

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 51

ОТЧЕТ ЗАЩИЩЕН С ОЦЕНКОЙ _____

ПРЕПОДАВАТЕЛЬ

доцент, канд. техн. наук

должность, уч. степень, звание

Е.М. Линский

подпись, дата

инициалы, фамилия

ОТЧЕТ О КУРСОВОЙ РАБОТЕ

Алгоритм Фано

по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 5013

подпись, дата

Д. В. Сергеев

инициалы, фамилия

Санкт-Петербург 2021

СОДЕРЖАНИЕ

1	ПОСТАНОВКА ЗАДАЧИ.....	3
1.1	ЗАДАНИЕ	3
1.2	СМЫСЛ ЗАДАЧИ.....	3
2	АЛГОРИТМ	3
2.1	ИДЕЯ	3
2.2	ОПИСАНИЕ АЛГОРИТМА	3
2.3	ПРИМЕР ВЫПОЛНЕНИЯ АЛГОРИТМА.....	4
2.4	СТРУКТУРЫ ДАННЫХ, ИСПОЛЬЗУЕМЫЕ В АЛГОРИТМЕ.....	6
2.5	ПСЕВДОКОД АРХИВАЦИИ.....	6
2.6	ПСЕВДОКОД РАЗАРХИВАЦИИ	9
2.7	АНАЛИЗ СЛОЖНОСТИ.....	9
3	ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ	10
3.1	ЗАПУСК ПРОГРАММЫ.....	10
3.2	ФОРМАТ ВХОДНОГО ФАЙЛА.....	10
3.3	ФОРМАТ ВЫХОДНОГО ФАЙЛА	11
4	ТЕСТОВЫЕ ПРИМЕРЫ.....	11
5	СПИСОК ЛИТЕРАТУРЫ.....	12

1 Постановка задачи

1.1 Задание

Задачей данной курсовой работы является разработка программы, которая выполняет архивацию и разархивацию файла алгоритмом Фано. Разархивированный файл должен в точности повторять файл до архивации.

Алгоритм, используемый программой описан в [1].

1.2 Смысл задачи

Задача имеет смысл, так как уменьшая занимаемое файлом место, экономится собственно место на диске.

2 Алгоритм

2.1 Идея

Алгоритм предложен американским математиком Роберто Фано, идея состоит в том, чтобы сократить избыточность таблицы символов ASCII заменив коды более часто встречающихся символов более короткими последовательностями.

2.2 Описание алгоритма

Алгоритм сжимает, опираясь на частоту встречаемости символов в данном файле. Он кодирует наиболее часто встречаемые символы кодами меньшей длины, а редко встречаемые символы кодами большей длины. Таким образом, заменяя символы на их соответствующие коды, полученные в ходе анализа дерева Фано можно сжать файл.

Самым главным отличием алгоритма Фано от других алгоритмов сжатия является построения дерева кодов (дерева Фано). Это дерево строиться следующим образом: отсортированные по возрастанию ребра делятся пополам и образуют две вершины дерева. Эти вершины в свою очередь тоже делятся

пополам по такому же принципу. Рекурсия продолжается до тех пор, пока вершины не станут листьями дерева.

Таким образом формируется дерево Фано. Чтобы узнать код символа, нужно пройти по дереву сверху вниз и записывать, какие вершины образуют путь к этому символу. И следует условиться, какие ребра считать “1”, а какие “0” (К примеру, правое ребро – это “1”, левое ребро – это “0”).

Разархивирование выполняется теми же функциями. За исключением того, что функция читает из сжатого файла символы и их частоту напрямую, потому что во время сжатия алгоритм записал в файл дополнительную информацию о символах.

2.3 Пример выполнения алгоритма

Архивировать алгоритмом Фано “test.txt”.

test.txt:

kursovaya

Алгоритм вычисляет частоту встречаемости каждого символа и отсортировать их по возрастанию (таблица 1)

Таблица 1 - Таблица символов и их частот

Символ	Частота встречаемости
‘k’	1
‘u’	1
‘r’	1
‘s’	1
‘o’	1
‘v’	1
‘y’	1
‘a’	2

Формирование дерева Фано (рисунок 1):

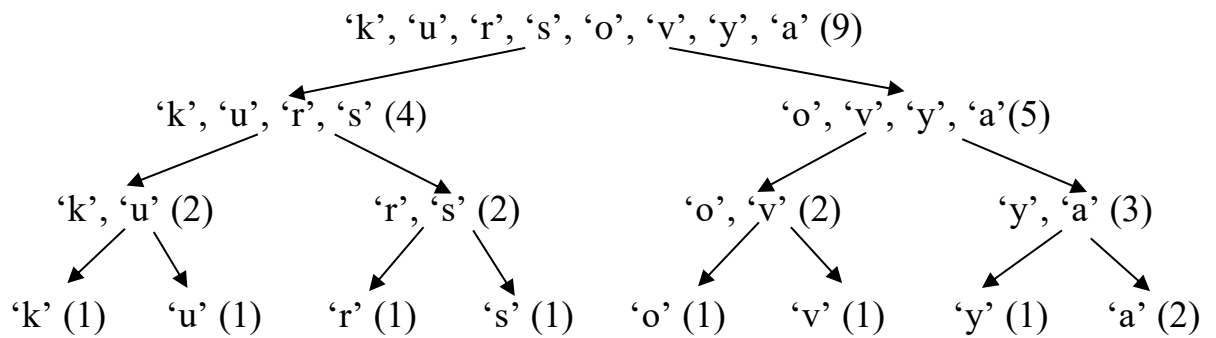


Рисунок 1 - Формирование дерева Фано

Формирование кодов Фано (таблица 2):

Таблица 2 - Таблица символов и их кодов

Символ	Код Фано
'k'	000
'u'	001
'r'	010
's'	011
'o'	100
'v'	101
'y'	110
'a'	111

Запись в файл с помощью кодов Фано:

000 001 01(‘\ENQ’) 0 011 100 1(‘9’) 01 111 110(‘}’) 111 00000(‘a’)
= _9}a (4 байта).

2.4 Структуры данных, используемые в алгоритме

В данном алгоритме используются структуры данных:

- class FanoCompression – главный класс, где происходят все преобразования
- struct SymbolRate– структура для хранения символа – его частоты встречаемости в тексте
- std::list<SymbolRate> SRate– структура данных для хранения всех символов с частотами
- std::map<char, vector<int>>* fMap– структура данных для хранения символов с их частотами
- class Tree– структура данных для создания дерева Фано
- class Node– структура данных для хранения узлов дерева

2.5 Псевдокод архивации

```
main() {  
    Подсчёт символов из файла;  
    Составление дерева Фано;  
    Составление кодов Фано;  
    Запись в новый файл частоту встречаемости символов и кодов  
    Фано;  
}
```

```
Подсчёт символов из файла {  
    создать List<символ, количество повторов>, для записи  
    символов;
```

```

while( не закончится файл ){
    Считать очередной символ;
    if( этот символ уже есть в List ){
        Прибавить 1 к количеству повторов этого символа;
    }
    else{
        Добавить новый узел с символами;
    }
}
Отсортировать по возрастанию полученный List;
}

```

```

Составление дерева Фано {
    Разделить list на две равные по количеству повторов части
    if( в первой половине не 1 элемент ){
        Рекурсивно вызвать эту же функцию с первой половиной
list;
    }
    if( во второй половине не 1 элемент )
        Рекурсивно вызвать эту же функцию со второй половиной
list;
    }
    Создать новый узел дерева с 2 сыновьями;
    Вернуть этот узел;
}

```

```

Составление кодов Фано {
    //Начать обход дерева с его корня
    if( если это первый шаг рекурсии ){

```

```

        Создать map для символов и их кодов Фано;
        Создать Buffer где будут храниться путь обхода дерева;
    }

    if ( текущий узел является листом дерева ){
        Добавить в map символ из текущего узла и значение
Buffer-a
    else{
        Добавить в Buffer 0;
        Вызвать рекурсивно функцию для левого потомка;
        Удалить из буфера 0;

        Добавить в Buffer 1;
        Вызвать рекурсивно функцию для правого потомка;
        Удалить из буфера 1;
    }
}
}

```

```

Запись в новый файл частоту встречаемости символов и кодов
Фано{
    Создать Buffer, где хранятся биты для записи в файл;
    while( не закончился файл ){
        Считать очередной символ;
        Добавить его код в Buffer;
        while (Buffer.size() > 8 (т.к. 1 символ состоит из 8
битов)){
            Записать первые 8 битов преобразовав их в
символ;

```



```

    }
}
}

```

2.6 Псевдокод разархивации

```

main() {
    Подсчёт символов и их частоты файла;
    Составление дерева Фано; //такое же, как и в архивации
    Составление кодов Фано; //такое же, как и в архивации
    Запись разархивированных символов;
}

Запись разархивированных символов {
    Создать Buffer для хранения кодов символов;
    Считать все символы из файла и поместить их коды в Buffer;
    while(Buffer.Size() != 0){
        Найти символ код которого совпадает с первыми N
        символами из Buffer;
    }
}

```

2.7 Анализ сложности

Пусть размер исходного файла N , а количество различных символов (длина алфавита) L , причём $1 \leq L \leq 256$

- Анализ сложности архивации:

Подсчёт символов имеет сложность $O(N)$

Составление дерева Фано имеет сложность $O(L \log_2 L)$ так как нужно выполнить задачу суммирования от 1 до $L-1$ объектов за каждый шаг, всего

$\log_2 L$ шагов (так как каждый шаг влечёт за собой выполнение 2 шагов размерности $L/2$)

Запись в файл имеет сложность $O(C_1)$ для алфавита и $O(2N)$ для исходного файла (так как нужно считать и записать символ N раз)

$C_1 = (1 + 1 + l) \cdot L$, где l – длина кода символа, учитывая, средняя длина кода символа примерно $\log_2 L$, то

$$C_1 = (1 + 1 + \log_2 L) \cdot L = L \log_2 L + 2L$$

Итоговая сложность:

$$O(N) + O(L \log_2 L) + O(L \log_2 L + 2L) + O(2N)$$

- Анализ сложности разархивации:

Считывание из файла и запись в файл имеют сложности $O(N)$

Построение дерева имеет сложность $O(L)$

Проход по дереву для интерпретации кода каждого символа в файле также имеет сложность $O(N)$

Итоговая сложность:

$$O(N) + O(N) + O(L) + O(N)$$

3 Инструкция пользователя

3.1 Запуск программы

Для запуска программы нужно запустить файл *KURSOVAYA.exe* из командной строки Windows. Формат запуска программы для архивации:

KURSOVAYA.exe compress /название файла для архивации/

Формат запуска программы для разархивации:

KURSOVAYA.exe decompress /название сжатого файла/ /имя выходного файла/

3.2 Формат входного файла

Любой одиночный файл.

3.3 Формат выходного файла

Файл с тем-же названием, но с расширением *.fano*.

4 Тестовые примеры

Тестовые примеры располагаются в папке test. Они представляют собой архивированные и разархивированные файлы следующего содержания.

- Test1 – роман Льва Толстого “Война и мир”

До сжатия 1508 килобайт, после сжатия 945 килобайт.

- Test2 – не монотонное изображение в формате bmp (рисунок 2)

До сжатия 2813 килобайт, после сжатия 2704 килобайт.



Рисунок 2 - Рисунок для test2

- Test3 – белое монотонное изображение в формате bmp

До сжатия 2113 килобайт, после сжатия 289 килобайт.

- Test4 – звуковой файл в формате mp3

До сжатия 2337 килобайт, после сжатия 2274 килобайт.

Список литературы

1. **М.Н. Аршинов, Л.Е. Садовский.** *Коды и математика.* Москва : Наука, 1983.