

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

ОТЧЕТ ПО ЗАДАНИЮ №1

«Методы сортировки»

Вариант 3 / 1 / 2 / 3

Выполнил:
студент 102 группы
Воробьев Е. Р.

Преподаватель:
Сенюкова О. В.

Москва
2020

Содержание

Постановка задачи	2
Результаты экспериментов	3
Сортировка методом простого выбора	3
Сортировка методом Шелла	4
Структура программы и спецификация функций	5
Отладка программы, тестирование функций	6
Тестирование сортировки методом простого выбора	6
Тестирование сортировки методом Шелла	7
Анализ допущенных ошибок	8
Список цитируемой литературы	9

Постановка задачи

В данном задании требуется:

- реализовать на языке Си два различных метода сортировки по неубыванию массива, состоящего из чисел двойной точности (типа `double`): методом простого выбора и методом Шелла в виде двух отдельных функций типа `void`,
- сравнить количество сравнений и перестановок элементов массива для каждой из этих сортировок (в отсортированном по неубыванию массиве, в отсортированном по невозрастанию массиве, в массиве из чисел, полученных случайным образом),
- проанализировать каждый из способов на эффективность,
- сделать выводы об их эффективности.

Результаты экспериментов

Сортировка методом простого выбора

Этот прием основан на следующих принципах:

1. Выбирается элемент с наименьшим ключом.
2. Он меняется местами с первым элементом.
3. Затем этот процесс повторяется с оставшимися $n - 1$ элементами, $n - 2$ элементами и т. д. до тех пор, пока не останется один самый большой элемент.

Число сравнений ключей (C), очевидно, не зависит от начального порядка ключей.

$$C = \frac{(n-1)n}{2}$$

Обмен элементов происходит только во внешнем цикле, следовательно, число обменов не превосходит $n - 1$ (худший случай). В лучшем случае (упорядоченный массив) количество обменов равно 0.

Алгоритм просматривает массив, сравнивая каждый элемент с только что обнаруженной минимальной величиной; если он меньше первого, то выполняется некоторое присваивание. Вероятность, что второй элемент окажется меньше первого равна $\frac{1}{2}$, с этой же вероятностью происходят присваивания минимуму. Вероятность, что третий элемент окажется меньше первых двух, равна $\frac{1}{3}$, а вероятность для четвертого оказаться наименьшим - $\frac{1}{4}$ и т. д. Поэтому полное среднее число пересылок равно H_{n-1} , где H_n - n -ое гармоническое число

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} \quad (1)$$

H_n можно представить как

$$H_n = \ln(n) + g + \frac{1}{2n} - \frac{1}{12n^2} + \dots \quad (2)$$

где $g = 0.577216\dots$ - константа Эйлера. Для достаточно больших n можно отбросить дробные члены и получить приближенное среднее число присваиваний на i -м проходе в виде

$$F_i = \ln(i) + g + 1 \quad (3)$$

Тогда среднее число пересылок M_{avg} в сортировке выбором равно сумме величин F_i с i , пробегающим от 1 до n :

$$M_{avg} = n(g + 1) + \sum_{i=1}^n \ln(i) \quad (4)$$

Аппроксимируя дискретную сумму интегралом

$$\int_1^n \ln(x) dx = n \ln(n) - n + 1 \quad (5)$$

получаем приближенное выражение

$$M_{avg} = n(\ln(n) + g) \quad (6)$$

n	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	45	45	45	45	45
	Перемещения	0	5	5	6	4
100	Сравнения	4950	4950	4950	4950	4950
	Перемещения	0	50	94	90	58.5
1000	Сравнения	49950	49950	49950	49950	49950
	Перемещения	0	500	992	989	620.25
10000	Сравнения	49995000	49995000	49995000	49995000	49995000
	Перемещения	0	5000	9989	9987	6244

Таблица 1: Результаты работы сортировки методом простого выбора

Сортировка методом Шелла

Метод, предложенный Дональдом Л. Шеллом, является неустойчивой сортировкой по месту.

В лучшем случае (на отсортированном массиве) условие внутри цикла не выполняется ни разу, значит в цикле будет всегда постоянное число итераций. В таком случае сложность фиксирована - $O(n \log(n))$.

В наихудшем случае можно оценить число сравнений/перемещений, как $n^2 + \frac{n^2}{2} + \frac{n^2}{4} + \frac{n^2}{8} + \dots \leq 2n^2 \in O(n^2)$.

Среднее время работы алгоритма зависит от длин промежутков — d , на которых будут находиться сортируемые элементы исходного массива ёмкостью N на каждом шаге алгоритма.

Первоначально используемая Шеллом последовательность длин промежутков: $d_1 = \frac{N}{2}, d_i = \frac{d_{i-1}}{2}, d_k = 1$ в худшем случае, сложность алгоритма составит $O(N^2)$;

n	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	22	35	27	31	28.75
	Перемещения	0	13	5	9	7.25
100	Сравнения	503	763	1005	920	797.75
	Перемещения	0	260	502	418	295
1000	Сравнения	8006	12706	15428	16331	13117.75
	Перемещения	0	4700	7422	8325	5111.75
10000	Сравнения	120005	182565	266921	278931	212105.5
	Перемещения	0	62560	146916	158926	92100.5

Таблица 2: Результаты работы сортировки методом Шелла

Структура программы и спецификация функций

- `void swap(double *a, double *b)`

Функция меняет местами значения двух переменных, ничего не возвращает.

- `void generate_random_array(int size, double *array)`

Данная функция заполняет получаемый массив `array` заданной величины `size` случайными числами типа `double`, ничего не возвращает.

- `void generate_ordered_array(int size, double *array)`

Данная функция заполняет получаемый массив `array` заданной величины `size` числами от $-\frac{size}{2}$ до $\frac{size}{2} - 1$, ничего не возвращает.

- `void generate_reversed_array(int size, double *array)`

Данная функция заполняет получаемый массив `array` заданной величины `size` числами от $\frac{size}{2}$ до $-(\frac{size}{2} + 1)$, ничего не возвращает.

- `void print_array(int size, double *array)`

Данная функция выводит получаемый массив `array` размером `size`, ничего не возвращает.

- `void selection_sort(int size, double *array)`

Данная функция сортирует получаемый на вход массив `array` величины `size` с помощью сортировки методом простого выбора, ничего не возвращает.

- `void shell_sort(int size, double *array)`

Данная функция принимает на вход массив `array` размером `size`. Функция реализует сортировку методом Шелла, в качестве 1-го шага взят предложенный самим Шеллом вариант $shift = \frac{N}{2}$, в качестве последующих шагов - предыдущий, поделенный целочисленно на 2, посредством битового сдвига. Функция ничего не возвращает.

Отладка программы, тестирование функций

Тестирование сортировки методом простого выбора

i	0	1	2	3	4	5	6	7	8	9
array[i](ввод)	-5	-4	-3	-2	-1	0	1	2	3	4
array[i](вывод)	-5	-4	-3	-2	-1	0	1	2	3	4

Таблица 3: Тест 1.1 (начальная последовательность отсортирована по неубыванию)

i	0	1	2	3	4	5	6	7	8	9
array[i](ввод)	4	3	2	1	0	-1	-2	-3	-4	-5
array[i](вывод)	-5	-4	-3	-2	-1	0	1	2	3	4

Таблица 4: Тест 1.2 (начальная последовательность отсортирована по невозростанию)

i	0	1	2	3	4	5	6	7	8	9
array[i](ввод)	-16	-26	12	-5	-11	-8	41	3	-29	45
array[i](вывод)	-29	-26	-16	-8	-5	3	11	12	41	45

Таблица 5: Тест 1.3 (начальная последовательность состоит из случайных чисел)

Тестирование сортировки методом Шелла

i	0	1	2	3	4	5	6	7	8	9
array[i](ввод)	-5	-4	-3	-2	-1	0	1	2	3	4
array[i](вывод)	-5	-4	-3	-2	-1	0	1	2	3	4

Таблица 6: Тест 2.1 (начальная последовательность отсортирована по неубыванию)

i	0	1	2	3	4	5	6	7	8	9
array[i](ввод)	4	3	2	1	0	-1	-2	-3	-4	-5
array[i](вывод)	-5	-4	-3	-2	-1	0	1	2	3	4

Таблица 7: Тест 2.2 (начальная последовательность отсортирована по невозрастанию)

i	0	1	2	3	4	5	6	7	8	9
array[i](ввод)	40	-44	14	-45	20	-49	-21	4	16	-42
array[i](вывод)	-49	-45	-44	-42	-21	4	14	16	20	40

Таблица 8: Тест 2.3 (начальная последовательность состоит из случайных чисел)

Анализ допущенных ошибок

Ошибок допущено не было.

Список литературы

- [1] Кормен Т., Лейзерсон Ч., Ривест Р, Штайн К. Алгоритмы: построение и анализ. Второе издание. — М.:«Вильямс», 2005.
- [2] Вирт Н. Алгоритмы и структуры данных. — М.: Мир, 1989
- [3] Д. Кнут. Искусство программирования. Том 3. Сортировка и поиск, 2-е изд. — М.:«Вильямс», 2017.