МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени. М. В. ЛОМОНОСОВА
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКЕ И КИБЕРНЕТИКИ

ОТЧЕТ ПО ЗАДАНИЮ №1

Методы сортировки

Вариант 3 3 2 5

Исполнитель: студент 106 группы
Марков И.В.
Преподаватель:
Корухова Л.С.
19.02.2019

Постановка задачи

Реализовать два метода сортировки массива чисел: метод простого выбора и пирамидальная сортировка, и провести их экспериментальное сравнение. Тип элементов массива - числа с плавающей точкой. Упорядочивание происходит по неубыванию модулей. При реализации каждого метода требуется вычислить число сравнений элементов и число их перемещений. Сравнение методов сортировки необходимо проводить на одних и тех же исходных массивах, при этом следует рассмотреть массивы разной длины. Память под массив выделяется динамически. Провести анализ работы алгоритмов при различной длине массива n = 10, 100, 1000, 10000. 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, 4) Расстановка элементов случайна

Результаты экспериментов оформить на основе нескольких запусков программы в виде сводной таблицы, содержащей информацию о проведенном экспериментальном анализе. Непосредственно таблица будет представлена ниже при описании результатов эксперимента.

Также требуется:

- предоставить список всех функций, используемых при решении поставленной задачи, и описать принцип их работы

-сообщить о методах проведения отладки численных методов и всей программы

-провести анализ допущенных ошибок

Результаты экспериментов

В результате проведенных экспериментов была подтверждена асимптотическая оценка алгоритмов. Оба алгоритма требуют O(1) дополнительной памяти, так как выделяемая программой память не зависит от входных данных, кроме, разумеется, выделения памяти для хранения самого массива. Экспериментально убедились в том, что пирамидальная сортировка в худшем случае работает за O(n * log n)[2], сортировка методом простого выбора как в лучшем, так и худшем случае работает за O(n * n)[1].

Также следует заметить, что сортировка методом простого выбора при достаточно малой длине массива n выигрывает по времени у пирамидальной сортировки за счет сложности алгоритма пирамидальной сортировки. Однако при больших n пирамидальная сортировка работает быстрее.

Сортировка методом простого выбора

n	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	45				45
	Перемещения	0	5	8	8	6
100	Сравнения	4950				4950
	Перемещения	0	50	93	97	61
1000	Сравнения	499500				499500
	Перемещения	0	500	991	997	622
10000	Сравнения	49995000				49995000
	Перемещения	0	5000	9989	9994	6246

Пирамидальная сортировка

n	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	23	19	20	23	22
	Перемещения	30	21	30	25	26
100	Сравнения	547	478	519	523	517
	Перемещения	640	516	571	592	580
1000	Сравнения	8813	7991	8425	8431	8415
	Перемещения	9708	8316	9082	9075	9045
10000	Сравнения	122288	113360	117717	117682	117762
	Перемещения	131956	116696	124162	124155	124242

Структура программы и спецификации функций

-void out(int n) и void out2(int n) используются для отладки программы для вывода массива на экран

-bool comp(double a, double b) является компоратором для алгоритмов сортировки

-void generate_straight (int sz) генерирует массив из первых sz натуральных чисел, расположенных по возрастанию

-void generate_reversed (int sz) генерирует массив из первых sz натуральных чисел, расположенных по убыванию

-void generate_random (int sz) генерирует массив из sz случайных чисел типа double

-void simple_choice (double *mas, int sz) представляет собой реализацию сортировки методом простого выбора

-void sift (int num, int n) реализует “просеивание” элемента под номером num в массиве из n элементов

-void heap_sort (double *mas, int n) является реализацией пирамидальной сортировки. Использует внутри себя описанную выше функцию sift

-int main(void) производит считывание размера массива, с которым впоследствии будет работать программа; а также реализует работу программы при помощи приведенных выше функций в виде генерации 4 массивов для каждого из алгоритмов, применения к ним методов сортировки и, при надобности, вывода результатов работы программы на экран

Отладка программы, тестирование функций

Отладка производилась поочередно для каждой функции.Отдельного описания требует отладка функций, реализующих алгоритмы сортировки. Первичная отладка использовала лишь вывод исходного и отсортированного массивов на экран, правильность работы алгоритмов проверялась самостоятельно. Далее, при прохождении данного этапа, проводилась конечная отладка функций: результаты их работы сравнивались с результатом работы встроенной функции qsort.

Правильность подсчетов количества сравнений и перемещений проверялась вручную на малых размерах массива

Анализ допущенных ошибок

1) При реализации пирамидальной сортировки была допущена следующая ошибка: программа была ориентирована на массив, индексация элементов в котором начинается с нуля. В последствии ошибка была исправлена путем переиндексации элементов массива нужным образом.
2) Добавлен компоратор для сравнения чисел в программе
3) Исправлена ошибка в алгоритме методом простой сортировки, при которой на очередном шаге цикла элемент массива менялся с самим собой. Соответствующие изменения внесены в таблицу
4) Приведены методы оценки сложности алгоритмов (указаны ссылки на литературу)

Литература

1. Кнут Д. Искусство программирования для ЭВМ. Том 3. — М.: Мир, 1978. стр. 169-171

2. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. Второе издание. — М.: «Вильямс», 2005. стр. 178-189