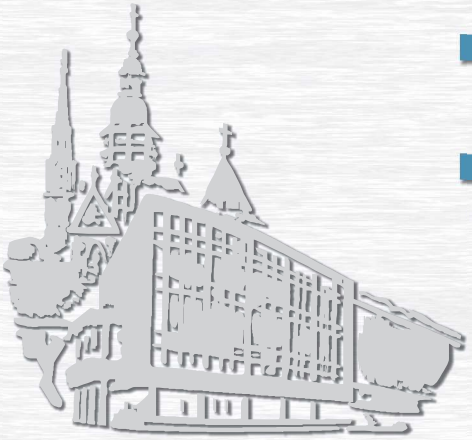

ICETA 2020

International Conference

18th International Conference on Emerging eLearning Technologies and Applications

IEEE Czechoslovakia Section

PROCEEDINGS

Information and Communication Technologies in Learning

November 12-13, 2020

Virtual Conference - Technical University of Košice

Slovak Republic

ICETA 2020

18th IEEE International Conference on Emerging eLearning Technologies and Applications

PROCEEDINGS

November 12 – 13, 2020

Virtual Conference – Technical University of Košice, Slovakia

Website: http://www.iceta.sk

ISBN: 978-0-7381-2366-0
IEEE Catalog number: CFP2038M-USB
Proceedings editor: František Jakab

TABLE OF CONTENTS

Web application for graph visualization purposes

A. Dudáš*, J. Janky** and J. Škrinárová***
*/*** Matej Bel University, Department of Computer Science, Banská Bystrica, Slovakia
** University of Žilina, Department of Computer Science, Žilina, Slovakia
*adam.dudas@umb.sk, **jakub1495@gmail.com, ***jarmila.skrinarova@umb.sk

***Abstract*—Graphs are one of several abstract topics presented in teaching of computer science at university. Graphs are part of various subjects such as optimization, compilers, networks, etc. Main objective of this paper is to present online application aimed at graph visualization. Implemented tool can be also used as a translator between various formats of graph description – adjacency matrix, graph6 format and diagram of graph. The greatest emphasis was placed on functionalities which are lacking in other online tools for similar purposes – mainly method of graphical representation of graph, possibility of various forms of input for graphical representation and possibility of multiple graphs on the input for simultaneous visualization of set of graphs. Presented application is first part of Moodle course project which is focused on graphs and parallel computing.**

I. Introduction

There is several abstract concepts presented during educational process in computer science and fields related to the computer science at universities. One of these abstract topics is graphs, which are part of subjects such as optimization, compilers, networks, etc. Other than these areas of computer science, graphs are important for visualization purposes in the process of education. There is wide variety of available tools for visualization of graphs but most of them lack important functionalities usable for easy to understand and effective graph visualization.

Main lacking functionalities of freely available graph visualization tools are:

- Symmetricity of diagram – unpaid tools for visualization of graphs rarely use algorithms for effective and symmetrical graph visualization. This leads to diagrams of graphs which are hard to read and therefore hard to interpret, understand and work with.
- Possibility of various input formats – even though adjacency matrix of graph is standard way to describe graph in computer, it is not effective. Matrices of graphs with high number of vertices are large and in most cases sparse. Most of online tools are able to create diagram of graph from adjacency matrix input into application by user (either typed or copied to the given field).
- Possibility of multiple graphs as input – none of online tools for graph visualization purposes are able to read input from file which can contain one or more graphs represented by adjacency matrix (or other format of graph description) and create diagrams of input set of graphs.

Main objective of this paper is to design and implement online tool for graph visualization purposes which emphasizes symmetrical diagrams of graphs, possibility of various input formats and possibility of multiple graphs as an input for application. This online application can also be used as a translator between chosen formats of graph description – adjacency matrix, grpah6 format and diagram of graph.

Our application for graph visualization purposes is first part of eLearning course focused on parallel and distributed computing in the area of graph problems.

The rest of the paper consists of three main sections – section III – V. The content of these sections is as follows.

Second section of this paper contains selection of past works related to the problem discussed in this paper.

In the section III, we introduce formats of graph description relevant for this paper – adjacency matrix, graph6 format and diagram of graph. We briefly describe these formats and discuss their importance in the relation to our application.

Section IV contains design and implementation of application which can be used for

- visualization of graphs,
- translation between formats of graph description.

In the section V, we test our application and compare results with other equivalent online tools usable for same purposes – graph visualization.

II. Related Works

Work presented in this paper is continuation of our previous research focused on graph coloring and online tools usable in university courses [1, 2].

Authors of [3, 4, 5] presented design and experiences with e-Learning courses in computer science similar to work presented in this paper, which is part of larger project (course focused on parallel computing of graph tasks).

Visualization of graphs and their diagrams is strongly based on work by authors of [6, 7] focused on graph drawing and permutation groups and extended by [8] which aims towards symmetrical visualization of diagram of graph.

Graph6 format of graph description is based on the work presented in [9].

III. Formats of Graph Descriptions

This section of the presented paper is focused on the formats of graph descriptions themselves. In the individual subsections we describe the formats necessary for our application.

Graph G is defined as a pair of sets V and E. Elements of set E (edges) are 2-element subsets of set V (vertices) [10].

In our case, graph G can be represented by:

- Adjacency matrix,
- Graph6 format,
- Diagram of graph.

A. Adjacecncy matrix

Matrix representation of graph is core representation of graph – simple format containing all necessary information about graph. Matrix representation is practical and often used in computer representation of graph and as an input for graph algorithms.

Let $G = (V, E)$ be graph. Adjacency matrix $M = (m_{ij})$ of this graph G (Fig. 1) is square matrix of $n \times n$ type, where $n = |V|$, indexes i, j represent vertices of graph G and its elements are defined as follows:

$$m_{ij} = \begin{cases} 1 \text{ in the case } \{i,j\} \in E \\ 0 \text{ otherwise} \end{cases} \quad (1)$$

$$M = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

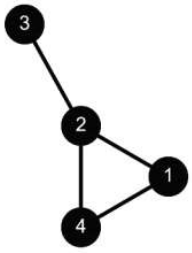

Figure 1. Adjacency matrix (left) of graph represented by diagram (right)

B. Graph6 format

Other interesting and light-on-memory description of graph is representation of adjacency matrix as bit vector transformed into the ASCII symbols. Each symbol represents decimal value in the range 63 - 126 (ASCII symbols with graphical representation).

Bit vector x of length k can be represented as follows:

- Add zero-valued bits to the end of vector x. Number of these bits needs to be such, that final vector is dividable by 6.
- Split this vector into the sets of 6 bits and convert these sets to decimal values.
- Add decimal value of 63 to all sets created in the step above.

Each of these resulting values is stored in one byte. It follows, that number of bytes needed is equal to $k/6$ [9].

Let $R(x)$ represent bit vector x as string of bytes. Let n be number of vertices of graph in the range of 0 - 68 719 476 735 (2^{36} - 1) and $N(n)$ is first element of vector which stored the information about number of vertices of graph, then [9]:

- $0 \leq n \leq 62$, set $N(n)$ as one byte $n + 63$;
- $63 \leq n \leq 258\,047$, set $N(n)$ as 4 bytes, 126 $R(x)$, where x is 18-bit form of n;
- $258\,048 \leq n \leq 68\,719\,476\,735$, set $N(n)$ as 8 bytes, 126 126 $R(x)$, where x is 36-bit form of n.

C. Graph diagram

Diagram of graph G is its graphical representation. Each graph has infinite number of diagrams. Graphical representation of graph G consists of point $D_V(u) \in R^2$ for each vertex $u \in V$, and curve $D_E(u, v) \in R^2$ for each edge $(u, v) \in E$ [8].

All diagrams of graph are isomorphic. Let $G = (V, E)$ and $G' = (V', E')$ be graphs (Fig. 2). Graph G is isomorphic to graph G', if there is mutual unambiguous bijective projection $f: V \leftrightarrow V'$ such, that for each pair of vertices $u, v \in V$ it holds [8]:

$$\{u, v\} \in E \text{ when } \{f(u), f(v)\} \in E' \quad (2)$$

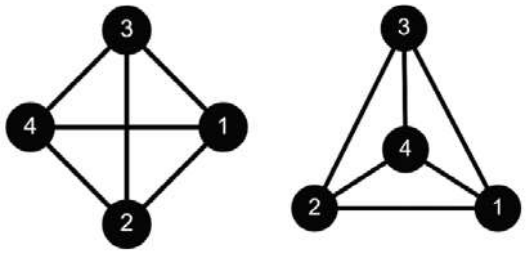

Figure 2. Two diagram visualizations representing same graph K_3

In the case, that graph project to itself, this projection is called automorphic projection (Fig. 3).

Automorphism of graph is representative of its symmetric properties. The more automorphic projections of the graph there is, the more symmetrical it can be drawn (represented as diagram). These properties are basis for one of drawing algorithms focused on finding symmetries in the graph and their projection [6].

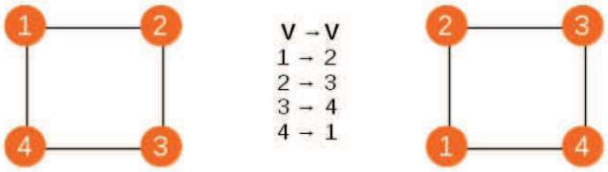

Figure 3. Automorphism of square graph

IV. Application for Visualizaiton of Graphs

This section is focused on design and implementation of online tool for graph visualizations purposes. Subsection A contains information about design of the application and description of main goals of the tool. Subsection B contains information about application implementation itself.

A. Design of visualization tool

As described in the introduction of this paper, there are some key functionalities that are missing in most online tools:

- Symmetricity of diagram,
- Possibility of various input formats,
- Possibility of multiple graphs as an input.

Our main focus is first of these points – symmetrical visualization of graphs. In the section III, we presented chosen formats of graph description – adjacency matrix, graph6 format and diagram of graph – which need to be implemented in our application.

As an input for the application, we can choose between inputting graphs to the dedicated field of the application, uploading file containing either graph6 or adjacency matrices of one or more graphs or creating diagram itself on canvas of the application.

Additional functionality which is not specified in these three points is possibility of downloading diagram of graph or graphs as .jpg file. This graph can also be modified by interacting with its vertices/edges on canvas of the application.

1) User interface of visualization application

From the point of view of user interface, we focus on basic concepts of user interfaces - readability, understandability and simplicity of usage.

Since we put emphasis on diagram of graph, canvas (interactive environment for diagrams of graph) should be largest part of user interface. Areas for inputting adjacency matrix and graph6 string are located on the bar on the left side of the application. This bar can be rolled to the left in order to maximize space for the canvas/diagram of graph.

In the case there is multiple graphs in the input file application creates new interactive canvas for each input graph.

Design for layout of user interface is presented in the figure 4:

- 1-AM – area for input in the form of adjacency matrix,
- 1-G6 – area for input in the form of graph6 format,
- 2 – interactive canvas for chosen graph,
- 3 – canvas bookmarks *1, ..., n* for *n* graphs in the input file.

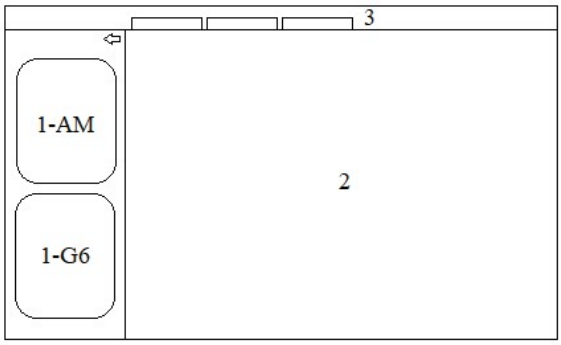

Figure 4. Design of user interface layout

2) Backend of visualization application

Functionalities of application can be divided into three groups of actions:

- Interactions with the components of user interface and subsequent reaction related to this interaction. In the case that user interacts with user interface (adds graph, works with diagram on canvas or alters adjacency matrix/graph6 format) application needs to react accordingly.
- Interaction with canvas and its modification. Backend of the application needs to contain methods for interaction with canvas itself – such as zooming in or out of canvas or tracking position of cursor on the canvas.
- Visualization of graph – drawing of the diagram. This part of the application contains algorithm for graph drawing [6] and data structures for representation of graph in the methods.

B. Impementation of visualization tool

The application was implemented as client side single page application, therefore backend of the application runs at user side o interaction. Tools used for implementation were:

- HTML was used for presentation of information on the basis of text – such as sections of the application, texts and buttons.
- CSS was used for the implementation of user interface and graphical parts of application.
- JavaScript for backend methods and functions. Since we chose to make the application client side whole program is sent by the website to the client browser.
- jQuery library was used for interaction between HTML and JavaScript.

Process of implementation brought some practical functionalities to the application, mainly:

- possibility of turning numbering of vertices on/off,
- button for refreshing of the diagram – since user can interact with vertices and edges of diagram, we implemented button usable to redraw graph with chosen drawing algorithm.

Figure 5 presents implementation of design for user interface created in the section IV, A.

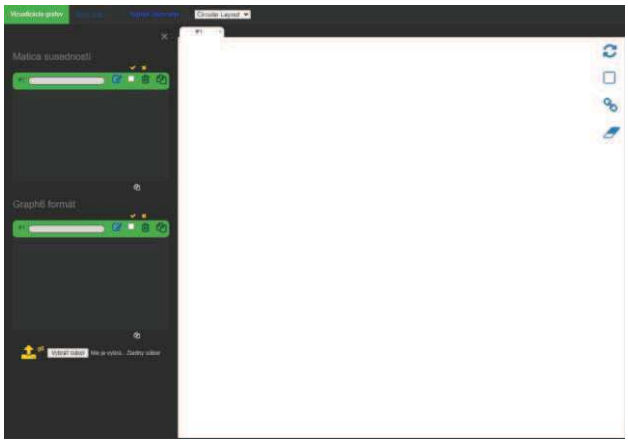

Figure 5. Implementation of design form the figure 4

Figures 6 and 7 contain example of visualization of simple graph. On the figure 7, we present same canvas as on the figure 6 with the left input panel rolled to the left.

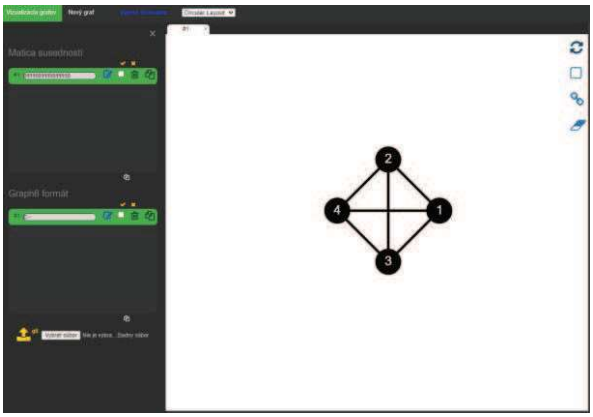

Figure 6. Example of visualization of simple graph

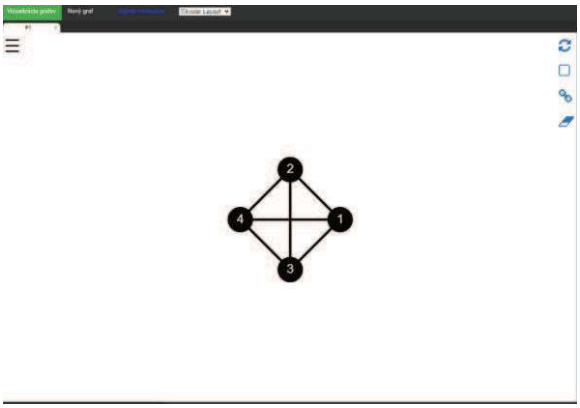

Figure 7. Example of rolled left panel

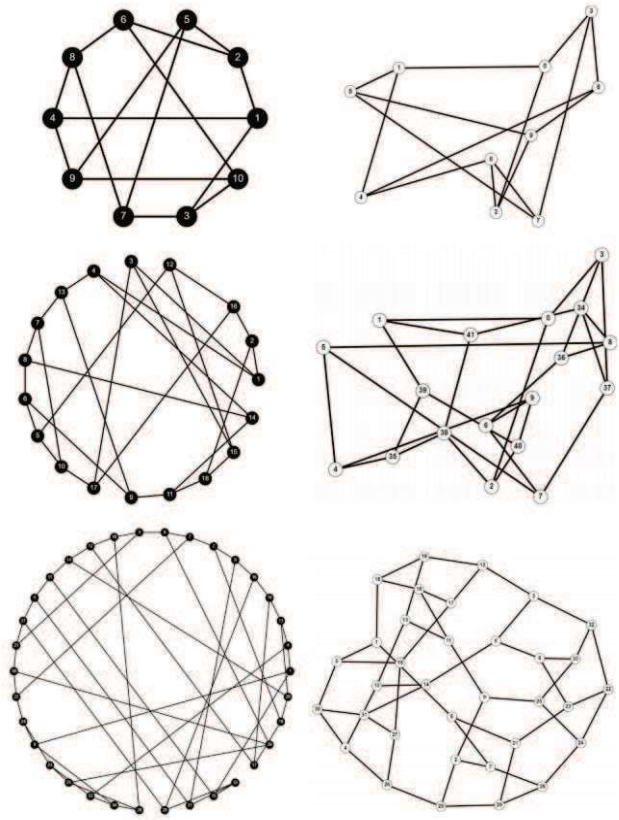

Figure 8. Comparison of diagrams of graphs using tool presented in this paper (left) and pre-existing online tool (right)

V. Testing of Application for Visualization of Graphs

For testing of designed application for visualization of graphs, we used group of graphs called snarks (cubic graphs that cannot be edge-colored with only three colors). These graphs do not contain bridges or articulations.

In the figure 8 we compare diagrams of three chosen snarks – diagrams on the left were created with the use of our application, diagrams on the right were created with online tool *graphonline*.

Since snarks presented in the figure 8 do not contain any bridges or articulations, we present additional graph designed on the basis of connecting several smaller graph into one large graph. This graph is presented on the figures 9 and 10. In the figure 9 the graph was visualized by online applications *graphonline* [11] and *csacademy* [12]. In the figure 10 we present same graph visualized with the use of our application.

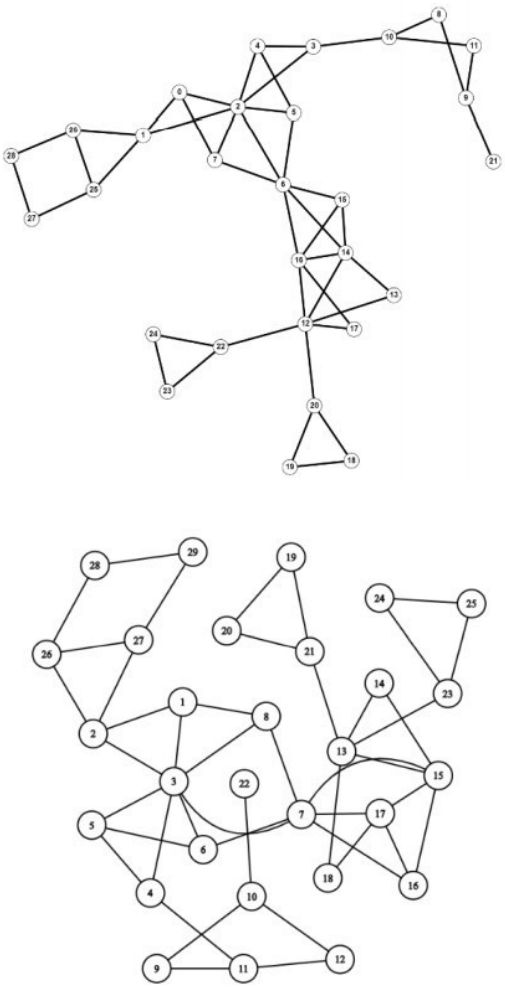

Figure 9. Diagram of graph containing bridges or articulations created with the application graphonline (top) and application csacademy (bottom)

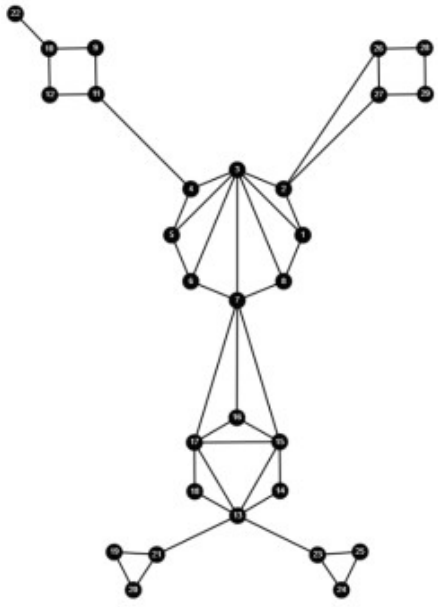

Figure 10. Diagram of graph containing bridges or articulations

As shown in the figure 10 diagram created with our application emphasizes readability of given diagram of graph and lowest possible crossing of edges. Even though diagrams in the figure 8 are not optimal (or standardized) readability of diagrams is much higher than readability of diagrams created with other online tools.

TABLE I. COMPARISON OF PROPERTIES OF ONLINE APPLICATIONS FOR GRAPHS

	Graph online	*CS academy*	*Presented application*
Adjacency matrix	✓	✓	✓
Graph6	×	×	✓
Programmability	×	×	✓
Downloadable diagram (.jpg)	×	×	✓
Drawing of sets of graphs	×	×	✓

In the table I, we compare properties of online graphing tools and presented application. Presented application contains three different formats of graph description, possibility of working with set of graphs, possibility of downloadable diagram in the .jpg format and is programmable – any graph algorithm can be added to the application.

VI. CONCLUSION

Aim of this paper was to present online application focused on graph visualization. Implemented tool can be also used as a translator between various formats of graph description – adjacency matrix, graph6 format and diagram of graph. The greatest emphasis was placed on functionalities which are lacking in other online tools for similar purposes – mainly method of graphical representation of graph, possibility of various forms of input for graphical representation and possibility of multiple graphs on the input for simultaneous visualization of set of graphs.

Future work contains additional optimization of the application for mobile devices, adding visualization of selected graph algorithms and new algorithms for graph drawing.

Since this application is part of course that is being created, future work contains also finalization of this course and all of its parts.

ACKNOWLEDGMENT

Computing was performed in the High Performance Computing Center of the Matej Bel University in Banská Bystrica using the HPC infrastructure acquired in project ITMS 26230120002 and 26210120002 (Slovak infrastructure for high-performance computing) supported by the Research & Development Operational Programme funded by the ERDF.

The research was partially supported by the grant of The Ministry of Education, Science, Research and Sport of the Slovak Republic – Implementation of new trends in computer science to the algorithmic thinking in subject of

Computer science in the secondary education, project number KEGA 018UMB-4/2020.

REFERENCES

[1] A. Dudáš, P. Voštinár, J. Škrinárová, J. Siláči. (2018) "Improved process of running tasks in the high performance computing system". Proceedings of 16th IEEE International Conference on Emerging eLearning Technologies and Applications. New Jersey: Institute of Electrical and Electronics Engineers. ISBN 978-1-5386-7912-8. pp. 133-140

[2] A. Dudáš, J. Škrinárová, E. Vesel. "Optimization design for parallel coloring of a set of graphs in the HighPerformance Computing". In: Proceedings of 2019 IEEE 15th International Scientific Conference on Informatics. pp 93-99. ISBN 978-1-7281-3178-8.

[3] Melicherčík, M., Siládi, V., Svítek, M., Huraj, L. "Spreading High Performance Computing Skills with E-Learning Support" (2018) ICETA 2018 - 16th IEEE International Conference on Emerging eLearning Technologies and Applications, pp. 361-366.

[4] Voštinár, P. "Interactive course for JavaScript in LMS Moodle" (2019) ICETA 2019 - 17th IEEE International Conference on Emerging eLearning Technologies and Applications, pp. 810-815.

[5] Pilarčíková, K., Rusnák, P., Rabčan, J., Kostolný, J. "User experience in the development of the education system" (2019) ICETA 2019 - 17th IEEE International Conference on Emerging eLearning Technologies and Applications, Proceedings, pp. 626-632.

[6] Tutte, W. T. How to draw a graph. Proc. London Math. Society, 1963.

[7] Wielandt, H. Finite Permutation Groups. Academic Press, 1964.

[8] Eades, P., Hong, Seok-H. Symmetric Graph Drawing. Kao MY. (eds) Encyclopedia of Algorithms. Springer, New York, NY, 2016.

[9] McKay, B., Piperno, A. Nauty and Traces User`s Guide. Version 2.6. Available online at: https://users.cecs.anu.edu.au/~bdm/nauty/nug26.pdf >.

[10] R Diestel, Graph theory. 5th edition. Springer - Verlag, Heidelberg, 2016.

[11] "Working with graphs online," [Online]. Available: https://graphonline.ru/.

[12] CS Academy. [Online]. Available: https://csacademy.com/app/graph_editor/

Organizer

Technically Sponsored by

Special guarantee

Partners

Media Partner

ISBN 978 - 0 - 7381 - 2366 - 0