МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

имени. М. В. ЛОМОНОСОВА

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

ОТЧЕТ ПО ЗАДАНИЮ №7

**«Сборка многомодульных программ. Методы сортировки»**

**Вариант 1214**

Исполнитель: студент 102 группы

Титова Анастасия Игоревна

Преподаватель: Сенюкова Ольга Викторовна

МОСКВА

2014

Содержание

[Содержание 2](#_Toc388609989)

[Постановка задачи 3](#_Toc388609990)

[Таблица сравнения результатов 4](#_Toc388609991)

[Быстрая сортировка 5](#_Toc388609992)

[Структура программы: набор файлов, спецификация функций 6](#_Toc388609993)

# Постановка задачи

Реализовать два метода сортировки массива чисел – метод «пузырька» и рекурсивную реализацию быстрой сортировки – и провести их экспериментальное сравнение. Тип элементов массива – целые числа(int), числа упорядочиваются по невозрастанию. Для каждого из реализуемых методов необходимо предусмотреть возможность работы с массивами длиной от 1 до N (N ≥ 1). Значение N в зависимости от варианта задания либо фиксировано, либо память под массив следует выделять динамически.

При реализации каждого метода вычислить число сравнений элементов, число обменов элементов и провести замер времени работы метода в процессорных тактах. Сравнение методов сортировки необходимо проводить на одних и тех же исходных массивах, при этом следует рассмотреть массивы разной длины. Для вариантов с динамическим выделением памяти рассмотреть n = 10, 100, 1000, 10000. Генерация исходных массивов для сортировки реализуется отдельной функцией, создающей, в зависимости от заданного параметра и заданной длины конкретный массив, в котором:

* элементы уже упорядочены (1);
* элементы упорядочены в обратном порядке (2);
* расстановка элементов случайна (3, 4).

Результаты экспериментов оформить на основе нескольких запусков программы в виде таблицы.

# Таблица сравнения результатов

**Метод сортировки «пузырёк»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **n** | **Параметр** | **Номер сгенерированного массива** | | | | **Среднее значение** |
| **1** | **2** | **3** | **4** |  |
| **10** | **Сравнения** | 54 | 54 | 54 | 54 | 54 |
| **Перемещения** | 0 | 45 | 25 | 15 | 21,25 |
| **Время, такты** | 1438 | 2426 | 2625 | 2310 | 2199,75 |
| **100** | **Сравнения** | 5049 | 5049 | 5049 | 5049 | 5049 |
| **Перемещения** | 0 | 4950 | 2444 | 2344 | 2434,5 |
| **Время, такты** | 94899 | 160031 | 171864 | 172085 | 149719,75 |
| **1000** | **Сравнения** | 500499 | 500499 | 500499 | 500499 | 5005499 |
| **Перемещения** | 0 | 49950 | 24874 | 24774 | 24899,5 |
| **Время, такты** | 11236922 | 16646353 | 18529581 | 18371577 | 16196108,25 |
| **10000** | **Сравнения** | 50004999 | 50004999 | 50004999 | 50004999 | 50004999 |
| **Перемещения** | 0 | 4995000 | 24932809 | 25034620 | 13740607,25 |
| **Время, такты** | 924017724 | 1562432476 | 1715065464 | 1721143315 | 1480664745 |

# Быстрая сортировка

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **n** | **Параметр** | **Номер сгенерированного массива** | | | | **Среднее значение** |
| **1** | **2** | **3** | **4** |  |
| **10** | **Сравнения** | 59 | 56 | 57 | 57 | 57,25 |
| **Перемещения** | 0 | 5 | 4 | 4 | 3,25 |
| **Время, такты** | 2667 | 2510 | 2930 | 2615 | 2680,5 |
| **100** | **Сравнения** | 880 | 877 | 995 | 995 | 936,75 |
| **Перемещения** | 0 | 50 | 126 | 126 | 75,5 |
| **Время, такты** | 23121 | 23583 | 38598 | 35133 | 30108,75 |
| **1000** | **Сравнения** | 11987 | 11984 | 15062 | 15062 | 13523,75 |
| **Перемещения** | 0 | 500 | 1974 | 1974 | 1112 |
| **Время, такты** | 268852 | 276822 | 479451 | 478894 | 376004,8 |
| **10000** | **Сравнения** | 153631 | 153628 | 201314 | 197949 | 176630,5 |
| **Перемещения** | 0 | 5000 | 27861 | 27885 | 15186,5 |
| **Время, такты** | 3286637 | 3679074 | 6593989 | 6523009 | 5020677 |

# Структура программы: набор файлов, спецификация функций

**Файлы:** main.c, functions.asm

**Функции:**

* **unsigned long long timestamp(void)** – возвращает количество тактов процессора, совершенных с начала работы программы.
* **int compare (int a, int b)** – сравнивает два числа, возвращает 1, если a<b, ноль в противном случае.
* **void xchng (int \* a, int \* b)** – меняет местами содержимое параметров
* **void sorted (int n, int \* a)** – сгенерировать массив размерности n, элементы которого упорядочены по не возрастанию.
* **void unsort(int n, int \* a)** - сгенерировать массив размерности n, элементы которого упорядочены по не убыванию.
* **void randomize (int n, int \* a)** - сгенерировать массив размерности n.
* **void bubble (int n, int \* a)** – отсортировать массив a размерности n методом «пузырька».
* **void qsrt(int n, int \* a)** – быстрая сортировка массива а размерности n.

**Глобальные переменные:**

* **int xchng\_cnt** – счетчик обменов элементов
* **int cmp\_cnt** – счетчик сравнений элементов

**Содержание Makefile**

program : main.o functions.o

gcc -march=i386 -o program functions.o main.o -m32

main.o : main.c

gcc -std=c99 -c -o main.o main.c -m32

functions.o : functions.asm

nasm -f elf32 -o functions.o functions.asm

clean :

rm program main.o functions.o

**Список литературы**

1. Трифонов Н.П., Пильщиков В.Н, Задания практикума на ЭВМ (1 курс). Методическая разработка (составители). – М.: ВМК МГУ, 2001.
2. Кормен Т., Лейзерсон С., Риверст Р., Штайн К. Алгоритмы: построение и анализ. Второе издание. – М.: «Вильямс», 2005.
3. Кнут Д. Искусство программирования для ЭВМ. том 3. – М.: Мир, 1978.
4. Лорин Г. Сортировка и системы сортировки. – М.: Наука, 1983.
5. Вирт Н. Алгоритмы и структуры данных. – М.: Мир, 1989
6. Столлман Р., МакГрат Р. GNU Make. - http://www.linuxlib.ru/prog/make\_379\_manual.html..