

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

ОТЧЁТ ПО ЗАДАНИЮ №1

«Методы сортировки»

Вариант 3 / 1 / 1 / 5

Выполнил:
студент 105 группы
Космынин И. Н.

Преподаватели:
Гуляев А. В.
Русол А. В.

Москва
2024

Содержание

1	Постановка задачи	2
2	Результаты экспериментов	3
3	Структура программы и спецификация функций	4
4	Отладка программы, тестирование функций	6
5	Анализ допущенных ошибок	7
	Список цитируемой литературы	8

1 Постановка задачи

Реализовать пузырьковую и пирамидальную сортировки по неубыванию, экспериментально сравнить их эффективность. Измерение эффективности каждого метода сортировки ведётся по числу перемещений и числу сравнений. Для сортировки используются динамические массивы вещественных чисел двойной точности (double) различной длины, элементы которых расположены в порядке неубывания, невозрастания и в случайном порядке.

2 Результаты экспериментов

Согласно теоретическим предположениям[1], время работы сортировки пузырьком равно $\mathcal{O}(n^2)$, а у пирамидальной сортировки — $\mathcal{O}(n \log_2 n)$. Таким образом, в случае пузырьковой сортировки число перемещений и сравнений элементов на больших наборах данных должны сильно превышать соответствующие параметры пирамидальной сортировки. Экспериментальные данные (при сиде 42), подтверждающие теоретические предположения, указаны в следующих таблицах:

n	Параметр	Номер сгенерированного массива				Среднее Значение
		1	2	3	4	
10	Сравнения	45	45	45	45	45
	Перемещения	0	45	21	31	24,25
100	Сравнения	4950	4950	4950	4950	4950
	Перемещения	0	4950	2405	2461	2454
1000	Сравнения	499500	499500	499500	499500	499500
	Перемещения	0	499500	250558	251849	250476,75
10000	Сравнения	49995000	49995000	49995000	49995000	49995000
	Перемещения	0	49995000	24977250	25031015	81325816,25

Таблица 1: Результаты работы пузырьковой сортировки

n	Параметр	Номер сгенерированного массива				Среднее Значение
		1	2	3	4	
10	Сравнения	70	52	64	64	62,5
	Перемещения	30	21	27	27	26,25
100	Сравнения	1380	1132	1278	1290	1270
	Перемещения	640	516	589	595	585
1000	Сравнения	20416	17632	19216	19226	19122,5
	Перемещения	9708	8316	9108	9113	9061,25
10000	Сравнения	273912	243392	258294	258118	258429
	Перемещения	131956	116696	124147	124059	124214,5

Таблица 2: Результаты работы пирамидальной сортировки

Таблицы показывают, что при малых значениях N пирамидальная сортировка не даёт никаких преимуществ, но при росте N число сравнений и перемещений растёт не так стремительно, то есть асимптотически она быстрее.

3 Структура программы и спецификация функций

При работе программы используются две глобальные переменные `comps` и `swaps` типа `long long`, необходимые для подсчёта количества сравнений и перемещений для каждой сортировки.

Функция `main` сначала анализирует аргументы, с которыми была запущена программа. При необходимости указывается верный вариант использования программы и выводится справка. Если пользователь запросил полную таблицу, вызывается функция `print_table`, если пользователь запросил лишь замеры быстродействия сортировок на определенном наборе данных, функция `main` вызывает `generate_arr` для генерации необходимых массивов, к ним применяются сортировки, а результаты печатаются в консоль.

Полный список используемых функций:

- `void swap(double *a, double *b)`
меняет значения переменных `a` и `b` между собой;
- `void bubble_sort(int n, double *arr)`
применяет алгоритм сортировки пузырьком к массиву `arr` длины `n` и считает кол-во сравнений и перемещений;
- `void heapify(int n, int i, double *arr)`
выполняет одну процедуру просеивания кучи `arr` длины `n` начиная с эл-та под индексом `i`, при этом считает число сравнений и перемещений;
- `void heap_sort(int n, double *arr)`
применяет алгоритм пирамидальной сортировки к массиву `arr` длины `n` и считает кол-во сравнений и перемещений;
- `double* generate_arr(int n, int type)`
создаёт и возвращает массивы эл-ов `double` длины `n`, причём изначальный порядок эл-ов зависит от параметра `type`:
 1. массив упорядочен по возрастанию,
 2. массив упорядочен по убыванию,
 3. элементы расположены в случайном порядке;
- `void print_table()`
вычисляет и выводит в таблице число сравнений и перемещений при пузырьковой и пирамидальной сортировках на разных наборах данных;
- `void print_usage(char *arg)`
выводит справку о вариантах использования программы, параметр `arg` отвечает за название исполняемого файла;

- `void print_help(char *arg)`

выводит полную справку по программе, параметр `arg` отвечает за название исполняемого файла.

4 Отладка программы, тестирование функций

Для проверки корректности реализации алгоритмов сортировки сравнивались несколько заранее подготовленных массивов до и после сортировки.

При тестировании функции генерации массивов особое внимание уделялось массивам со случайным порядком элементов. Необходимо было добиться генерации различных массивов при нескольких запусках программы.

Большое количество времени ушло на написание и отладку функции для вывода полной таблицы, т.к. размер ячеек и форматирование вывода подбирались опытным путём.

5 Анализ допущенных ошибок

Первоначально были допущены ошибки при обработке аргументов программы, из-за чего она могла запускаться при неверных параметрах. К примеру, можно было одновременно указать параметры `-t` и `-bs`, что приводило к неадекватной работе приложения.

Вторая проблема была связана с тем, что я забыл задать сид для функции генерации псевдослучайных чисел. Из-за этого при каждом запуске программы массивы, которые должны были состоять из случайных элементов, создавались одни и те же.

Список литературы

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