МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Национальный исследовательский
Нижегородский государственный университет им. Н.И. Лобачевского»
(ННГУ)**

Институт информационных технологий, математики и механики

Направление подготовки: «Прикладная математика и информатика»

ОТЧЕТ

по лабораторной работе №5

Тема:

«Разные способы сортировки массивов»

Выполнил:
студент группы 3824Б1ПМ4
Пышкина А.С.

_____
подпись

Преподаватель:
Преподаватель по предмету
«Языки и методы программирования»
А.Е. Куклин

_____
подпись

Нижний Новгород
2024

Содержание:

Введение

Сортировка представляет собой одну из ключевых задач в сфере алгоритмов и структур данных. Этот процесс включает в себя организование элементов в массиве или списке согласно определённому критерию, например, в порядке возрастания или убывания. Эффективные алгоритмы сортировки играют значимую роль в программировании, так как они находят применение в различных областях, включая базы данных, поисковые системы и системы обработки информации.

Я рассматриваю три изученных метода сортировки:
1. Сортировка «пузырьком» (BubbleSort)
2. Сортировка «выбором» (SelectionSort)
3. Сортировка «вставками» (InsertionSort)

Постановка задачи

Задача: разработать программу, которая сортирует заранее созданный массив с использованием трех различных алгоритмов: пузырьковой сортировки, сортировки выбором и сортировки вставками. После выполнения сортировки программа должна предоставить информацию о времени сортировки массива. Целью является определить, какой из методов сортировки наиболее эффективен в зависимости от размера обрабатываемого массива.

Описание алгоритмов

1. Сортировка пузырьком (BubbleSort)
   Пузырьковая сортировка - это простой алгоритм сортировки, который повторно сканирует массив, сравнивает соседние элементы изменяет их, если они расположены в неправильном порядке.
   В случае неправильного порядка элементов сортировка происходит наоборот. Тогда можно сказать, что массив отсортирован.

   Порядок действий:
   1) Берем первый элемент массива и сравниваем его со следующим.
   2) Если наш элемент больше, то меняем их местами.
   3) Делаем так же и с последующими элементами.

2. Сортировка выбором (SelectionSort)
   Сортировка выбором – это алгоритм сортировки массива путем нахождения наименьшего элемента из неотсортированной части массива и перемещения его в начало отсортированной части. Он находит элемент

в неотсортированной части массива и перемещает его в начало отсортированной части. Этот процесс повторяется для всех элементов массива.

Порядок действий:
1) Разделите массив на отсортированную и неотсортированную части.
2) На каждой итерации находите наименьший элемент неотсортированной части.
3) Замените найденный элемент первым элементом неотсортированной части.
4) Увеличьте границу отсортированной части на один элемент и повторите этот процесс.

3. Сортировка вставкой (InsertionSort)

Сортировка вставкой - это алгоритм, который строит отсортированный массив, вставляя новый элемент в правильную позицию для уже отсортированного элемента. Этот алгоритм особенно эффективен для небольших массивов и частично отсортированных данных.

Порядок действий:
1) Начинаем с первого элемента, который считается отсортированным.
2) Берем следующий элемент и сравниваем его с отсортированной частью.
3) Вставляем элемент в правильное положение, сдвигая все большие элементы вправо.
4) Повторяем процесс для всех элементов массива.

Описание программной реализации

В этой программе реализовано три метода сортировки: пузырьковая сортировка, сортировка выбором и сортировка вставками. Пользователь имеет возможность выбрать один из этих методов для упорядочивания массива случайных целых чисел. Далее приведено подробное описание каждой составляющей программы.

1. *Подключение библиотек и файлов*

- stdio.h: Библиотека для ввода и вывода данных.
- stdlib.h: Библиотека для работы с памятью и генерации случайных чисел.
- time.h: Библиотека для работы с временем, используется для измерения времени выполнения
сортировок.
- iostream: Библиотека предоставляет средства ввода-вывода для стандартной консоли.

- function.h: файл, содержащий все функции.

2. *Алгоритмы сортировки*

- Сортировка пузырьком (BubbleSort):

Проходит по массиву и сравнивает соседние элементы, меняя их местами, если они находятся в неправильном порядке. Процесс повторяется до тех пор, пока не будет выполнен полный проход без изменений.

- Сортировка выбором (SelectionSort):

На каждой итерации находит наименьший элемент в неотсортированной части массива и перемещает его в начало отсортированной части.

- Сортировка вставками (Insertion_Sort):

Строит отсортированный массив, вставляя каждый элемент в правильное положение относительно уже отсортированных элементов.

Каждый из алгоритмов реализован в отдельной функции, принимающей массив и его размер в качестве аргументов.

3.*Генерация*

Для генерации случайных чисел используется функция rand(), инициализированная с помощью srand(time(NULL)), что обеспечивает разнообразие значений при каждом запуске программы.

4. *Основная функция*

- В основной функции происходит: запрос размера массива у пользователя.
- Объявление переменных.
- Использовании функции для замера времени.

5. *Цикл выбора сортировки*

В этом блоке программа предлагает пользователю выбрать тип сортировки. В зависимости от выбора, массив «a» копируется во временный массив «b»\ «c»\ «d», чтобы сохранить исходные данные для каждой сортировки.

6. *Измерение времени выполнения*

Для каждой сортировки используется функция clock() для измерения времени выполнения. Время выполнения сортировки сохраняется в переменной t.

7. *Вывод результатов*

После завершения сортировок программа выводит время выполнения каждого алгоритма на экран.

Результаты экспериментов

Я проводила эксперименты над массивами, содержащими минимум 1000 элементов, так как при меньшем их содержании выводиться время, которое трудно сравнивать, ведь различия небольшие.

1000 элементов:

BubbleSort – 0,005с
SelectionSort – 0,004с
InsertionSort – 0,005с

Лучший результат в этом тесте показал SelectionSort.

5000 элементов:

BubbleSort – 0,009с
SelectionSort – 0,005с
InsertionSort – 0,006с

Лучший результат в этом тесте показал SelectionSort.

7000 элементов:

BubbleSort – 0,005с
SelectionSort – 0,006с
InsertionSort – 0,007с

Лучший результат в этом тесте показал BubbleSort.

10000 элементов:

BubbleSort – 0,007с
SelectionSort – 0,007с
InsertionSort – 0,006с

Лучший результат в этом тесте показал InsertionSort.

Заключение

Из результатов экспериментов видно, что присутствует разница иногда в долю секунды, иногда в пару секунд, но эту разницу можно считать незначительной. В итоге я считаю, что для маленьких массивов можно

выбрать SearchSort, но для больших массивов стоит пользоваться более улучшенными сортировками.

Литература

https://www.cyberforum.ru/c-beginners/thread2268835.html

Приложение

https://github.com/nasti2x/sort