

Федеральное государственное автономное образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

Прикладная математика
Лабораторная работа №2

Выполнила: Калугина Марина

Группа: Р3302

г. Санкт-Петербург

2019 г.

Задание:

Реализовать процедуру построения арифметического кода.

Вычислить коэффициент сжатия.

Определить зависимость коэффициента сжатия от длины кода.

Декодировать полученное сообщение.

Код программы:

```
struct symbol {          // структура для символа
    char c;              // символ
    int n;               // количество в тексте
    long double p;       // вероятность
    long double a_beg;   // начало отрезка
    long double a_end;   // конец отрезка

    symbol(char c) : c(c) {
        n = 1;
    }
};

void read_from_file(std::string filename) {                                //
    считывание с файла
    std::ifstream in(filename);
    int sum = 0;                                                           // кол-во символов
    if (in.is_open()) {                                                   // чтение посимвольно с
        файла и сохранение количества каждого типа символов в тар
        while (!in.eof()) {
            char c = in.get();
            auto iter = symbols.find(c);
            if (iter != symbols.end()) {                                    // если символ уже
                встречался -- увеличим счетчик
                iter->second.n++;
            } else {
                symbol s = symbol(c);                                     // если символ еще не
                встречался, создадим элемент в карте
                symbols.insert({c, s});
            }
            sum++;
            size++;
        }
        for (auto &symbol : symbols) {                                    // расчет вероятностей
            symbol.second.p = (double) symbol.second.n / (double) sum;
        }
    } else {
        std::cout << "Файл не найден";
    }
    size--;
    in.close();
}

void read_string(std::string filename) {
    std::ifstream in(filename);
    if (in.is_open()) {
        getline(in, str);
    }
}
```

```

    }
    in.close();
}

void make_segments() { // создание границ
отрезков
    double courser = 0;
    for (auto &s : symbols) {
        s.second.a_beg = courser;
        courser += s.second.p;
        s.second.a_end = courser;
    }
}

double code(long double a_beg, long double a_end, char *c) { // кодирование
информации
    auto it = symbols.find(*c);
    long double new_a_beg = a_beg + (a_end - a_beg) * it->second.a_beg; // новые
границы
    long double new_a_end = a_beg + (a_end - a_beg) * it->second.a_end;
    printf("%c [%.53Lf; %.53Lf) -> [%.53Lf; %.53Lf)\n", *c, a_beg, a_end, new_a_beg,
new_a_end);
    c++;
    if (*c) { //
продолжить рекурсивно, пока не найден конец файла
        return code(new_a_beg, new_a_end, c);
    } else {
        return (new_a_beg + new_a_end) / 2;
    }
}

std::string decode(long double code, std::string res) { //декодирование фразы
    for (auto &character : symbols) {
        if (code >= character.second.a_beg && code < character.second.a_end) {
// ПОИСК СИМВОЛА
            res += character.first;
            code = (code - character.second.a_beg)/(character.second.a_end -
character.second.a_beg); //нормализация и новый код
            size--;
            if (!size) {
                return res;
            }
            return decode(code, res); // рекурсивно, пока не найдем конец
строки
        }
    }
}

double H() { // энтропия
    double res = 0;
    for (auto &item : symbols) {
        res -= item.second.p * std::log2(item.second.p);
    }
    res *= str.size();
}

```

```

    return res;
}

int main() {
    std::cout << "Введите имя файла\n";
    std::string filename = "../input/test.txt";
    std::cin >> filename;
    read_from_file(filename);
    read_string(filename);
    make_segments();
    long double res = code(0, 1, &str[0]);
    std::cout << "Арифметическое кодирование фразы: \"" << str << "\"\n";
    std::cout << "Результат: ";
    printf("%.53Lf\n", res);
    std::cout << "Декодированная строка: " << decode(res, "") << "\n";

    std::cout << "Коэффициент сжатия: " << H()/sizeof(res) << "\n";
    return 0;
}

```

В результате работы программы было выявлено, что для выбранной фразы арифметическое кодирование работает с коэффициентом сжатия 2.81752. При уменьшении длины кодируемой строки значение коэффициента сжатия уменьшается.