

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

ОТЧЕТ ПО ЗАДАНИЮ №1

«Методы сортировки»

Вариант 2 / 3 / 1 / 5

Выполнил:
студент 119 группы
Трубецкой С.А.

Преподаватель:
Сковорода Н. А.

Москва
2020

Содержание

Постановка задачи	2
Результаты экспериментов	3
Структура программы и спецификация функций	5
Отладка программы, тестирование функций	6
Анализ допущенных ошибок	7
В процессе написания программы сложности возникли только с генерацией уже отсортированных массивов и массивов, отсортированных в обратном порядке. Функции 2 и 3 работали неправильно из-за того, что был неправильно выбран диапазон значений и составлена формула для генерации псевдорандомных чисел.	7
Список литературы	8

Постановка задачи

В данной работе необходимо было реализовать два метода сортировки массива чисел и провести их экспериментальное сравнение. В моем варианте элементы массива должны были быть 64-разрядными целыми числами (long long int). В программе необходимо было рассмотреть массивы различных размеров: 10, 100, 1000, 10000 элементов. Конкретно в моем варианте необходимо было сравнить метод сортировки «пузырек» и пирамидальную сортировку, причем числа должны быть упорядочены по неубыванию модулей, т.е. при сравнении элементов не учитывается знак. Преимущество одного из методов устанавливается непосредственным вычислением числа сравнений элементов и числа перемещений (обменов) элементов при реализации каждого метода сортировки.

Отдельными функциями в программе должны быть реализованы:

- генерация исходных массивов;
- каждый из предложенных методов сортировки.

Результаты экспериментов

В данном разделе приведены результаты экспериментов, теоретические оценки методов и их сравнение. Каждый метод сортировки был применен к четырем массивам каждого размера – 10, 100, 1000, 10000 элементов. В первом массиве элементы уже были упорядочены нужным образом, во втором – упорядочены в обратном порядке, в третьем и четвертом расположение элементов было случайным. В Таблице 1 приведены результаты работы сортировки по методу «пузырек», в Таблице 2 – по методу пирамидальной сортировки.

N	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	45	45	45	45	45
	Перемещения	0	45	12	19	19
100	Сравнения	4950	4950	4950	4950	4950
	Перемещения	0	4238	2651	2224	2278,25
1000	Сравнения	499500	499500	499500	499500	499500
	Перемещения	0	60046	252166	243223	138858,75
10000	Сравнения	49995000	49995000	49995000	49995000	49995000
	Перемещения	0	568333	24775649	24884741	12557180,75

Таблица 1: Результаты работы сортировки методом «пузырек».

В сортировке методом «пузырек» число сравнений равно $N = (n - 1) \frac{n}{2}$, где n - кол-во элементов массива ; сложность - $O(n^2)$. Результаты, полученные эмпирическим путем, получились сравнимыми с теоретической оценкой.

N	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	41	35	40	39	38,75
	Перемещения	30	21	29	27	26,75
100	Сравнения	1029	703	1041	1015	947
	Перемещения	601	318	596	565	520
1000	Сравнения	6036	3440	16849	16819	10786
	Перемещения	2785	1222	9100	9053	5540
10000	Сравнения	36812	30330	235614	235351	134526,75
	Перемещения	13833	10167	124365	124121	68121,5

Таблица 2: Результаты работы пирамидальной сортировки.

Оценка сложности пирамидальной сортировки приведена в [1] и определяется в О-нотации: $O(n\log(n))$. Результаты, полученные эмпирическим путем, показывают, что на разных выборках они разнятся, однако средние значения близки к теоретической оценке.

Отсюда можно сделать вывод, что пирамидальная сортировка будет гораздо эффективнее, чем сортировка методом «пузырек». Для достаточно больших значений N, кол-ва элементов массива, количество сравнений в сортировке «пузырьком» превышает кол-во сравнение в пирамидальной сортировке примерно в 370 раз, а кол-во перемещений - почти в 200 раз.

Структура программы и спецификация функций

В данном разделе я приведу полный список функций, используемых в моей программе.

Следующие три функции использовались для генерации массивов различных типов: Функция 1 – для генерации отсортированного массива; Функция 2 – массива с элементами, расположенными в обратном порядке; Функция 3 – с элементами, расположенными случайным образом. На вход каждой из этих функций подается указатель на выделенное в памяти место и размер генерируемого массива.

```
void funcright(long long int* a, int n)
```

Функция 1.

```
void funcback(long long int* a, int n)
```

Функция 2.

```
void funcrandom(long long int* a, int n)
```

Функция 3.

Функция 4 осуществляет сортировку методом «пузырек». На вход функции также подается указатель на выделенное в памяти место и размер генерируемого массива. Переменные `str` и `mov`, объявленные вне функций, являются счетчиками соответственно сравнений и перемещений элементов.

```
void bubble(long long int* a, int n)
```

Функция 4.

Прежде чем отсортировать массив, требуется построить из его элементов бинарное дерево, называемое кучей или пирамидой, что и выполняет функция 5. Помимо указателя на выделенное в памяти место и размера генерируемого массива функции также подается индекс элемента, который ей нужно «просеять» вниз кучи.

```
void heapify(long long int* a, int n, int i)
```

Функция 5.

Функция 6 вызывает внутри себя функцию 5, чтобы из исходного массива построить кучу. Затем на каждом шаге функция 6 меняет местами корень(как максимальный по модулю элемент) и последний элемент кучи местами и «выкидывает» последний из кучи. Далее применяет функцию 5, но уже для кучи из $n - 1$ элемента(на каждом i -ом шаге функция применяется к $n - i$ элементам кучи). Процесс повторяется, пока мы не получим кучу, отсортированную нужным образом.

```
void pyramid(long long int* a, int n)
```

Функция 6.

Отладка программы, тестирование функций

Тестирование и отладка методов сортировки производились в несколько этапов:

1. Сначала функции генерации массивов (функции 1, 2, 3) тестировались на маленьких значениях N - число элементов массива. Сгенерированные массивы выводились на экран.

2. Потом проверялась правильность работы функций сортировки (4, 5, 6) на сгенерированных на предыдущем этапе массивах. Полученные массивы также выводились на экран. Позже эта часть кода была убрана в комментарий.

3. Все проблемы и неточности были выявлены уже на первых двух этапах. Далее были проведены тесты для $N = 100, 1000, 10000$. При таких значениях правильность сортировки определялась функцией `qsort`, элементы массива на экран не выводились.

Анализ допущенных ошибок

В процессе написания программы сложности возникли только с генерацией уже отсортированных массивов и массивов, отсортированных в обратном порядке. Функции 2 и 3 работали неправильно из-за того, что был неправильно выбран диапазон значений и составлена формула для генерации псевдорандомных чисел.

Список литературы

- [1] Кнут Д. Искусство программирования для ЭВМ. Том3 - Москва.:Мир, 1978.