**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра АПУ**

**ОТЧЕТ**

**по лабораторной работе №3**

**на тему: «Последовательный и бинарный поиск   
в одномерном массиве»**

**по дисциплине «Алгоритмы и структуры данных»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3391 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Николаев В.Ю. |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Ряскова Е.Б. |

Санкт-Петербург

2024 г.

## Цель работы

Получить практические навыки работы с улучшенными методами сортировки массивов структур и применением алгоритмов поиска.

## Задачи

В программу, написанную в лабораторной работе № 2, добавьте возможность осуществлять поиск элемента в **отсортированном** массиве структур (поиск проводится по тому же полю (полям), по которому осуществлялась сортировка в соответствии с вариантом задания) двумя способами:

* прямой (последовательный) поиск,
* бинарный поиск.

## Описание программы

Программа реализует структуру данных «Студент», включающую следующие поля:

* ФИО
* Курс
* Факультет

## Алгоритмы поиска

1. Прямой поиск:

|  |
| --- |
| void sequential\_search(Student\* sorted\_students, int size, const int& key, int& left\_index, int& right\_index) {  left\_index = -1;  right\_index = -1;  for (int i = 0; i < size; ++i) {  if (sorted\_students[i].course == key) {  if (left\_index == -1) {  left\_index = i;  }  right\_index = i;  }  }  if (left\_index == -1) {  std::cout << "Элемент с ключом " << key << " не найден.\n";  }  } |

1. Бинарный поиск:

|  |
| --- |
| void binary\_search(Student\* sorted\_students, int size, const int& key, int& left\_index, int& right\_index) {  left\_index = -1;  right\_index = -1;  for (int i = 1; i < size; ++i) {  if (sorted\_students[i].course < sorted\_students[i - 1].course) {  std::cout << "Ошибка: массив не отсортирован. Бинарный поиск невозможен.\n";  return;  }  }  int left = 0, right = size - 1;  while (left <= right) {  int mid = (left + right) / 2;  if (sorted\_students[mid].course == key) {  left\_index = mid;  right = mid - 1;  } else if (sorted\_students[mid].course > key) {  right = mid - 1;  } else {  left = mid + 1;  }  }  if (left\_index != -1) {  left = left\_index;  right = size - 1;  while (left <= right) {  int mid = (left + right) / 2;  if (sorted\_students[mid].course == key) {  right\_index = mid;  left = mid + 1;  } else if (sorted\_students[mid].course > key) {  right = mid - 1;  } else {  left = mid + 1;  }  }  }  if (left\_index == -1) {  std::cout << "Элемент с ключом " << key << " не найден.\n";  }  } |

## Меню программы

Программа имеет следующее меню для взаимодействия с пользователем:

1. Добавить студента
2. Показать студентов
3. Выполнить сортировку
4. Поиск студентов по курсу
5. Выход

При выборе пункта «Показать студентов» показывается следующая страница меню:

1. Сортировка пузырьком
2. Сортировка выбором
3. Быстрая сортировка
4. Пирамидальная сортировка
5. Отмена

При выборе пункта «Выполнить сортировку» показывается следующая страница меню:

1. Прямой (последовательный) поиск
2. Бинарный поиск
3. Отмена

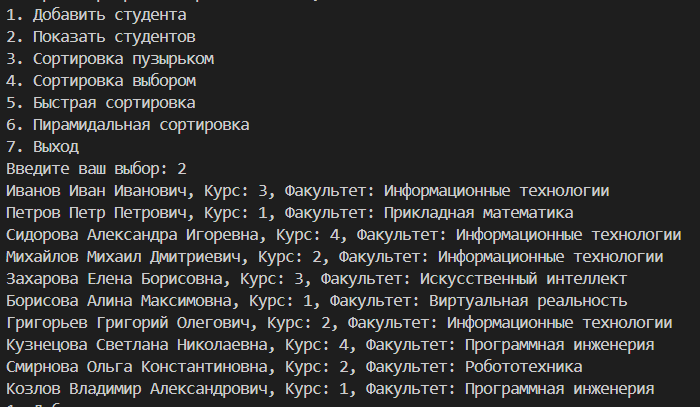
## Результаты работы программы

Пример ввода данных:

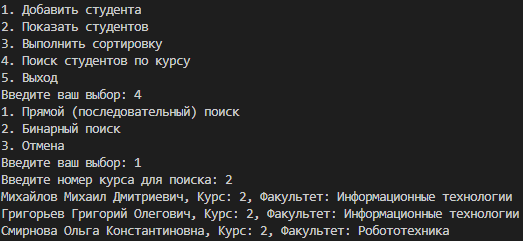


|  |  |  |
| --- | --- | --- |
| ФИО | Курс | Факультет |
| Иванов Иван Иванович | 3 | Информационные технологии |
| Петров Петр Петрович | 1 | Прикладная математика |
| Сидорова Александра Игоревна | 4 | Информационные технологии |
| Михайлов Михаил Дмитриевич | 2 | Информационные технологии |
| Захарова Елена Борисовна | 3 | Искусственный интеллект |
| Борисова Алина Максимовна | 1 | Виртуальная реальность |
| Григорьев Григорий Олегович | 2 | Информационные технологии |
| Кузнецова Светлана Николаевна | 4 | Программная инженерия |
| Смирнова Ольга Константиновна | 2 | Робототехника |
| Козлов Владимир Александрович | 1 | Программная инженерия |

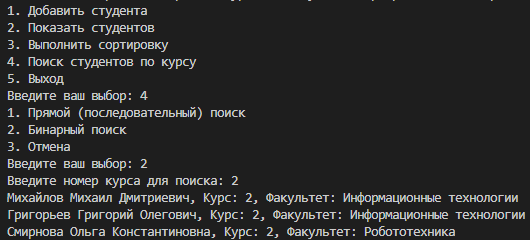
Исходный массив студентов:



После прямого (последовательного) поиска:



После бинарного поиска:



## Выводы

В ходе выполнения лабораторной работы №3 были получены практические навыки работы с методами поиска в отсортированных массивах данных, а именно последовательным и бинарным поиском. Программа, разработанная в рамках работы, успешно реализует структуру «Студент», позволяя проводить сортировку и поиск по ключевому полю — курсу студентов. Последовательный поиск продемонстрировал простоту реализации, однако при увеличении объема данных данный метод теряет свою эффективность. Бинарный поиск показал более высокую производительность, но требует предварительной сортировки массива. Результаты работы программы показали, что оба метода поиска корректно выполняют свои функции.