Московский государственный университет
имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики

Отчет по заданию №1

«Методы сортировки»

Вариант 3 / 4 / 1 / 3

Выполнил:
студент 102 группы
Шутков Г. А.

Преподаватель:
Кулагин А. В.

Москва
2020

Содержание

Постановка задачи

Требуется реализовать два метода сортировки: "пузырьком"и сортировку Шелла. Затем необходимо сравнить их эффективность на тестовом наборе данных и произвести теоретическую оценку сложности работы алгоритмов. Входными данными являются десятичные числа с плавающей точкой, а сортировка происходит по невозрастанию модулей элементов.

Результаты экспериментов

Сортировка пузырьком

n	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	45	45	45	45	45
	Перемещения	9	0	36	21	16.5
100	Сравнения	4095	4095	4095	4095	4095
	Перемещения	4095	0	2498	2599	2298
1000	Сравнения	499500	499500	499500	499500	499500
	Перемещения	499500	0	250170	245301	248742
10000	Сравнения	49995000	49995000	49995000	49995000	49995000
	Перемещения	49995000	0	25185157	25017928	25049521

Таблица 1: Результаты работы сортировки пузырьком

Оценка сложности алгоритма сортировки пузырьком

Данная сортировка совершает последовательные проходы от начала массива до его конца и производит поочередное сравнение соседних элементов. Элементы меняются местами, если их порядок не соответствует выбранному. Так как в алгоритме меняться местами могут только соседние элементы, то каждый обмен уменьшает количество инверсий на единицу. Следовательно, количество обменов равно количеству инверсий в исходном массиве вне зависимости от реализации сортировки. Максимальное количество инверсий содержится в массиве, элементы которого отсортированы в обратном порядке, их в нем $\frac{n(n-1)}{2}$. Получаем, что в худшем случае ассимптотика алгоритма составит $O(n^2)$. Лучшим случаем является отсортированный массив и в зависимости от реализиции ассимптотика составит $O(n)$ или $O(n^2)$. В среднем этот алгоритм сортирует произвольный массив за $O(n^2)$, поскольку в нем $\frac{n(n-1)}{4}$ инверсий.

Сортировка Шелла

n	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	45	9	28	13	24
	Перемещения	45	0	19	4	19.25
100	Сравнения	620	275	753	711	590
	Перемещения	430	0	511	471	353
1000	Сравнения	9217	5616	13229	13419	10370
	Перемещения	4478	0	7977	8175	5158
10000	Сравнения	132571	86021	197172	195560	152831
	Перемещения	55174	0	115024	113436	70909

Таблица 2: Результаты работы сортировки Шелла

Оценка сложности алгоритма сортировки Шелла

Сортировка Шелла является усовершенствованной версией сортировки вставками. На каждом проходе алгоритма по массиву выберается велечина L_i, описывающая расстояние между соседними элементами. На последнем проходе $L_0 = 1$. После определения L_i, массив разбивается на несколько списков, в каждом их которых расстоние между соседними элементами равно L_i. Списков ровно L_i штук. Далее каждый из получившихся списков сортируется при помощи сортировки вставками. В конуе итерации списки снова объединяются в один массив. Легко понять, что ассимптотик алгоитма зависит от способа выбора L_i на каждом шаге. В данной реализации используется формула, полученная Робертом Седжвиком:

$$L_i = \begin{cases} 9 * (2^i - 2^{\frac{i}{2}}) + 1 & \text{если i - четно} \\ 8 * 2^i - 6 * 2^{\frac{i+1}{2}} + 1 & \text{если i - нечетно} \end{cases}$$

При таком методе выбора шага, алгоритм работает в среднем за $O(n^{\frac{7}{6}})$, а в худшем случае не медленнее, чем за $O(n^{\frac{4}{3}})$.

Структура программы и спецификация функций

- int comp(const void *a, const void *b)

  Функция принимает на вход указатели на два сравниваемых элемента массива, преобразует их к типу double и возвращает:

$$\begin{cases} 1, |a| > |b| \\ -1, |a| < |b| \\ 0, |a| = |b| \end{cases}$$

- int *step_precnt(int n)

  Функция принимает на вход одно целое число типа int, формирует массив сдвигов для сортировки Шелла и возвращает указатель на его начало.

- void shell_sort(double *arr, int n)

  Функция принимает на вход массив чисел типа double и его размер, а затем сортирует его алгоритмом Шелла.

- void bubble_sort(double *arr, int n)

  Функция принимает на вход массив чисел типа double и его размер, а затем сортирует его алгоритмом "пузырек".

- void gen_arr(double *arr, int n, int k, int min, int max)

  Функция принимает указатель на генерируемый массив типа double, его длину, параметр, задающий упорядоченность элементов и границы, в которых лежат элементы этого массива.

Отладка программы, тестирование функций

Отладка программы производилась при помощи вывода на экран генерируемых и уже отсортированных массивов. Проверка корректности подсчета числа сравнений и обменов делалась вручную на данных маленького объема. Также производилось измерение времени работы сортировок для лучшего понимания их эффективности.

Анализ допущенных ошибок

В ходе написания программы была допущена критическая ошибка, вследствии которой сортировка Шелла работала за $O(n^2)$, но после проведения тестов с замерением времени и сопоставления выходных данных програмы она была устранена.

Список литературы

[1] Кормен Т., Лейзерсон Ч., Ривест Р, Штайн К. Алгоритмы: построение и анализ. Второе издание. — М.:«Вильямс», 2005.