Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ КОСМИЧЕСКИХ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

ОТЧЕТ
по дисциплине Алгоритмы и структуры данных
Практическая работа №3 — Поиск

Преподаватель _____ Матковский И. В.
подпись, дата инициалы, фамилия

Студент КИ19-07Б, 031941597 _____ Горбацевич А. А.
номер группы, зачётной книжки подпись, дата инициалы, фамилия

Красноярск 2020

Содержание

1. Задание на работу.....3
1.1 Разработать для решения поставленной задачи алгоритм; решить поставленную задачу с помощью последовательного поиска и поиска, предусмотренного вариантом. Оценить алгоритмы.....3
2. Задание на вариант.....4
2.1 Найти заданные элемент отсортированной целочисленной последовательности двоичным и линейным поиском.....4
3. Исходный код программы.....5
4. Теоретические оценки временной сложности алгоритмов.....8
4.1 Линейный поиск.....8
4.2 Интерполирующий поиск.....8
5. Экспериментальные оценки временной и пространственной сложности программы.....9
Приложение А Результаты работы программы.....11

1. Задание на работу

1.1 Разработать для решения поставленной задачи алгоритм; решить поставленную задачу с помощью последовательного поиска и поиска, предусмотренного вариантом. Оценить алгоритмы.

2. Задание на вариант

2.1 Найти заданные элемент отсортированной целочисленной последовательности двоичным и линейным поиском.

3. Исходный код программы

```
// dsaa_03.cpp
// Горбацевич Андрей
#include <iostream>
#include <fstream>
#include <vector>
#include <chrono>
#include <cmath>
#include <cassert>
#include <algorithm>


inline void time_passed(std::chrono::system_clock::time_point start, double &holder) {
    auto stop = std::chrono::high_resolution_clock::now();
    auto duration = std::chrono::duration_cast<std::chrono::microseconds>(stop - start);
    holder = duration.count();
}

typedef long long num_type;
typedef std::vector<num_type> num_list;

void test(const std::string& filename, int num);
void algo_01(const num_list &cont, const num_type &num, num_type &found);  // linear search
void algo_02(const num_list &cont, const num_type &num, num_type &found);  // interpolating search


int main() {
    for (auto num : {68, 288, 610, 666, 665, 73720, 95210, 99998, 0, -1}) {
        test("tests/snba1.txt", num);
    }

    for (auto num : {68, 288, 610, 666, 665, 73720, 95210, 99999, 1, 0}) {
        test("tests/snba2.txt", num);
    }

    for (auto num : {3, 69, 610, 6666, 6656, 73720, 95210, 99999, 0, 2}) {
        test("tests/snbb1.txt", num);
    }

    for (auto num : {3, 69, 610, 6666, 6656, 73720, 95210, 99998, 1, 0}) {
        test("tests/snbb2.txt", num);
    }

    for (auto num : {3, 69, 610, 6666, 6656, 73720, 95210, 99991, 1, 99999}) {
        test("tests/snbc1.txt", num);
    }

    for (auto num : {3, 69, 610, 6666, 6656, 73720, 95210, 99999, 4, 1}) {
        test("tests/snbc2.txt", num);
    }
```

```
    for (auto num : {1, 2, 3, 4}) {
        test("tests/snla1.txt", num);
        test("tests/snla2.txt", num);
        test("tests/snlb1.txt", num);
        test("tests/snlb2.txt", num);
        test("tests/snlc1.txt", num);
        test("tests/snlc2.txt", num);
    }

    return 0;
}

void test(const std::string& filename, int num) {
    std::ifstream data_file(filename);
    int elc;
    data_file >> elc;

    num_list data(elc);
    for (int i = 0; i < elc; i++) {
        data_file >> data[i];
    }

    double fft, sft = fft = 0;
    num_type fff, sff = fff = -1;

    {
        auto start = std::chrono::high_resolution_clock::now();
        algo_01(data, num, fff);
        time_passed(start, fft);
    }
    {
        auto start = std::chrono::high_resolution_clock::now();
        algo_02(data, num, sff);
        time_passed(start, sft);
    }
    {
        assert(fff == sff);
    }

    printf(
            "algo_01 is %s than algo_02 by %.0f microseconds\nFound %d at: %s in %s\n\n",
            (fft > sft? "slower" : "faster"), fabs(fft - sft), num,
            (fff == -1? "not found" : std::to_string(fff).c_str()), filename.c_str()
    );
}

void algo_01(const num_list &cont, const num_type &num, num_type &found) {  // O(n)
    for (int i = 0; i < cont.size(); i++) {
        num_type item = cont[i];
        if (item == num) {
            found = i;
            break;
        }
    }
}
```

```
void algo_02(const num_list &cont, const num_type &num, num_type &found) {  // O(log(log(n))); worst O(n)
   num_type c,
            l = 0,
            r = cont.size()-1;
   while (cont[l] < num && cont[r] > num) {
      num_type rv = cont[r];
      num_type lv = cont[l];
      if (rv == lv) {
         break;
      }

      c = l + ((num - lv) * (r - l)) / (rv - lv);
      if (cont[c] < num) {
         l = c + 1;
      }
      else if (cont[c] > num) {
         r = c - 1;
      }
      else {
         found = c;
         return;
      }
   }

   if (cont[l] == num) {
      found = l;
   }
   else if (cont[r] == num) {
      found = r;
   }
}
```

4. Теоретические оценки временной сложности алгоритмов

4.1 Линейный поиск.

Временная сложность алгоритма: O(n).

4.2 Интерполирующий поиск.

Временная сложность алгоритма: O(log(log(n))); в худшем: O(n), но, исходя из входных данных, можно понять, что такого не случится (на наборе данных).

5. Экспериментальные оценки временной и пространственной сложности программы

Размер входного набора данных	Искомое число	Время, algo_01, микросекунды	Занимаемое пространство, algo_01, байты	Время, algo_02, микросекунды	Занимаемое пространство, algo_02, байты
50000	68	0	8	0	40
50000	288	0	8	0	40
50000	610	0	8	0	40
50000	666	0	8	0	40
50000	665	997	8	0	40
50000	73720	1022	8	0	40
50000	95210	0	8	0	40
50000	99998	992	8	0	40
50000	0	0	8	0	40
50000	-1	0	8	0	40
50000	68	0	8	0	40
50000	288	0	8	0	40
50000	610	0	8	0	40
50000	666	0	8	0	40
50000	665	0	8	0	40
50000	73720	0	8	0	40
50000	95210	997	8	0	40
50000	99999	0	8	0	40
50000	1	0	8	0	40
50000	0	0	8	0	40
33334	3	0	8	0	40
33334	69	0	8	0	40
33334	610	0	8	0	40
33334	6666	0	8	0	40
33334	6656	0	8	0	40
33334	73720	999	8	0	40
33334	95210	0	8	0	40
33334	99999	0	8	0	40
33334	0	0	8	0	40

33334	2	0	8	0	40
66666	3	0	8	0	40
66666	69	961	8	0	40
66666	610	0	8	0	40
66666	6666	0	8	0	40
66666	6656	0	8	0	40
66666	73720	0	8	0	40
66666	95210	996	8	0	40
66666	99998	0	8	0	40
66666	1	0	8	0	40
66666	0	0	8	0	40
9594	3	0	8	0	40
9594	69	0	8	0	40
9594	610	0	8	0	40
9594	6666	0	8	0	40
9594	6656	0	8	0	40
9594	73720	0	8	0	40
9594	95210	0	8	0	40
9594	99991	0	8	0	40
9594	1	0	8	0	40
9594	99999	0	8	0	40
90406	3	0	8	0	40
90406	69	0	8	0	40
90406	610	0	8	0	40
90406	6666	0	8	0	40
90406	6656	0	8	0	40
90406	73720	0	8	0	40
90406	95210	0	8	0	40
90406	99999	1015	8	0	40
90406	4	0	8	0	40
90406	1	0	8	0	40
1	1	0	8	0	40

Приложение А
Результаты работы программы

```
C:\Users\Admin\CLionProjects\instp_01\cmake-build-debug\instp_01.exe
algo_01 is faster than algo_02 by 0 microseconds
Found 68 at: 34 in tests/snba1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 288 at: 144 in tests/snba1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 610 at: 305 in tests/snba1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 666 at: 333 in tests/snba1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 665 at: not found in tests/snba1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 73720 at: 36860 in tests/snba1.txt

algo_01 is slower than algo_02 by 997 microseconds
Found 95210 at: 47605 in tests/snba1.txt

algo_01 is slower than algo_02 by 1000 microseconds
Found 99998 at: 49999 in tests/snba1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 0 at: 0 in tests/snba1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found -1 at: not found in tests/snba1.txt
```

Рисунок 1: Результат работы программы (1)

```
algo_01 is faster than algo_02 by 0 microseconds
Found 68 at: not found in tests/snba2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 288 at: not found in tests/snba2.txt

algo_01 is slower than algo_02 by 996 microseconds
Found 610 at: not found in tests/snba2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 666 at: not found in tests/snba2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 665 at: 332 in tests/snba2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 73720 at: not found in tests/snba2.txt

algo_01 is slower than algo_02 by 997 microseconds
Found 95210 at: not found in tests/snba2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 99999 at: 49999 in tests/snba2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 1 at: 0 in tests/snba2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 0 at: not found in tests/snba2.txt
```

Рисунок 2: Результат работы программы (2)

```
algo_01 is faster than algo_02 by 0 microseconds
Found 3 at: 1 in tests/snbb1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 69 at: 23 in tests/snbb1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 610 at: not found in tests/snbb1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 6666 at: 2222 in tests/snbb1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 6656 at: not found in tests/snbb1.txt

algo_01 is slower than algo_02 by 996 microseconds
Found 73720 at: not found in tests/snbb1.txt

algo_01 is slower than algo_02 by 997 microseconds
Found 95210 at: not found in tests/snbb1.txt

algo_01 is slower than algo_02 by 970 microseconds
Found 99999 at: 33333 in tests/snbb1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 0 at: 0 in tests/snbb1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 2 at: not found in tests/snbb1.txt
```

Рисунок 3: Результат работы программы (3)

```
algo_01 is slower than algo_02 by 968 microseconds
Found 3 at: not found in tests/snbb2.txt

algo_01 is slower than algo_02 by 993 microseconds
Found 69 at: not found in tests/snbb2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 610 at: 406 in tests/snbb2.txt

algo_01 is slower than algo_02 by 987 microseconds
Found 6666 at: not found in tests/snbb2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 6656 at: 4437 in tests/snbb2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 73720 at: 49146 in tests/snbb2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 95210 at: 63473 in tests/snbb2.txt

algo_01 is slower than algo_02 by 997 microseconds
Found 99998 at: 66665 in tests/snbb2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 1 at: 0 in tests/snbb2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 0 at: not found in tests/snbb2.txt
```

Рисунок 4: Результат работы программы (4)

```
algo_01 is faster than algo_02 by 0 microseconds
Found 3 at: 3 in tests/snbc1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 69 at: not found in tests/snbc1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 610 at: not found in tests/snbc1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 6666 at: not found in tests/snbc1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 6656 at: not found in tests/snbc1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 73720 at: not found in tests/snbc1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 95210 at: not found in tests/snbc1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 99991 at: 9593 in tests/snbc1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 1 at: 1 in tests/snbc1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 99999 at: not found in tests/snbc1.txt
```

Рисунок 5: Результат работы программы (5)

```
algo_01 is slower than algo_02 by 994 microseconds
Found 3 at: not found in tests/snbc2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 69 at: 48 in tests/snbc2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 610 at: 497 in tests/snbc2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 6666 at: 5805 in tests/snbc2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 6656 at: 5797 in tests/snbc2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 73720 at: 66439 in tests/snbc2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 95210 at: 86031 in tests/snbc2.txt

algo_01 is slower than algo_02 by 1021 microseconds
Found 99999 at: 90405 in tests/snbc2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 4 at: 0 in tests/snbc2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 1 at: not found in tests/snbc2.txt
```

Рисунок 6: Результат работы программы (6)

```
algo_01 is faster than algo_02 by 0 microseconds
Found 1 at: not found in tests/snla1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 1 at: 0 in tests/snla2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 1 at: not found in tests/snlb1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 1 at: 0 in tests/snlb2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 1 at: 0 in tests/snlc1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 1 at: not found in tests/snlc2.txt
```

Рисунок 7: Результат работы программы (7)

```
algo_01 is faster than algo_02 by 0 microseconds
Found 2 at: 0 in tests/snla1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 2 at: not found in tests/snla2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 2 at: not found in tests/snlb1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 2 at: not found in tests/snlb2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 2 at: not found in tests/snlc1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 2 at: not found in tests/snlc2.txt
```

Рисунок 8: Результат работы программы (8)

```
algo_01 is faster than algo_02 by 0 microseconds
Found 3 at: not found in tests/snla1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 3 at: not found in tests/snla2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 3 at: 0 in tests/snlb1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 3 at: not found in tests/snlb2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 3 at: not found in tests/snlc1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 3 at: not found in tests/snlc2.txt
```

Рисунок 9: Результат работы программы (9)

```
algo_01 is faster than algo_02 by 0 microseconds
Found 4 at: not found in tests/snla1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 4 at: not found in tests/snla2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 4 at: not found in tests/snlb1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 4 at: not found in tests/snlb2.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 4 at: not found in tests/snlc1.txt

algo_01 is faster than algo_02 by 0 microseconds
Found 4 at: 0 in tests/snlc2.txt


Process finished with exit code 0
```

Рисунок 10: Результат работы программы (10)