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

Отчет по заданию №1

«Методы сортировки»

Вариант 1 / 4 / 2 / 4

Выполнила:
студентка 119 группы
Голубкова М. С.

Преподаватель:
Сковорода Н. А.

Москва
2021

Содержание

Постановка задачи

Задача заключалась в реализации двух методов сортировки массива чисел и проведении их экспериментального сравнения.

В данном варианте рассматриваются метод простого выбора и быстрая сортировка в рекурсивной реализации, элементами массива являются числа типа integer.

Массив заполняется случайными числами.

Числа упорядочиваются по невозрастанию модулей, т.е. при сравнении элементов не учитывается знак.

Эксперимент заключался в приведении количества сравнений и перемещений, производимых в сортировке уже упорядоченных массивов, массивов, упорядоченных в обратном порядке и массивов с случайным порядком элементов.

Результаты экспериментов

В книге "Алгоритмы. Введение в разработку и анализ"Левитина А. В. [1] для сортировки методом простого выбора приводятся такие теоретические оценки:

Число сравнений постоянно и вычисляется по формуле $\frac{n(n-1)}{2}$

Общее число операций (суммма числа сравнений и перемещений) - $O(n^2)$

Из таблицы видно, что число сравнений постоянно и соответствует теоритической оценке. Примерное количество операций сравнения и перемещения для каждого n можно вычислить по формуле $\frac{1}{2}n^2$, что также соответствует теореческой оценке с небольшой погрешностью.

n	**Параметр**	**Номер сгенерированного массива**				**Среднее значение**
		1	2	3	4	
10	Сравнения	45	45	45	45	45
	Перемещения	0	5	8	7	5
100	Сравнения	4950	4950	4950	4950	4950
	Перемещения	0	50	95	96	60,25
1000	Сравнения	499500	499500	499500	499500	499500
	Перемещения	0	500	994	989	620,75
10000	Сравнения	49995000	49995000	49995000	49995000	49995000
	Перемещения	0	5000	9991	9987	6244,5

Таблица 1: Результаты работы сортировки методом простого выбора

В книге "Алгоритмы и структуры данных."Вирта Н. [2] для быстрой сортировки приводятся такие теоретические оценки:

Общее число сравнений с погрешностью описывается формулой $n * log_2 n$

Общее число перемещений с погрешностью описывается формулой $\frac{n*log_2 n}{6}$

Общее число операций (суммма числа сравнений и перемещений) примерно $O(n * log_2 n)$

Приведённые формулы точнее всего работают для идеального случая, когда в качестве главного элемента выбирается медиана. Но на практике это довольно редкое явление, поэтому средние значения сравнений и перемещений в таблице несколько больше значений, вычесленных по формулам. Примерное количество операций сравнения и перемещения для каждого n можно вычислить по формуле $\frac{8*n*log_2 n}{6}$, что также соответствует теореческой оценке с небольшой погрешностью.

n	**Параметр**	**Номер сгенерированного массива**				**Среднее значение**
		1	2	3	4	
10	Сравнения	31	34	43	32	35
	Перемещения	6	11	13	12	10,5
100	Сравнения	606	610	971	780	741,75
	Перемещения	63	112	183	187	136,25
1000	Сравнения	9009	9016	12868	14049	11235,5
	Перемещения	511	1010	2600	2556	1669,25
10000	Сравнения	125439	125452	183194	183618	154425,75
	Перемещения	5904	10904	33770	33287	20966,25

Таблица 2: Результаты работы быстрой сортировки в рекурсивной реализации

Структура программы и спецификация функций

Полный список функций и описание их работы:

- cmp1

  Принимает 2 указателя на числа типа integer.

  Возвращает разницу абсолютных значений параметров.

  Тип возвращаемого значения - integer.

  Используется для вызова встроенной функции qsort, для сортировки в порядке невозрастания абсолютных значений эллементов сортируемого массива.

- cmp2

  Принимает 2 указателя на числа типа integer.

  Возвращает разницу абсолютных значений параметров.

  Тип возвращаемого значения - integer.

  Используется для вызова встроенной функции qsort, для сортировки в порядке неубывания абсолютных значений эллементов сортируемого массива.

- fill

  Принимает указатель на массив, количество элементов и параметр (номер столбца в таблицах).

  Заполняет массив случайными числами с помощью функции rand(), из возвращаемого ею значения вычитается половина её максимального возвращаемого значения для получения как положительных так и отрицательных чисел в массиве. Затем, в зависимости от параметра, массив сортируется с помощью встроенной функции qsort по возрастанию, убыванию или не сортируется вообще.

- sort2

  Реализация сортировки методом простого выбора.

  Принимает указатель на массив и количество его элементов.

  Функция сортирует массив в порядке неубывания элементов по модулю.

  Сначала ищется максимальный по абсолютному значению элемент в части массива, начинающейся с элемента на который указывает переменная bgn до последнего элемента, (неотсортированная часть массива). Затем этот максимальный элемент ставится в конец отсортированной части массива (левая часть исходного массива), то есть на своё место, а начало неотсортированной части массива сдвигается на один элемент вправо. Алгоритм повторяется до тех пор, пока значение переменной bgn не будет совпадать с индексом последнего элемента.

- sort4

  Быстрая сортировка в рекурсивной реализации.

  Принимает указатель на начало сортируемой части массива и количество элементов в ней.

  Функция сортирует массив в порядке неубывания элементов по модулю.

  Выбирается "главный"элемент, который до рекурсивного вызова функции должен будет встать на своё место в отсортированном массиве. Левый и правый указатели двигаются к "главному"элементу до тех пор, пока слева не найдётся элемент, меньший "главного а справа больший "главного затем эти элементы меняются местами. Указатели двигаются до тех пор, пока они не встретятся или не поменяются местами. В ходе работы цикла while "главный"элемент тоже может сдвинуться, к концу работы цикла он будет стоять на своём месте. Затем функция вызывается рекурсивно для левой и правой относительно "главного"элемента частей массива, если они существуют.

- main

  Главная функция, в которой начинается работа программы.

  3 массива выделяются динамически, первый заполняется функциеё fill, а затем копируется в оставшиеся 2 массива. Первый сортируется с помощью sort2, второй - sort4, а 3 служит для проверки корректности этих алгоритмов и сортируется встроенной функцией qsort. Для корректной работы программы нужно ввести 2 числа: количество элементов в массиве и параметр для заполнения массива определённым образом (соответствует номерам столбцов в вышеприведённых таблицах). Программы выводит 4 числа: количество сравнений и перемещенний при работе двух функций сортировки.

Отладка программы, тестирование функций

При тестировании функции fill сформированный массив с небольшим количеством элементов выводился на экран, позволяя проверить правильность заполнения по невозрастанию и неубыванию и наличие отрицательных элементов в нём.

При тестировании функций сортировки помимо вывода отсортированных массивов использовался массив Check. Этот массив сортируется с помощью функции qsort по невозрастанию абсолютных значений элементов. Затем модуль каждого элемента массива Chek сравнивался с модулем соответствующего элемента в отсортированных функциями, реализованными в программе, массивах, при несовпадении этих значений должно выводиться сообщение об ошибке, однако такого ни разу не произошло.

Анализ допущенных ошибок

Все ошибки при работе над данным домашним заданием были допущенны по невнимательности.

Список литературы

[1] Левитин А. В. Глава 3. Метод грубой силы: Сортировка выбором // Алгоритмы. Введение в разработку и анализ — М.: Вильямс, 2006

[2] Вирт Н. Алгоритмы и структуры данных. —М.: Мир, 1989.