



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ» (ИУ7)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

О Т Ч Е Т

по лабораторной работе № 3

Название: Трудоёмкость алгоритмов сортировки

Дисциплина: Анализ алгоритмов

Студент

ИУ7-52Б

(Группа)

(Подпись, дата)

Е.В. Брянская

(И.О. Фамилия)

Преподаватель

Л.Л. Волкова

(Подпись, дата)

(И.О. Фамилия)

Москва, 2020

Оглавление

Введение	3
1 Аналитическая часть	4
1.1 Пузырьковая сортировка	4
1.2 Сортировка вставками	4
2 Конструкторская часть	5
2.1 Сортировка пузырьком	5
2.2 Сортировка вставками	5
2.3 Требования к ПО	5
2.4 Заготовки тестов	6
3 Технологическая часть	7
3.1 Выбранный язык программирования	7
3.2 Листинг кода	7
3.3 Результаты тестов	7
3.4 Оценка трудоёмкости	7
3.5 Оценка времени	8
4 Исследовательская часть	9
Заключение	10
Список литературы	11

Введение

Трудоёмкость алгоритма - это зависимость стоимости операций от линейного(ых) размера(ов) входа(ов) [1].

Модель вычислений трудоёмкости должна учитывать следующие оценки.

- 1) Стоимость базовых операций. К ним относятся: =, +, -, *, /, ==, !=, <, <=, >, >=, %, +=, -=, *=, /=, [], < <, > >. Каждая из операций имеет стоимость равную 1.
- 2) Оценка цикла. Она складывается из стоимости тела, инкремента и сравнения.
- 3) Оценка условного оператора if. Положим, что стоимость перехода к одной из веток равной 0. В таком случае, общая стоимость складывается из подсчета условия и рассмотрения худшего и лучшего случаев.

Оценка характера трудоёмкости даётся по наиболее быстрорастущему слагаемому.

Сортировка - процесс перегруппировки заданного множества объектов в некотором определенном порядке. Сортировка предпринимается для того, чтобы облегчить последующий поиск элементов в отсортированном множестве.

В этой лабораторной работе будет оцениваться трудоёмкость алгоритмов сортировки. Будут рассмотрены следующие алгоритмы:

- 1) сортировка пузырьком;
- 2) сортировка вставками;
- 3) *****

1. Аналитическая часть

Цель данной работы – оценить трудоёмкость алгоритмов сортировки.

Для достижения поставленной цели необходимо решить ряд следующих **задач**:

- 1) дать математическое описание;
- 2) описать алгоритмы сортировки;
- 3) дать теоретическую оценку трудоёмкости алгоритмов;
- 4) реализовать эти алгоритмы ;
- 5) провести замеры процессорного времени работы алгоритмов на материале серии экспериментов;
- 6) провести сравнительный анализ алгоритмов.

Задача сортировки состоит в перестановке (переупорядочивании) входной последовательности из n чисел a_1, a_2, \dots, a_n , так, чтобы выполнялось условие

$$a'_1 \leq a'_2 \leq \dots \leq a'_n \quad (1.1)$$

(в случае сортировки по неубыванию). Аналогичным образом осуществляется сортировка по невозрастанию.

1.1. Пузырьковая сортировка

Представляет собой популярный, но не эффективный алгоритм сортировки. В его основе лежит многократная перестановка соседних элементов, нарушающих порядок сортировки [2].

1.2. Сортировка вставками

Считается простым алгоритмом сортировки. На каждом шаге алгоритма для очередного элемента находится подходящая позиция в уже отсортированной части массива и осуществляется вставка этого элемента.

2. Конструкторская часть

Рассмотрим выбранные алгоритмы сортировки. Для упрощения задачи будем сортировать последовательность по неубыванию.

2.1. Сортировка пузырьком

Осуществляется проход по массиву от начала до конца, в процессе меняя местами неотсортированные соседние элементы.

В результате первого прохода на последнем месте окажется максимальный элемент. Далее снова делается проход по неотсортированной части массива (от первого до предпоследнего) и так же меняются неупорядоченные соседние элементы. Таким образом, на предпоследнее место будет помещён второй по величине элемент.

Действия повторяются до тех пор, пока не обработается вся неотсортированная часть.

Схема алгоритма представлена на Рис.??.

2.2. Сортировка вставками

В этом алгоритме рассматриваемый массив условно делится на две части: отсортированная и нет.

В начале работы отсортированной частью считается нулевой элемент. Далее берётся каждый следующий и сравнивается с уже отсортированной частью. Находится подходящая для текущего элемента позиция в ней, осуществляется сдвиг уже отсортированных элементов, но больших по величине, чем рассматриваемый. И затем рассматриваемый элемент помещается на найденную позицию.

И так до тех пор, пока не просмотрится вся неотсортированная часть.

Схема алгоритма представлена на Рис.??.

2.3. Требования к ПО

Для корректной работы алгоритмов и проведения тестов необходимо выполнить следующее.

- Обеспечить возможность ввода массива через консоль и выбора алгоритма сортировки.

- В случае ввода некорректных данных вывести соответствующее сообщение. Программа не должна аварийно завершаться.
- Программа должна отсортировать массив и вывести результат на экран.
- Реализовать функцию замера процессорного времени, которое выбранный метод затрачивает на вычисление результатов. Для этого дать возможность ввода размера тестируемого массива. Вывести результаты замеров на экран.

2.4. Заготовки тестов

При проверке на корректность работы реализованных функций необходимо провести следующие тесты:

- отсортировать массив размером в один элемент;
- упорядоченный по невозрастанию массив;
- упорядоченный по неубыванию массив;
- массив, состоящий из одинаковых элементов.

3. Технологическая часть

3.1. Выбранный язык программирования

Для выполнения этой лабораторной работы был выбран язык программирования C++, так как есть большой навык работы с ним и с подключаемыми библиотеками, которые также использовались для проведения тестирования и замеров.

Используемая среда разработки - Visual Studio.

3.2. Листинг кода

Ниже представлены Листинги 3.1 - 3.3 функций, реализующих сортировку массива.

Листинг 3.1 — Сортировка пузырьком

Листинг 3.2 — Сортировка вставками

```
1
```

Листинг 3.3 — ****

3.3. Результаты тестов

Для тестирования были написаны функции, проверяющие, согласно заготовкам выше, случаи. Выводы о корректности работы делаются на основе сравнения результатов.

Все тесты пройдены успешно. Сами тесты представлены ниже (Листинг 3.4).

Листинг 3.4 — Тесты

```
1
```

3.4. Оценка трудоёмкости

Произведём оценку трудоёмкости приведённых алгоритмов. Рассмотрим сортировку массива $A[N]$.

3.5. Оценка времени

Процессорное время измеряется с помощью функции `QueryPerformanceCounter` библиотеки `windows.h` [3]. Осуществление замеров показано ниже (Листинг 3.5).

Листинг 3.5 — Замеры процессорного времени

```
1
```


4. Исследовательская часть

Заключение

Список литературы

1. Трудоёмкость алгоритмов и временные оценки [Электронный ресурс]. Режим доступа: <http://techn.sstu.ru/kafedri/подразделения/1/MetMat/shaturn/theoralg/5.htm>, свободный (дата обращения: 29.09.20).
2. Кормен, Томас Х. и др Алгоритмы: построение и анализ, 3-е изд. : Пер. с англ. - М. : ООО "И.Д. Вильямс 2018. - 1328 с. : ил. - Парал. тит. англ. - ISBN 978-5-8459-2016-4 (рус.).
3. QueryPerformanceCounter function [Электронный ресурс]. Режим доступа: <https://docs.microsoft.com/en-us/windows/win32/api/profileapi/nf-profileapi-queryperformancescounter>, свободный (дата обращения: 01.10.2020).