

poate manifesta profesorul și / sau elevul, utilizând instrumentele performante și multiple ale unui astfel de MD.

Bibliografie

1. Balmuș, Nicloae, Borș, Alexandru, Burlacu, Natalia, Chirilov, Vaisile, Activități valorificate în manualului digital de educație muzicală pentru clasa a II-a, în: Conferința de totalizare a muncii științifice și științifico-didactice a corpului profesoral-didactic pentru anul 2015, Chișinău, 2016, volumul I, pp. 185-190.
2. Burlacu, Natalia, Balmuș, Nicolae, Vacarciuc, Mariana, Role and configuration of digital textbook for musical education from the perspective of initial training for primary school teachers, în: The-12-th INTERNATIONAL CONFERENCE ON VIRTUAL LEARNING, October 28, 2017. Editura Universității din București, 2017, pp. 398-405.
3. Burlacu, Natalia, Balmuș, Nicolae, Digital handbook of music: development principles, in: The-11-th INTERNATIONAL CONFERENCE ON VIRTUAL LEARNING, October 29, 2016. Editura Universității din București, 2016.

METODOLOGIA APLICĂRII SISTEMULUI ELECTRONIC „MAPLE” ÎN STUDIAREA CURSULUI „TEORIA GRAFURILOR”

Marina BOSTAN, lector universitar

Summary

Many problems computers are used to solve involve representing relationships between objects, places, or concepts. Because graphs can be either directed or undirected, they are flexible method of showing connections. Maple has a large collection of commands related to Graph Theory, all found in the GraphTheory package. The GraphTheory package is presently intended for teaching and research usage an dis designed to handle graphs of up to 1000 vertices. Some of standard operations for graphs we describe in this paper with examples.

Originile teoriei grafurilor se regăsesc în rezolvarea unor probleme distractive din matematică, care au atras atenția unor matematicieni cum ar fi: Euler, Hamilton, Cayley, Sylvester, Birkoff. „Data nașterii” a teoriei grafurilor poate fi considerată anul 1736, când elvețianul Leonard Euler, fizician-matematician, a publicat un articol în care a clarificat problema „Celor 7 poduri” și a prezentat metode pentru rezolvarea altor probleme de același tip. Tradițional, metodele teoriei grafurilor sunt folosite în cercetările din domeniile științelor exacte: informatică, fizică, electrotehnică, chimie ș. a. Datorită investigațiilor teoretice profunde, efectuate în ultimele decenii, precum și dezvoltării aspectului algoritmic, care a avut loc odată cu apariția tehnicii de calcul moderne, astăzi sunt cunoscuți un șir de algoritmi eficienți, folosiți în soluționarea problemelor aplicative din domeniul informaticii, economiei, sociologiei etc.

Grafurile oferă cele mai potrivite metode de a exprima relațiile dintre obiecte, astfel aria lor de utilizare practică este foarte vastă, de la economie la psihologie socială. În scopul familiarizării cu diversele domenii de aplicare, utilizăm probleme a căror rezolvare implică noțiuni legate de teoria grafurilor [5].

Scopul urmărit în cadrul predării acestei discipline este de a aduce la cunoștință studenților, fundamentele teoretice ale teoriei grafurilor, precum și aspectul algoritmic al acesteia, implementarea deprinderilor de a folosi metode și algoritmi eficienți în soluționarea problemelor cu caracter teoretic și practic din diverse domenii. În mod special se va accentua rolul Teoriei grafurilor în dezvoltarea informaticii, în elaborarea metodelor eficiente și a algoritmilor respectivi pentru soluționarea problemelor practice [6].

Maple este un sistem electronic de calcule matematice, dezvoltat de firma Maplesoft (<http://maplesoft.com>)[4], care poate fi utilizat și la soluționarea majorității problemelor din Teoria Grafurilor. Pachetele din sistemul Maple sunt colecții de funcții care permit efectuarea calculelor specifice. Apelarea acestora se face cu ajutorul comenzii *with* (nume_pachet). Pachetul **GraphTheory**

reprezintă o colecție de comenzi destinate pentru crearea, construirea, manipularea și testarea grafurilor și determinarea proprietăților acestora.

Funcțiile de bază, care se utilizează la rezolvarea problemelor cu grafuri sunt: **Graph()** – crearea unui graf neorientat; **DGraph()** – crearea unui graf orientat; **AddEdge()** – adăugarea muchiei(-iilor) într-un graf; **AddVertex()** – adăugarea nodului(-rilor) într-un graf; **DrawGraph()** – desenarea (construirea, prezentarea) unui graf; **AdjacencyMatrix()** – returnează matrice de adiacență; **IncidenceMatrix()** – returnează matrice de incidență; **WeightMatrix()** – returnează matricea ponderată pentru garful dat și altele [4].

Prezentăm careva exemple de utilizarea pachetului **GraphTheory** din sistemul Maple 18.

Exemplul 1. Grafuri neorientate

Apelăm pachetul **GraphTheory**: *with(GraphTheory)*

Definim graful neorientat cu 5 noduri și 7 muchii:

$G := \text{Graph}(5, \text{undirected}, \{[1, 2], [1, 3], [1, 4], [2, 3], [2, 5], [3, 4], [3, 5]\})$

Graph 6: an undirected unweighted graph with 5 vertices and 7 edge(s)

Construim graful definit: *DrawGraph(G)*

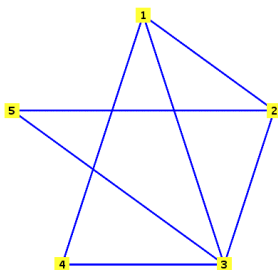


Fig. 1 Graful neorientat cu 5 noduri și 7 muchii

Afișăm lista muchiilor: *Edges(G)*

$\{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 5\}, \{3, 4\}, \{3, 5\}\}$

Determinăm gradul nodurilor 1, 3 și 5:

$Degree(G, 1) 3$

$Degree(G, 3) 4$

$Degree(G, 5) 2$

Adaugăm nodul 6: $G := AddVertex(G, 6)$

Graph 8: an undirected unweighted graph with 6 vertices and 8 edge(s)

Adaugăm muchiile (2,6) și (4,6): $AddEdge(G, \{(2, 6), \{4, 6\})\}$

Graph 8: an undirected unweighted graph with 6 vertices and 11 edge(s)

Construim graful obținut: $DrawGraph(G)$

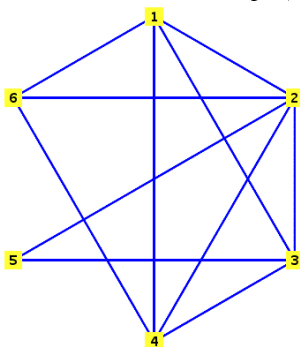


Fig. 2 Graful neorientat cu 6 noduri și 11 muchii

Eliminăm muchiile (1, 4) și

(3,5): $G := DeleteEdge(G, \{1, 4\}, \{3, 5\})$

Graph 8: an undirected unweighted graph with 6 vertices and 10 edge(s)

Construim graful obținut: $DrawGraph(G)$

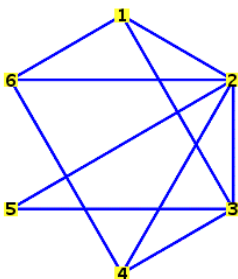


Fig. 3 Graful neorientat cu 6 noduri și 9 muchii

Afișăm matricea de adiacență: $A := AdjacencyMatrix(G)$

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Generăm un subgraf din mulțimea nodurilor [1, 2, 3, 4, 6]:

$E := \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 4\}, \{2, 6\}, \{3, 4\}, \{4, 6\}, \{6, 1\}\}$
 $\{\{1, 2\}, \{1, 3\}, \{1, 6\}, \{2, 3\}, \{2, 4\}, \{2, 6\}, \{3, 4\}, \{4, 6\}\}$

$G1 := \text{Subgraph}(G, E)$

Graph 12: an undirected unweighted graph with 5 vertices and 8 edge(s)

Construim subgraful G1 din graful G :

HighlightSubgraph(G, G1)

DrawGraph(G)

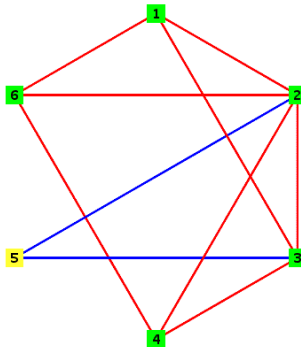


Fig. 4 Subgraful G1 (roșu) din graful G

Generăm graful complet din 4 noduri: $H := \text{CompleteGraph}(4)$

$H := \text{Graph 10: an undirected unweighted graph with 4 vertices and 6 edge(s)}$

Construim graful obținut: *DrawGraph(H)*

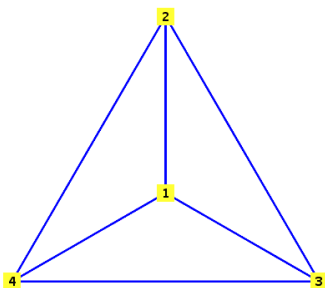


Fig. 5 Graful complet din 4 noduri

Exemplul 2. Drumuri minime în graf

Problema: Aplicând algoritmul lui Dijkstra și Roy-Floyd [2, p. 56] să se determine drumurile de cost minim de la nodul 1 la toate celelalte noduri.

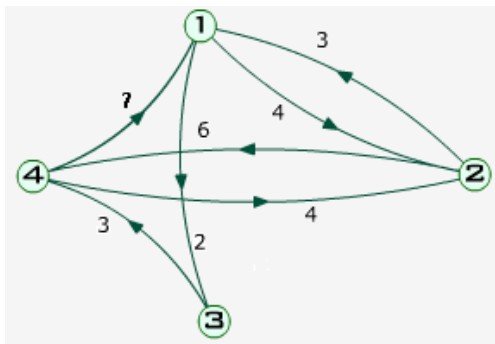


Fig. 6 Graful orientat cu 4 noduri și 7 arce

Apelăm pachetul **GraphTheory**: `with(GraphTheory)`

Definim graful cu 4 noduri și costurile date:

```
G4 := Digraph( ([[1, 2], 4], [[1, 3], 2], [[2, 1], 3], [[2, 4], 6],
  [[3, 4], 3], [[4, 1], 7], [[4, 2], 4]])
```

Graph 1: a directed weighted graph with 4 vertices and 7 arc(s)

Construim graful definit: $DrawGraph(G4)$

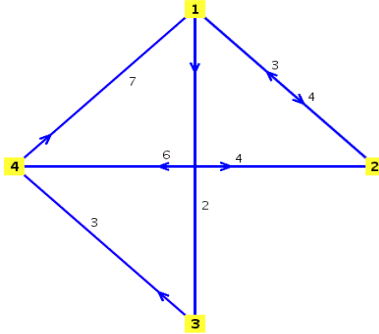


Fig. 7 Graful orientat $G4$

Apelăm procedura **DijkstraAlgorithm** pentru aflarea drumurilor de cost minim de la vârful 1 până la celelalte:

$DijkstraAlgorithm(G4, 1)$

$[[[1], 0], [[1, 2], 4], [[1, 3], 2], [[1, 3, 4], 5]]$

Observăm că obținem nu numai costul minim pentru drumurile de la vârful 1 până la celelalte, dar și traseele acestora (noduri intermediare, prin care ajungem de la nodul 1 până la nodul final).

Apelăm procedura **Distance ($G4, s, t$)** pentru a returna numărul de muchii din cel mai scurt drum de la nodul s la nodul t .

$Distance(G4, 1, 2) \quad 1$

Prin analogie determinăm numărul de muchii din cel mai scurt drum de la nodul 1 până la nodurile 3 și 4.

$Distance(G4, 1, 3) \quad 1$

$Distance(G4, 1, 4) \quad 2$

Pentru utilizarea algoritmului Roy-Floyd se aplică comanda **AllPairsDistance** pentru a obține valorile costurilor minime a tuturor drumurilor:

$AllPairsDistance(G4)$

$$\begin{bmatrix} 0 & 4 & 2 & 5 \\ 3 & 0 & 5 & 6 \\ 10 & 7 & 0 & 3 \\ 7 & 4 & 9 & 0 \end{bmatrix}$$

Se construiește graful pentru nouă matricea obținută:

DrawGraph(G4);

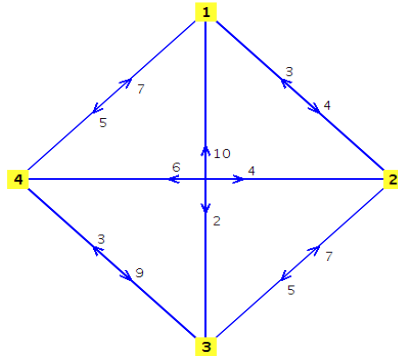


Fig. 8 Graful orientat $G4$ îmbunătățit

La implementarea TIC în procesul de studii trebuie să atragem atenția la următoarele aspecte: vizibilitatea, interactivitatea și vizualizarea etapelor de rezolvare a problemelor. Acestor criterii le corespunde sistemul electronic Maple, din acest considerent el poate fi aplicat în cadrul studierii și predării Teoriei Grafurilor.

Bibliografie

1. Corlat, S., Gremalschi, A., Grafuri: Metodologia predării în cadrul instruirii de performanță la disciplinele Matematică & Informatică, Univ. de Stat din Tiraspol, Chișinău, 2014.
2. Кирсанов, М. Н., Графы в Maple, Физматлит, М., 2007.
3. Кристофидес, Н., Теория графов. Алгоритмический подход, Мир, М., 1978.
4. <https://www.scribd.com/doc/249423721/Methodologie-Teoria-Grafurilor>.
5. <https://www.maplesoft.com/support/help/index.aspx>.

6. Grafuri neorientate, <http://www.rasfoiesc.com/educatie/matematica/Grafuri-neorientate43.php>.

7. Algoritmica grafurilor, <http://fmi.usm.md/node/171>.

UNELE METODE DE PROTEJARE A OCHILOR ÎN TIMPUL LUCRULUI LA CALCULATOR

Olga CHIRCHINA, dr., lector universitar

Zinaida GHILAN, dr., conf.

Summary

In this paper we presented some methods that will reduce the negative impact of the computer on human eyesight with the help of simple eye exercises, which could be done while working.

Calculatorul a devenit, în ultimul timp, un adevărat „partener” de lucru al omului în multe domenii de activitate. Totuși, există un șir de factori negativi, care ar trebui să se ia în considerație când se lucrează la calculator mai mult de 2-3 ore pe zi: mușchii ochilor obolesc de tensiunea constantă a calculatorului, care provoacă dureri oculare și spasme [1].

Primele semne de presiune asupra ochilor se manifestă prin uscăciune, inflamație, roșeață, sensibilitate crescută la lumină, dureri în gât, umeri, spate, cap, oboseală și nervozitate – toate acestea pot duce la o slăbire drastică a vederii. Conform statisticilor s-a estimat că fiecare al patrulea elev se confruntă cu probleme deficiente de vedere, iar până în anul 2020 o treime din populație va fi afectată de miopie, ce rezultă că fiecare al treilea copil va fi purtător de ochelari [2].

Să analizăm mai detaliat cauzele tulburărilor vizuale.

În primul rând, este dovedit faptul că în timpul lucrului scade clipirea ochilor la ecranului monitorului. În momentul când se lucrează la monitor, pleoapele ochilor sunt închise de câteva ori mai puțin, pentru a vedea imaginea în mod natural. În situația normală, ochii clipesc de 20-25 de ori pe minut, în timp când se lucrează la