Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

**ОТЧЕТ О ПРАКТИЧЕСКОЙ** **РАБОТЕ № 2**

Линейные структуры данных.  
Вариант № 10

Преподаватель \_\_\_\_\_\_\_\_ И.В. Матковский

подпись, дата

Студент КИ18-09б, 031830645 \_\_\_\_\_\_\_\_ С.А. Котов

подпись, дата

Красноярск 2019

**1 Задание**

Разработать для решения поставленной задачи алгоритм; реализовать полученный алгоритм с использованием линейных структур данных заданных типов. Одна из заданных структур должна быть реализована самостоятельно, без использования готовых библиотек; вторая структура может быть как реализована самостоятельно, так и взята из STL.

Задача: Умножить все элементы последовательности на элемент, находящийся в её середине.

Структуры данных: Массив и двусвязный список.

**2 Код программы**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | auto iter = sequence.cbegin();  for (int i = 0; i < sequence.size() / 2; i++) {  iter++;  }  int middleIndexElement = \*iter;  // Прогон двухсвязного списка  for (long int& i : sequence) i \*= middleIndexElement;  // Прогон массива  for (int i = 0; i < n; i++) arr[i] \*= middleIndexElement; |

**3 Теоретические оценки сложности алгоритмов**

Временная сложность поиска срединного элемента в массиве равна O(1), а временная сложность перемножение всех элементов массива на значение срединного элемента равна O(n).

Временная сложность поиска срединного элемента в двусвязном списке равна O(n), а временная сложность перемножение всех элементов двусвязного списка на значение срединного элемента равна O(n). Таким образом временная сложность выполнения задания равно O(n).

Доступ к любому элементу массива всегда равно O(1), а перемножение происходит путем прохода по всему массиву с поочередным умножением.

В двусвязном списке невозможен моментальный доступ к любому элементу списка, из-за чего приходится изначально проходить по нему до середины, запоминать итератор срединного элемента. Далее проходим по всему списку, поочередно перемножая элементы.

**4 Экспериментальные оценки сложности алгоритмов**

|  |  |  |
| --- | --- | --- |
| n | Время 1, сек | Время 2, сек |
| 100 | 0 | 0 |
| 50000 | 0 | 0 |
| 1000000 | 0 | 0 |
| 10000000 | 0 | 2 |