МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ

по лабораторной работе №5

по дисциплине «Алгоритмы и структуры данных»

Тема: Кодирование Фано-Шеннона

Студентка гр. 7383 _____ Чемова К.А.

Преподаватель _____ Размочаева Н.В.

Санкт-Петербург

2018

Содержание

Цель работы

Цель работы: познакомиться с алгоритмом кодирования Фано-Шеннона и его реализацией на языке программирования C++.

Формулировка задачи:

На вход подаётся файл, содержимое которого требуется закодировать алгоритмом Фано-Шеннона.

Реализация задачи

Структура `struct Elem` представляет собой элемент бинарного дерева.

```
struct Elem {
    char c;
    string value;
    Elem* left;
    Elem* right;
     };
```

Структура `struct Node` представляет собой элемент массива структур, содержащий в себе букву, число ее повторений и указатель на соответствующий элемент бинарного дерева.

```
struct Node {
    int num;
    Elem* ptr;
    char value;
     };
```

Функция `void SearchTree` кодирует дерево и создает таблицу кодирования.

Функция `void Encode` кодирует дерево.

Функция `MakeList` печатает дерево.

Тестирование

Программа была собрана в компиляторе g++ в OS Linux Ubuntu 16.04 при помощи g++. В других системах тестирование не проводилось. Результаты тестирования приведены в приложении А.

Выводы

В ходе выполнения работы были изучены алгоритм кодирования. Была написана программа, кодирующая текст алгоритмом Фано-Шеннона

ПРИЛОЖЕНИЕ А

ТЕСТОВЫЕ СЛУЧАИ

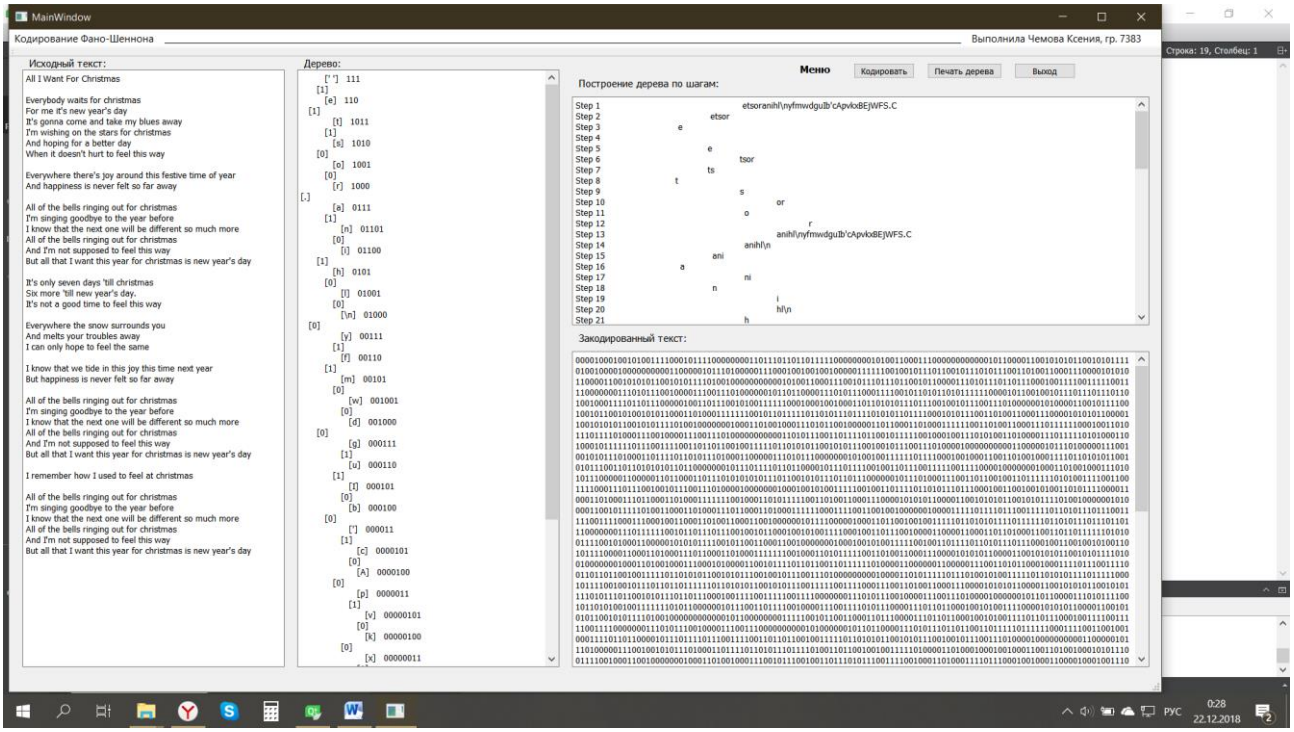

Рисунок 1 – Тестовый случай 1

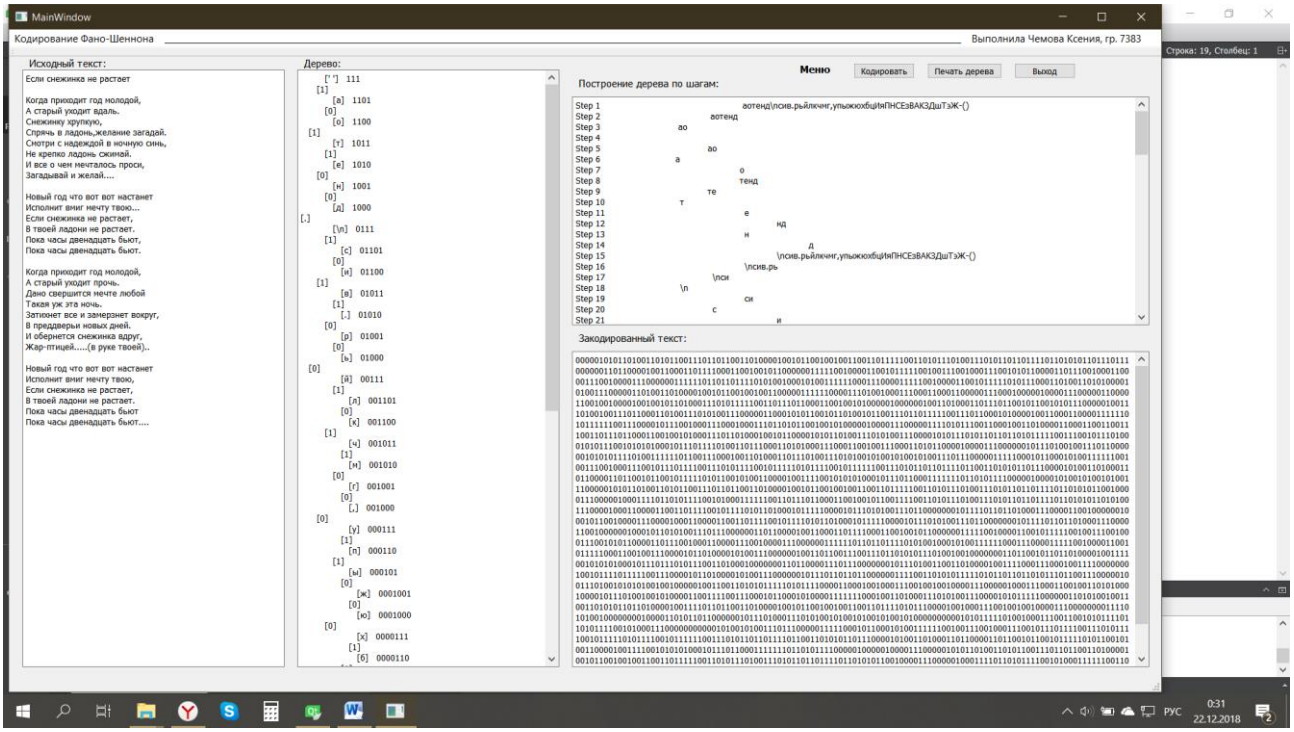

Рисунок 1 – Тестовый случай 2

ПРИЛОЖЕНИЕ Б

КОД ПРОГРАММЫ

lab5.pro:

```
#-------------------------------------------------
#
# Project created by QtCreator 2018-12-21T21:17:00
#
#-------------------------------------------------

QT       += core gui

greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

TARGET = lab5
TEMPLATE = app


SOURCES += main.cpp\
        mainwindow.cpp

HEADERS  += mainwindow.h

FORMS    += mainwindow.ui
```

mainwindow.h:

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H
#include <iostream>
#include <cstdlib>
#include <vector>
#include <list>
#include <QFileDialog>
#include <QMessageBox>
#include <QDebug>
#include <QMainWindow>

using namespace std;

struct Elem{
    QChar c;
    QString value;
    Elem* parent;
```

```
    Elem* left;
    Elem* right;
};

struct Node{
    int num;
    Elem* ptr;
    QChar value;
};

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow {
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();
    void MakeList(Elem* t);

private slots:
    void ChooseANDencode();

    void on_close();

    void PrintTree();

private:
    Ui::MainWindow *ui;
};

#endif // MAINWINDOW_H
```

main.cpp:

```
#include "mainwindow.h"
#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
```

```
    return a.exec();
}
```

mainwindow.cpp:

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QString>
using namespace std;

Node* arr = NULL;
Elem* head = NULL;
int index = 0;
QString binary = "";

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow){
    ui->setupUi(this);
    connect(ui-
>ChooseEncode,SIGNAL(clicked()),this,SLOT(ChooseANDencode()));
    connect(ui->PrintTree,SIGNAL(clicked()),this,SLOT(PrintTree()));
    connect(ui->on_close,SIGNAL(clicked()),this,SLOT(on_close()));
}

MainWindow::~MainWindow(){
    delete ui;
}

int cmp(const void* a,const void* b){
    return (*(Node*)b).num-(*(Node*)a).num;
}

void iteration (Elem** root){
    if (*root){
        iteration(&(*root)->left);
        iteration(&(*root)->right);
                free(*root);
    }
}

Elem* buildTree(QChar el){
    Elem* head = new Elem;
    head->c = el;
    head->parent = NULL;
```

```
    head->value = "";
    head->left = NULL;
    head->right = NULL;
    return head;
}

void SearchTree(Elem** t,Node* el,QString &branch,QString &fullBranch,int
start,int end,QString &steps,QChar uzel,int &ind,int tab){
    char st[10];
    steps = steps + "Step " + itoa(ind++,st,10) + "  ";
    for (int i=0;i<tab;i++)
        steps = steps + "  ";
    for (int i = start;i<=end;i++){
        if (el[i].value == '\n')
            steps = steps + "\\n";
        else steps = steps + el[i].value;
    }
    steps = steps + "\n";
    if (*t == NULL){
        *t = new Elem;
        (*t)->parent = NULL;
        (*t)->left = NULL;
        (*t)->right = NULL;
        (*t)->value = "";
        (*t)->c = uzel;
    }
    double dS=0;
    int i,S=0;
    QString cBranch="";
    cBranch=fullBranch+branch;
    if (start==end)
    {
        (*t)->value += cBranch;
        (*t)->c = el[start].value;
        el[start].ptr = *t;
        return;
    }
    for (i=start;i<=end;i++)
        dS+=el[i].num;
    dS/=2.;
    i=start+1;
    S+=el[start].num;
    while (fabs(dS-(S+el[i].num))<fabs(dS-S) && (i<end))
    {
```

```
        S+=el[i].num;
        i++;
    }
    QChar o = '1';
    QChar z = '0';
    QString zero="0";
    QString one="1";
    SearchTree(&(*t)->left,el,one,cBranch,start,i-1,steps,o,ind,tab-6);
    SearchTree(&(*t)->right,el,zero,cBranch,i,end,steps,z,ind,tab+6);
}

void Encode(Node* arr,int index,QString copy,QString &binary){
    for (int i=0;copy[i]!='\0';i++){
        for (int j=0;j<index;j++){
            if (copy[i] == arr[j].value)
                binary += arr[j].ptr->value;
        }
    }
}

void print(Elem *t,QString &out,int count){
   if (t != NULL){
       print(t->left,out,count+1);
       for (int i = 0;i < count;i++)
           out = out + "   ";
       if (t->c == '\n')
           out = out + "[\\n]" + "   " + t->value + "\n";
       else if (t->c == ' ')
           out = out + "[' ']" + "   " + t->value + "\n";
       else out = out + "[" + t->c + "]   " + t->value + "\n";
       print(t->right,out,count+1);
   }
}

void MainWindow::MakeList(Elem* t){
    if (t == NULL){
        ui->ResultEdit->setText("Tree is empty");
    }
    else{
        QString tr;
        print(t,tr,0);
        ui->ResultEdit->setText(QString(tr.data()));
    }
}
```

```
void Decode(Node* arr,QString &answer,QString binary,int index){
//Декодирование дерева
    int len = 0;
    QString ptr = "";
    while (len <= binary.length()){
        ptr += binary[len++];
        for (int i = 0; i < index;i++){
            if(ptr == arr[i].ptr->value) {
                answer += arr[i].value;
                ptr.clear();
            }
        }
    }
}

void MainWindow::ChooseANDencode(){ //Кодирует дерево
    int k;
    QString str = "",copy = "",all = "";
    QString filename = QFileDialog::getOpenFileName(this,tr("Open
file"),"D://lab5//text//","Text file (*.txt)");
    QFile file(filename.toStdString().data());
    if (!file.open(QIODevice::ReadOnly))
        ui->ResultEdit->setText("File can not be opened");
    else{
        free(arr);
        iteration(&head);
        head = NULL;
        binary.clear();
        ui->ResultEdit->clear();
        ui->BinaryEdit->clear();
        ui->CodesEdit->clear();
        ui->PreEdit->clear();
        index = 0;
        arr = (Node*) malloc(sizeof(Node));
        QTextStream in(&file);
        all = in.readLine();
        if (in.atEnd()){
            ui->ResultEdit->setText("File is empty");
            ui->BinaryEdit->setText("File is empty");
            ui->CodesEdit->setText("File is empty");
            ui->PreEdit->setText("File is empty");
            return;
        }
```

```
int c=0;
str += all + "\n";
while (!in.atEnd()){
    all = in.readLine();
    str += all + "\n";
}
ui->PreEdit->setText(str);
copy += str;
for (int j=0;str[j] != '\0';j++){
    k = 0;
    for (int i=j+1;str[i] != '\0';i++){
        if (str[j] == str[i]){
            while (str[j] == str[i]){
                str.remove(i,1);
                k++;
            }
        }
    }
    k++;
    index++;
    arr = (Node*) realloc (arr,index*sizeof(Node));
    if (arr != NULL){
        arr[index-1].num = k;
        arr[index-1].value = str[j];
        arr[index-1].ptr = NULL;
    }
    else {
        free(arr);
        ui->ResultEdit->setText("Error with allocation");
        break;
    }
}
if (c != index){
    iteration(&head);
    head = NULL;
}
file.close();
qsort(arr,index,sizeof(Node),cmp);
int ind = 1;
QChar sym = '.';
QString a = "",b = "",symbols = "";
ind = 1;
int tab = 25;
SearchTree(&head,arr,a,b,0,index-1,symbols,sym,ind,tab);
```

```
        ui->CodesEdit->setText(symbols);
        Encode(arr,index,copy,binary);
        if (binary.isEmpty())
            ui->BinaryEdit->setText("Binary is empty");
        else ui->BinaryEdit->setText(binary);
   }
}

void MainWindow::on_close(){
    this->close();
}

void MainWindow::PrintTree(){
    MakeList(head);
}
```

mainwindow.ui:

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
 <class>MainWindow</class>
 <widget class="QMainWindow" name="MainWindow">
  <property name="geometry">
   <rect>
    <x>0</x>
    <y>0</y>
    <width>1719</width>
    <height>988</height>
   </rect>
  </property>
  <property name="windowTitle">
   <string>MainWindow</string>
  </property>
  <widget class="QWidget" name="centralWidget">
   <widget class="QPushButton" name="ChooseEncode">
    <property name="geometry">
     <rect>
      <x>1260</x>
      <y>10</y>
      <width>91</width>
      <height>23</height>
     </rect>
    </property>
    <property name="text">
     <string>Кодировать</string>
```

```
 </property>
</widget>
<widget class="QTextEdit" name="ResultEdit">
 <property name="geometry">
  <rect>
   <x>430</x>
   <y>20</y>
   <width>391</width>
   <height>891</height>
  </rect>
 </property>
 <property name="readOnly">
  <bool>true</bool>
 </property>
</widget>
<widget class="QTextEdit" name="CodesEdit">
 <property name="geometry">
  <rect>
   <x>840</x>
   <y>60</y>
   <width>861</width>
   <height>341</height>
  </rect>
 </property>
 <property name="readOnly">
  <bool>true</bool>
 </property>
</widget>
<widget class="QTextEdit" name="BinaryEdit">
 <property name="geometry">
  <rect>
   <x>840</x>
   <y>440</y>
   <width>861</width>
   <height>471</height>
  </rect>
 </property>
 <property name="readOnly">
  <bool>true</bool>
 </property>
</widget>
<widget class="QPushButton" name="on_close">
 <property name="geometry">
  <rect>
```

```
   <x>1500</x>
   <y>10</y>
   <width>91</width>
   <height>23</height>
  </rect>
 </property>
 <property name="text">
  <string>Выход</string>
 </property>
</widget>
<widget class="QTextEdit" name="PreEdit">
 <property name="geometry">
  <rect>
   <x>20</x>
   <y>20</y>
   <width>391</width>
   <height>891</height>
  </rect>
 </property>
 <property name="readOnly">
  <bool>true</bool>
 </property>
</widget>
<widget class="QPushButton" name="PrintTree">
 <property name="geometry">
  <rect>
   <x>1370</x>
   <y>10</y>
   <width>111</width>
   <height>23</height>
  </rect>
 </property>
 <property name="text">
  <string>Печать дерева</string>
 </property>
</widget>
<widget class="QLabel" name="label_3">
 <property name="geometry">
  <rect>
   <x>850</x>
   <y>410</y>
   <width>161</width>
   <height>21</height>
  </rect>
```

```
    </property>
    <property name="text">
     <string>&lt;html&gt;&lt;head/&gt;&lt;body&gt;&lt;p&gt;&lt;span
style=&quot; font-size:9pt;&quot;&gt;Закодированный
текст:&lt;/span&gt;&lt;/p&gt;&lt;/body&gt;&lt;/html&gt;</string>
    </property>
   </widget>
   <widget class="QLabel" name="label">
    <property name="geometry">
     <rect>
      <x>850</x>
      <y>30</y>
      <width>221</width>
      <height>21</height>
     </rect>
    </property>
    <property name="text">
     <string>&lt;html&gt;&lt;head/&gt;&lt;body&gt;&lt;p&gt;&lt;span
style=&quot; font-size:9pt;&quot;&gt;Построение дерева по
шагам:&lt;/span&gt;&lt;/p&gt;&lt;/body&gt;&lt;/html&gt;</string>
    </property>
   </widget>
   <widget class="QLabel" name="label_5">
    <property name="geometry">
     <rect>
      <x>30</x>
      <y>0</y>
      <width>121</width>
      <height>21</height>
     </rect>
    </property>
    <property name="text">
     <string>&lt;html&gt;&lt;head/&gt;&lt;body&gt;&lt;p&gt;&lt;span
style=&quot; font-size:9pt;&quot;&gt;Исходный
текст:&lt;/span&gt;&lt;/p&gt;&lt;/body&gt;&lt;/html&gt;</string>
    </property>
   </widget>
   <widget class="QLabel" name="label_2">
    <property name="geometry">
     <rect>
      <x>440</x>
      <y>0</y>
      <width>61</width>
      <height>21</height>
```

```
     </rect>
    </property>
    <property name="text">
     <string>&lt;html&gt;&lt;head/&gt;&lt;body&gt;&lt;p&gt;&lt;span
style=&quot; font-
size:9pt;&quot;&gt;Дерево:&lt;/span&gt;&lt;/p&gt;&lt;/body&gt;&lt;/html&g
t;</string>
    </property>
   </widget>
   <widget class="QLabel" name="label_4">
    <property name="geometry">
     <rect>
      <x>1180</x>
      <y>10</y>
      <width>55</width>
      <height>16</height>
     </rect>
    </property>
    <property name="text">
     <string>&lt;html&gt;&lt;head/&gt;&lt;body&gt;&lt;p&gt;&lt;span
style=&quot; font-size:9pt; font-
weight:600;&quot;&gt;Меню&lt;/span&gt;&lt;/p&gt;&lt;/body&gt;&lt;/html&gt
;</string>
    </property>
   </widget>
  </widget>
  <widget class="QMenuBar" name="menuBar">
   <property name="geometry">
    <rect>
     <x>0</x>
     <y>0</y>
     <width>1719</width>
     <height>26</height>
    </rect>
   </property>
   <widget class="QMenu" name="menu">
    <property name="title">
     <string>Кодирование Фано-Шеннона</string>
    </property>
   </widget>
   <widget class="QMenu" name="menu_7383">
    <property name="title">

<string>_____
```

```
_____
_____</string>
    </property>
   </widget>
   <widget class="QMenu" name="menu_7384">
    <property name="title">
     <string>Выполнила Чемова Ксения, гр. 7383</string>
    </property>
   </widget>
   <addaction name="menu"/>
   <addaction name="menu_7383"/>
   <addaction name="menu_7384"/>
  </widget>
  <widget class="QToolBar" name="mainToolBar">
   <attribute name="toolBarArea">
    <enum>TopToolBarArea</enum>
   </attribute>
   <attribute name="toolBarBreak">
    <bool>false</bool>
   </attribute>
  </widget>
  <widget class="QStatusBar" name="statusBar"/>
 </widget>
 <layoutdefault spacing="6" margin="11"/>
 <resources/>
 <connections/>
</ui>
```