Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра компьютерных систем в управлении и проектировании (КСУП)

Двусвязный список

Лабораторная работа №2 по

Дисциплине «Структуры данных»

Студент гр. 571-2
_____ К.Д. Никитин
(подпись)
_____
(дата)
Руководитель
к.т.н., доцент каф. КСУП
_____ _____ А.А. Калентьев
(дата) (подпись)
_____
М.П. (дата)

Томск 2022

Содержание

1 ВВЕДЕНИЕ

Цель работы: Реализовать структуру данных «Двусвязный список» и набор функций для работы с ней.

Необходимо обеспечить безопасность функций и всей программы в целом.

Необходимо реализовать следующие функции:

- Функция создания и инициализации полей списка;
- Добавления элемента;
- Удаление элемента;
- Вставка элемента в начало;
- Вставка элемента в конец;
- Вставка после определенного элемента;
- Вставка перед определенным элементом;
- Сортировка списка;
- Линейный поиск элемента в массиве.

2 **ОСНОВНАЯ ЧАСТЬ**

2.1 Описание функций

AddElement – добавляет элемент в конец списка и в структуре двусвязного списка устанавливает указатель этого элемента в качестве последнего. Сложность O(1).

RemoveElement – удаляет элемент по переданном индексу. При удалении начального указатель на голову списка перемещается на следующий элемент. При удалении последнего элемента, указатель на последний элемент перемещается на предыдущий.

InsertElementInBegin – вставка элемента в начало. Указатель начала списка заносится адрес этого элемента. Сложность O(1).

InsertElementBeforeCertainElement – вставка элемента перед переданным в метод индексом. Сложность O(n).

InsertElementAfterCertainElement – вставка элемента после переданного в метод индекса. Сложность O(n).

SortList – сортировка списка. Сложность $O(n^2)$.

LinearSearch – производит поиск элемента по указанному индексу. Возвращает число больше или равно 0, если это индекс искомого числа, иначе возвращает -1, указывая, что элемента в списке нет.

2.2 Исследование сложности алгоритмов

1. Оценка сложности алгоритма для операции вставки элемента перед указанным индексом.

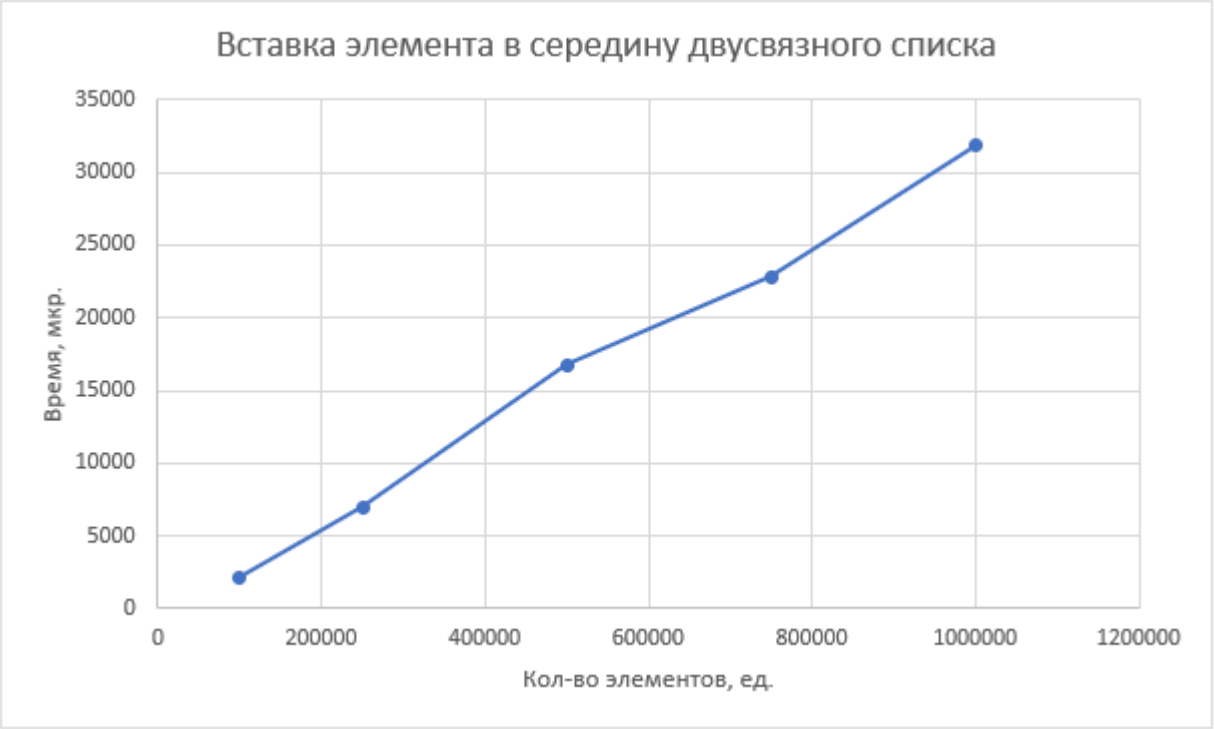

График 1. Отношение между временем выполнения алгоритма и кол-вом элементов.

Время, мкр.	Кол-во элементов, ед.
2145	100000
6957	250000
16726	500000
22816	750000
31836	1000000

Таблица 1. Отношение между временем выполнения алгоритма и кол-вом элементов

По графику видно, что сложность данного алгоритма равна O(n).

2. Оценка сложности алгоритма для операции вставки элемента в начало.

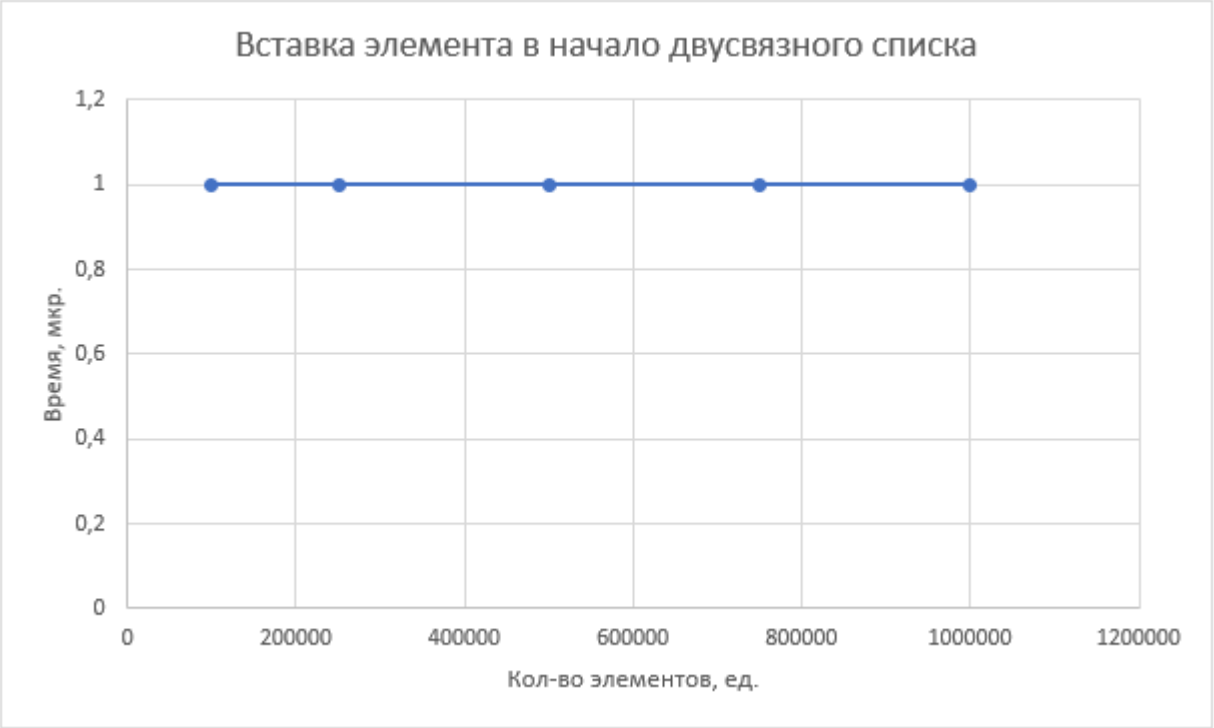

График 2. Отношение между временем выполнения алгоритма и кол-вом элементов.

Время, мкр.	Кол-во элементов, ед.
1	100000
1	250000
1	500000
1	750000
1	1000000

Таблица 2 Результат работы алгоритма вставки элемента в начало.

По графику видно, что сложность данного алгоритма равна O(1).

3. Операция удаление элемента из списка.

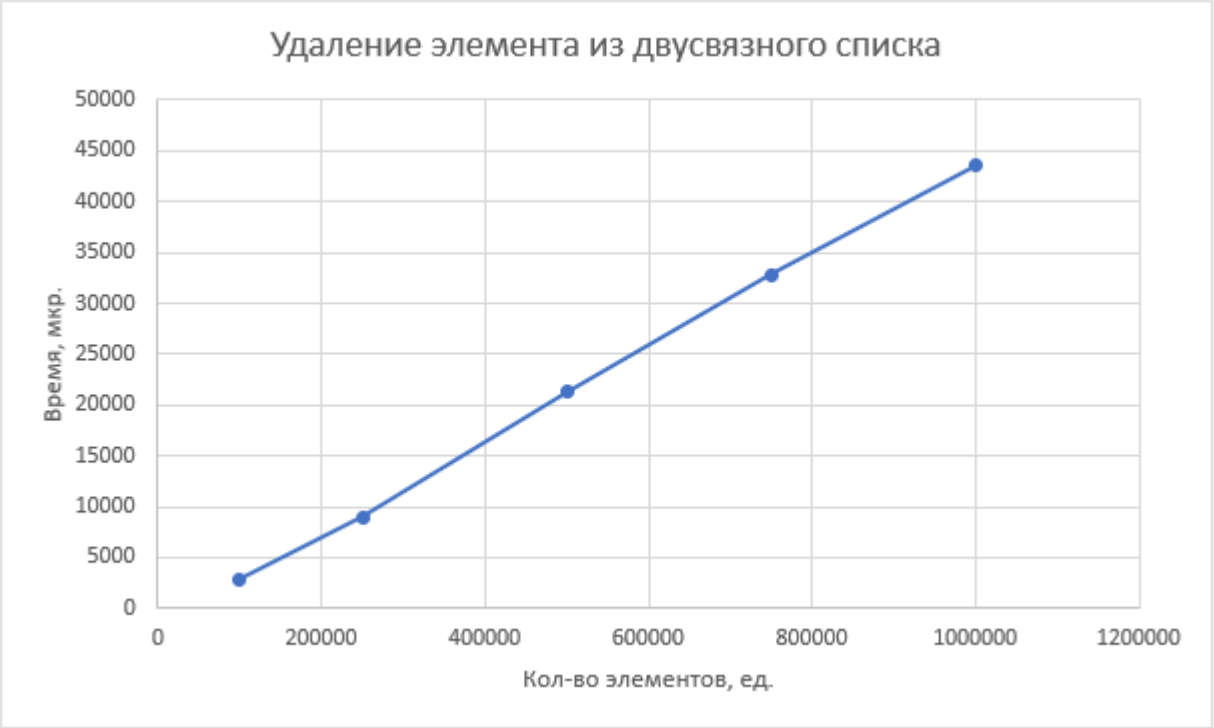

График 3. Отношение между временем выполнения алгоритма и кол-вом элементов.

1462

Время, мкр.	Кол-во элементов, ед.
2836	100000
8957	250000
21306	500000
32850	750000
43539	1000000

Таблица 3. Результаты работы алгоритма удаления элемента

По графику видно, что сложность данного алгоритма равна O(n).

3 Заключение

В ходе данной лабораторной работы, мной был разработана структура данных двусвязный список, а также базовые функции для работы с ним.

Были проведены исследования, в ходе которых я установил сложность некоторых алгоритмов:

1. Алгоритм вставки элемента перед определенным индексом имеет сложность O(n).
2. Алгоритм вставки элемента в начало имеет сложность O(1).
3. Алгоритм удаления элемента имеет сложность O(n).