МИНОБРНАУКИ РФ

ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»

Кафедра «Программное обеспечение»

Отчет

по лабораторной работе № 2 по программированию

на тему «Разработка структур данных и алгоритмов на основе векторов и списков. Итераторы для контейнеров последовательностей»

Выполнил студент группы Б03-191-3: Р.А. Гумметов

Принял: В.Г. Тарасов

Ижевск 2019

1. Условие задачи

Квадратный шифр очень прост как к шифрованию, так и к дешифрованию. Предположим, у нас имеется строка s = s0...sn−1. Квадратный шифр передвинет все символы, стоящие на позициях с номерами, являющимися полными квадратами, в начало строки, причем относительный порядок сдвинутых символов не изменится. Порядок же остальных символов останется без изменений. Предположим, мы хотим зашифровать строку s =«thisisacontest» квадратным шифром. Мы передвинем символы, стоящие на позициях 0, 1, 4, 9 в начало. Таким образом, зашифрованная строка будет иметь вид «thinissacotest». Вам дана зашифрованная строка. Дешифруйте её и выведите оригинальную строку.

2. Постановка задачи

Первая строка содержит строку s (1 ≤ |s| ≤ 50), состоящую только из строчных букв латинского алфавита.

3. Идея решение задачи

Сначала переведем исходную строку в вектор char (Str), в вектор char (perest) запишем символы из начала строки, которые сдвинуты, в вектор int (el) запишем индексы, равные полным квадратам и посчитаем количество этих индексов. Удалим из вектора Str начальные символы, которые сдвинуты. В массив char (end) запишем нужную строку, проверяя, если индекс равен полному квадрату, то берем символ из вектора perest, если нет, то берем символ из вектора Str.

4. Код программы

#include <iostream>

#include <string>

#include <vector>

using namespace std;

bool f(int i) //проверка на полный квадрат

{

if (sqrt(i) == static\_cast<int>(sqrt(i)))

return true;

else

return false;

}

int main()

{

string str;

getline(cin, str); //зашифрованная строка

vector<char> Str;

char \*end = new char[str.length()];

for (int i = 0; i < str.length(); i++)

{

Str.push\_back(str[i]); //перевод строки в вектор char

}

int count = 0;

vector<int> el(51); //вектор для индексов, равных полным квадратам

vector<char> perest(51); //

int m = 0; //

int n = 0; //

for (int i = 0; i < str.length(); i++)

{

bool sq = f(i); //проверка индекса на полный квадрат

if (sq)

{

perest[n] = str[count]; //вектор char для начальных эл., которые сдвинуты в начало

n++;

el[m] = i;

m++;

count++; //считаем кол-во индексов, являющихся полными квадратом

}

}

Str.erase(Str.begin(), Str.begin() + count); //удаление из вектора Str сдвинутых в начало элементов

int k = 0;

int l = 0;

int j = 0;

for (int i = 0; i < str.length(); i++)

{

if (el[j] == i) //проверка индекса на полный квадрат

{

end[i] = perest[l];

j++;

l++;

}

else

{

end[i] = Str[k];

k++;

}

}

for (int i = 0; i < str.length(); i++)

{

cout << end[i];

}

}

5. Вывод

Таким образом написана программа, дешифрующая зашифрованную строку.

Временная сложность программы составляет O(n).

Пространственная сложность программы составляет O(n).