

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 №5 — Сортировка

Преподаватель

подпись, дата

Матковский И. В.

инициалы, фамилия

Студент

КИ19-07Б, 031941597

номер группы, зачётной книжки

подпись, дата

Горбацевич А. А.

инициалы, фамилия

Красноярск 2020

Содержание

| | |
|---|---|
| 1. Задание на работу..... | 3 |
| 1.1 Пример второй уровень..... | 3 |
| 2. Задание на вариант..... | 4 |
| 3. Исходный код программы..... | 5 |
| 4. Теоретические оценки временной сложности алгоритмов..... | 6 |
| 5. Экспериментальные оценки временной и пространственной сложности программы..... | 7 |
| Приложение А Результаты работы программы..... | 8 |

1. Задание на работу

1.1 Разработать программы, демонстрирующие два заданных метода сортировки; сравнить их эффективность.

2. Задание на вариант

2.1 Алгоритмы сортировки: Stupid Sort и Shell Sort.

3. Исходный код программы

```
// dsaa_05.cpp
// Горбачев Андрей
#include <iostream>
#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 double num_type;
typedef std::vector<num_type> num_list;

void test();
void algo_01(num_list &cont); // Stupid sort
void algo_02(num_list &cont); // Shell sort

int main() {
    test();

    return 0;
}

void test() {
    int n;
    std::cout << "Count of items:\n";
    std::cin >> n;
    num_list data;
    data.reserve(n);
    for (int i = 0; i < n; i++) {
        int d;
        std::cin >> d;
        data.push_back(d);
    }

    num_list copy_1(data);
    num_list copy_2(data);

    double fft, sft = fft = 0;

    {
        auto start = std::chrono::high_resolution_clock::now();
        algo_01(copy_1);
        time_passed(start, fft);
    }
```

```

{
    auto start = std::chrono::high_resolution_clock::now();
    algo_02(copy_2);
    time_passed(start, sft);
}
{
    assert(copy_1 == copy_2);
}

printf(
    "algo_01 (%.0f) is %s than algo_02 (%.0f) by %.0f microseconds\n",
    fft, (fft > sft? "slower" : "faster"), sft, fabs(fft - sft)
);
}

void algo_01(num_list &cont) { // Stupid sort
    auto vit_b = cont.begin() + 1;

    while (vit_b != cont.end()) {
        auto vit_p = vit_b - 1;
        if (*vit_p > *vit_b) {
            std::swap(*vit_p, *vit_b);
            vit_b = cont.begin() + 1;
        }
        else {
            vit_b += 1;
        }
    }
}

void algo_02(num_list &cont) { // Shell sort
    auto vit_first = cont.begin();
    auto vit_last = cont.end();
    for(auto d = (vit_last - vit_first) / 2; d != 0; d /= 2) {
        for(auto i = vit_first + d; i != vit_last; ++i) {
            for(auto j = i; j - vit_first >= d; j -= d) {
                if (*j < *(j - d)) {
                    std::swap(*j, *(j - d));
                }
            }
        }
    }
}
}

```

4. Теоретические оценки временной сложности алгоритмов

4.1 Stupid Sort: $O(n^3)$ в худшем

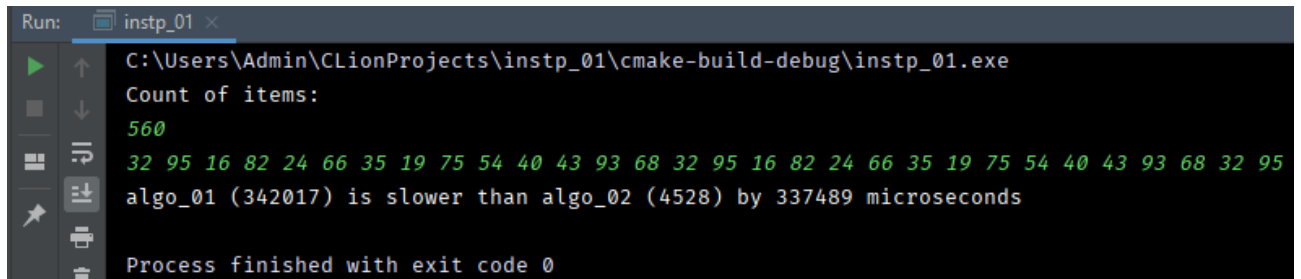
4.2 Shell Sort: $O(n^2)$ в худшем

5. Экспериментальные оценки временной и пространственной сложности программы

| N элементов | algo01, время, микросекунды | algo02, время, микросекунды |
|------------------------|------------------------------------|------------------------------------|
| 0 | 0 | 0 |
| 14 | 0 | 0 |
| 140 | 5063 | 0 |
| 560 | 342017 | 4528 |

Приложение А

Результаты работы программы



The screenshot shows a terminal window titled 'Run: instp_01'. The output text is as follows:

```
C:\Users\Admin\CLionProjects\instp_01\cmake-build-debug\instp_01.exe
Count of items:
560
32 95 16 82 24 66 35 19 75 54 40 43 93 68 32 95 16 82 24 66 35 19 75 54 40 43 93 68 32 95
algo_01 (342017) is slower than algo_02 (4528) by 337489 microseconds
Process finished with exit code 0
```

Рисунок 1: Результат работы программы на 560 элементах