**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

**Кафедра информационных управляющих систем**

**ОТЧЁТ**

по лабораторной работе №1

по дисциплине «Структура и алгоритмы обработки данных  
в информационных системах и сетях»

Выполнил: студент группы ИСТ-813, /Кравец А.Ю./

«06» ноября 2020 г. \_\_\_\_\_\_\_\_\_\_\_/А.Ю. Кравец/

Принял: ст. преподаватель Антонов В.В.

«07» ноября 2020 г. \_\_\_\_\_\_\_\_\_\_\_/В.В. Антонов/

**Задание**: описать алгоритмы пузырьковой сортировки (Bubble sort), сортировки выбором (Selection sort), сортировки вставками (Insertion sort), быстрой сортировки (Quick sort), реализовать их программно, провести сравнение быстродействия алгоритмов.

# 1Описание работы алгоритмов

## 1.1 Пузырьковая сортировка (Bubble sort)

Алгоритм пузырьковой сортировки:

1. Попарное сравнение элементов несортированной части массива: при неверном порядке элементов в паре происходит обмен местами;
2. Несортированная часть массива уменьшается на один элемент. Выполняется пункт 1.

## 1.2 Сортировка выбором (Selection sort)

Сортировка может производится как по выбору максимального значения, так и по выбору минимального значения.

Алгоритм сортировки выбором:

1. Проход по несортированной части массива в поиске минимального (максимального) значения;
2. Найденный минимальный (максимальный) элемент меняется местами с первым (последним, если по максимальному значению) элементом несортированной части массива;
3. Несортированная часть массива уменьшается на один элемент. Выполняется пункт 1.

## 1.3 Сортировка вставками (Insertion sort)

Алгоритм сортировки вставками:

1. Выбирается элемент несортированной части массива;
2. Избранный элемент размещается в нужную позицию в сортированной части массива;
   1. Если шаги 1 и 2 выполняются впервые, избранный элемент становится первым в сортированной части массива.
3. Несортированная часть массива уменьшается на один элемент. Выполняется пункт 1.

## 1.4 Быстрая сортировка (Quick sort)

Алгоритм быстрой сортировки:

1. Выбирается опорный элемент;
2. Массив делится на два подмассива: в одном все элементы меньше или равны опорному элементу (или строго меньше опорного элемента), а во втором все элементы больше опорного элемента (или больше или равны опорному элементу). Причем опорный элемент становится границей между двумя подмассивами;
3. Пункт 1 выполняется для каждого подмассива.

# 2Сравнение алгоритмов

Для сравнения замерялось время сортировки массивов со случайными значениями элементов. Полученные значения для алгоритмов пузырьковой сортировки, сортировки выбором и сортировки вставками представлены в таблице 1. Полученные значения для алгоритма быстрой сортировки представлены в таблице 2.

Таблица 1 – Время выполнения алгоритмов пузырьковой сортировки, сортировки выбором, сортировки вставками

|  |  |  |  |
| --- | --- | --- | --- |
| **Размер массива** | **Время выполнения, с** | | |
| **Bubble Sort** | **Selection Sort** | **Insertion Sort** |
| 10000 | 0,869 | 0,531 | 0,109 |
| 16000 | 1,266 | 0,797 | 0,14 |
| 22000 | 2,047 | 0,829 | 0,234 |
| 28000 | 4,516 | 0,921 | 0,39 |
| 34000 | 6,688 | 1,844 | 0,641 |
| 40000 | 9,344 | 2,844 | 0,719 |
| 46000 | 12,345 | 3,438 | 0,922 |
| 52000 | 15,938 | 4,301 | 0,937 |
| 58000 | 19,798 | 5,36 | 1,625 |
| 64000 | 24,223 | 6,532 | 1,954 |
| 70000 | 28,923 | 7,797 | 2,344 |

Таблица 2 – Время выполнения алгоритма быстрой сортировки

|  |  |  |  |
| --- | --- | --- | --- |
| **Размер массива** | **Время выполнения, с** | **Размер массива** | **Время выполнения, с** |
| 1000000 | 0,172 | 11000000 | 5,328 |
| 2000000 | 0,437 | 12000000 | 5,891 |
| 3000000 | 0,922 | 13000000 | 6,266 |
| 4000000 | 1,282 | 14000000 | 6,954 |
| 5000000 | 2,282 | 15000000 | 7,375 |
| 6000000 | 2,812 | 16000000 | 7,813 |
| 7000000 | 3,234 | 17000000 | 8,516 |
| 8000000 | 3,766 | 18000000 | 8,844 |
| 9000000 | 4,376 | 19000000 | 9,333 |
| 10000000 | 4,797 | 20000000 | 10,017 |

Для сравнения были составлены графики по таблицам №1 и №2.

Рисунок 1 – Время выполнения алгоритмов пузырьковой сортировки, сортировки выбором, сортировки вставками

Рисунок 2 – Время выполнения алгоритма быстрой сортировки

# Вывод:

В данной работе были рассмотрены следующие алгоритмы сортировки: пузырьковая сортировка, сортировка выбором, сортировка вставками, быстрая сортировка.

Данные алгоритмы были реализованы программно. Были получены данные о времени выполнения каждого алгоритма при различных размерах исходного массива.

Исходя из собранных данных (рисунки 1 и 2) ранжирование алгоритмов сортировки в порядке убывания быстродействия имеет вид:

1. Быстрая сортировка (Quick sort);

2. Сортировка вставками (Insertion Sort);

3. Сортировка выбором (Selection Sort);

4. Пузырьковая сортировка (Bubble Sort).