

МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение высшего образования*

***«МИРЭА – Российский технологический университет»***

**РТУ МИРЭА**

Отчет по выполнению практического задания № 7

**Тема:**

«Рекурсивные алгоритмы и их реализации»

Дисциплина: «Структуры и алгоритмы обработки данных»

Выполнил студент: Моисенко М. О.

Фамилия И.О.

Группа: ИКБО-00-22

Номер группы

Москва – 2023

СОДЕРЖАНИЕ

[1 ПОСТАНОВКА ЗАДАЧИ 3](#_Toc135272203)

[1.1 Условие задания 3](#_Toc135272204)

[2 РЕШЕНИЕ ЗАДАНИЯ 1 3](#_Toc135272205)

[2.1 Реализация бинарного поиска иттерационно 3](#_Toc135272206)

[2.2 Реализация бинарного поиска рекурсивно 4](#_Toc135272207)

[2.3 Тестирование работы алгоритма 4](#_Toc135272208)

[3 РЕШЕНИЕ ЗАДАНИЯ 2 4](#_Toc135272209)

[3.1 Рекурсивная реализация создания двунаправленного списка 4](#_Toc135272210)

[3.2 Тестирование работы алгоритма 5](#_Toc135272211)

[4 ВЫВОД 5](#_Toc135272212)

[5 СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 5](#_Toc135272213)

# ПОСТАНОВКА ЗАДАЧИ

Цель:получить знания и практические навыки по разработке и реализации рекурсивных процессов.

## Условие задания

Разработать и протестировать рекурсивные функции в соответствии с задачами варианта.

Индивидуальный вариант под номеров 9.

Функции:

1. Бинарный поиск элемента в массиве.

2. Создание двунаправленного списка.

# РЕШЕНИЕ ЗАДАНИЯ 1

## Реализация бинарного поиска иттерационно

int ClassicBinSearch(const int\* arr, int count, int key)
{
 int l = 0;
 int u = count - 1;

 while (l <= u)
 {
 int m = (l + u) / 2;
 if (arr[m] == key)
 return m;
 if (arr[m] < key)
 l = m + 1;
 if (arr[m] > key)
 u = m - 1;
 }
 return -1;
}

## Реализация бинарного поиска рекурсивно

На каждом шаге цикла итерационного алгоритма сравнивается текущий элемент, и наибольший из всех предыдущих, что делает возможным выполнение задачи с помощью рекурсивного алгоритма. На каждом шаге рекурсивной функции возвращается наибольшее значение из текущего и наибольшего их всех предыдущих, т.е. значения этой же функции для массива, длинной на единицу меньшей текущей. Условием выхода из рекурсии является наличие только одного элемента в массиве. В это случае возвращается сам элемент.



Рисунок 3 – Входные данные тестирования функций

int RecBinSearch(int\* arr, int b, int count, int key)
{
 if(b <= count)
 {
 int mid = b + (count - b) / 2;
 if(arr[mid] == key)
 return mid;
 else if(arr[mid] < key)
 return RecBinSearch(arr, mid + 1, count, key);
 else if(arr[mid] > key)
 return RecBinSearch(arr, b, mid - 1, key);
 }
 return -1;
}

## Тестирование работы алгоритма

Результаты тестирования функций программы при различных входных данных представлены на рисунке 3 и 4.



Рисунок 3 – Входные данные тестирования функций



Рисунок 4 – Вывод в консоли

По результатам тестирования можно сделать вывод, что все функции являются рабочими.

# РЕШЕНИЕ ЗАДАНИЯ 2

## Рекурсивная реализация создания двунаправленного списка

void manu\_gen(listik& a, int v, int& i)
{
 cout << "Entry number " << i << endl;
 i++;
 if (v == 0)
 return;
 else
 {
 cin >> v;
 manu\_gen(a, v, i);
 if (v != 0)
 a.add\_l(v);
 }
}

## Тестирование работы алгоритма

Результаты тестирования функций программы при различных входных данных представлены на рисунке 5.



По результатам тестирования можно сделать вывод, что все функции являются рабочими.

# ВЫВОД

В ходе выполнения работы были получены знания и освены практические навыки работы с рекурсивными функциями.

# СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Структуры данных и проектирование программ : Пер. с англ. / Р. Круз. — М.: БИНОМ. Лаборатория знаний, 2017. — 766 с.

2. Полный справочник по C++ : Пер. с англ. / Г. Шилдт. — М.: ООО "И.Д.Вильямс", 2016. — 796 с.: ил. — Предм. указ.: с. 787-796