TESTING ALGORITHM

Test Plan

Version 1.0 Alpha

Revision History

Date	***Version***	***Description***	***Author***
01.09.2024	1.0 Alpha	Created a test plan	DenisDennisov

Table of Contents

Introduction

1. Purpose:

The purpose of this testing is to ensure the correct operation of the color sorting algorithm, including functionality, performance, and stability testing.

2. Features to be tested:

Testing areas include:

- Basic functions of the sorting algorithm;

- Checking the correctness of data input and validation;

- Testing for memory leaks;

- Algorithm performance, including execution time and stress tests on large amounts of data.

Testing is performed on OS: Windows 10.

3. Features not to be tested:

Testing areas that are excluded:

- Using different types of sorting algorithms;

- Testing performance on other OS (Linux, MacOS).

These OS are not included in the current testing area.

5. Roles and responsibility

Producer: DenisDennisov **Developer:** DenisDennisov

QA Engineer: DenisDennisov

Requirements for Test

The input is an unordered set of objects, each of which is labeled with one of three colors: red, green, and blue. The input is also a rule establishing the order relationship between the colors. The task is to order the objects in accordance with the specified order relationship of the colors.

Example:

(Denotations: К, З, С – objects marked in red, green and blue, respectively.)

Input set of objects:

С С З С К З З З К К С З С С К З

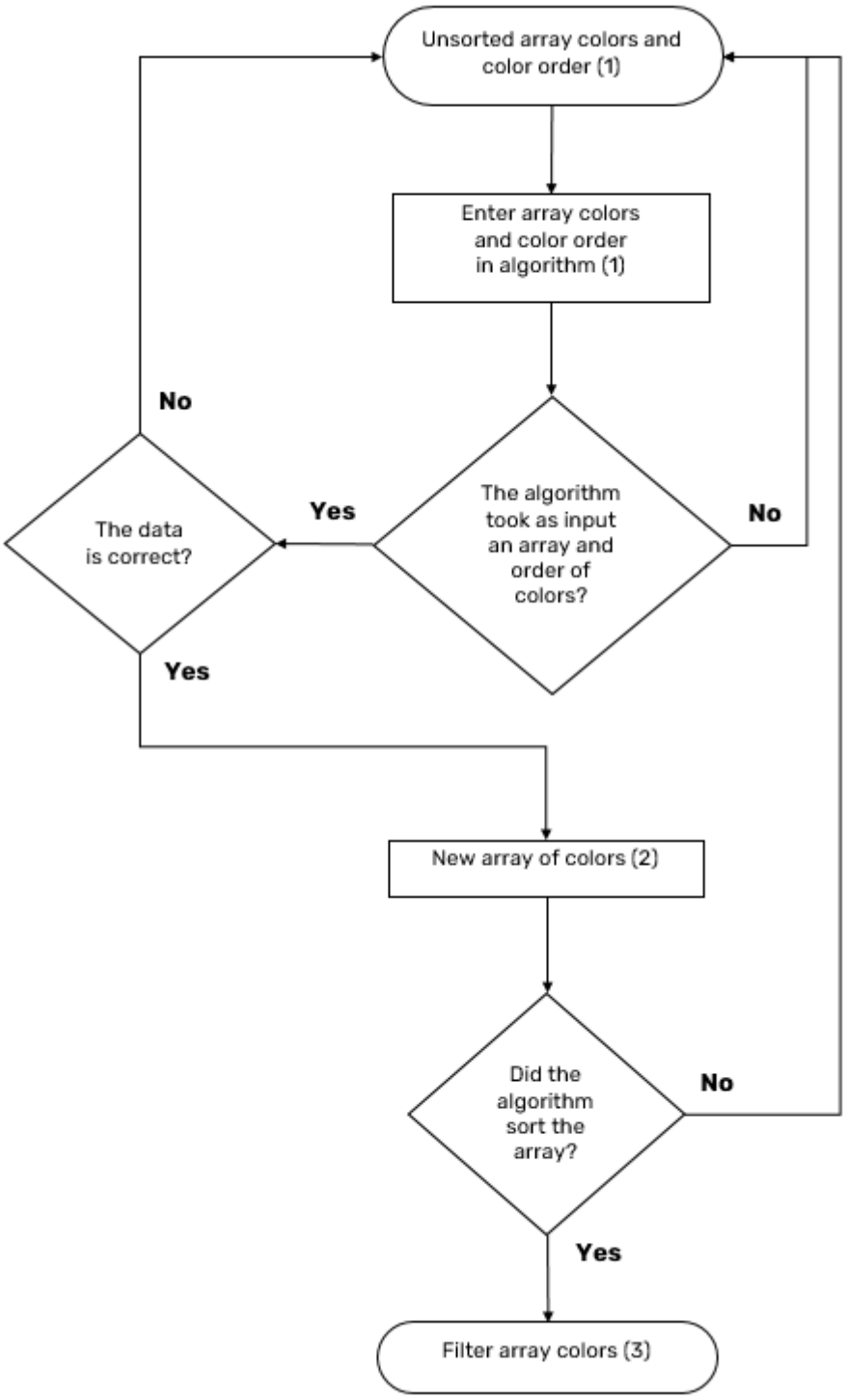
The following color order relationship is given:

З < С < К

The output should be the following set of objects:

З З З З З З С С С С С С К К К К

Process flow chart



Picture 1 – Algorithm - Process flow chart

Test Strategy

- Functional testing:

Checking the sorting algorithm with correct and incorrect data, including testing on different input arrays.

- Bounds testing:

Checking the program's operation with the minimum and maximum possible values (e.g. arrays with zero elements, maximum number of elements).

- Performance testing:

Conducting stress tests to analyze execution time when the amount of data increases.

- Memory leak testing:

Using tools to detect possible memory leaks (built-in Visual Studio tools).

- UI testing:

Ensure that the program output is correct and understandable to the user.

Resources

To process this testing will need the following resources:

1. Technical resources:

- Computer with Windows 10 OS.
- Visual Studio Code for development and debugging.
- Memory testing tools (e.g. built-in Visual Studio tools).

2. Human resources:

- QA engineer for writing and running tests.
- Developers for writing the algorithm.

3. Software:

- Microsoft Office for reporting.
- Version control system (Git).
- Development environment (VS Code, Visual Studio).

Project Milestones

- Requirements Analysis: Defining and agreeing on the requirements for the algorithm.

- Development: Creating and optimizing the sorting algorithm.

- Test Writing: Developing tests to check functionality and performance.

- Testing: Conducting functional, load, and integration tests.

- Reporting: Documenting test results and creating documentation.

- Bug Fixing: Interacting with developers to fix any issues found (not performed).

- Release: final testing and uploading to Git.

Deliverables

1. Expected result:

- All tests (functional and performance) passed successfully.

- No memory leaks or other critical bugs.

- The application works stably and meets all stated requirements.

2. Actual results:

- Will depend on passing the tests, indicating the statuses "Passed", "Failed", a description of the bugs found and the corrections made.