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

Отчет по заданию №1

«Методы сортировки»

Вариант 2 / 2 / 1 / 5

Выполнил:
студент 102 группы
Никитин В. В.

Преподаватель:
Смирнов А. В.

Москва
2017

Содержание

Постановка задачи

Необходимо реализовать два метода сортировки массива чисел и провести их экспериментальное сравнение. Для каждого из реализуемых методов необходимо предусмотреть возможность работы с массивами длины от 1 до N (N $\geq$ 1). При реализации каждого метода вычислить число сравнений элементов и число перемещений (обменов) элементов.

Сравнение методов сортировки необходимо проводить на одних и тех же исходных массивах, при этом следует рассмотреть массивы разной длины. Для вариантов с фиксированным значением N рассмотреть, как минимум, n = 10, 20, 50, 100. Для вариантов с динамическим выделением памяти — 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:

- элементы уже упорядочены (1);
- элементы упорядочены в обратном порядке (2);
- расстановка элементов случайна (3, 4).

Результаты экспериментов оформить на основе нескольких запусков программы в виде таблицы.

Тип данных: 64-битные целые числа (long long int)
Вид сортировки: числа упорядочиваются по невозрастанию
Методы сортировки: метод «пузырька», пирамидальная сортировка.

Более подробное описание методов сортировки можно прочесть в программе, либо в разделе «Структура программы и спецификация функций»

Результаты экспериментов

Элементы массива с номером 1 уже упорядочены (лучший случай), с номером 2 расположены в обратном порядке (худший случай), с номерами 3-4 расположены в случайном порядке.

n	**Параметр**	**Номер сгенерированного массива**				**Среднее значение**
		1	2	3	4	
10	Сравнения	45	45	45	45	45
	Перемещения	0	45	13	29	22
100	Сравнения	4990	4990	4990	4990	4990
	Перемещения	0	4990	2676	2551	2555
1000	Сравнения	499500	499500	499500	499500	499500
	Перемещения	0	499500	265677	245980	254038
10000	Сравнения	49995000	49995000	49995000	49995000	49995000
	Перемещения	0	49995000	25076575	25485962	25265508

Таблица 1: Результаты работы метода «пузырька»

n	**Параметр**	**Номер сгенерированного массива**				**Среднее значение**
		1	2	3	4	
10	Сравнения	60	52	58	62	58
	Перемещения	38	29	33	34	34
100	Сравнения	1288	1134	1232	1228	1221
	Перемещения	738	598	676	677	672
1000	Сравнения	19660	17968	18786	18820	18809
	Перемещения	10032	10710	9312	10030	10077
10000	Сравнения	264526	246744	255330	255386	255497
	Перемещения	141424	126976	134258	134158	134204

Таблица 2: Результаты работы пирамидальной сортировки

Стоить заметить, что метод «пузырька» сравнивает значения элементов массива $(n-1)*n/2$ раз, количество перемещений элементов в лучшем случае равно 0, в худшем равно $(n-1)*n/2$. Сложность алгоритма $O(n^2)$.

Совершенно иные данные выдает нам метод пирамидальной сортировки. Функция RightPyramid просеивает элемент через дерево, сложность $O(log(n))$, повторяется данное действие $O(n)$ раз. Сложность алгоритма $O(n\ log(n))$.

Исходя из тестов можно сказать, что пирамидальная сортировка однозначно лидирует на больших наборах данных. Метод "пузырька"имеет преимущество только на полностью отсортированном наборе, что на практике встречается довольно редко.

Структура программы и спецификация функций

int main (void);
Специальная функция. Является начальной точкой выполнения программы.

void CreateArray(int param, int n);
Функция, которая создает статический массив с элементами, порядок которых зависит от параметра (подробнее о параметре на стр. 2).
Входные данные: 'n' - количество элеметов в массиве, 'param' - параметр.

void BubbleSort(long long int a[], int n);
Функция, которая сортирует массив методом "пузырька". Упрощенно: сравнивает элементы a[i], a[i+1] и, если a[i]<a[i+1], то меняет их местами.
Входные данные: 'a' - массив, 'n' - количество элементов массива

void PyramidSort(long long int a[], int n);
Функция, которая сортирует массив с помощью метода пирамидальной сортировки. Создает правильную пирамиду, удаляет высоту, переносит на место высоты последний элемент, заново создает правильную пирамиду.
Входные данные: 'a' - массив, 'n' - количество элементов массива

void RightPyramid(long long int a[], int k, int m);
Функция, которая создает правильную пирамиду. Просеиваем a[k] элемент через пирамиду.
Входные данные: 'a' - массив, 'k' - последний элемент пирамиды, 'n' - первый элемент пирамиды

Отладка программы, тестирование функций

В процессе отладки были устранены недочеты (пропущенные знаки, табуляция) и неиспользуемые переменные, добавлены комментарии к коду.
Тестирование программы проводилось на 4 массивах: {0}, {10000, 9999, 9998, 9997}, {0,1,2,4}, {-2126464088, 737385984, -1010079636}

Анализ допущенных ошибок

Ошибок не было допущено.

Список литературы

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

[2] Белеванцев А.А. Конспект лекций по программированию для студентов 1 курса ВМК МГУ, 2016-2017.