

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

ОТЧЕТ ПО ЗАДАНИЮ №1

«Методы сортировки»

Вариант 1 / 3 / 1 / 3

Выполнил:
студент 104 группы
Каратыщев Д. И.

Преподаватель:
Гуляев Д. А.

Москва
2022

Содержание

Постановка задачи	2
Результаты экспериментов	3
Структура программы и спецификация функций	4
Отладка программы, тестирование функций	6
Анализ допущенных ошибок	7
Список цитируемой литературы	8

Постановка задачи

Требуется реализовать два метода сортировки массива и провести их экспериментальное сравнение. Должны быть использованы сортировка «Пузырьком» и сортировка Шелла. Тип элементов массива – целые числа `int`. Элементы должны быть отсортированы в порядке неубывания модулей, то есть при сравнении не учитывается знак. Каждое экспериментальное сравнение должно показать число сравнений элементов и число их перемещений (обменов). Сравнение методов сортировки проводится на одних и тех же массивах, при этом используется разное число элементов, выделяемое динамически, – 10, 100, 1000, 10000. Генерация элементов – случайная. Всего следует рассмотреть 4 типа генерации – элементы уже упорядочены, элементы упорядочены в обратном порядке, расстановка элементов случайна (2 разных массива). Результаты экспериментов требуется оформить на основе нескольких запусков программы в виде таблицы, которая будет содержать информацию по каждому эксперименту для соответствующего числа элементов, в конце нужно вычислить среднее значение. Также требуется реализовать возможность запуска программы через командную строку, используя аргументы командной строки.

Результаты экспериментов

В данном разделе приведены результаты экспериментов, теоретические оценки и проведено их сравнение. Теоретические оценки количества сравнений и обменов взяты из литературы. Ссылки на неё в разделе 7.

n	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	45	45	45	45	45
	Перемещения	0	45	19	21	21
100	Сравнения	4950	4950	4950	4950	4950
	Перемещения	0	4950	2428	2281	2415
1000	Сравнения	499500	499500	499500	499500	499500
	Перемещения	443889	250777	252510	236794	236794
10000	Сравнения	49995000	49995000	49995000	49995000	49995000
	Перемещения	0	2712050	25175231	25099846	13246782

Таблица 1: Результаты работы сортировки Пузырьком

n	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	22	53	36	38	37
	Перемещения	0	45	19	21	21
100	Сравнения	503	5147	2807	2663	2780
	Перемещения	0	4950	2428	2281	2415
1000	Сравнения	8006	447614	256684	258587	242723
	Перемещения	443889	250777	252510	236794	236794
10000	Сравнения	120005	2812247	25265944	25190656	13347213
	Перемещения	0	2712050	25175231	25099846	13246782

Таблица 2: Результаты работы сортировки Шелла

Согласно теоретическим оценкам, среднее количество операций сортировки Шелла - $O(n^{7/6})$, в худшем случае оно составляет $O(n^{4/3})$, а в лучшем $O(n)$. Это справедливо при использовании формулы Седжвика для вычисления шагов. Для обычного вычисления (дихотомия) получаем: лучший и средний случаи - $O(n \log(n))$, худший случай - $O(n^2)$. Для сортировки "Пузырьком" сложность в среднем и сложном случаях составляет $O(n^2)$. В лучшем случае получаем $O(n)$. Практика показывает, что сортировка Шелла лучше сортировки Пузырьком по сравнениям. Это особенно заметно на больших данных.

Структура программы и спецификация функций

```
/**
 * Prints help documentation
 */
void print_help();

/**
 * Exchanges the values of two variables
 */
void swap_elems(int *a, int *b);

/**
 * Checks if two strings are equal
 * @return 0 if string is not equal to sample and 1 otherwise
 */
int is_equal(const char* string, const char* sample);

/**
 * Checks if the string is the help flag
 * @return 1 if string is the help flag and 0 otherwise
 */
int is_help(const char *string);

/**
 * Finds the help flag
 * @return 1 if the help flag is found and 0 otherwise
 */
int help_is_found(int argc, char **args);

/**
 * Converts a string into the integer type and sets the value
 * @return 1 if value is not a flag and 0 otherwise
 */
int set_str_value(int* number, char* value);

/**
 * Generates elements for the first mode
 */
void gen_el_1(int size, int* arr);

/**
 * Generates elements for the second mode
 */
void gen_el_2(int size, int* arr);
```

```

/**
 * Generates elements for the third and fourth modes
 */
void gen_el_34(int size, int* arr);

/**
 * Generates elements according to the mode
 */
void generate_elements(int size, int* arr, int mode);

/**
 * Bubble Sort
 */
void BubbleSort(int size, int* arr, int* comparisons, int* moves);

/**
 * Shell Sort
 */
void ShellSort(int size, int* arr, int* comparisons, int* moves);

/**
 * It's for copying one array to another
 */
void copy_arr(int size, int* arr, int* clear_arr);

/**
 * The main function
 * @return 1 if the program works correctly and 0 otherwise
 */
int main(int argc, char **args);

```

Отладка программы, тестирование функций

Тестирование сортировок проводилось с использованием аргументов командой строки, каждый из которых задавал начальное значение числу элементов массива, а также вариант заполнения элементов массива. Далее в программе подсчитывались число сравнений и обменов для каждой сортировки, что в дальнейшем выводилось на экран. Отладка сортировок проводилась на массивах различной длины, а именно: 10, 100 и 1000. Элементы создавались случайно (3 - 4 типа генерации). Использовался отладчик IDE, до этого исходный и впоследствии отсортированный массивы выводились на экран.

Анализ допущенных ошибок

Были допущены ошибки в подсчёте сравнений, так как не учитывались случаи выхода из цикла, а также в реализации флагов для командной строки - функция `getopt` не считает ошибкой два подряд идущих флага, где первый имеет свой числовой аргумент - этим числовым аргументом становится следующий флаг. Допущены и исправлены ошибки, связанные с генерацией элементов массива, уже упорядоченных в соответствующем порядке. Они связаны с реализацией данной генерации.

Список литературы

- [1] Кормен Т., Лейзерсон Ч., Ривест Р, Штайн К. Алгоритмы: построение и анализ. Второе издание. — М.: «Вильямс», 2005.
- [2] Кнут Д. Искусство программирования для ЭВМ. Том 3. — М.: Мир, 1978.