EasyChair Preprint

№ 4751

Development and using of a virtual laboratory to study the graph algorithms for bachelors of software engineering

Andrii Striuk, Olena Rybalchenko and Svitlana Bilashenko

December 20, 2020

Development and Using of a Virtual Laboratory to Study the Graph Algorithms for Bachelors of Software Engineering

Andrii Striuk[0000-0001-9240-1976], Olena Rybalchenko[0000-0001-8691-5401] and Svitlana Bilashenko[0000-0002-4331-7425]

Kryvyi Rih National University, 11 Vitalii Matusevych Str., Kryvyi Rih, 50027, Ukraine
andrey.n.stryuk@gmail.com

Abstract. The paper presents an analysis of the importance of studying graph algorithms, the reasons for the need to implement this project and its subsequent use. The existing analogues analysis is carried out, due to which a list of advantages and disadvantages is formed and taken into account in developing the virtual laboratory. A web application is created that clearly illustrates the work of graph algorithms, such as Depth-First Search, Dijkstra's Shortest Path, Floyd-Warshall, Kruskal Minimum Cost Spanning Tree Algorithm. A simple and user-friendly interface is developed and it is supported by all popular browsers. The software product is provided with user registration and authorization functions, chat communication, personal cabinet editing and viewing the statistics on web-application use. An additional condition is taken into account at the design stage, namely the flexibility of the architecture, which envisaged the possibility of easy expansion of an existing functionality. Virtual laboratory is used at Kryvyi Rih National University to training students of specialty 121 Software Engineering in the disciplines "Algorithms and Data Structures" and "Discrete Structures".

Keywords: Virtual laboratory, graph theory, graph algorithms, visualization, software engineering.

1 Introduction

The graphs theory is an important part of discrete mathematics and one of the components of a fundamental bachelor's degree in software engineering [1]. The graph theory is studied in higher educational institutions individually or as part of other disciplines [2, 3]. Analysis of educational programs [4] shows that the greatest attention is paid to the following elements of the graph theory practical application:

1. Overview of classical algorithms: breadth-first search (BFS) and depth-first search (DFS), finding the shortest paths and maximum flow, etc.
2. Solving typical tasks of practical nature (about telecommunication towers, color mapping, some logical tasks related to graphical representation of relations).

3. Determination of the main ways of applying graphs: the search for related components in communication networks; search for the shortest and cheapest routes in communication networks.
4. Construction of skeletal tree: search for the maximum flow for the transport network, which identifies the sources, drains and bandwidth of the ribs.
5. Graph isomorphism.
6. Finding cycle in a graph: the Hamiltonian cycle (the traveling salesman problem (TSP); Euler cycle (network capability control).
7. Graph coloring (geographical map coloring, creating schedules of training, resource allocation, etc.).
8. Planar graph (printed electronic design on electrical circuits, transport solutions, etc.).
9. Finding the centers of a graph (vertices, the maximum distance from which to all other vertices of the graph is minimal).

The main goals of our project are the development of an online system that would provide a detailed visualization of an algorithm functioning; the possibility of using it as a demonstration material during lectures, use as a platform for performing independent work with an additional opportunity of textual communication with mentors (teachers) to facilitate the assimilation of new material and to speed up the search for answers to questions that arose during training.

According to the goals, we need to solve the following tasks:

1. Build a hierarchical model of actor systems.
2. Create library modules of graph algorithms.
3. Develop a module for their visualization.
4. Develop a web resource map.
5. Modify the system of web application classes.
6. Develop data structures.
7. Design a database model.
8. Design a visual interface.
9. Implement project software modules.

2 Analysis of existing developments

Studying graph algorithms is not possible without a visual demonstration. Virtual laboratories are able to visualize graphs and step-by-step execute the algorithms. Such virtual laboratories can be used by teachers during lectures, students during independent study of the material, the implementation of computational experiments and practical tasks. Not surprisingly, the development of graph visualization tools and algorithms is always given special attention.

One of the most interesting examples of such virtual laboratories is the “Date Structure Visualizations of the University of San Francisco” [5]. This resource, which can be seen in Figure 1 and Figure 2, positions itself as a web application of the University of San Francisco website which demonstrates the algorithms visually.

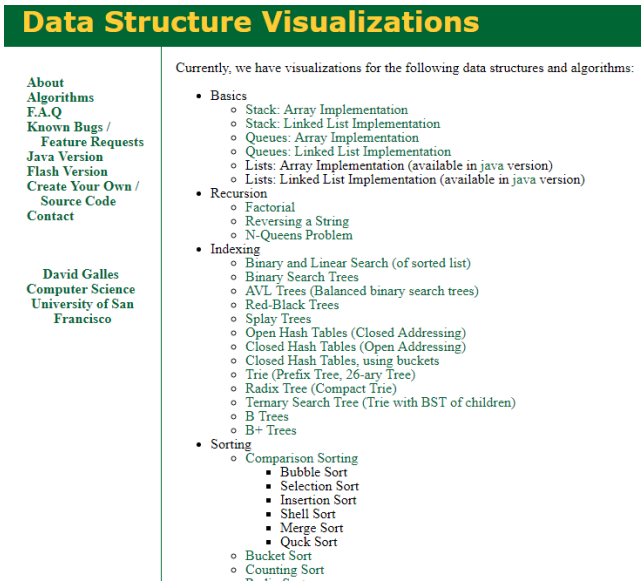

Fig. 1. The University of San Francisco web application

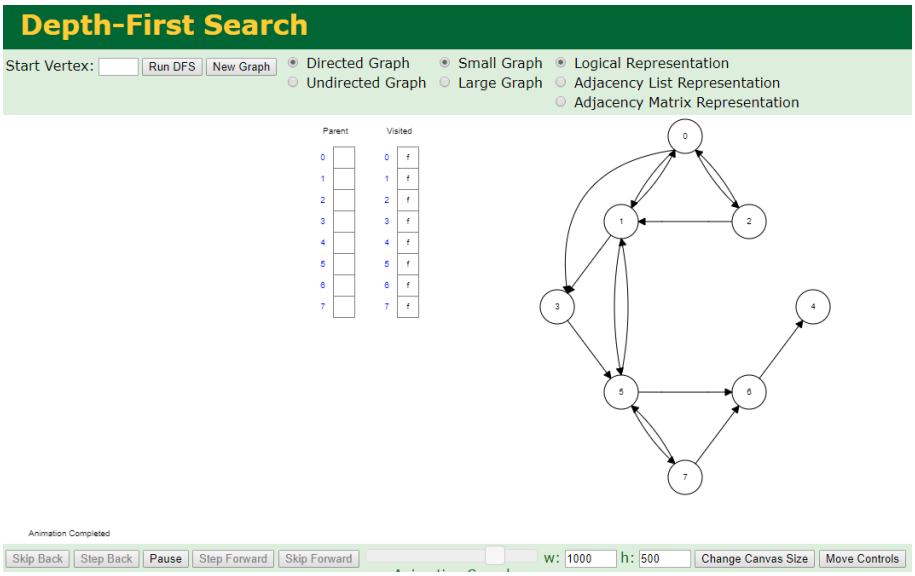

Fig. 2. Algorithm demonstration

This web resource presents a wide variety of algorithms, among which: basics algorithms, recursive algorithms, indexing algorithms, sorting algorithms, heap-like data structures, graph algorithms, dynamic programming, geometric algorithms.

Particular attention is paid to the following graph algorithms: Breadth-First Search, Depth-First Search, Dijkstra's Shortest Path, Prim's Minimum Cost Spanning Tree,

Topological Sort (Using Indegree array and DFS), Floyd-Warshall (all pairs shortest paths), Kruskal Minimum Cost Spanning Tree Algorithm.

The main advantages of [5] can be considered:

— Clear, not overloaded with interface elements.
— Convenient structured list of algorithms
— Sufficiently wide range of different algorithms.

Among disadvantages we can list the following:

- Outdated design of the pages, style is not pleasant to the eye.
- Absence of any textual description of algorithms.
- Presence only of English localization.
- The lack of returning option on the main page.

Another example of a web-based graph and graph visualization program is Greuler [6]. This resource (Fig. 3) positions itself as a web application that demonstrates the capabilities of the library which renders graphs and was created by Mauricio Popp.

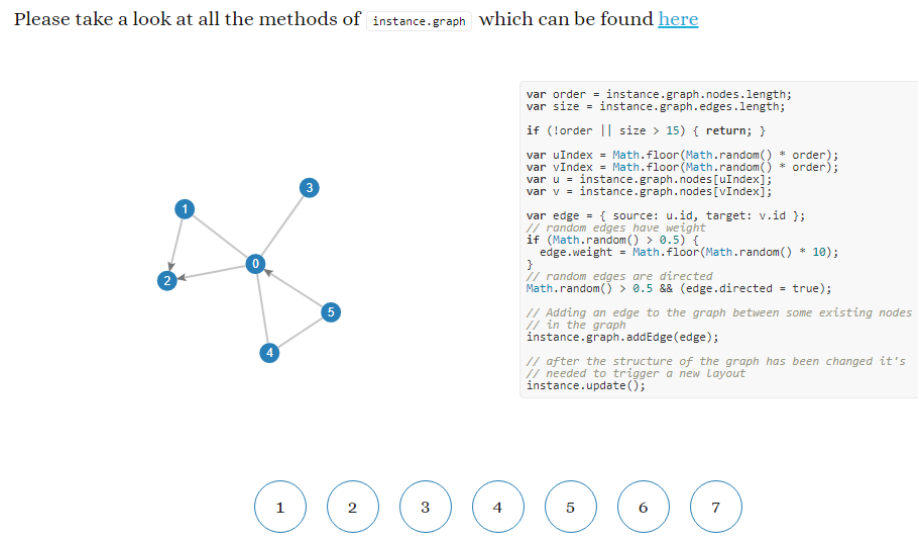

Fig. 3. Demo page

The main advantages of [6] can be considered:

— Clear, not overloaded with interface elements.
— A clear and concise presentation of information.
— Intuitive interface.

Among disadvantages we can list the following:

— Absence of a list of algorithms.

– Presence only of English localization.

Based on given the advantages and disadvantages of existing programs, we decided to develop a virtual laboratory for studying graph algorithms that would satisfy the following requirements:

– a simple and adaptive web interface due to which it would be equally convenient to work on both desktop PCs and mobile devices;
– the ability to authorize users and provide control over student activity;
– support for multiple user roles;
– a demonstration of the basic graph algorithms, the study of which is provided by the training program for students of specialty 121 Software Engineering;
– Ukrainian-language localization, facilitating work with a virtual laboratory for students of Ukrainian universities;
– the ability of students to communicate with teachers and among themselves during laboratory work.

3 Results

The developed system should contain the following functions:

(1) User authorization.
(2) Providing users access to their personal profile.
(3) Ability to flexibly edit user's personal data.
(4) Sharing short text messages (chat).
(5) Presence in the system of roles "Administrator", "Mentor", "Student", "Guest".
(6) "Administrator" – a role in the system, which provides unlimited access to user profiles, ability to chat with them and overview the general statistics of the web application.
(7) "Mentor" – a role in the system, envisaging the possibility of viewing the user profiles with a role of "Student", chat-communication with them and viewing their statistics of study graph algorithms presented in the system.
(8) "Student" – a role in the system, envisaging the possibility of studying the graph algorithms represented in the system, and conducting chat-communication with the users with the role of "Mentor".
(9) "Guest" – a role in a system that is automatically assigned to all unregistered users. It provides only the possibility of studying the graph algorithms represented in the system.
(10) Availability in the system of visualization modules of such graph algorithms: "Depth-first search", "Dijkstra's algorithm", "Floyd-Warshall Algorithm" and "Kruskal's algorithm".
(11) Maintaining statistics on the time spent by users with the role of "guest" on the study of algorithms.
(12) Provide the system with a feedback module.

Technological requirements:

1. The server and client parts of the web application should not be dependent on the operating system or version of the operating system.
2. The web application must be compatible with browsers like Google Chrome, Opera, and Mozilla Firefox.
3. The server part should be developed using PHP (version 7) and MySQL (version not lower than 5.0).
4. When developing the system, you need to use a modular approach: the source texts must be divided into modules depending on the implemented functionality.
5. To build the user interface, the Model-View-Controller (MVC) approach should be used. The developer should not mix within one implementation file, responsible for building the interface and business logic.
6. User rights must be consolidated in one file, module or table, external, in relation to the functions themselves.

Non-functional requirements:

1. The system interface should be built to take into account the latest trends in web application development.
2. The interface should not be overloaded with visual elements.
3. The system should ensure the simultaneous operation of many users, without its rejection. It is allowed to increase the response time of the system.

Additional requirements:

1. To enhance the usability of the system, all features of the web interface of AJAX technology, where applicable, such as chat communication and graphical display of statistics, should be implemented without over-loading the page after each action.
2. The system should be developed “open” for change, with the submission of exclusive rights.

Figure 4 shows the main window of the program, on which the user is prompted to select an algorithm for study.

Figure 5 shows a page demonstrating one of the graph algorithms (Depth-first search). Working with a virtual laboratory, the user can use the prepared graphs or create a new arbitrary graph. The visual display of the graph is supported, as well as its presentation in the form of an adjacency matrix or adjacency list. It is possible to work with both oriented and non-oriented graphs. For most algorithms, such as graph search algorithms, it is possible to set a start vertex. Otherwise, the program will consider the initial vertex 0.

The visualization of the algorithm takes place in the form of an animation that can be viewed step-by-step or continuously. The user can adjust the animation speed independently.

An additional feature of the program is the display of theoretical information about the algorithm. To go to theoretical materials, you must click the Description button (Fig. 5).

Fig. 4. The page for choosing an algorithm to study

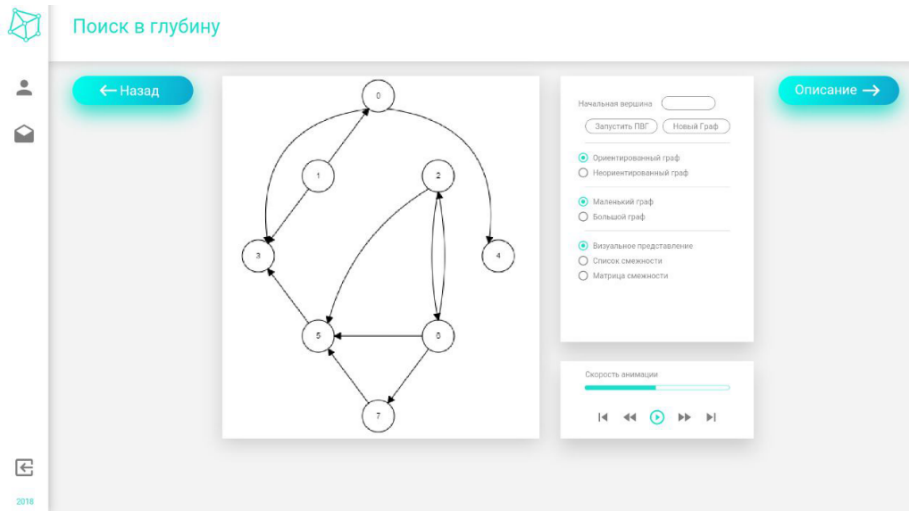

Fig. 5. Depth-First Search Algorithm Demo

The program records the activity of each student. For this purpose, a user profile is created (Fig. 6). The user can control the general data independently.

Authorized users have the opportunity to communicate in a special chat (Fig. 7). Virtual laboratory chat has primarily educational purpose. With its help, students can receive teacher advice, receive assignments or recommendations, and communicate with each other.

It should be noted that students of specialty 121 Software Engineering were involved in the development and testing of the virtual laboratory individual components. This provided them with the opportunity to gain practical experience in team work on

software products, as well as to deepen their own knowledge in the field of graph theory. A subsequent group of students continues to work with the program as users, studying graph algorithms and experimenting with them, as well as finalizing the program, developing new components, implementing additional algorithms, etc.

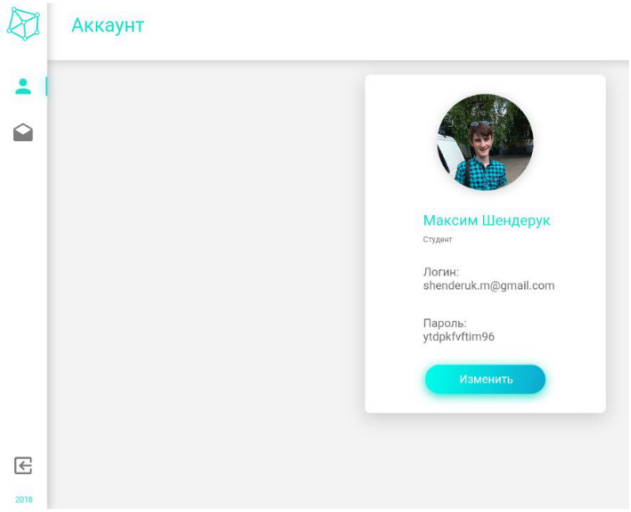

Fig. 6. Virtual laboratory user profile

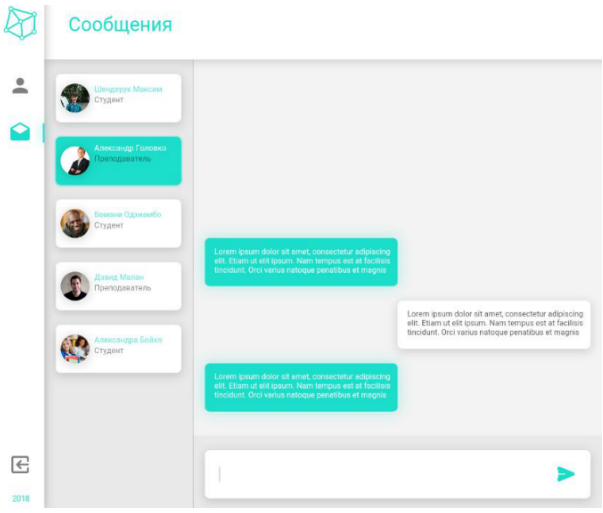

Fig. 7. Virtual laboratory chat

The simple and minimalistic interface of the virtual laboratory simplifies the work with it and provides the opportunity to adapt it for both desktop and mobile devices. This program feature creates the conditions for its effective use in the mobile training of software engineers [7; 8].

Today, a virtual laboratory is used at Kryvyi Rih National University to training students of specialty 121 Software Engineering in the disciplines “Algorithms and Data Structures” and “Discrete Structures”. The development of additional components is included in the topics of qualification work of bachelors in software engineering.

4 Conclusions

The precedence diagrams, class diagrams, and flowcharts and site interface were created in the process of designing a software product. Programming technologies and database architecture were selected during the development. The most important modules of the software complex and the typical scenario of its use were considered.

The following series of tasks are successfully completed:

1. A web application is created that clearly illustrates the work of graph algorithms.
2. A simple and user-friendly interface is developed and it is supported by all popular browsers.
3. The software product is provided with the function of user registration and authorization, chat communication, personal cabinet editing and viewing the statistics on web-application use.
4. A reliable user data protection system is developed. The security system is based on modern data encryption technology.

An additional condition was taken into account at the design stage, namely the flexibility of the architecture, which envisaged the possibility of easy expansion of an existing functionality.

The software package has been tested and is fully ready for further use.

References

1. Shchedrolosiev, D.Ye.: Metodychna systema navchannia dyskretnoi matematyky maibutnikh inzheneriv-prohramistiv zasobamy informatsiinykh tekhnolohii (Methodical system of teaching discrete mathematics of future software engineers by means of information technologies). Dissertation, Kherson State University (2011)
2. Bourque, P., Fairley, R.E. (eds.): SWEBOK V3.0: Guide to the Software Engineering Body of Knowledge. IEEE Computer Society (2014)
3. Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. https://computingcurricula.com/files/SE2014.pdf (2015). Accessed 21 Mar 2020
4. Semerikov, S., Striuk, A., Striuk, L., Striuk, M., Shalatska, H.: Sustainability in Software Engineering Education: a case of general professional competencies. In: Semerikov, S., Chukharev, S., Sakhno, S., Striuk, A., Osadchyi, V., Solovieva, V., Vakaliuk, T., Nechypurenko, P., Bondarenko, O., Danylchuk, H. (eds.) The International Conference on Sustainable Futures: Environmental, Technological, Social and Economic Matters (ICSF 2020). Kryvyi Rih, Ukraine, May 20-22, 2020. E3S Web of Conferences **166**, 10036 (2020). doi:10.1051/e3sconf/202016610036

5. Galles, D.: Data Structure Visualizations. Computer Science. University of San Francisco https://www.cs.usfca.edu/~galles/visualization/Algorithms.html (2011). Accessed 21 Mar 2020
6. Poppe, M.: greuler - graph theory visualizations. https://mauriciopoppe.github.io/greuler (2017). Accessed 21 Mar 2020
7. Tkachuk, V.V.: Mobilni informatsiino-komunikatsiini tekhnolohii navchannia informatychnykh dystsyplin maibutnikh inzheneriv-pedahohiv (Mobile information and communication technologies for learning informatics of future professionals in engineering pedagogy). Dissertation, Kryvyi Rih State Pedagogical University (2019)
8. Tkachuk, V.V., Shchokin, V.P., Tron, V.V.: The Model of Use of Mobile Information and Communication Technologies in Learning Computer Sciences to Future Professionals in Engineering Pedagogy. In: Kiv, A.E., Soloviev, V.N. (eds.) Proceedings of the 1st International Workshop on Augmented Reality in Education (AREdu 2018), Kryvyi Rih, Ukraine, October 2, 2018. CEUR Workshop Proceedings **2257**, 103–111. http://ceur-ws.org/Vol-2257/paper12.pdf (2018). Accessed 30 Nov 2018