## Общее описание:

Общий взгляд на продукт. Данный продукт является сортировкой 4 методами строки чисел. С его помощью можно будет указать размерность массива, сам массив и выбрать метод, которым будет проводится сортировка, при этом если пользователь захочет, то будет указано время, потраченное на выбранный метод. Продукт должен обладать простым и удобным интерфейсом, в котором должны быть учтены все требования пользователя.

Основные функции:

* Возможность ввода количества элементов;
* Возможность ввода строки чисел;
* Возможность выбора одного из четырех методов сортировки и применение его к строке чисел;
* Возможность вкл/выкл отображения времени;
* Отображение отсортированного массива.

Классы и характеристики пользователей. Основными пользователями будут являться:

* Оператор: Обычный человек с базовыми знаниями работы с компьтером.
* Оператор-тестировщик: среди технических навыков тестировщика можно выделить знание одного или нескольких языков программирования, умение работать с базами данных и с консолью. Так же необходимо базовое понимание клиент-серверной архитектуры, умение тестировать API и пользоваться снифферами трафика.

Операционная среда. Приложение рассчитано на кроссплатформенность и рассчитано на следующие браузеры:

* MS IE 5.0 и выше;
* Opera 6.0 и выше;
* Mozilla Firefox 1.0;
* Mozilla 1.7.

Ограничения дизайна и реализации. Возможности дизайна ограничены возможностями браузеров, в частности версиями CSS и HTML.

Предположения и зависимости. До установки данного продукта может потребоваться установка Node.js.

## Функции системы

1. Сортировка пузырьком.

Описание сортировки пузырьком. Будем идти по массиву слева направо. Если текущий элемент больше следующего, меняем их местами. Делаем так, пока массив не будет отсортирован. Заметим, что после первой итерации самый большой элемент будет находиться в конце массива, на правильном месте. После двух итераций на правильном месте будут стоять два наибольших элемента, и так далее. Очевидно, не более чем после n итераций массив будет отсортирован. Таким образом, асимптотика в худшем и среднем случае – O(n2), в лучшем случае – O(n).

Функциональные требования сортировки пузырьком. К таким требованиям отнесем:

* Корректный массив чисел
* Корректно заданный размер массива

1. Быстрая сортировка

В методе быстрой сортировки выбирается значение из основного списка, которое называется опорным значением. Остальные значения разделяются на два списка:

* Первый список содержит значения, которые меньше либо равны опорному значению. Эти значения располагаются слева от опорного значения.
* Второй список содержит значения, которые больше опорного значения. Эти значения располагаются справа от опорного значения.

Метод быстрой сортировки повторяется для всех результирующих списков, пока не останется только одно значение или пустой список значений.

После этого вы выбираете последнее одиночное значение, и если значение располагается слева от опорного значения, оно остается таким, пока вы не дойдете до первого опорного значения вверху.

Функциональные требования быстрой сортировки. К таким требованиям отнесем:

* Корректный массив чисел
* Корректно заданный размер массива

1. Сортировка выбором

Алгоритм сортировки выбором заключается в поиске на необработанном срезе массива или списка минимального значения и в дальнейшем обмене этого значения с первым элементом необработанного среза. На следующем шаге необработанный срез уменьшается на один элемент.

* Найти наименьшее значение в списке.
* Записать его в начало списка, а первый элемент - на место, где раньше стоял наименьший.
* Снова найти наименьший элемент в списке. При этом в поиске не участвует первый элемент.
* Второй минимум поместить на второе место списка. Второй элемент при этом перемещается на освободившееся место.
* Продолжать выполнять поиcк и обмен, пока не будет достигнут конец списка.

Функциональные требования сортировки выбором. К таким требованиям отнесем:

* Корректный массив чисел
* Корректно заданный размер массива

1. Сортировка вставками

Описание сортировки вставками. Создадим массив, в котором после завершения алгоритма будет лежать ответ. Будем поочередно вставлять элементы из исходного массива так, чтобы элементы в массиве-ответе всегда были отсортированы. Асимптотика в среднем и худшем случае – O(n2), в лучшем – O(n). Реализовывать алгоритм удобнее по-другому (создавать новый массив и реально что-то вставлять в него относительно сложно): просто сделаем так, чтобы отсортирован был некоторый префикс исходного массива, вместо вставки будем менять текущий элемент с предыдущим, пока они стоят в неправильном порядке.

Функциональные требования сортировки вставками. К таким требованиям отнесем:

* Корректный массив чисел
* Корректно заданный размер массива

## Требования к внешним интерфейсам

Графический интерфейс реализован в файлаx «main.html» и «main.css». Выглядит он следующим образом:

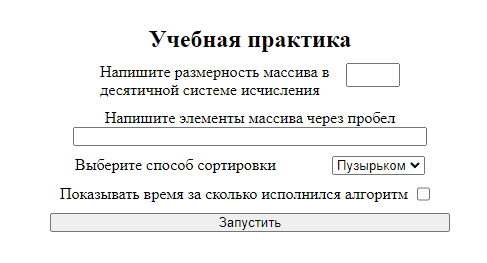


Рисунок 1 – графический интерфейс программы\*

\*Источник: выполнено автором

Размерность массива, строка ввода массива, выпадающий список, флажок для вкл/выкл отображения времени и кнопка «Запустить» – это элементы графического интерфейса, позволяющие пользователю ориентироваться в программе и корректно вводить данные.

Запустить – функциональная кнопка, запускающая сам процесс сортировки.

Поля для ввода и вывода представлены – input в теге form.

Аппаратные интерфейсы это: стандартное оборудование компьютера, включающее монитор, клавиатуру, мышь, модем.

## Атрибуты качества

Удобство использования. Минималистичный дизайн интерфейса, позволяет легко ориентироваться в программе.

Производительность. Программа была оптимизирована для работы на слабых ПК.

Безопасность. Так как программа не работает с данными, которые несут конфиденциальную вопрос безопасности не поднимался.

Техника безопасности. Особых требований к ТБ нет.