

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

ОТЧЕТ ПО ЗАДАНИЮ №1
«Методы сортировки»
Вариант 2 / 4 / 1 / 5

Выполнил:
студент 104 группы
Лозинский И. П.

Преподаватель:
Гуляев Д. А.

Москва
2017

Содержание

Постановка задачи	2
Результаты экспериментов	3
Сортировка методом «пузырька»: [2]	3
Пирамидальная сортировка: [1]	4
Структура программы и спецификация функций	5
Отладка программы, тестирование функций	6
Анализ допущенных ошибок	7
Список цитируемой литературы	8

Постановка задачи

- Реализовать метод «пузырька» сортировки массива чисел
- Реализовать пирамидальную сортировку массива чисел

Провести экспериментальное сравнение эффективности методов при условии, что:

- Массив содержит элементы типа «long long int»
- Числа сортируются по невозрастанию их модуля

Анализ эффективности производится путём подсчёта операций сравнения элементов и операций перестановки элементов.

Для сравнения использовать динамические массивы длиной 10, 100, 1000 и 10 000 элементов.

Для каждого количества элементов сгенерировать:

- Массив отсортированный по возрастанию
- Массив отсортированный по убыванию
- 2 массива отсортированных случайным образом

Результаты экспериментов

Сортировка методом «пузырька»: [2]

Теоретические оценки числа сравнений:

- Наилучший случай: $N - 1$
- Наихудший случай: $(N - 1)\frac{N}{2}$

Теоретические оценки числа обменов:

- Наилучший случай: 0
- Наихудший случай: $(N - 1)\frac{N}{2}$

Практические результаты представлены в таблице:

n	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	90	9	72	81	63
	Перемещения	45	0	23	29	24
100	Сравнения	9900	99	8514	8316	6707
	Перемещения	4950	0	2541	2674	2541
1000	Сравнения	999000	999	949050	964035	728271
	Перемещения	499500	0	248447	243591	247884
10000	Сравнения	99990000	9999	99110088	99760023	74717527
	Перемещения	49995000	0	25123967	25068395	25046840

Таблица 1: Результаты работы сортировки методом «пузырька»

Пирамида́льная сортировка: [1]

Теоретические оценки числа сравнений:

- Наилучший случай: $N \log N$
- Наихудший случай: $N \log N$

Теоретические оценки числа обменов:

- Наилучший случай: $\frac{N}{2} \log N$
- Наихудший случай: $N \log N$

Практические результаты представлены в таблице:

n	Параметр	Номер сгенерированного массива				Среднее значение
		1	2	3	4	
10	Сравнения	35	41	34	37	36
	Перемещения	21	30	25	28	26
100	Сравнения	944	1081	1020	1037	1020
	Перемещения	516	640	572	590	579
1000	Сравнения	15965	17583	16897	16794	16809
	Перемещения	8316	9708	9097	9045	9041
10000	Сравнения	226682	244460	235373	235358	235468
	Перемещения	116696	131956	124162	124191	124251

Таблица 2: Результаты работы пирамидальной сортировки

Структура программы и спецификация функций

- `void genArray(size_t n, ll *arr, int order)` – генерирует массив из `n` элементов в порядке `order` и сохраняет его по указателю `arr`.
`order` принимает следующие значения:
 - `-1` – массив отсортирован по убыванию
 - `1` – массив отсортирован по возрастанию
 - `0` – массив отсортирован по случайно
- `void swap(ll *a, ll *b)` – меняет местами значения 2х указателей `a` и `b` типа *long long int* и увеличивает значение счётчика `swaps`
- `bool isLess(ll a, ll b)` – сравнивает модули аргументов и увеличивает счётчик `compares`. Возвращает `True` если $|a| < |b|$.
- `void reset(void)` – сбрасывает значения счётчиков `compares` и `swaps` .
- `void bubbleSort(size_t n, ll *arr)` – реализует сортировку методом «пузырька»
- `void heapify(size_t i, size_t n, ll *arr)` – восстанавливает свойство кучи для элемента с индексом `i` массива `arr` из `n` элементов.
- `void heapSort(size_t n, ll *arr)` – реализует пирамидальную сортировку
- `void printarr(size_t n, ll *arr)` – отладочный метод для распечатки массива
- `void compareSorts(int type, size_t n, unsigned int *resultbc, unsigned int *resulthc, unsigned int *resultbs, unsigned int *resulths)` – проводит тестирование сортировок на массивах типа `j` (см. параметр `order` в функции `genArray`) и длины `n`. Результаты записывает по переданным указателям:
 - `resultbc` – количество сравнений в методе «пузырька»
 - `resulthc` – количество сравнений в пирамидальной сортировке
 - `resultbs` – количество перестановок элементов в методе «пузырька»
 - `resulths` – количество перестановок элементов в пирамидальной сортировке
- `bool test()` – запускает 3 раза методы сортировки на случайных массивах длины 1000, если в результате какой-либо из массивов оказался не отсортирован – возвращает `False` , иначе `True`.
- `int main(int argc, char *argv[])` – при запуске без аргументов выводит читаемый текст с результатами, иначе текст для вставки в TeX отчёт.

Отладка программы, тестирование функций

Тестирование функций сортировки производится при помощи `test()`.
Отладочный вывод массивов производится при помощи функции `printarr()`.

Анализ допущенных ошибок

Полученные практические результаты подтверждают теоретические оценки. Однако для более корректной оценки среднего числа сравнений/перемещений элементов следует провести больше тестов на случайных входных данных. Возможна погрешность связанная с выбранным способом генерации случайных массивов, т.к. он далеко не идеален.

Список литературы

- [1] Кормен Т., Лейзерсон Ч., Ривест Р, Штайн К. Алгоритмы: построение и анализ. Второе издание. — М.:«Вильямс», 2005.
- [2] Лорин Г. Сортировка и системы сортировки. — М.: Наука, 1983.