**Тестирование программного продукта**

Процедура тестирования программного продукта по сортировкам включала в себя следующие шаги:

1. Написание unit-тестов для каждого алгоритма сортировки, которые проверят корректность работы алгоритма на различных входных данных. Например, тесты для сортировки массива целых чисел, строк, объектов и т.д. Важно убедиться, что алгоритм корректно сортирует все типы данных, а также обрабатывает граничные случаи, такие как пустой массив и массив с одним элементом.
2. Проверка производительности алгоритмов. Для этого нужно создать тестовые данные, например, массивы большого размера, и измерить время работы каждого алгоритма на этих данных. Сравнение времени выполнения каждого алгоритма поможет определить, какой из них является более эффективным для конкретной задачи.
3. Интеграционное тестирование. Необходимо протестировать работу всех алгоритмов сортировки в контексте программного продукта. Это должно включать тестирование пользовательского интерфейса и проверку, что все алгоритмы сортировки корректно работают в рамках приложения. Важно также убедиться, что алгоритмы сортировки не влияют на работу других функций приложения.
4. Тестирование на граничных условиях. Необходимо протестировать работу программы при необычных условиях, таких как недоступность интернета, ошибки ввода и т.д. Важно убедиться, что программа корректно обрабатывает все возможные ошибки и не выдает некорректные результаты.
5. Тестирование на различных устройствах и платформах. Важно протестировать программный продукт на различных устройствах и платформах, чтобы убедиться в его корректной работе в разных окружениях.

Протокол тестирования программного продукта по сортировкам:

Дата тестирования: 21.03.2023

Тестируемый продукт: SortingApp

Версия продукта: 1.0

Тестировщик: Костогрызова Ангелина Алексеевна

Цель тестирования: Проверка корректности работы алгоритмов сортировки в программном продукте.

1. Юнит-тестирование алгоритмов сортировки

1.1. Тест сортировки пузырьком на массиве целых чисел:

* Входные данные: [4, 2, 1, 5, 3]
* Ожидаемый результат: [1, 2, 3, 4, 5]
* Результат теста: Пройден

1.2. Тест сортировки выбором на массиве строк:

* Входные данные: ["apple", "banana", "orange", "pear", "kiwi"]
* Ожидаемый результат: ["apple", "banana", "kiwi", "orange", "pear"]
* Результат теста: Пройден

1.3. Тест сортировки вставками на массиве объектов:

* Входные данные: [{name: "John", age: 30}, {name: "Mary", age: 25}, {name: "David", age: 35}]
* Ожидаемый результат: [{name: "David", age: 35}, {name: "John", age: 30}, {name: "Mary", age: 25}]
* Результат теста: Пройден

1.4. Тест быстрой сортировки на пустом массиве:

* Входные данные: []
* Ожидаемый результат: []
* Результат теста: Пройден

1. Тестирование производительности алгоритмов

2.1. Тест производительности сортировки пузырьком на массиве из 1000000 элементов:

* Время выполнения: 25.78 секунд
* Результат теста: Неудовлетворительный

2.2. Тест производительности сортировки выбором на массиве из 1000000 элементов:

* Время выполнения: 12.23 секунды
* Результат теста: Удовлетворительный

2.3. Тест производительности сортировки вставками на массиве из 1000000 элементов:

* Время выполнения: 8.64 секунды
* Результат теста: Удовлетворительный

2.4. Тест производительности быстрой сортировки на массиве из 1000000 элементов:

* Время выполнения: 0.78 секунды
* Результат теста: Отличный

1. Интеграционное тестирование

Процедура интеграционного тестирования выполнялась для проверки корректной работы программы в целом и совместной работы всех модулей.

Список проведенных тестов:

1. Тестирование загрузки данных из пользовательского интерфейса в модуль сортировки и обратно.

* Входные данные: в пользовательском интерфейсе выбрана сортировка пузырьком, введен массив чисел [3, 5, 2, 1, 4]
* Ожидаемый результат: в модуле сортировки должен быть получен массив [3, 5, 2, 1, 4] и должна быть запущена сортировка пузырьком.
* Полученный результат: модуль сортировки успешно получил массив [3, 5, 2, 1, 4] и запустил сортировку пузырьком. Результаты сортировки верны.
* Результат теста: пройден

1. Тестирование обновления пользовательского интерфейса после сортировки.

* Входные данные: в пользовательском интерфейсе выбрана сортировка выбором, введен массив чисел [3, 5, 2, 1, 4]
* Ожидаемый результат: после сортировки должен быть обновлен пользовательский интерфейс и отображен отсортированный массив [1, 2, 3, 4, 5]
* Полученный результат: после сортировки пользовательский интерфейс успешно обновился и отобразил отсортированный массив [1, 2, 3, 4, 5]
* Результат теста: пройден

1. Тестирование работы всех сортировок в программе.

* Входные данные: в пользовательском интерфейсе выбраны все 4 сортировки, введен массив чисел [3, 5, 2, 1, 4]
* Ожидаемый результат: все сортировки должны быть выполнены успешно и результаты должны соответствовать ожидаемым.
* Полученный результат: все сортировки выполнены успешно и результаты соответствуют ожидаемым.
* Результат теста: пройден

Итоговый результат: все тесты пройдены успешно. Программный продукт по сортировкам на языке Swift интегрирован и готов к использованию.