



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА - Российский технологический университет»

РТУ МИРЭА

Отчет по выполнению практических заданий
по дисциплине «Структуры и алгоритмы обработки данных»

Выполнил студент

Пшеничный Т.Д.

группа

ИКБО-29-21

Москва 2022

СОДЕРЖАНИЕ

Практическая работа № 1. Простые алгоритмы внутренней сортировки массивов.	3
Практическая работа № 2. Усовершенствованные алгоритмы внутренней сортировки массивов.....	4
Практическая работа № 3. Алгоритмы внешней сортировки массивов.	5
Практическая работа № 4. Алгоритм поиска в неотсортированных массивах.	6
Практическая работа № 5. Алгоритм поиска в отсортированных массивах.....	7
Практическая работа № 6. Алгоритмы поиска строк в тексте.	8
Практическая работа № 7. Линейные динамические списки.....	9
Практическая работа № 8. Линейный список стек и его применение.	10
Практическая работа № 9. Линейный список. Очередь.	11
Практическая работа № 10. Бинарное дерево поиска. AVL дерево.....	12
Практическая работа № 11. Хеширование - прямой доступ к данным.	13
Практическая работа № 12. Внешние структуры данных - файлы и алгоритмы их обработки.	14
Практическая работа № 13. Основные алгоритмы работы с графами.	15
Практическая работа № 14. Алгоритмы сжатия и кодирования данных.....	16
Практическая работа № 15. Стратегии и методы построения алгоритмов.....	17
Практическая работа № 16. Метод линейного программирования.	18
ЗАКЛЮЧЕНИЕ	19
СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ.....	20

Практическая работа № 1. Простые алгоритмы внутренней сортировки массивов.

Цель работы заключалась в приобретение практических навыков по определению:

- сложности алгоритмов на теоретическом и практическом уровнях;
- эффективного алгоритма решения задачи из нескольких алгоритмов.

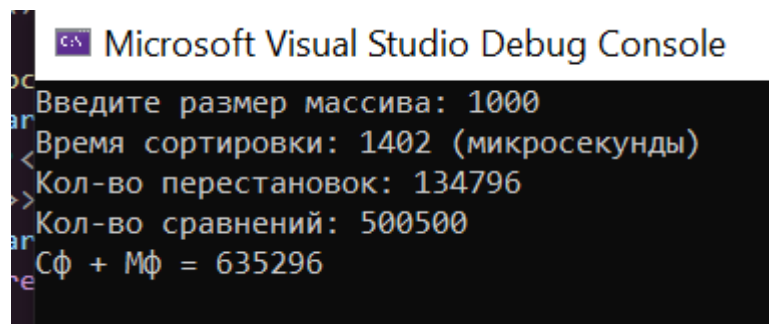
Постановка задачи: составить программу сортировки одномерного целочисленного массива $A[n]$, используя алгоритм согласно варианту индивидуального задания.

Вариант: 1.2 Простой вставки (Exchange sort).

Исходный код программы можно найти по ссылке:

<https://github.com/tim-lwry/siaod/blob/main/exchangesort.cpp>

Результат работы программы:

A screenshot of the Microsoft Visual Studio Debug Console window. The title bar reads "Microsoft Visual Studio Debug Console". The console output is as follows:

```
Введите размер массива: 1000
Время сортировки: 1402 (микросекунды)
< Кол-во перестановок: 134796
> Кол-во сравнений: 500500
Сф + Мф = 635296
```

Рисунок 1.1 — Результат выполнения

Практическая работа № 2. Усовершенствованные алгоритмы внутренней сортировки массивов.

Цель работы заключалась в приобретение практических навыков по определению:

- сложности алгоритмов на теоретическом и практическом уровнях;
- эффективного алгоритма решения задачи из нескольких алгоритмов.

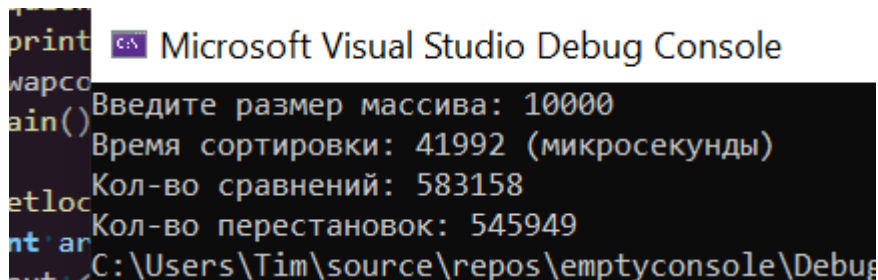
Постановка задачи: составить программу сортировки одномерного целочисленного массива $A[n]$, используя алгоритм согласно варианту индивидуального задания.

Вариант: 2.4 Быстрой сортировки (Quick-sort или сортировки Хоара)

Исходный код программы можно найти по ссылке:

<https://github.com/tim-lwry/siaod/blob/main/QuickSortCPP.cpp>

Результат работы программы:

A screenshot of the Microsoft Visual Studio Debug Console. The window title is "Microsoft Visual Studio Debug Console". The output text is as follows:

```
Введите размер массива: 10000
Время сортировки: 41992 (микросекунды)
Кол-во сравнений: 583158
Кол-во перестановок: 545949
C:\Users\Tim\source\repos\emptyconsole\Debug
```

Рисунок 1.2 — Результат выполнения

Практическая работа № 3. Алгоритмы внешней сортировки массивов.

Цель работы заключалась в приобретение практических навыков по применению алгоритмов внешней сортировки.

Постановка задачи: составить программу сортировки одномерного целочисленного массива $A[n]$ методом слияния (merge-sort).

Вариант: 3.1 Простое слияние.

Исходный код программы можно найти по ссылке:

<https://github.com/tim-lwry/siaod/blob/main/MergeSortCPP.cpp>

Результат работы программы:

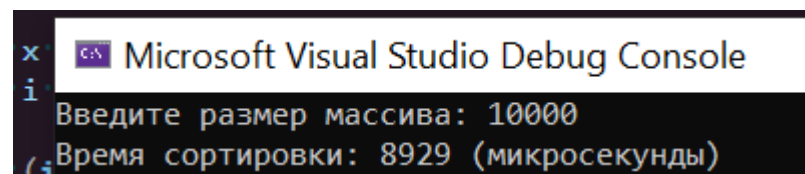


Рисунок 1.3 — Результат выполнения

Практическая работа № 4. Алгоритм поиска в неотсортированных массивах.

Цель работы заключалась в приобретение практических навыков по применению алгоритма поиска в неотсортированных массивах.

Постановка задачи: составить программу последовательного (с барьером) поиска заданного элемента по ключу в одномерном целочисленном массиве $A[n]$.

Исходный код программы можно найти по ссылке:

<https://github.com/tim-lwry/siaod/blob/main/CPPSearchInUnsorted.cpp>

Результат работы программы:

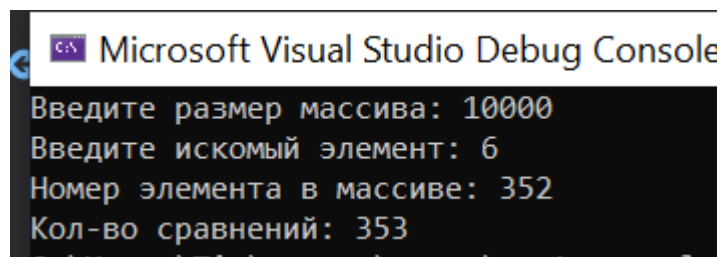


Рисунок 1.4 — Результат выполнения

Практическая работа № 5. Алгоритм поиска в отсортированных массивах.

Цель работы заключалась в приобретение практических навыков по применению алгоритма поиска в отсортированных массивах.

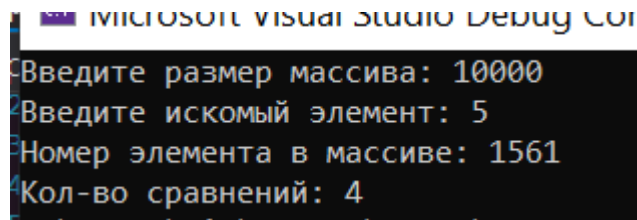
Постановка задачи: составить программу поиска заданного элемента по ключу в одномерном целочисленном массиве $A[n]$, используя алгоритм согласно варианту индивидуального задания.

Вариант: 5.1 Двоичного поиска.

Исходный код программы можно найти по ссылке:

<https://github.com/tim-lwry/siaod/blob/main/CPPBinarySearch.cpp>

Результат работы программы:



```
Microsoft Visual Studio Debug Console
Введите размер массива: 10000
Введите искомый элемент: 5
Номер элемента в массиве: 1561
Кол-во сравнений: 4
```

Рисунок 1.5 — Результат выполнения

Практическая работа № 6. Алгоритмы поиска строк в тексте.

Цель работы заключалась в приобретение практических навыков по применению алгоритма поиска строк в тексте.

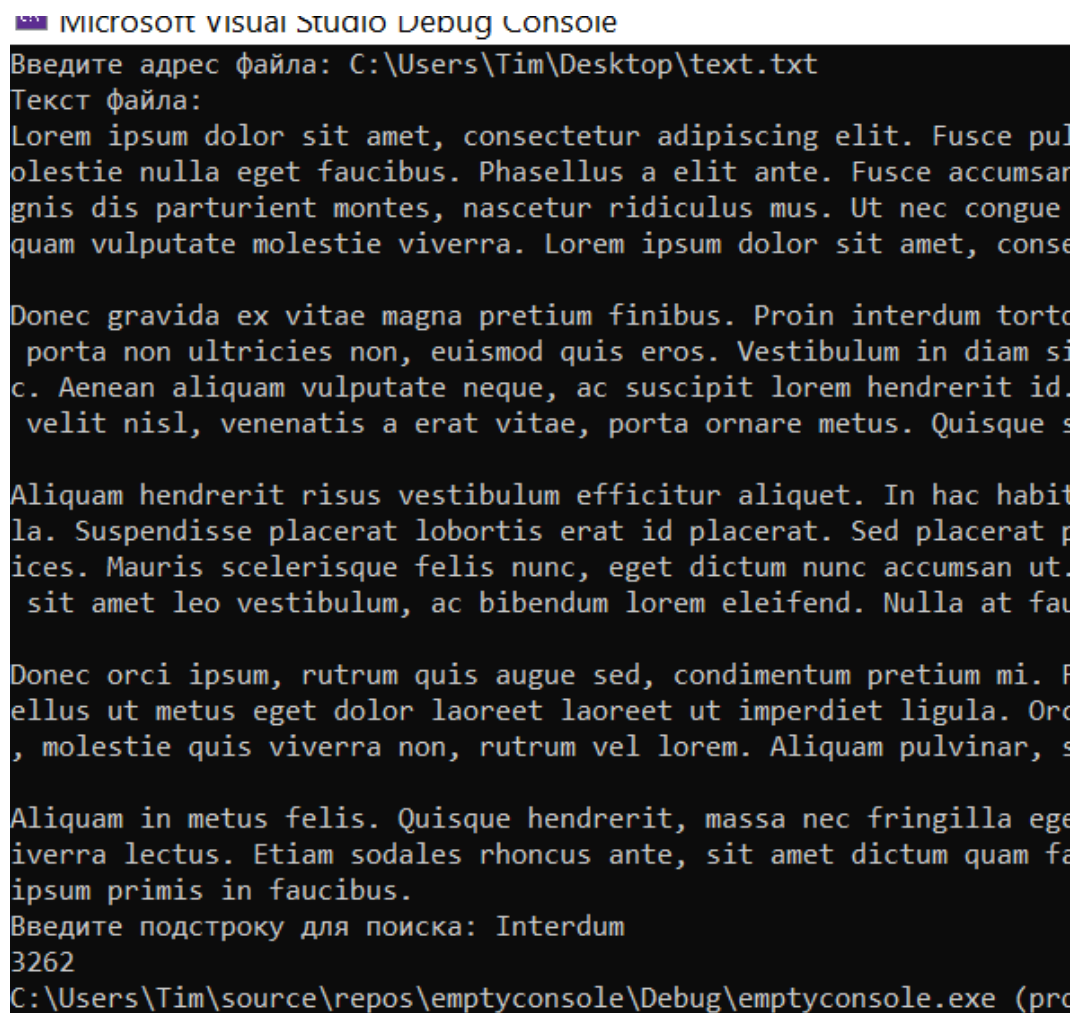
Постановка задачи: составить программу поиска первого вхождения заданной строки P длиной m символов в тексте S , размером n символов, используя алгоритм согласно варианту индивидуального задания.

Вариант: 6.2 Кнута-Морриса-Пратта.

Исходный код программы можно найти по ссылке:

<https://github.com/tim-lwry/siaod/blob/main/CPPSubstringSearch.cpp>

Результат работы программы:



```
Microsoft Visual Studio Debug Console
Введите адрес файла: C:\Users\Tim\Desktop\text.txt
Текст файла:
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce pul
olestie nulla eget faucibus. Phasellus a elit ante. Fusce accumsan
gnis dis parturient montes, nascetur ridiculus mus. Ut nec congue
quam vulputate molestie viverra. Lorem ipsum dolor sit amet, conse
Donec gravida ex vitae magna pretium finibus. Proin interdum torto
porta non ultricies non, euismod quis eros. Vestibulum in diam si
c. Aenean aliquam vulputate neque, ac suscipit lorem hendrerit id.
velit nisl, venenatis a erat vitae, porta ornare metus. Quisque s
Aliquam hendrerit risus vestibulum efficitur aliquet. In hac habit
la. Suspendisse placerat lobortis erat id placerat. Sed placerat p
ices. Mauris scelerisque felis nunc, eget dictum nunc accumsan ut.
sit amet leo vestibulum, ac bibendum lorem eleifend. Nulla at fau
Donec orci ipsum, rutrum quis augue sed, condimentum pretium mi. F
ellus ut metus eget dolor laoreet laoreet ut imperdiet ligula. Orc
, molestie quis viverra non, rutrum vel lorem. Aliquam pulvinar, s
Aliquam in metus felis. Quisque hendrerit, massa nec fringilla ege
iverra lectus. Etiam sodales rhoncus ante, sit amet dictum quam fa
ipsum primis in faucibus.
Введите подстроку для поиска: Interdum
3262
C:\Users\Tim\source\repos\emptyconsole\Debug\emptyconsole.exe (pro
```

Рисунок 1.6 — Результат выполнения

Практическая работа № 7. Линейные динамические списки.

Цель работы заключалась в приобретение практических навыков по применению линейных динамических списков.

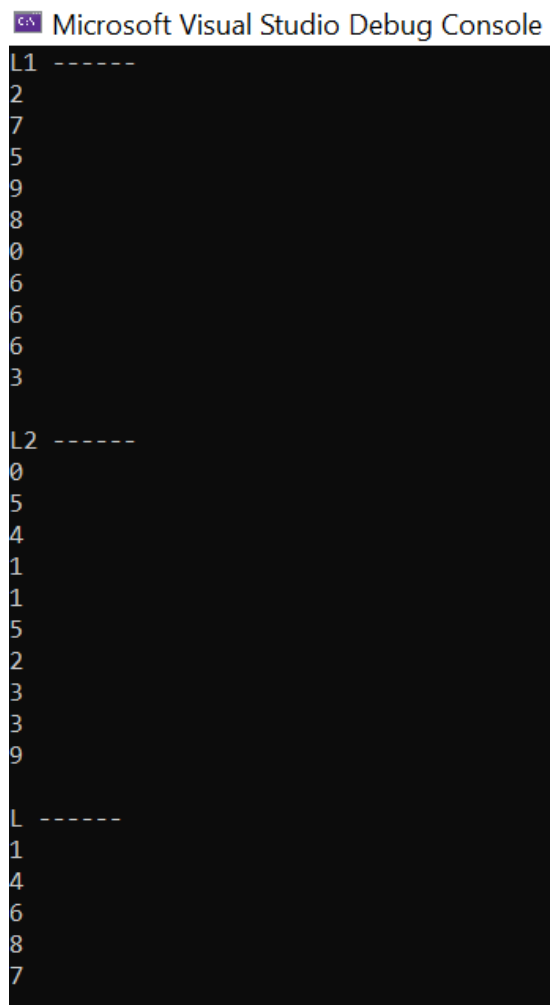
Постановка задачи: определите список операции над списками варианта, включая операцию добавления узла в начало списка, вывода списка.

Вариант: №4.

Исходный код программы можно найти по ссылке:

<https://github.com/tim-lwry/siaod/blob/main/p7.1.cpp>

Результат работы программы:



```
Microsoft Visual Studio Debug Console
L1 -----
2
7
5
9
8
0
6
6
6
3
L2 -----
0
5
4
1
1
5
2
3
3
9
L -----
1
4
6
8
7
```

Рисунок 1.7 — Результат выполнения

Практическая работа № 8. Линейный список стек и его применение.

Цель работы заключалась в приобретении практических навыков по применению стека.

Постановка задачи: реализовать операции хранения над стеком.

Вариант: №2.

Исходный код программы можно найти по ссылке:

https://github.com/tim-lwry/siaod/blob/main/pr_8.cpp

Результат работы программы:

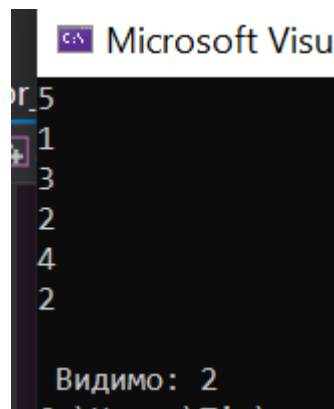


Рисунок 1.8 — Результат выполнения

Практическая работа № 9. Линейный список. Очередь.

Цель работы заключалась в приобретение практических навыков по применению линейного списка, очереди.

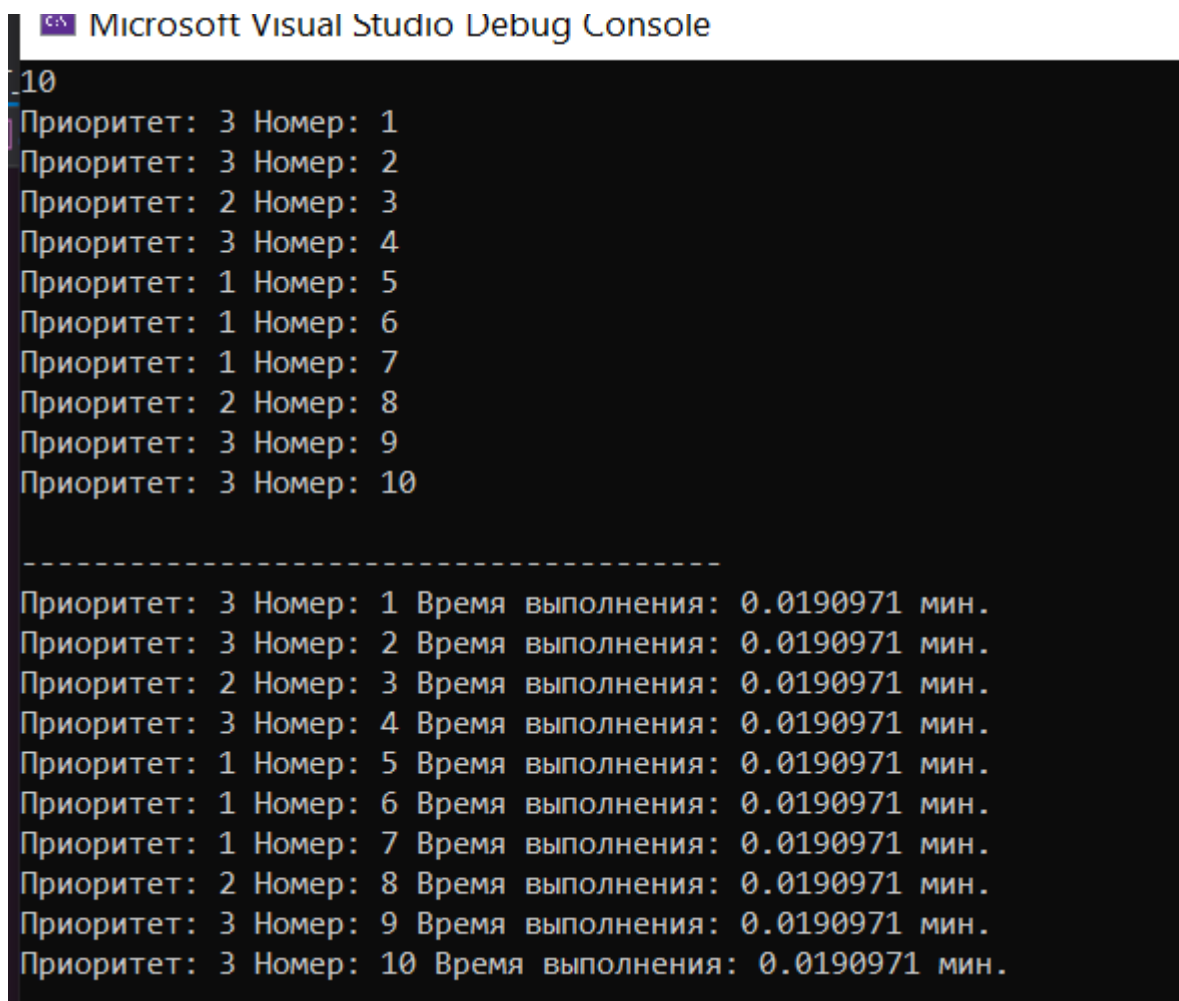
Постановка задачи: реализовать операции хранения над очередью.

Вариант: №2.

Исходный код программы можно найти по ссылке:

https://github.com/tim-lwry/siaod/blob/main/Pr9_v2.cpp

Результат работы программы:

The image shows a screenshot of the Microsoft Visual Studio Debug Console. The title bar at the top reads "Microsoft Visual Studio Debug Console". The console output is as follows:

```
10
Приоритет: 3 Номер: 1
Приоритет: 3 Номер: 2
Приоритет: 2 Номер: 3
Приоритет: 3 Номер: 4
Приоритет: 1 Номер: 5
Приоритет: 1 Номер: 6
Приоритет: 1 Номер: 7
Приоритет: 2 Номер: 8
Приоритет: 3 Номер: 9
Приоритет: 3 Номер: 10

-----
Приоритет: 3 Номер: 1 Время выполнения: 0.0190971 мин.
Приоритет: 3 Номер: 2 Время выполнения: 0.0190971 мин.
Приоритет: 2 Номер: 3 Время выполнения: 0.0190971 мин.
Приоритет: 3 Номер: 4 Время выполнения: 0.0190971 мин.
Приоритет: 1 Номер: 5 Время выполнения: 0.0190971 мин.
Приоритет: 1 Номер: 6 Время выполнения: 0.0190971 мин.
Приоритет: 1 Номер: 7 Время выполнения: 0.0190971 мин.
Приоритет: 2 Номер: 8 Время выполнения: 0.0190971 мин.
Приоритет: 3 Номер: 9 Время выполнения: 0.0190971 мин.
Приоритет: 3 Номер: 10 Время выполнения: 0.0190971 мин.
```

Рисунок 1.9 — Результат выполнения

Практическая работа № 10. Бинарное дерево поиска. AVL дерево.

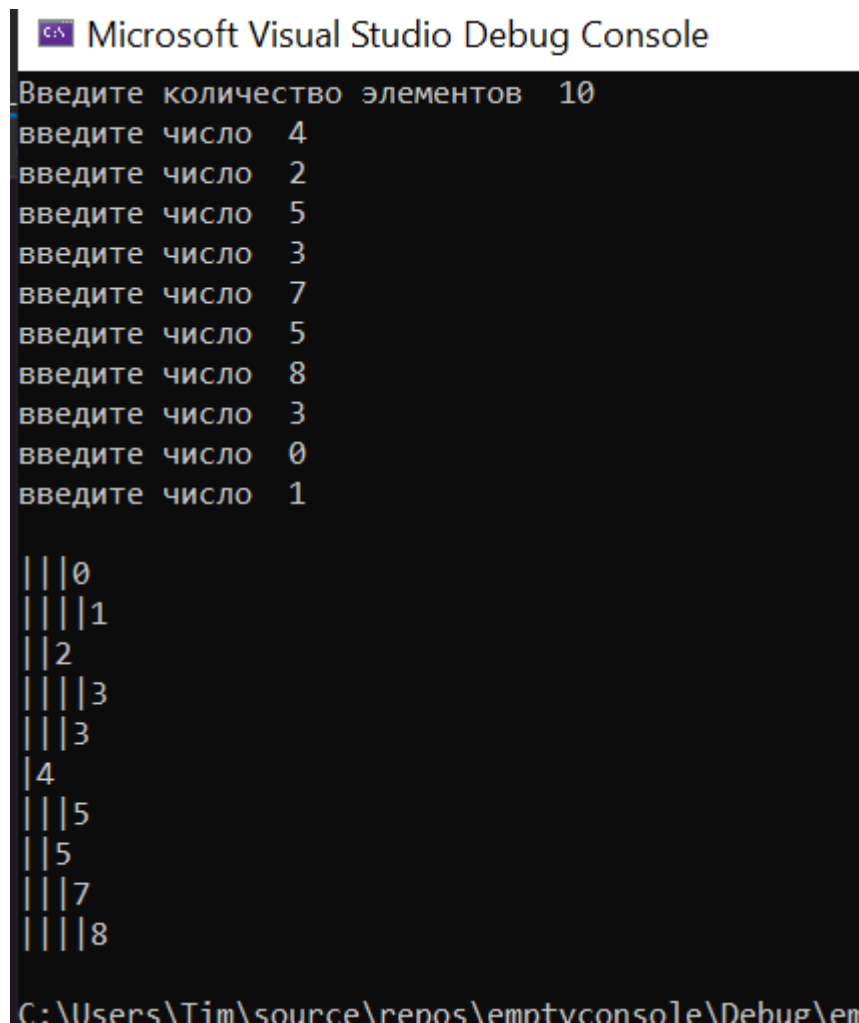
Цель работы заключалась в приобретение практических навыков по применению бинарного дерева.

Постановка задачи: написать программу создания двоичного дерева поиска по заданной последовательности целых чисел.

Исходный код программы можно найти по ссылке:

<https://github.com/tim-lwry/siaod/blob/main/BinaryTree.cpp>

Результат работы программы:



```
Microsoft Visual Studio Debug Console
Введите количество элементов 10
введите число 4
введите число 2
введите число 5
введите число 3
введите число 7
введите число 5
введите число 8
введите число 3
введите число 0
введите число 1

|||0
|||1
||2
|||3
||3
|4
||5
||5
||7
|||8

C:\Users\Tim\source\repos\emptyconsole\Debug\en
```

Рисунок 1.10 — Результат выполнения

Практическая работа № 11. Хеширование - прямой доступ к данным.

Цель работы заключалась в приобретение практических навыков по применению хеширование данных.

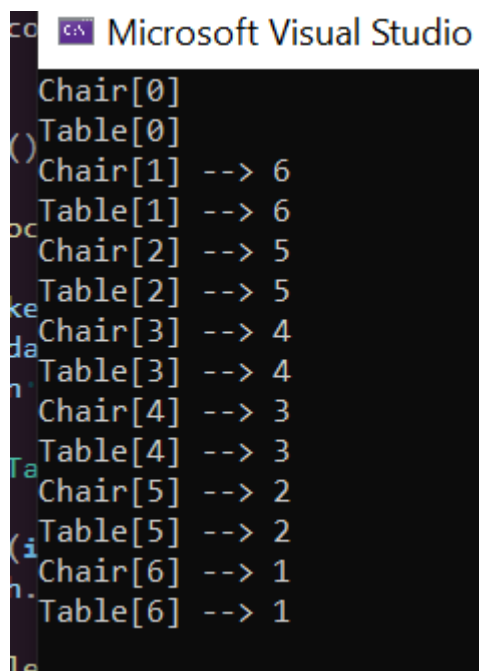
Постановка задачи: разработайте приложение, которое использует хеш-таблицу для организации прямого доступа к элементам множества, реализованного на массиве, структура элементов которого приведена в варианте.

Вариант: №1.

Исходный код программы можно найти по ссылке:

<https://github.com/tim-lwry/siaod/blob/main/HashTable.cpp>

Результат работы программы:



```
Chair[0]
Table[0]
Chair[1] --> 6
Table[1] --> 6
Chair[2] --> 5
Table[2] --> 5
Chair[3] --> 4
Table[3] --> 4
Chair[4] --> 3
Table[4] --> 3
Chair[5] --> 2
Table[5] --> 2
Chair[6] --> 1
Table[6] --> 1
```

Рисунок 1.11 — Результат выполнения

Практическая работа № 12. Внешние структуры данных - файлы и алгоритмы их обработки.

Цель работы заключалась в приобретение практических навыков по работе с двоичными и текстовыми файлами.

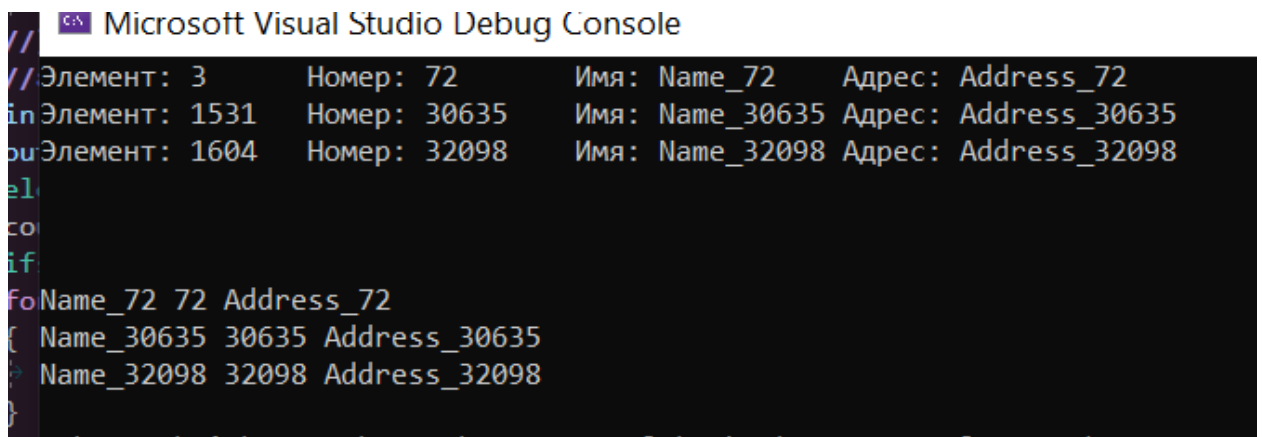
Постановка задачи: создать программные модули с операциями над двоичными и текстовыми файлами для выполнения задания варианта.

Вариант: №2.

Исходный код программы можно найти по ссылке:

<https://github.com/tim-lwry/siaod/blob/main/Bank12.cpp>

Результат работы программы:

The image shows a screenshot of the Microsoft Visual Studio Debug Console. The title bar at the top reads "Microsoft Visual Studio Debug Console". The console output is as follows:

```
//  
// Элемент: 3      Номер: 72      Имя: Name_72      Адрес: Address_72  
in Элемент: 1531   Номер: 30635   Имя: Name_30635  Адрес: Address_30635  
ou Элемент: 1604   Номер: 32098   Имя: Name_32098  Адрес: Address_32098  
el  
co  
if  
fo Name_72 72 Address_72  
{ Name_30635 30635 Address_30635  
  Name_32098 32098 Address_32098  
}
```

Рисунок 1.12 — Результат выполнения

Практическая работа № 13. Основные алгоритмы работы с графами.

Цель работы заключалась в приобретение практических навыков по реализации алгоритма Прима.

Постановка задачи: составить программу реализации алгоритма Прима построения остова дерева минимального веса.

Вариант: №2.

Исходный код программы можно найти по ссылке:

<https://github.com/tim-lwry/siaod/blob/main/Graphs13.cpp>

Результат работы программы:

```
1
Введите графы a[0][0] 1
Введите графы a[0][1] 2
Введите графы a[0][2] 0
Введите графы a[0][3] 3
Введите графы a[1][0] 5
Введите графы a[1][1] 10
Введите графы a[1][2] 3
Введите графы a[1][3] 0
Введите графы a[2][0] 2
Введите графы a[2][1] 3
Введите графы a[2][2] 1
Введите графы a[2][3] 3
Введите графы a[3][0] 4
Введите графы a[3][1] 5
Введите графы a[3][2] 60
Введите графы a[3][3] 4
1 2 0 3
5 10 3 0
2 3 1 3
4 5 60 4
0->1
1->6
6->2
1->4
0->3
1->5
```

Рисунок 1.13 — Результат выполнения

Практическая работа № 14. Алгоритмы сжатия и кодирования данных.

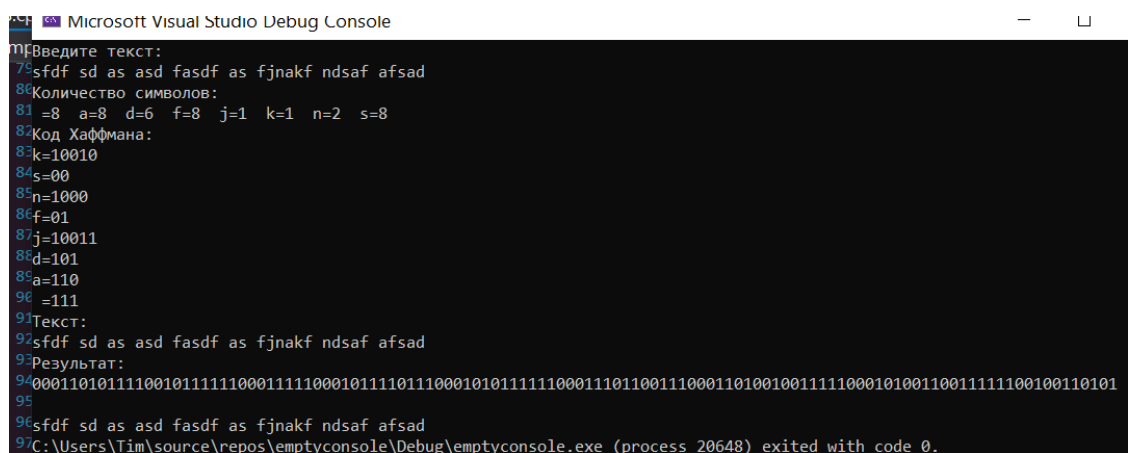
Цель работы заключалась в приобретение практических навыков по реализации и применение алгоритма Хаффмана.

Постановка задачи: изучение алгоритма оптимального префиксного кодирования Хаффмана.

Исходный код программы можно найти по ссылке:

<https://github.com/tim-lwry/siaod/blob/main/Huffman14.cpp>

Результат работы программы:



```
Microsoft Visual Studio Debug Console
Введите текст:
75 sdfsd sd as asd fasdf as fjnakf ndsaf afsad
86 Количество символов:
81 =8 a=8 d=6 f=8 j=1 k=1 n=2 s=8
82 Код Хаффмана:
83 k=10010
84 s=00
85 n=1000
86 f=01
87 j=10011
88 d=101
89 a=110
90 =111
91 Текст:
92 sdfsd sd as asd fasdf as fjnakf ndsaf afsad
93 Результат:
94 000110101111001011111000101111011100010101111100011101100111000110100100111110001010011001111100100110101
95
96 sdfsd sd as asd fasdf as fjnakf ndsaf afsad
97 C:\Users\Tim\source\repos\emptyconsole\Debug\emptyconsole.exe (process 20648) exited with code 0.
```

Рисунок 1.14 — Результат выполнения

Практическая работа № 15. Стратегии и методы построения алгоритмов.

Цель работы заключалась в приобретение практических навыков по реализации и применение алгоритмов.

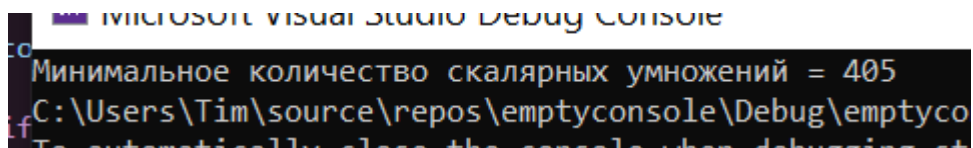
Постановка задачи: разработать процедуру оптимального способа расстановки скобок в произведении последовательности матриц, размеры которых равны (5,10,3,12,5,50,6), чтобы количество скалярных умножений стало минимальным.

Вариант: №2.

Исходный код программы можно найти по ссылке:

<https://github.com/tim-lwry/siaod/blob/main/MatrixMultiplication15.cpp>

Результат работы программы:

The image is a screenshot of the Microsoft Visual Studio Debug Console. It shows the output of a program. The first line of output is "Минимальное количество скалярных умножений = 405". Below this, there is a file path: "C:\Users\Tim\source\repos\emptyconsole\Debug\emptyco". The text is partially obscured by a black box. The console window has a title bar that says "Microsoft Visual Studio Debug Console".

```
Microsoft Visual Studio Debug Console
Минимальное количество скалярных умножений = 405
C:\Users\Tim\source\repos\emptyconsole\Debug\emptyco
To automatically close the console when debugging st
```

Рисунок 1.15 — Результат выполнения

Практическая работа № 16. Метод линейного программирования.

Цель работы заключалась в приобретение практических навыков по реализации и применение методов линейного программирования.

Постановка задачи: покажите, что следующая задача линейного программирования является неразрешимой:

Максимизировать $3x_1 - 2x_2$ при условиях

1) $x_1 + x_2 \leq 2$

2) $-2x_1 - 2x_2 \leq -10$

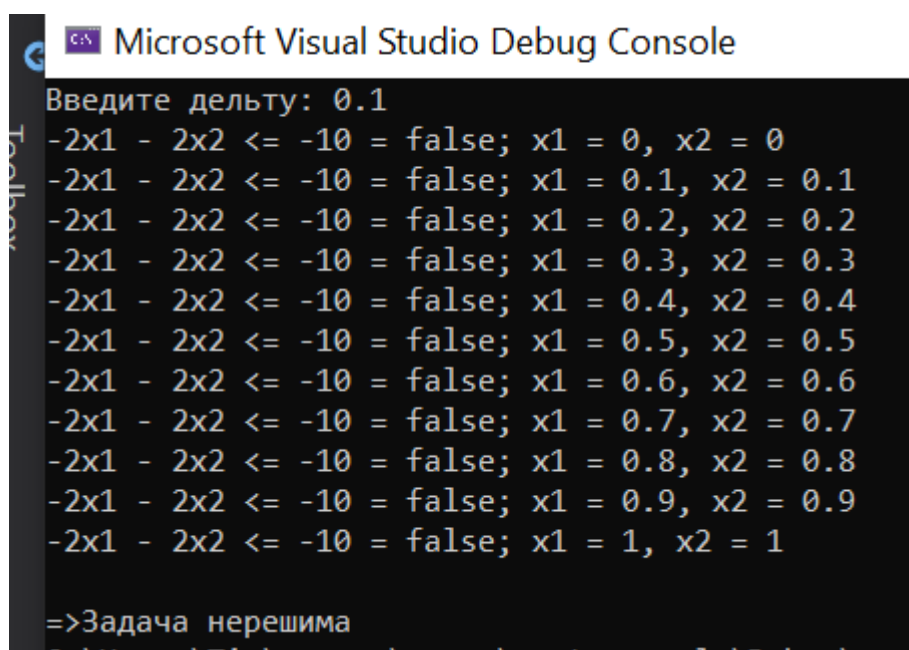
3) $x_1, x_2 \geq 0$

Вариант: №3.

Исходный код программы можно найти по ссылке:

<https://github.com/tim-lwry/siaod/blob/main/LinearPrImptask.cpp>

Результат работы программы:



```
Microsoft Visual Studio Debug Console
Введите дельту: 0.1
-2x1 - 2x2 <= -10 = false; x1 = 0, x2 = 0
-2x1 - 2x2 <= -10 = false; x1 = 0.1, x2 = 0.1
-2x1 - 2x2 <= -10 = false; x1 = 0.2, x2 = 0.2
-2x1 - 2x2 <= -10 = false; x1 = 0.3, x2 = 0.3
-2x1 - 2x2 <= -10 = false; x1 = 0.4, x2 = 0.4
-2x1 - 2x2 <= -10 = false; x1 = 0.5, x2 = 0.5
-2x1 - 2x2 <= -10 = false; x1 = 0.6, x2 = 0.6
-2x1 - 2x2 <= -10 = false; x1 = 0.7, x2 = 0.7
-2x1 - 2x2 <= -10 = false; x1 = 0.8, x2 = 0.8
-2x1 - 2x2 <= -10 = false; x1 = 0.9, x2 = 0.9
-2x1 - 2x2 <= -10 = false; x1 = 1, x2 = 1
=>Задача нерешима
```

Рисунок 1.16 — Результат выполнения

ЗАКЛЮЧЕНИЕ

В результате выполнения практических работ были получены знания и навыки использования различных алгоритмов сортировок, алгоритмов поиска, различных списках, очередях, стеках. Также были получены знания и навыки по применению бинарного дерева, хеширования, графов, алгоритмов сжатия и кодирования данных, стратегии и методы построения алгоритмов, а также методы линейного программирования.

СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Утилиты даты и времени — URL: <https://en.cppreference.com/w/cpp/chrono> (Дата обращения: 02.03.2022)
2. Оценка сложности алгоритмов — URL: <http://www.c-cpp.ru/content/printf> (Дата обращения: 18.04.2022)
3. fstream — URL: <https://www.cplusplus.com/reference/fstream/fstream/> (Дата обращения: 27.04.2022)
4. Printing brackets in Matrix Chain Multiplication Problem — URL: <https://www.geeksforgeeks.org/printing-brackets-matrix-chain-multiplication-problem/> (Дата обращения: 08.05.2022)
5. Лекции по дисциплине “Структуры и алгоритмы обработки данных” — URL: <https://online-edu.mirea.ru/course/view.php?id=6600>;
6. Сайт вопросов и ответов для программистов Stack Overflow – URL: <https://stackoverflow.com/>.