

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ М. В. ЛОМОНОСОВА
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

ОТЧЕТ ПО ЗАДАНИЮ №1

«Методы сортировки»

Вариант 2 / 3 / 2 / 3

Выполнил:
студент 104 группы
Воробьев С. Ю.

Преподаватель:
Сенюкова О. В.

Москва
2020

Содержание

Постановка задачи	2
Результаты экспериментов	3
Структура программы и спецификация функций	4
Отладка программы, тестирование функций	5
Анализ допущенных ошибок	6
Список цитируемой литературы	7

Постановка задачи

В данной задаче нужно реализовать два метода сортировки массива чисел и провести их экспериментальное сравнение. В моем варианте нужно реализовать сортировку выбором и сортировку Шелла для 64-битных целых чисел. После сортировки массивы должны быть упорядочены по неубыванию модулей элементов. Для каждого из реализуемых методов необходимо предусмотреть возможность работы с массивами длины от 1 до N ($N \geq 1$). Значение N пробегает числа 10, 100, 1000, 10000. При реализации каждого метода нужно вычислить число сравнений элементов и число перемещений (обменов) элементов. Сравнение методов сортировки необходимо проводить на одних и тех же исходных массивах, при этом следует рассмотреть массивы разной длины. Генерация исходных массивов для сортировки реализуется отдельной функцией, создающей в зависимости от заданного параметра и заданной длины конкретный массив, в котором:

1. элементы уже упорядочены
2. элементы упорядочены в обратном порядке
3. расстановка элементов случайна

Все полученные данные необходимо занести в таблицы, представленные на следующей странице.

Результаты экспериментов

n	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	45	45	45	45	-
	Перемещения	0	5	6	6	4.25
100	Сравнения	4950	4959	4950	4950	-
	Перемещения	0	50	96	92	59.5
1000	Сравнения	499500	499500	499500	499500	-
	Перемещения	0	500	992	994	865.75
10000	Сравнения	49995000	49995000	49995000	49995000	-
	Перемещения	0	5000	9989	9995	6246

Таблица 1: Результаты работы сортировки выбором

n	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	22	35	29	36	30.5
	Перемещения	0	13	7	14	8.5
100	Сравнения	503	763	818	758	710.5
	Перемещения	0	260	315	255	207.5
1000	Сравнения	8006	12706	14431	14390	12383.25
	Перемещения	0	4700	6425	6384	4377.25
10000	Сравнения	120005	182565	265588	258141	206574.75
	Перемещения	0	62560	145583	138136	86569.75

Таблица 2: Результаты работы сортировки методом Шелла

Теоретические оценки:

Оценка работы сортировки простым выбором [1]:

Число сравнений всегда одно и то же из-за специфики алгоритма и равняется $\frac{n(n-1)}{2} = O(n^2)$

Число перемещений зависит от изначального порядка в массиве. Наилучший случай – 0, наихудший – $n - 1$

Итого: $O(n^2) + O(n) = O(n^2)$

Оценка работы сортировки Шелла [2]:

В наилучшем случае сортировка работает за n .

В наихудшем за $n(\log n)^2$

В среднем $n(\log n)^2$

Структура программы и спецификация функций

1. `int abs64(int64_t a);`
2. `void check_array(int64_t *arr, int n)`
3. `void print_arr(int64_t *arr, int n)`
4. `int64_t *generate_array(int n, int type)`
5. `void selection_sort(int64_t *arr, int n, int *comps, int *swaps)`
6. `void shell_sort (int64_t *arr, int n, int *comps, int *swaps)`
7. `void cmp (int n, int type)`

1. `int abs64(int64_t a)` `a` – 64 битное число, функция возвращает его модуль
2. `void check_array(int64_t *arr, int n)` `arr` – указатель на массив, `n` – его размерность. Функция проверяет массив на корректность сортировки и выдает ошибку в случае неверного ответа.
3. `void print_arr(int64_t *arr, int n)` `arr` – указатель на массив, `n` – его размерность. Функция печатает массив.
4. `int64_t *generate_array(int n, int type)` `n` – размерность запрашиваемого массива, `type` – его тип. Функция возвращает указатель на начало массива.
5. `void selection_sort(int64_t *arr, int n, int *comps, int *swaps)` `arr` – указатель на массив, `n` – количество элементов, `comps` – указатель на переменную с количеством сравнений, `swaps` на переменную с количеством обменов. Функция сортирует массив методом простого выбора.
6. `void shell_sort (int64_t *arr, int n, int *comps, int *swaps)` – аргументы аналогичные, функция сортирует массив методом Шелла.
7. `void cmp (int n, int type)` `n` – количество элементов в массиве, `type` – его тип. Функция согласует работу всей программы, генерируя и сортируя массивы, и выводя результаты в консоль.

Отладка программы, тестирование функций

Тестирование и отладка производились с помощью функций
`check_array(int64_t *arr, int n)` и `print_arr(int64_t *arr, int n)`.

0 1 2 3 4 5 6 7 8 9 - Исходный массив

0 1 2 3 4 5 6 7 8 9 - Shell sort

0 1 2 3 4 5 6 7 8 9 - Selection sort

9 8 7 6 5 4 3 2 1 0 - Исходный массив

0 1 2 3 4 5 6 7 8 9 - Shell sort

0 1 2 3 4 5 6 7 8 9 - Selection sort

Далее значения разделены на 10^6

Исходный:

753.2 -63.8 53.2 -798.2 -846.7 2.0 1357.8 1057.2 -1134.9 525.8

Результаты у обоих сортировок:

2.0 53.2 -63.8 525.8 753.2 -798.2 -846.7 1057.2 -1134.9 1357.8

Анализ допущенных ошибок

Для генерации чисел методом `rand()` не учитывалось, что такие числа не являются 64-битными, а помещаются в 32 бита, исправлено это было перемножением 5 таких получаемых чисел.

Список литературы

- [1] Кормен Т., Лейзерсон Ч., Ривест Р, Штайн К. Алгоритмы: построение и анализ. Второе издание. — М.:«Вильямс», 2005.
- [2] Д. Кнут. Искусство программирования. Том 3. Сортировка и поиск 2-е изд.