

3.2 Visualization and analysis of sorting algorithms

Introduction. Creating a program to visualize sorting algorithms [92-94] can make this topic easier for students [93-96] or others starting a career in programming or a related field.

The visualization aspect is essential as it enables decision makers to analyze the proximity of the optimal solution [97-101]. Using different types of visualizations will increase the visibility of the algorithm [100, 102-105] and deriving various additional data will help the analysis process [93, 100, 106].

We all know that sorting is used everywhere these days [92, 95, 107, 108, 109]. Because it's obvious – it's much easier to work with sorted data [100, 106]. There is no website that does not use sorting. Not to mention software and even video games. However, many developers don't care about the choice of algorithm [106], and often even use libraries that make programs overloaded. However, with the right choices, you can reduce the resources and time required for processing.

Thus, the software that visualizes the sorting process will help people to see clearly [107] in which case it would be more appropriate to use the appropriate algorithm [93, 100, 106, 110] and obtain accurate data, which, with proper statistical processing [97, 103], will allow you to make sure of your choice [92, 111, 112].

Actuality of theme. The main problem of our software:

- small target audience (interest and demand for the product);
- large number of programs of a similar direction (competitors);
- low possibility of earning on the product (lack of the ability to insert advertising and low demand).

To solve these problems, you need a strong marketing company:

- an offer to schools from different cities of Ukraine, and maybe other countries, to use our application for learning in computer science lessons of middle classes;
- collaboration with other applications;
- expansion of functionality in subsequent updates (possible addition of sorting

user data with the integration of sql, excel and other files).

The object of research is the visualization of algorithms.

The subject of research is sorting algorithms.

Research methods: comparison of visualization methods with competitors.

The aim of the work is to create an application that would help the young and not only generation to be interested in the world of IT by showing the work of the initial algorithms, namely sorting algorithms. The software should have a convenient and modern interface and be more functional than that of competitors.

With our application, people will be able to expand their horizons by learning different digital ways of sorting numbers and, thanks to the statistics we display, determine the appropriate one for a particular case.

Results and discussion.

(a) Scope. The project team is working hard and doing everything possible to create software for visualizing sorting algorithms and data processing of these algorithms, which:

- during its work will not load the computer too much. This is necessary in order to make the application more convenient to use and give the most accurate data. The program should display the coefficient of time that is spent on sorting the array. Our task is to make this coefficient as low as possible even for the slowest computers;
- showed correctly sorted data without any errors and inaccuracies. This issue is dealt with by the tester in cooperation with the main developer;
- had a user-friendly interface that would be easy and efficient to work with. It should also have pleasant colors so that when working with the application for a long time, the user's eyes do not get tired and he can continue working as long as possible. The software must have a beautiful, stylish design (fonts, buttons, colors, images) in order to attract new users while maintaining a high download speed;
- clearly shows the work of sorting algorithms. The introduction of several types of visualization will increase the visibility effect and attract more users;
- is different from other similar applications of competitors to maintain interest in the product;

PROSPECTIVE DIRECTIONS OF SCIENTIFIC RESEARCH IN ENGINEERING AND AGRICULTURE

- has a navigation system for easier and faster mastering of the interface and the operation of the product as a whole;
- will have a convenient placement of information.

This software will not have:

- sorting text data (only arrays of numbers);
- operation of all existing sorting algorithms at the initial stage. Our task is to release our product as quickly as possible, and new algorithms can be introduced with subsequent updates;
- detailed and complex customization of algorithms by the user. The audience we are counting on are people who are just starting their way into algorithms and programming in general.

Stakeholders. Several Stakeholders were involved in the creation of the software and its documentation. They are divided into development (Table 1) and testing groups (Table 2-3)

Development group (Table 1).

Table 1.

Stakeholder Main software developer

Name	Main software developer
Description	Lead Creator of the Product
Responsibilities	Writes code and creates an interface for the entire application. Decides what specific goals are to be achieved and what the overall software should look like
Skills	Knowledge of the C# programming language, knowledge of Windows Forms, understanding of the operation of sorting algorithms

Testing group (Table 2-3).

Table 2.

Stakeholder Main tester

Name	Main tester
Description	Lead Product Tester
Responsibilities	Tests the entire code for bugs, looks for parts of the code in which the user can enter data and intentionally enters incorrect values
Skills	Knowledge of the C# programming language, knowledge of Windows Forms, tester skills

Table 3.

Stakeholder Text editor

Name	Text editor
Description	Makes amendments to the text of the documentation
Responsibilities	Rereads the text written by Main software developer in search of grammatical, spelling and semantic errors
Skills	Perfect knowledge of Ukrainian and English, ability to use Microsoft Word, general understanding of the topic

Non-functional requirements:

1. User friendly. Our users will be people who are just starting their journey in IT (and maybe in computer technology), so the interface and the program itself should be easy to use.
2. No bugs due to user errors. The places where the user can enter any data should be rechecked and give the user an internal error, not a code error.
3. Ability to change / update / read the code. Updates are planned in our software with the further addition of new functions and algorithms. Therefore, the code should be understandable and be able to add something to it.
4. High performance. The program must perform its functions without any delay, otherwise it may affect the analysis of sorting algorithms or the overall user experience.

Terms and acronyms. Terms:

- C# – object-oriented programming language.
- Bug – coding error in a computer program
- Stakeholders – people involved in the project
- Interface – set of means by which the user interacts with various

programs and devices:

- Excel – spreadsheet program

Acronyms:

- SQL – structured query language
- IT – information technology

(b) Application Component. The project implements only one component – application. This is because no database, web server, or any other external component is connected to the software.

The application works in C# using Windows forms (Figure 1).

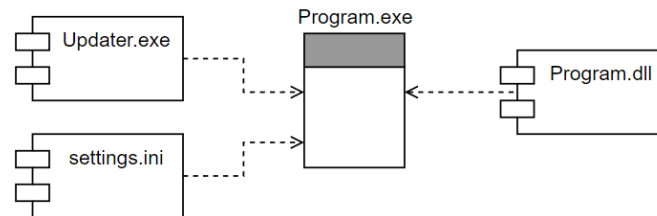


Figure 1. Component diagram

It consists of three parts: control panel (entering the number of elements in the array box, entering the range of possible numbers box, array generation button, drop-down list with sorting algorithms, array sorting button, pause button, delay increase and decrease buttons), text visualization panel and panel graphic visualization.

Responsibilities. The application should work continuously and smoothly, not crash, not wait for a response. Also, there should be no errors associated with user input. All algorithms must work correctly and output the result with minimal delay.

Features. Windows Forms is very handy for creating applications. Its peculiarity is that the creation of the interface is much easier and intuitive. To assign a code when you press any of the buttons, you can simply double-tap it. It is also very easy to modify the interface itself in the future (for more detailed modification, there are separate libraries).

Setup. In order to setup the application, we recommend that you install Visual Studio. When installing, check the boxes next to .net and Windows Forms.

Usage. The code runs in a separate window. To use it, a short instruction was created:

First, enter the number of array elements (minimum 3, maximum 200). After we enter in what range random numbers will be generated for our array. Next, select the

type of sorting and click the button to start sorting. While the algorithm is running, we can change the delay and pause.

(c) Use-Case View. The Use-Case View of our project is very simple due to the limited functionality (due to the uselessness of extra functions) and the small size of the project itself (Figure 2).

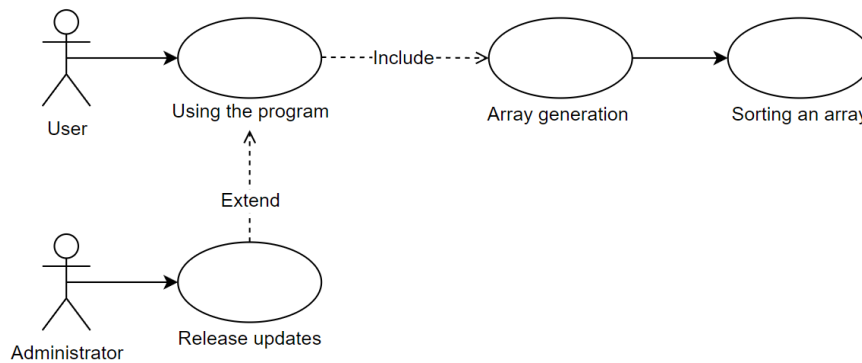


Figure 2. User Case diagram

Using the program. Using the program includes any user actions aimed at obtaining the result of the application (viewing the visualization of the sorting of the array during the program's operation and obtaining a sorted array). Includes array generation and sorting.

Array generation. Random array generation by pressing a button. At this stage, the application has no other way to generate array (from user documents, or by user input).

Sorting an array. Start sorting by pressing the button and wait for sorting to finish. Until the end – viewing visualizations of different types (text and diagram).

Release updates

Release by the administrator of updates created by the main software developer. Expands the possibilities of using the program by the user.

(d) Logical View. In our project, there is no Business Services Package due to the lack (at this stage of development) of possible profit and any user authorization.

The user interface package includes classes to simplify the user's work with the program and better visual display.

The application package includes classes for processing user input and outputting a specific desired result from that processing. Processing is the sorting of the data that is in the data package.

The data package includes classes for storing data such as a generated array consisting of random numbers from a range entered by the user (Figure 3).

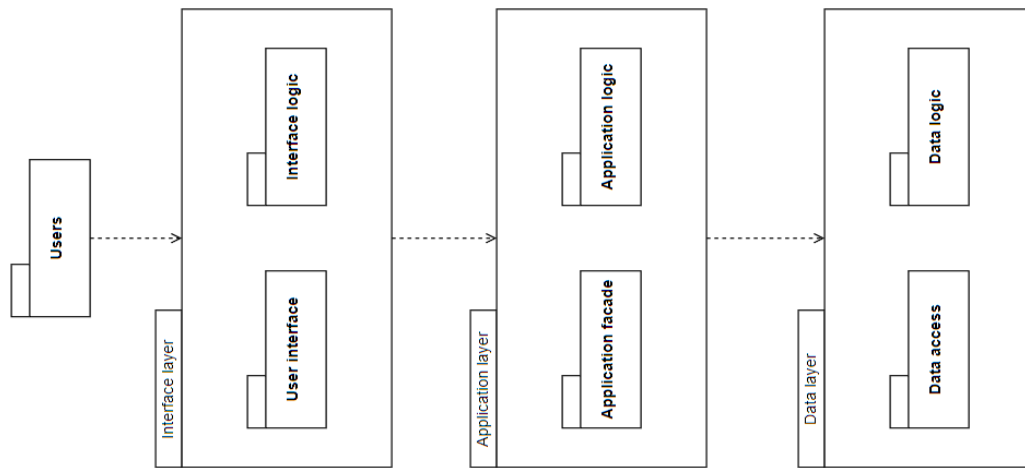


Figure 3. Package layering diagram

(e) Testing. Testing was done on a single component: the application. The testing was not done by a special program, but by a person: the main tester. He made the following checks:

- Checking all possible user inputs (number of array elements, minimum array number, maximum array number);
- Checking for pressing different buttons in different orders and situations (generate button, start sorting button (together with Drop down list), pauses, increase delay, decrease delay);
- Check the performance of the program on different computers with different characteristics (processor, video card, monitor (screen size));
- Test the program on different operating systems;
- Check step by step the correct output of the program;
- Check the interface for anomalies (important when drawing diagrams).

PROSPECTIVE DIRECTIONS OF SCIENTIFIC RESEARCH IN ENGINEERING AND AGRICULTURE

We do this by giving the user the opportunity to demonstrate the working of the following algorithms: Bubble Sort [109], Shaker Sort, Insertion Sort [109], Selection Sort [109], Shell Sort, Quick Sort, BogoSort. The user can see this on different arrays (by length and by number in the array) and at different speeds. Based on the analysis, users can choose the algorithm that best suits their purpose (Figure 4).

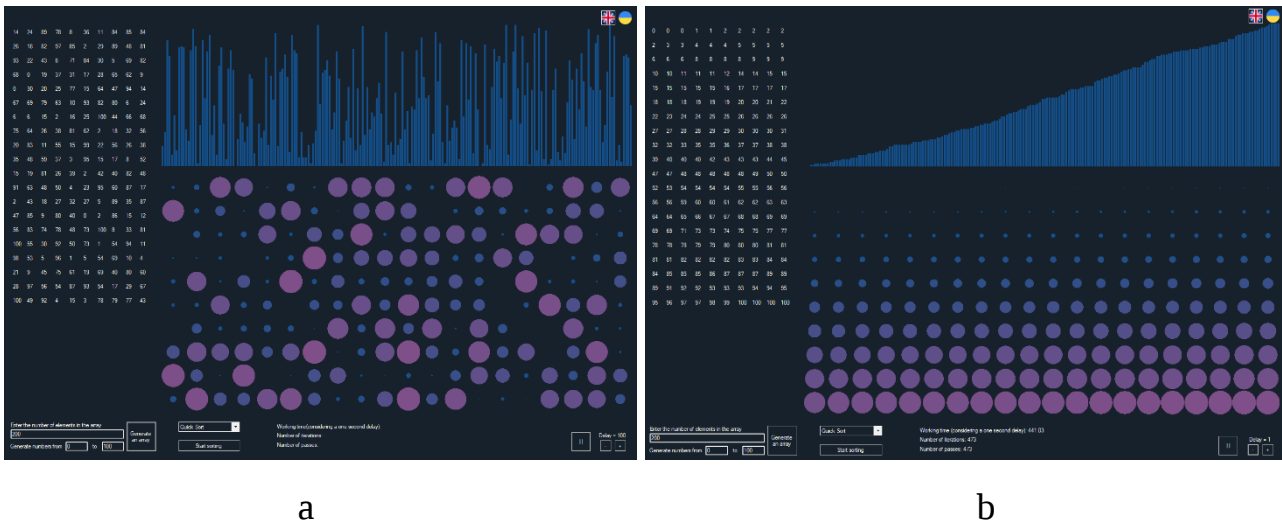


Figure 4. Application interface: a – array before sorting; b – sorted array

Conclusions. The research-developed software has full functionality and appropriate visual support to easily perceive and understand the ranking algorithm. The software also enables data output, making it possible to explore and analyze sorting algorithms.