000 011 0010 1 0011 000 011 011 1 1 1 000 1 1 1 0011 011 0101 1 0101 0101 011 1 000 1 1".

Оценим теперь степень сжатия. В исходном сообщении S было 36 символов, на каждый из которых отводилось по $[\log_2|A|]=3$ бита (здесь и далее будем понимать квадратные скобки [] как целую часть, округленную в положительную сторону, т.е. [3,018]=4). Таким образом, размер S равен 36*3=108 бит

Размер закодированного сообщения $S^/$ можно получить воспользовавшись замечанием 2 к определению 1, или непосредственно, подсчитав количество бит в $S^/$. И в том и другом случае мы получим 89 бит.

Итак, нам удалось сжать 108 в 89 бит.

Теперь декодируем сообщение $S^/$. Начиная с корня дерева будем двигаться вниз, выбирая левое поддерево, если очередной бит в потоке равен "0", и правое - если "1". Дойдя до листового узла мы декодируем соответствующий ему символ.

Ясно, что следуя этому алгоритму мы в точности получим исходное сообщение S.

Метод RLE.

Наиболее известный простой подход и алгоритм сжатия информации обратимым путем - это кодирование серий последовательностей (Run Length Encoding - RLE). Суть методов данного подхода состоит в замене цепочек или серий повторяющихся байтов или их последовательностей на один кодирующий байт и счетчик числа их повторений. Проблема всех аналогичных методов заключается лишь в определении способа, при помощи которого распаковывающий алгоритм мог бы отличить в результирующем потоке байтов кодированную серию от других - некодированных последовательностей байтов. Решение проблемы достигается обычно простановкой меток в начале кодированных цепочек. Такими метками могут быть, например, характерные значения битов в первом байте кодированной серии, значения первого байта кодированной серии и т.п. Данные методы, как правило, достаточно эффективны для сжатия растровых графических изображений (BMP, PCX, TIF, GIF), т.к. последние содержат достаточно много длинных серий повторяющихся последовательностей байтов. Недостатком метода RLE является достаточно низкая степень сжатия или стоимость кодирования файлов с малым числом серий и, что еще хуже - с малым числом повторяющихся байтов в сериях.

Задание

1. Сжатие методом Хаффмана

«КАКАЯ ЗИМА ЗОЛОТАЯ!
КАК БУДТО ИЗ ДЕТСКИХ ВРЕМЕН...
НЕ НАДО НИ СОЛНЦА, НИ МАЯ –
ПУСТЬ ДЛИТСЯ ТОРЖЕСТВЕНИЫЙ СОН.

ПУСТЬ Я В ЭТОМ СНЕ ПОЗАБУДУ
КОГДА-ТО МАНИВШИЙ ОГОНЬ,
И ЛЕТО ПРЕДАМ, КАК ИУДА,
ЗА ТРИДЦАТЬ СНЕЖИНОК В ЛАДОНЬ.

ЗАТЕМ, ЧТО И Я ХОЛОДЕЮ,
ТЕПЛО УЖЕ СТРАШНО ПРИНЯТЬ:
Я СЛИШКОМ ДАВНО НЕ УМЕЮ
НИ ТЛЕТЬ, НИ ГОРЕТЬ, НИ СЖИГАТЬ…

ВСЕ ЧАЩЕ, ВСЕ ДОЛЬШЕ НЕМЕЮ:
К ЗИМЕ УЖЕ ДЕЛО, К ЗИМЕ...
И ТОЛЬКО ТОГО ОТОГРЕЮ,
КОМУ ХОЛОДНЕЕ, ЧЕМ МНЕ»

2. С помощью сжатия по методу RLE.

1 последовательность:

SSSSOOOEEERROOOAAYYYYYDDDDOEUUUUUWWWWJJORRUUUUUUUUUUXXXKHHHHHHMMMMMMGGGLLLLLLLJJJJ

2 последовательность:

FFFFFFFFKKKKKSSSSUURERRRRRRRRRPPPPPPPPDDDDKKKKKKGLDDDDDDDDKKKKKKKKGGGGMGMMMM

3. Создайте презентацию по теме «Алгоритмы сжатия изображений». Используйте ресурсы Интернет.

Контрольные вопросы:

1. Что такое код Хаффмана?
2. Что называется деревом Хаффмана?
3. Как происходит сжатие методом Хаффмана?
4. Как происходит сжатие по методу RLE?
5. Назовите расширения растровых графических изображений?