

Lucrare practică de atestare a cunoștințelor la informatică

Soft Educațional

Profesor îndrumător:

Martin Elena Liliana

Elev:

Manea Andrei Alexandru

Mai 2017

Cerințe și specificații

Să se realizeze o pagină web de tip soft educațional care să permită studierea capitolului „**Drumuri minime de sursă unică în graf**”, din domeniul informaticii. Pagina va permite utilizatorului să acceseze noțiuni teoretice, exerciții propuse, exerciții rezolvate, precum și verificarea gradului de însușire a cunoștințelor prin intermediul unui test de tip grilă cu butoane radio, realizat în limbajul **PHP**. Pentru a fi util unui profesor la clasă, rezultatele se vor transfera automat într-o bază de date de tip **SQL**. Totodată lucrarea va trebui să conțină un program în C++ care să rezolve o cerință specifică temei. Se vor preciza explicit sursele bibliografice precum și autorul cu date de contact.

Tema lucrării:

Drumuri minime de sursă unică în graf

Soft educațional

Cuprins

Capitolul 1 - Prezentarea lucrării	1
1.1 Introducere	1
1.2 Limbajul HTML.....	2
1.3 Limbajul PHP.....	3
1.4 Limbajul CSS.....	4
1.5 Limbajul MySQL.....	5
Capitolul 2 - Schema conceptuală a lucrării	6
Capitolul 3 - Testare în timp real	7
Capitolul 4 - Instalarea aplicației	20
4.1 Resurse minimale.....	20
4.2 Instalare – dezinstalare.....	20
Capitolul 5 - Bibliografie	21
Capitolul 6 – Anexe	22
□ Fragment din pagina „Home”:	22
□ Fragment din pagina „Test”:	23

Capitolul 1 - Prezentarea lucrării

1.1 Introducere

Softul creat se dorește a fi un instrument educațional proiectat pentru însușirea și evaluarea cunoștințelor privind un capitol important al din „Teoria Grafurilor” , cu ajutorul computerului. Lucrarea conține elemente teoretice: terminologie legată de grafuri, arbori, structuri de date necesare pentru a lucra cu astfel de „concepte” și algoritmi importanți din cadrul acestui capitol precum Algoritmul lui Dijkstra sau Bellman-Ford, fiecare însoțit de demonstrația teoretică a acestuia. În plus, este disponibil un test pentru a verifica cunoștințele dobândite după parcurgerea site-ului.

Interfața lucrării este prietenoasă, putând fi folosită cu ușurință de către orice persoană, indiferent de calificare sau dacă a mai folosit sau nu un asemenea soft educațional.

Utilizatorul poate accesa fiecare secțiune unde va găsi informații utile pentru completarea cunoștințelor sale. Când va face click pe un link, acesta îl va trimite la secțiunea respectivă unde poate căuta informațiile dorite, fiecare informație fiind însoțită de o imagine corespunzătoare care să reflecte ideile descrise și să faciliteze înțelegerea informațiilor prezentate.

Proiectul mai conține, de asemenea, încă o secțiune destinată evaluării cunoștințelor dobândite prin parcurgerea lucrării. Modulul de testare conține noua întrebări selectate din întregul proiect, utilizatorul primind la final procentajul obținut în urma răspunsurilor date. Acesta este realizat în **HTML** și **PHP** și consta într-un fișiere complex ce îmbină structuri **HTML** pentru setarea parțială a aspectului testului și structuri **PHP** ce afișează și procesează întrebările.

În realizarea acestui proiect s-au utilizat: limbajul **HTML**, limbajul **CSS** și limbajul **PHP** însoțit de instrucțiuni din limbajul **MySQL** pentru a păstra rezultatele într-o bază de date.

1.2 Limbajul HTML

HTML (Hypertext Markup Language) este un limbaj creat în scopul de a descrie, în mod text, formatul paginilor Web; fișierele create în acest limbaj vor fi interpretate de navigatoare, care vor afișa paginile în forma dorită (cu texte formate, liste, tabele, formule, imagini, hiperlegături, obiecte multimedia etc.). **HTML** a apărut ca o aplicație ISO standard (aparține standardului SGML - Standard Generalized Markup Language, specializat pentru hipertext și adaptat la Web).

Așa cum se poate deduce din numele limbajului, **HTML** descrie caracteristicile de format ale elementelor incluse prin procedee de marcare. Acestea pot fi asemănate intuitiv cu marcasele folosite în tipografie pentru a indica scrierea unui text cu un anumit tip de caractere. Fiecare element va fi introdus între două marcase ("tags", în limba engleză) - de început și sfârșit - (uzual) de forma `<marcaj> ... </marcaj>`. Caracterele speciale de delimitare a marcajelor "`<`", "`>`" permit deosebirea acestora de textul propriu-zis. De exemplu, pentru textele aldine (îngroșate), marcajul de început este ``, iar marcajul de sfârșit este ``.

Procesele de standardizare și de includere a comenzilor de marcare în fișierele **HTML** permit navigatoarelor să citească și să formateze paginile Web, lucru foarte important în condițiile în care ele conțin nu numai texte alb-negru, ci și culori, imagini, hiperlegături, diverse obiecte. Practic, marcasele **HTML** asigură controlul asupra modului de afișare a obiectelor corespunzătoare în cadrul programelor de vizualizare a documentelor **HTML** - navigatoarele.

Limbajul **HTML** a evoluat în versiuni succesive, odată cu evoluția protocolului HTTP și a programelor de navigare. Astfel, **HTML 1.0** era compatibil cu Mosaic, primul program de navigare, dar după apariția unor navigatoare noi, a fost necesară introducerea unui standard oficial Internet pentru construirea paginilor (**HTML 2.0**) și extinderea sa cu noi facilități: formule matematice, tabele, moduri avansate de descriere a organizării paginilor (începând cu **HTML 3.0**).

Standardizarea oficială a limbajului **HTML** a fost realizată de consorțiul WWW și dezvoltată de diverși producători de soft (unii dintre aceștia urmăresc chiar promovarea navigatoarelor proprii prin introducerea unor particularități în formatele oficiale). Paginile **HTML** se pot crea cu orice editor de texte de către utilizatorii care cunosc limbajul **HTML** sau, mai simplu, se pot utiliza editoare speciale, în care obiectele se introduc interactiv, iar codul **HTML** se generează automat.

HTML 5 are niște avantaje față de celelalte versiuni:

- Drag and Drop
- Fișiere Audio și Video
- Aplicații Web ce pot fi rulate offline
- Istoric
- Geolocalizare (GPS)
- Mesaje Web

Începutul unui astfel de document va conține marcajul : `<!DOCTYPE HTML>`.

1.3 Limbajul PHP

PHP (Hypertext Preprocessor), cunoscut în versiunile mai vechi și sub numele de **PHP/FI** (Personal Homepage/Form Interpreter), inițial a fost gândit a fi o simplă aplicație CGI pentru interpretarea formularelor definite prin **HTML** și procesate de un program scris într-un limbaj Perl, script shell, executat pe server. În cazul interfeței CGI era necesară permisiunea de a rula programe pe server, ceea ce ducea la lacune în securitate și în plus la disocierea de documentul **HTML** a programului care procesa datele.

PHP (în versiunea 4.0) reprezintă un pachet puternic care oferă un limbaj de programare accesibil din cadrul fișierelor **HTML**, limbaj asemănător cu Perl sau C, plus suport pentru manipularea bazelor de date într-un dialect **SQL** (**dBase**, **Informix**, **MySQL**, **mSQL**, **Oracle**, **PostgreSQL**, **Solid**, **Sybase**, **ODBC** etc.) și acces la sisteme hipermedia precum Hyperwave. De asemenea, **PHP** suportă încărcarea fișierelor de pe calculatorul client: upload (standard propus de E. Nebel și L. Masinter de la Xerox, descris în RFC 1867) și oferă suport pentru cookies (mecanism de stocare a datelor în navigatorul client pentru identificarea utilizatorilor, propus de Netscape).

Versiunea 5.0, a modificat anumite funcții ce țin de interacțiunea cu o bază de date: de exemplu prefixul **mysql** a devenit **mysqli**.

Această aplicație este disponibilă gratuit pe Internet, pentru medii Unix și pentru medii Windows (inclusiv sursele), integrându-se în popularul sever Apache.

Istoria **PHP**-ului începe la sfârșitul anului 1994, când Rasmus Lerdorf dezvoltă prima versiune, ca proiect personal. **PHP**-ul este făcut public în debutul anului 1995 sub denumirea de Personal Home Page Tools, fiind considerat un analizor simplist care interpretează câteva macrouri ce puteau fi incluse în cadrul documentelor **HTML**, permițând contorizarea accesului la paginile Web sau accesarea unei cărți de oaspeți (guestbook). Analizorul a fost rescris la mijlocul aceluiași an și denumit **PHP/FI 2.0**, unde FI era o altă aplicație scrisă de Rasmus Lenford, un interpretor de formulare **HTML**. A fost adăugat și suportul pentru bazele de date **SQL** și astfel **PHP/FI** a început să aibă succes, fiind disponibil gratuit pe Web. Se estimează că la sfârșitul lui 1996 cel puțin 15 mii de site-uri Web utilizau **PHP/FI**, iar în anul 1997 numărul acestora era de 50 de mii.

Codul **PHP** este inclus între tag-urile speciale de început și de sfârșit care permit utilizatorului să treacă din "modul **HTML**" în "modul **PHP**".

Față de alte limbaje script, precum **Javascript**, la **PHP** codul se execută pe server. Dacă pe serverul Web se află un script similar, clientul nu va primi decât rezultatul execuției scriptului, fără a avea nici o posibilitate de acces la codul care a produs rezultatul. Vă puteți configura serverul de Web să prelucreze (analizeze) toate fișierele **HTML** ca fișiere **PHP**. Astfel nu există nici un mijloc de a distinge paginile care sunt produse dinamic de paginile statice.

Limbajul **PHP** posedă aceleași funcții ca alte limbaje, permițând să se scrie scripturi CGI, să colecteze date și să genereze dinamic pagini web sau să trimită și să primească cookies.

Marea calitate și cel mai mare avantaj al limbajului **PHP** este suportul cu un număr mare de baze de date. A realiza o pagină web dinamic cu o bază de date este extrem de simplu.

1.4 Limbajul CSS

CSS este un acronim provenind din Cascading Style Sheets, care înseamnă "foi de stil în cascadă". În documentele W3C, **CSS** nu e definit ca un nou limbaj, ci ca un mecanism care permite formatarea documentului **HTML**.

CSS-ul nu exclude **HTML**-ul din pagina web. Nu se poate realiza o pagină web folosind numai **CSS**, care a fost proiectat astfel încât să conlucreze cu **HTML**-ul. Proiectanții **CSS**-ului au urmărit îndeosebi separarea între:

- conținutul paginii (textul destinat vizitatorului și imaginile din pagină);
- codul-sursă.

Folosind **CSS**, se ajunge la:

- un control mai fin asupra paginii web;
- scăderea dimensiunii în octeți a paginii web, atunci când codul **CSS** e conținut într-un fișier extern;
- mai mare comoditate: modificând fișierul **CSS** extern, modificăm simultan toate paginile web în care acesta e inclus;
- efecte mai sofisticate decât cele produse de codul **HTML**: suprapunerea unei imagini peste altă imagine, a unui text peste alt text, impresia de relief, efectul hover, afișarea unor fonturi mai mari decât h1 etc.

Adăugând cod **JavaScript**, se obțin efecte și mai sofisticate, chiar de animație. **CSS** + **JavaScript** = **DHTML** (Dynamic **HTML**). Aceste efecte spectaculoase justifică titulatura de "artiști **CSS**" sau "artizani **CSS**" acordată unor creatori de pagini web.

1.5 Limbajul MySQL

În ultimii ani, utilizarea bazelor de date pe Internet a luat o amploare deosebită. Există o mulțime de aplicații, extrem de utile (care le utilizează), cum ar fi:

- ✓ **Aplicații de contorizare**, aplicații prin care cei care au creat un site o posibilitatea să-și contorizeze numărul de vizitatori ai site-ului respectiv sau, mai mult, contorizarea se poate extinde la nivel de pagină afișată. Pentru a evita cu un vizitator, care a deschis de mai multe ori pagina respectivă, să nu fie contorizat de mai multe ori, se va reține adresa **IP** a acestuia.
- ✓ **Comerț electronic**, aplicații prin care anumite firme își promovează și își vând produsele de care dispun. În acest caz, bazele de date vor reține, pe de o parte, informații despre produse, iar pe de altă parte, comenzile clienților. Avantajele comerțului electronic sunt uriașe pentru că nu implică cheltuieli pentru spațiile de desfacere (magazine) și teoretic, oferta se adresează clienților din toată lumea.
- ✓ **Votul electronic**, aplicații prin care se poate afla părerea vizitatorilor site-ului respectiv referitoare la tema supusă votului. Și aici este important ca un vizitator să nu voteze de mai multe ori, motiv pentru care baza de date reține adresa **IP** a vizitatorului.
- ✓ **Aplicații de comunicare** – aplicații prin care diverși vizitatori ai site-ului respectiv dezbate o anumită problemă, supusă discuției. În astfel de cazuri, bazele de date vor reține informații despre cei care scriu (de multe ori aceștia, mai întâi, trebuie să se înscrie în baza de date a site-ului) și mesajele acestora.

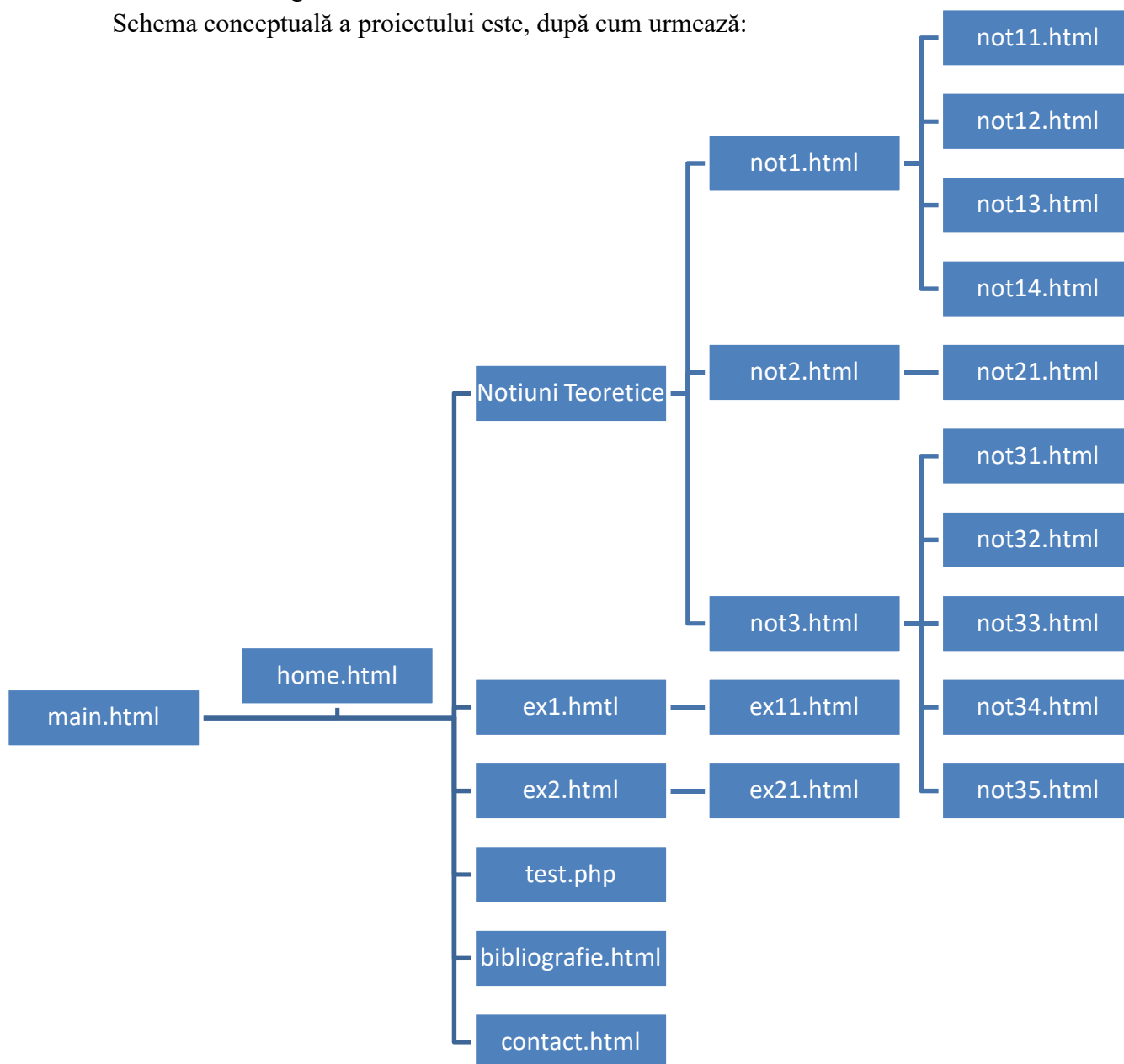
Ce este SQL? În practica creării și utilizării bazelor de date relaționale s-a impus necesitatea existenței unui limbaj standard care permite efectuarea acestor operații. Astfel, a apărut **SQL** – Structured Query Language. Limbajul este supervizat de comisia de standardizare **ANSI** (American National Standards Institute), motiv pentru care se mai numește și ANSI SQL. SQL nu este un limbaj de firmă, el este implementat de o mulțime de SGBD-uri, de aceasta dată consacrate, cum ar fi: **Microsoft Access, Oracle, Microsoft SQL Server** și bineînțeles, **MySQL**.

Ce este MySQL ? **MySQL** este un limbaj specializat pentru gestiunea bazelor de date relaționale pe Internet. Așa cum s-a mai spus, are la bază limbajul **SQL**. **MySQL** gestionează bazele de date cu ajutorul limbajului **PHP**, dar și cu alte limbaje, de exemplu **Java**.

Capitolul 2 - Schema conceptuală a lucrării

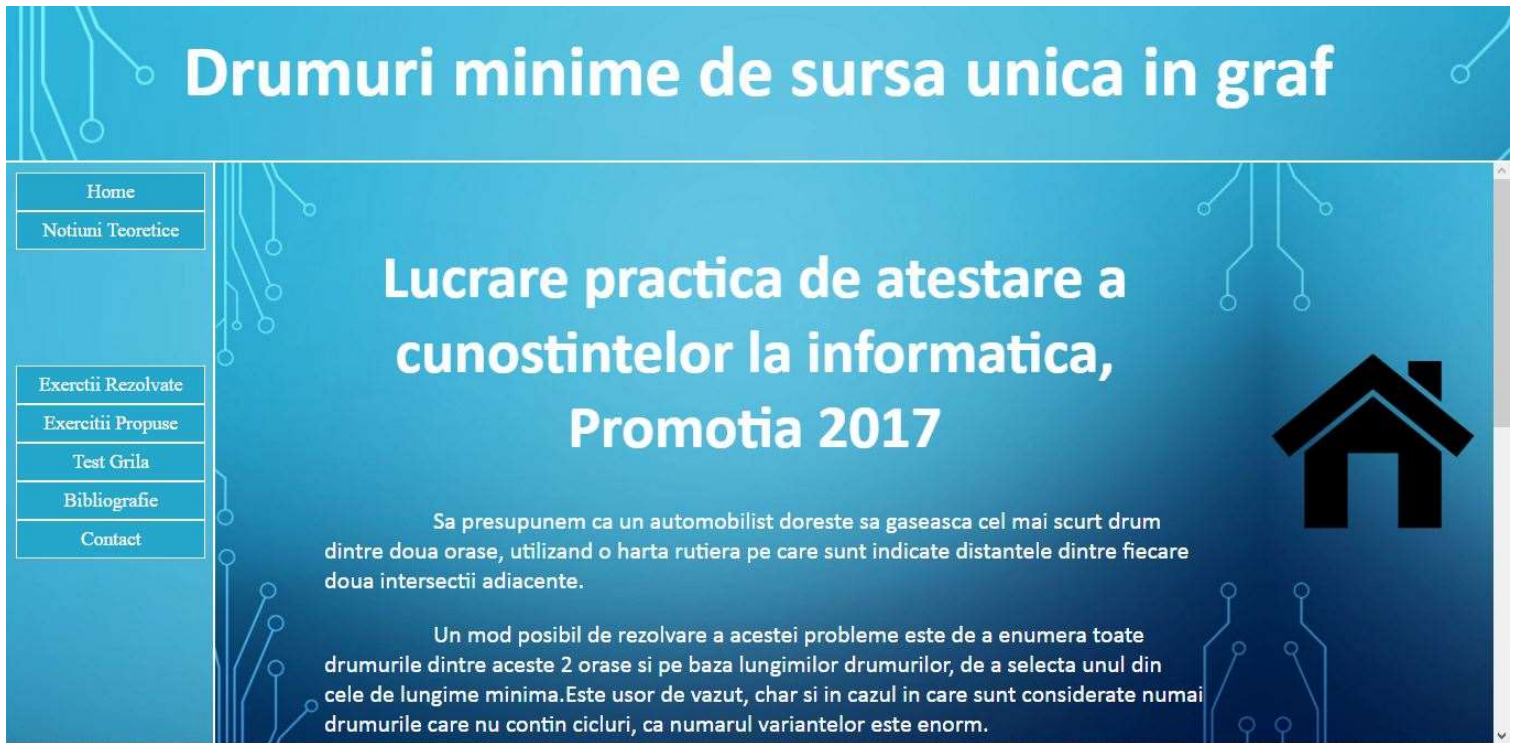
Pagina principală a proiectului se numește **main.html**. Paginile fiind **html** și **php**, proiectul trebuie copiat în folderul “www” al aplicației WampServer. Sunt două modalități în care se poate intra în pagina principală: fie deschizând browserul de internet (Internet Explorer, Opera, Mozilla, Google Chrome etc.) și scriind întreaga cale până la fișierul **index.php**, fie pur și simplu scriind localhost și apăsând enter, iar apoi navigând cu ajutorul cursorului până la folderul dorit. Asupra acestei adrese, se adăugă „/main.html”.

Schema conceptuală a proiectului este, după cum urmează:



Capitolul 3 - Testare în timp real

La deschiderea documentului **main.html** , se află 3 frame-uri, unul deasupra cu titlul (**topic.html**), unul în stânga cu cuprinsul (**cuprins2.html**) și unul care ocupă cel mai mare spațiu în care va apărea fiecare document selectat (**desk.html**) prin intermediul comenzii „target”. Implicit, în „desk” va apărea **home.html**.



Din cadrul cuprinsului, se pot accesa documentele care conțin doar o cifră sau niciuna (**home.html**, **not1.html**, **not2.html**, **not3.html**, **ex1.html**, **ex2.html**, **test.php**, **bibliografie.html**, **contact.html**).


Așezând cursorul pe caseta „Noțiuni introductive”, se poate accesa una dintre cele 3 pagini : **not1.html** (Noțiuni teoretice) , **not2.html** (Structuri de Date), **not3.html**(Algoritmi).

Drumuri minime de sursa unica in graf

[Home](#)
[Noțiuni Teoretice](#)

[Exercitii Rezolvate](#)
[Exercitii Propuse](#)
[Test Grila](#)
[Bibliografie](#)
[Contact](#)

Lucrare practica de atestare a cunostintelor la informatica a elevului Manea Andrei, Promotia 2017



Sa presupunem ca un automobilist doreste sa gaseasca cel mai scurt drum dintre doua orase, utilizand o harta rutiera pe care sunt indicate distantele dintre fiecare doua intersectii adiacente.

Un mod posibil de rezolvare a acestei probleme este de a enumera toate drumurile dintre aceste 2 orase si pe baza lungimilor drumurilor, de a selecta unul din cele de lungime minima. Este usor de vazut, chiar si in cazul in care sunt considerate numai drumurile care nu contin cicluri, ca numarul variantelor este enorm.

În pagina **not1.html** se află o definiție cu exemple ale termenilor de graf orientat și neorientat. Pentru a putea accesa documentul următor (**not11.html**), se apasă butonul „Next”.

Drumuri minime de sursa unica in graf


[Home](#)
[Noțiuni Teoretice](#)

[Exercitii Rezolvate](#)
[Exercitii Propuse](#)
[Test Grila](#)
[Bibliografie](#)
[Contact](#)

Intr-o *problema de drum minim* este dat un graf orientat $G = (V, E)$ si functia de cost $\omega : E \rightarrow \mathbb{R}$.

Grafuri

Un graf orientat G este o pereche de multimi (V, E) , unde V este o multime nevida si finita de elemente numite **varfuri**, iar E este o multime de perechi ordonate formate din elemente distincte din multimea V , numite **arce**.



Next

Aici este definit gradul unui nod într-un graf neorientat, precum și gradul interior și exterior ale unui vârf într-un graf orientat. Pentru a putea accesa documentul următor (**not12.html**), se apasă butonul „Next”. De asemenea, pentru a reveni la pagina anterioară (**not1.html**), se apasă butonul „Previous”.

Home

Notiuni Teoretice

Exercitii Rezolvate

Exercitii Propuse

Test Grila

Bibliografie

Contact

Drumuri minime de sursa unica in graf


Gradul unui varf/nod

Gradul exterior al unui varf u , notat $d^+(u)$, reprezinta numarul arcelor care ies din nodul u , adica numarul arcelor de forma $(u,v) \in E$.

Analog, se defineste **gradul interior** al unui varf u , notat $d^-(u)$, ca fiind numarul arcelor care intra in nodul u (de forma $(v,u) \in E$).

De exemplu, in graful din figura pentru nodul $v = 4$ avem:

- $d^+(4)=1 \rightarrow$ din nodul 4 iese un singur arc, in speta arcul $(4, 5)$
- $d^-(4)=3 \rightarrow$ exista trei arce care intra in nodul 4 si anume $(1, 4), (2, 4), (3, 4)$.



Previous

Next

Aici este definit lanțul și ciclul într-un graf neorientat, iar pentru cel orientat, lanțul, drumul și circuitul. Pentru a putea accesa documentul următor (**not13.html**), se apasă butonul „Next”. De asemenea, pentru a reveni la pagina anterioară (**not11.html**), se apasă butonul „Previous”.

Home

Notiuni Teoretice

Exercitii Rezolvate

Exercitii Propuse

Test Grila

Bibliografie

Contact

Drumuri minime de sursa unica in graf


Lanturi. Drumuri. Circuite. Cicluri

Se numeste **lant** într-un graf orientat o multime de arce $L = \{ x_1, x_2, \dots, x_k \}$ cu proprietatea ca \forall doua arce vecine au o extremitate comuna.

De exemplu pentru graful din prima figura:

- $L_1 = \{ (2,4), (3,4), (2,3) \}$
- $L_2 = \{ (1,5), (5,3), (2,3) \}$

Observam defapt ca un lant este un traseu care uneste prin arce, doua noduri (extremitatile lantului), fara a tine cont de orientarea arcelor componente.



Previous

Next

Aici este definit conceptul de arbore și alte noțiuni precum fiu, tată, nepot etc. Pentru a putea accesa documentul următor (**not14.html**), se apasă „Next”. De asemenea, pentru a reveni la pagina anterioară (**not12.html**), se apasă butonul „Previous”.

The screenshot shows a web page with a blue header and a sidebar on the left. The sidebar contains a menu with the following items: Home, Notiuni Teoretice, Exerciții Rezolvate, Exerciții Propuse, Test Grila, Bibliografie, and Contact. The main content area is titled "Arbori binari. Terminologie". It contains the following text:

O categorie importanta de grafuri neorientate sunt acelea in care muchiile reprezinta niste legaturi de tip "parinte-fiu". Un astfel de graf se numeste **arbore**. Ca sa intelegeti semnificatia unei legaturi de tipul "parinte-fiu", este suficient sa va ganditi la urmatorul exemplu. Sa zicem ca o persoana oarecare x are 3 copii; alti copii devenind astfel la randul sau parinte. Dar si nepotii lui x vor urma acelasi drum firesc si vor avea la randul sau alti copii, etc... Am descris astfel ceea ce numim **arbore genealogic** al unei familii.

Un graf neorientat conex si fara cicluri se numeste arbore. Un graf conex este unul care nu contine niciun nod izolat.

Teorema: un arbore cu n noduri are $n-1$ muchii.

Detaliem si sistematizam in continuare cateva proprietati care caracterizeaza un arbore.

- ✓ Exista un nod care nu are niciun parinte(tata), numit **radacina arborelui**
- ✓ Cu exceptia radacinii, fiecare nod are proprietatea ca el este "fiu" unui nod din multime.
- ✓ Nodurile care nu au niciun fiu se numesc **frunze**.
- ✓ Nodurile sunt organizate pe nivele, primul nivel fiind ocupat de nodul radacina.

On the right side of the main content area, there are two buttons: "Previous" and "Next".

Aici este definită o coadă, apoi ce este o coadă de priorități și pentru amândouă structurile, cum se pot folosi. Pentru a reveni la pagina anterioară (**not13.html**), se apasă butonul „Previous”.

The screenshot shows a web page with a blue header and a sidebar on the left. The sidebar contains a menu with the following items: Home, Notiuni Teoretice, Exerciții Rezolvate, Exerciții Propuse, Test Grila, Bibliografie, and Contact. The main content area is titled "Cozi obisnuite. Cozi de prioritate". It contains the following text:

O **coada** este o succesiune ordonata de elemente, in care adaugarea elementelor se face pe la un capat, numit **capat de introducere**(p_1), iar eliminarea elementelor se realizeaza pe la celalalt capat, numit **capat de extragere**(p_2). In orice moment, putem scoate din coada numai elementul aflat la capatul de extragere. Altfel spus, elementele ies din coada in ordinea in care au fost introduse. Datorita acestei proprietati se vor numi cozi **FIFO** ("First In First Out" = "Primul Intrat Primul Iesit").

Prezentam exemplu concret de coada: pentru fiecare zbor de calatori, o companie aviatica intocmeste o lista a persoanelor care mai solicita bilete dupa epuizarea acestora. Lista este organizata sub forma unei cozi de asteptare, ce cuprinde numele si numarul actului de identitate al fiecarui solicitant. In momentul in care o persoana doreste un bilet, este introdusa in coada. Daca in urma unei reuntari apare un bilet disponibil, acesta este atribuit primei persoane care a fost introdusa in coada. Beneficiarul iese din coada, iar capatul de extragere avanseaza cu o pozitie.

O coada se poate implementa cu ajutorul unui tablou unidimensional. Fie $maxn$ numarul maxim de elemente care pot incapa in coada.

Vom numi o **coada vida**, o coada in care nu avem elemente ($p_1 > p_2$) si o **coada plina** o coada in care nu se mai pot adauga elemente ($p_1 = maxn$).

On the right side of the main content area, there is a button: "Previous".

În pagina **not2.html** se află o metoda de reprezentare a grafurilor, anume Matricea de adiacență. Pentru a putea accesa documentul următor (**not21.html**), se apasă butonul „Next”.

Drumuri minime de sursa unica in graf

Metode de reprezentare

Matricea de adiacenta este cea mai simpla forma de reprezentare. Pentru un graf care are n noduri, matricea de adiacenta este un tablou bidimensional cu n linii si n coloane unde pentru orice i si j ($1 \leq i, j \leq n$):

$$A[i][j] = \begin{cases} \omega(i, j), & \text{daca } \exists \text{ arcul } (i, j), i \neq j \\ \infty, & \text{in caz contrar} \end{cases}$$

- unde $\omega(i, j)$ reprezinta costul arcului (i, j) . Datorita orientarii, asa cum am mai spus, arcul (i, j) nu este acelasi lucru cu (j, i) . Prin urmare $A[i][j] \neq A[j][i]$, iar matricea este asimetrica fata de diagonala principala.

Next

Aici este o altă metodă de reprezentare ale grafurilor (listele de adiacență) atât pentru un graf neorientat, cât și pentru un graf orientat. Apoi este explicat ce este vectorul de tați al unui arbore. Pentru a revedea pagina anterioară (**not2.html**), se apasă butonul „Previous”.

Drumuri minime de sursa unica in graf

O alta modalitate de reprezentare o constituie **listele de vecini**. Spre deosebire de matricea de adiacenta aceasta se foloseste pentru a reduce consumul de memorie si de timp. Aceasta consta in - Pentru fiecare varf $u \in \{1, 2, 3, \dots, n\}$:

1. se precizeaza gradul exterior ($d^+(u)$)
2. se precizeaza multimea succesorilor lui u ($\Gamma^+(u)$)
3. se precizeaza pentru fiecare succesor, costul arcului respectiv

Practic acestea se retin sub forma unui tablou bidimensional A , cu doua campuri: unul pentru varfuri de adiacenta (v) si altul pentru costul muchiei respective (c). Mai exact pentru fiecare linie i , $A[i][0].v$ reprezinta $d^+(u)$, iar elementele $A[i][j].v$ $1 \leq j \leq A[i][0].v$ succesori lui i , respectiv $A[i][j].c$ ($1 \leq j \leq A[i][0].v$) reprezinta costul arcului (i, j) .

Previous

În documentul **not3.html** se găsește prezentat, conceptul de „Relaxare a drumurilor”, susținut prin 2 teoreme și 2 subprograme necesare pentru prezentarea algoritmilor următori. Pentru a putea accesa documentul următor (**not31.html**), trebuie se apasă butonul „Next”.

The screenshot shows a web application interface. At the top, there's a header with the title "Drumuri minime de sursa unica in graf". Below the header, on the left, is a sidebar menu with links: "Home", "Notiuni Teoretice", "Exercitii Rezolvate", "Exercitii Propuse", "Test Grila", "Bibliografie", and "Contact". The main content area is titled "Algoritmi" and contains a section "Relaxarea drumurilor". This section explains the relaxation technique and lists two theorems. Theorem 1 states that subpaths of a minimum path are also minimum paths. Theorem 2 relates the distance from a source to a node with the distance to its parent plus the edge weight. A "Next" button is visible on the right side of the content area.

Aici sunt enumerate câteva proprietăți ale Relaxării prin alte 2 teoreme, prima dintre ele fiind demonstrată. Pentru a putea accesa documentul următor (**not32.html**), se apasă butonul „Next”. De asemenea, pentru a vedea pagina anterioară (**not3.html**), se apasă butonul „Previous”.

This screenshot shows the same web application interface as the previous one, but at a different page. The sidebar menu is identical. The main content area is still titled "Algoritmi" but now displays the section "Proprietati ale relaxarii". It discusses the correctness of the algorithms based on the properties of relaxation. It lists Theorem 3, which states that the distance from the source to a node is non-decreasing as relaxation steps are performed. A "Previous" and "Next" button are visible on the right side of the content area.

Aici este explicat Algoritmul Dijkstra (afișat în pseudocod), alături de o poză a lui Edsger W. Dijkstra. Ulterior, ca apăsarea demonstrația acestui algoritm, precum și cazurile în care nu se poate aplica algoritmul Dijkstra

Drumuri minime de sursa unica in graf

[Home](#)
[Notiuni Teoretice](#)


[Exercitii Rezolvate](#)
[Exercitii Propuse](#)
[Test Grila](#)
[Bibliografie](#)
[Contact](#)

Algoritmul Dijkstra

Algoritmul Dijkstra rezolva problema drumurilor minime de sursa unica intr-un graf orientat $G=(V, E)$ cu costuri pozitive. In cadrul acestei secțiuni vom presupune ca $\omega(u,v) \geq 0$ pentru fiecare arc $(u,v) \in E$. Algoritmul a fost conceput de matematicianul Edsger W. Dijkstra in 1956, intr-o varianta care nu folosea coada de prioritati.

Dijkstra(G, ω, s)

1. Initializeaza-Sursa-Unica(G, s)
2. $S \leftarrow \emptyset$
3. $Q \leftarrow V$
4. cat timp $Q \neq \emptyset$ executa
5. $u \leftarrow \text{Extrage-Min}(Q)$
6. $S \leftarrow S \cup \{u\}$
7. pentru fiecare varf $v \in \Gamma^+(u)$ executa
8. Relaxeaza(u, v, ω)



[Previous](#)
[Next](#)

Tot aici se poate vedea un scurt film în care se aplică algoritmul asupra aceluși graf. Pentru a putea accesa documentul următor (**not33.html**), se apasă butonul „Next”. De asemenea, pentru a reveni la pagina anterioară (**not31.html**), se apasă butonul „Previous”.

Drumuri minime de sursa unica in graf

[Home](#)
[Notiuni Teoretice](#)

[Exercitii Rezolvate](#)
[Exercitii Propuse](#)
[Test Grila](#)
[Bibliografie](#)
[Contact](#)

Algoritmul opereaza prin mentinerea unei multimi S de varfuri pentru care costurile finale corepunzatoare drumurilor minime de la sursa s au fost deja determinate respectiv pentru toate varfurile $v \in S$ avem $d[v] = \delta(s,v)$. Algoritmul itereaza selectarea unui varf $v \in V - S$ pentru care estimarea drumului minim este minima, introduce u in S si relaxeaza arcele adiacente lui u . In implementarea care va fi prezentata se va mentine o coada de prioritati Q ce contine toate varfurile din $V - S$ indexate prin valorile din d corespunzatoare.



Aici este explicat Algoritmul Bellman-Ford (afișat în pseudocod), alături de o poză a lui Richard Bellman. Ulterior, ca apăsarea demonstrația acestui algoritm. Pentru a putea accesa documentul următor (**not34.html**), se apasă butonul „Next”. De asemenea, pentru a reveni la pagina anterioară (**not32.html**), se apasă butonul „Previous”.

The screenshot shows a web page with a blue header and a sidebar on the left. The main content area is titled "Algoritmul Bellman-Ford". The sidebar contains links: Home, Notiuni Teoretice, Exerciții Rezolvate, Exerciții Propuse, Test Grila, Bibliografie, and Contact. The main content area contains the following text:

Algoritmul Bellman-Ford

Algoritmul Bellman-Ford rezolva problema drumurilor minime de sursa unica in cazul mai general, in care costurile arcelor pot fi negative. Fiind dat un graf orientat $G = (V, E)$ cu functia de cost $\omega : E \rightarrow \mathbb{R}$, algoritmul returneaza o valoare booleana ce indica daca exista sau nu un circuit de cost negativ, accesibil din varful sursa considerat. In cazul in care un astfel de circuit exista, algoritmul semnaleaza ca nu exista solutie, iar daca nu exista, acesta returneaza drumurile minime si costurile corespunzatoare lor.

Algoritmul Bellman-Ford preia numele a 2 matematicieni dintre cei care l-au conceput, Richard Bellman si Lester Ford Jr., care l-au publicat in 1958, respectiv in 1956. Cu toate acestea, Edward F. Moore a publicat, de asemenea, acelasi algoritm in 1957 si de aceea mai este numit Algoritmul Bellman-Ford-Moore.

La fel ca algoritmul Dijkstra, algoritmul Bellman-Ford utilizeaza tehnica de relaxare, prin intermediul decrementarii estimarii $d[v]$ a drumului minim de la sursa s la fiecare varf $v \in V$ pana este obtinut costul adevarat $\delta(s, v)$ corespunzator unui drum minim. Algoritmul returneaza ADEVARAT daca si numai daca nu contine circuite de cost negativ accesibile din sursa.

On the right side of the main content area, there are two buttons: "Previous" and "Next".

În documentul **not34.html** este prezentată o abordare optimizată a algoritmului Bellman-Ford, apoi sunt explicate câteva subprograme pentru folosirea cozilor de prioritate. Pentru a putea accesa documentul următor (**not35.html**), se apasă butonul „Next”. De asemenea, pentru a vedea pagina anterioară (**not33.html**), se apasă butonul „Previous”.

The screenshot shows a web page with a blue header and a sidebar on the left. The main content area is titled "Drumuri minime de sursa unica in graf". The sidebar contains links: Home, Notiuni Teoretice, Exerciții Rezolvate, Exerciții Propuse, Test Grila, Bibliografie, and Contact. The main content area contains the following text:

Acesta poate fi optimizat prin intermediul utilizării unei cozi de prioritate Q , a unui tablou unidimensional N care contorizeaza pentru orice varf $v \in V$ de cate ori a fost introdus in Q si a unei variabile booleane C .

Bellman-Ford2(G, s)

1. $C \leftarrow \text{ADEVARAT}$
2. $\text{Initializeaza-Sursa-Unica}(G, s)$
3. $Q \leftarrow \{s\}$
4. **pentru** fiecare varf $v \in V$ **executa**
5. $N[v] \leftarrow 0$
6. $N[s] \leftarrow 1$
7. **cat timp** $Q \neq \emptyset$ **si** $C = \text{ADEVARAT}$ **executa**
8. $u \leftarrow \text{Extrage-Min}(Q)$
9. **pentru** fiecare varf $v \in \Gamma^+(u)$ **executa**
10. **daca** $d[v] \leq d[u] + \omega(u, v)$ **atunci**
11. $d[v] \leftarrow d[u] + \omega(u, v)$

On the right side of the main content area, there are two buttons: "Previous" and "Next".

Aici se poate vedea cum este creat și folosit un heap. Tot aici, apar și anumite subprograme apelate în paginile anterioare. Pentru a putea accesa documentul anterior (**not34.html**), se apasă butonul „Previous”.

Drumuri minime de sursa unica in graf

[Home](#)
[Notiuni Teoretice](#)

[Exercitii Rezolvate](#)
[Exercitii Propuse](#)
[Test Grila](#)
[Bibliografie](#)
[Contact](#)

Crearea unui Heap

Dupa cum am observat, putem crea un heap prin apeluri succesive ale procedurii **Insereaza(H, cheie)**; daca am insera n elemente, atunci complexitatea ar ajunge la $O(n \log_2 n)$. O strategie mai eficienta de constructie a unui heap se bazeaza pe ideea de echilibrare. Apelul functiei **Insereaza(H, cheie)** poate fi interpretat ca o combinatie de doua heap-uri: un heap cu n elemente si un heap format doar din elementul cheie. Putem construi heap-ul cu radacina $H[i]$ combinand la fiecare pas i , doua heap-uri de dimensiuni apropiate: heap-ul de radacina $2i$ si heap-ul cu radacina $2i+1$ si cu elementul $H[i]$ asa cum se intampla in procedura **CreareHeap(n)**:

CreareHeap(H)

1. pentru $i \leftarrow [n/2, 1, -1]$ executa
2. **CombHeap(H, i)**

Functia **CombHeap(H, i)** combina elementul $H[i]$ cu heapurile de radacina $2i$ si $2i+1$. Sa analizam combinarea heap-urilor (din figura) cu radacina in nodul 2, respectiv 3 cu nodul de pe pozitia 1. (avand valoarea 12):

[Previous](#)

A doua parte a acestui soft este constituită din partea practică. În documentul **ex1.html** se află printscreen-uri dintr-o problemă ce se rezolvă prin intermediul algoritmului Dijkstra. Programul se afla De asemenea, în folderul cu proiectul (proiect). Pentru a putea accesa documentul următor (**ex11.html**), se apasă butonul „Next”.

Drumuri minime de sursa unica in graf

[Home](#)
[Notiuni Teoretice](#)

[Exercitii Rezolvate](#)
[Exercitii Propuse](#)
[Test Grila](#)
[Bibliografie](#)
[Contact](#)

Exercitii rezolvate

1. Se da un graf orientat cu N noduri si M arce. Sa se determine lungimea minima a drumului de la nodul 1 la fiecare din nodurile 2, 3, ..., $N-1$, N si sa se afiseze aceste lungimi. Lungimea unui drum este data de suma lungimilor arcelor care constituie drumul.

```
1. #include <fstream>
2. #include <vector>
3. #include <queue>
4. #include <limits>
5. #define w 52001
6. using namespace std;
7. struct nod{unsigned int c,v;};
8. class prio
9. {
10. public:
11. bool operator <()(nod &p1 ,nod&p2)
12. {
13. return p1.c>p2.c;
14. }
15. };
16. priority_queue <nod, vector <nod>, prio> q;
17. vector <nod> a[w];
18. unsigned int cost;
```

```
37. int main()
38. {
39. ifstream f("dijkstra.in");
40. ofstream g("dijkstra.out");
41. unsigned int n,i,m,x,y,cost;
42. f>>n>>m;
43. for (i=1;i<=m;i++)
44. {
45. f>>x>>y>>cost;
46. a[x].push_back(nod{cost,y});
47. }
48. for (i=2;i<=n;i++) d[i]=INT_MAX;
49. q.push(nod{0,1});
50. dij();
51. for (i=2;i<=n;i++)
52. {
53. if (d[i]==INT_MAX) g<<"0 ";
54. else g<<d[i]<<" ";
55. }
```

[Next](#)

În documentul **ex1.html** se află printscreen-uri dintr-o altă problemă ce se rezolvă prin intermediul algoritmului Bellman-Ford. Programul se află, de asemenea, în folderul cu proiectul(proiect). Pentru a putea accesa documentul anterior (**ex1.html**), se apasă butonul „Previous”.

The screenshot shows a web application titled "Drumuri minime de sursa unica in graf". On the left is a navigation menu with links: Home, Notiuni Teoretice, Exercitii Rezolvate (active), Exercitii Propuse, Test Grila, Bibliografie, and Contact. The main content area is titled "Exercitii rezolvate" and displays problem 2: "Se da un graf orientat conex cu N noduri si M muchii cu costuri. Definim un lant ca fiind un sir de noduri cu proprietatea ca intre oricare doua consecutive exista o muchie. Costul unui lant este dat de suma costurilor muchiilor care unesc nodurile ce il formeaza. Definim un ciclu ca fiind un lant cu proprietatea că primul element al sau este egal cu ultimul. Sa se determine dacă în graful dat exista un ciclu de cost negativ. Daca nu exista, sa se determine costul minim al unui lant de la nodul 1 la fiecare dintre nodurile 2, 3, ..., N-1, N." Below the text are two code snippets in C++ for a Bellman-Ford algorithm. A "Previous" button is visible on the right.

În documentele **ex2.html** și **ex21.html** se află în fiecare cate o problema propusă.

The screenshot shows the same web application but with the "Exercitii propuse" section active. The navigation menu remains the same. The main content area is titled "Exercitii propuse" and displays problem 1, "Speedy": "Speedy Gonzales este fermier in Texas. El are un ranch situat intr-o zona denivelata care se poate reprezenta printr-o matrice A cu N linii si M coloane, elementul A[i, j] reprezentand inaltimea zonei de coordonate (i, j). Speedy nu are astampar. El porneste din zona initiala de coordonate (A, B) si se poate deplasa pe cele patru directii: sus, jos, stanga, dreapta. Daca prin deplasare intr-o celula invecinata Speedy urca K unitati de nivel, atunci viteza lui scade cu 2K unitati de viteza. Daca prin deplasarea intr-o celula invecinata el coboara K unitati de nivel, atunci viteza sa creste cu K unitati de viteza. In ambele situatii, din cauza oboselii viteza scade constant cu o unitate. Deoarece nu vrea sa se opreasca, el nu se va putea deplasa intr-o celula vecina decat daca viteza ramane strict pozitiva. Cunoscand viteza initiala V a lui Speedy, determinati viteza maxima cu care poate ajunge intr-o zona aflata pe marginea ranch-ului." Below the text are sections for "Date de intrare" and "Date de iesire". A "Next" button is visible on the right.

Home

Notiuni Teoretice

Exercitii Rezolvate

Exercitii Propuse

Test Grila

Bibliografie

Contact

Drumuri minime de sursa unica in graf

Exercitii propuse

2.Dragonii(OJI 2015)

Suparati ca lansarea partii a treia a filmului lor preferat s-a amanat pana in iunie 2018, Henry si Hetty s-au gandit la propriul scenariu pentru finalul trilogiei: Intr-o lume in care vikingii pot zbura cu dragonii exista n insule. Hiccup, seful tribului de vikingi aflat pe insula 1, stie m rute directe de zbor **bidirectionale** intre insule. Pentru fiecare j intre 1 si m , ruta j uneste insulele A_i si B_j si are lungime D_j .

Pe fiecare insula i , ($1 \leq i \leq n$) exista dragoni din specia i care pot zbura fara a se opri pentru odihna o distanta maxima D_{maxi} . Cu alte cuvinte, dragonii de pe insula i vor putea parcurge orice ruta j , ($1 \leq j \leq m$) pentru care $D_j \leq D_{maxi}$ indiferent de ce alte drumuri au facut anterior.

Hiccup doreste sa ajunga de pe insula 1 pe insula N pentru a-l salva pe Toothless, dragonul lui. Pentru a ajunge acolo, el va lua initial un dragon din specia 1 (de pe insula 1). Apoi, daca la un moment dat Hiccup se afla pe o insula i , ($1 \leq i \leq n$) avand cu el un dragon din specia t , el poate:

1. Sa zboare de pe insula i pe o alta insula x cu dragonul pe care il are, folosind o ruta directa j intre insulele i si x , bineinteles doar daca $D_j \leq D_{maxt}$.
2. Sa schimbe dragonul din specia t cu un dragon din specia i (de pe insula i).

Previous

În documentul **test.php** se găsește un test grilă prin care utilizatorul își poate verifica cunoștințele dobândite. Acesta a fost realizat folosind butoane radio, checkbox-uri, chiar și tabele. Utilizatorul trebuie să își completeze câmpurile cu Nume, Prenume și Clasa, apoi sa selecteze butoanele radio, respectiv checkbox-urile din dreptul fiecărei întrebări. Dacă se procedează astfel, se va afișa nota și posibilele răspunsuri greșite, altfel vă vor apărea câteva mesaje de validare în colțul din dreapta sus al paginii.

Home

Notiuni Teoretice

Exercitii Rezolvate

Exercitii Propuse

Test Grila

Bibliografie

Contact

Drumuri minime de sursa unica in graf

Test

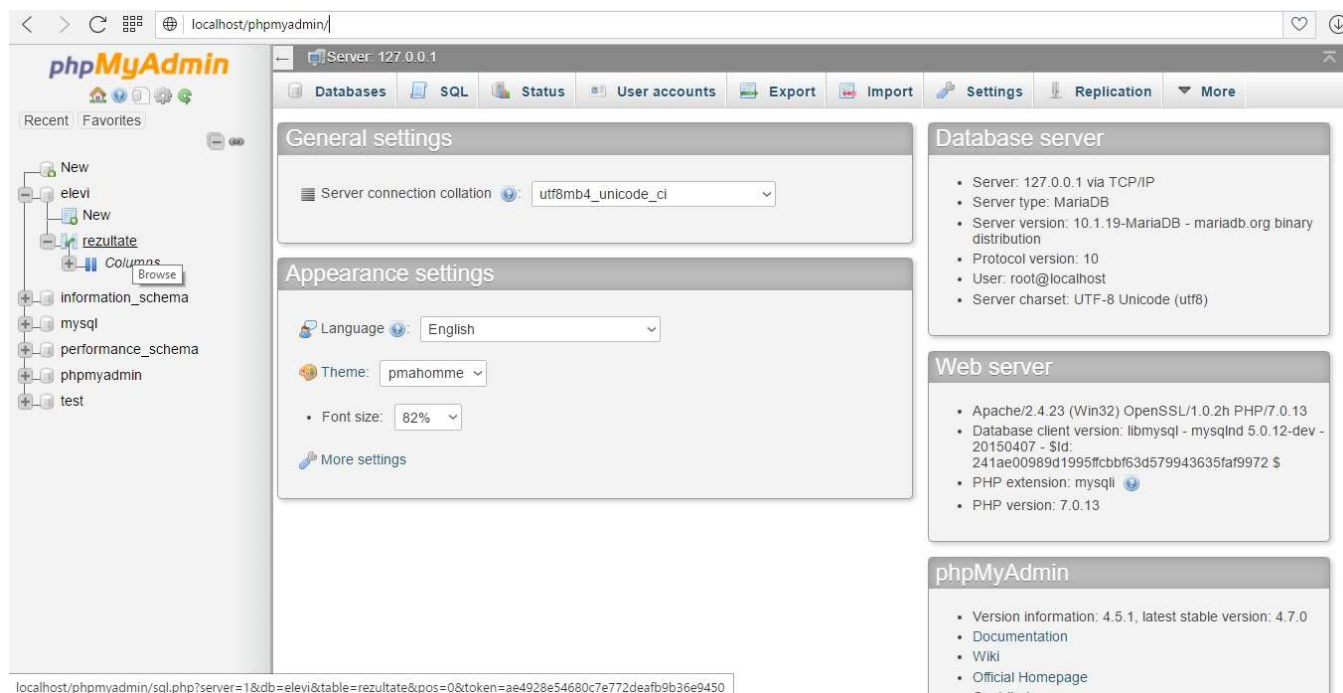
Nume :

Prenume :

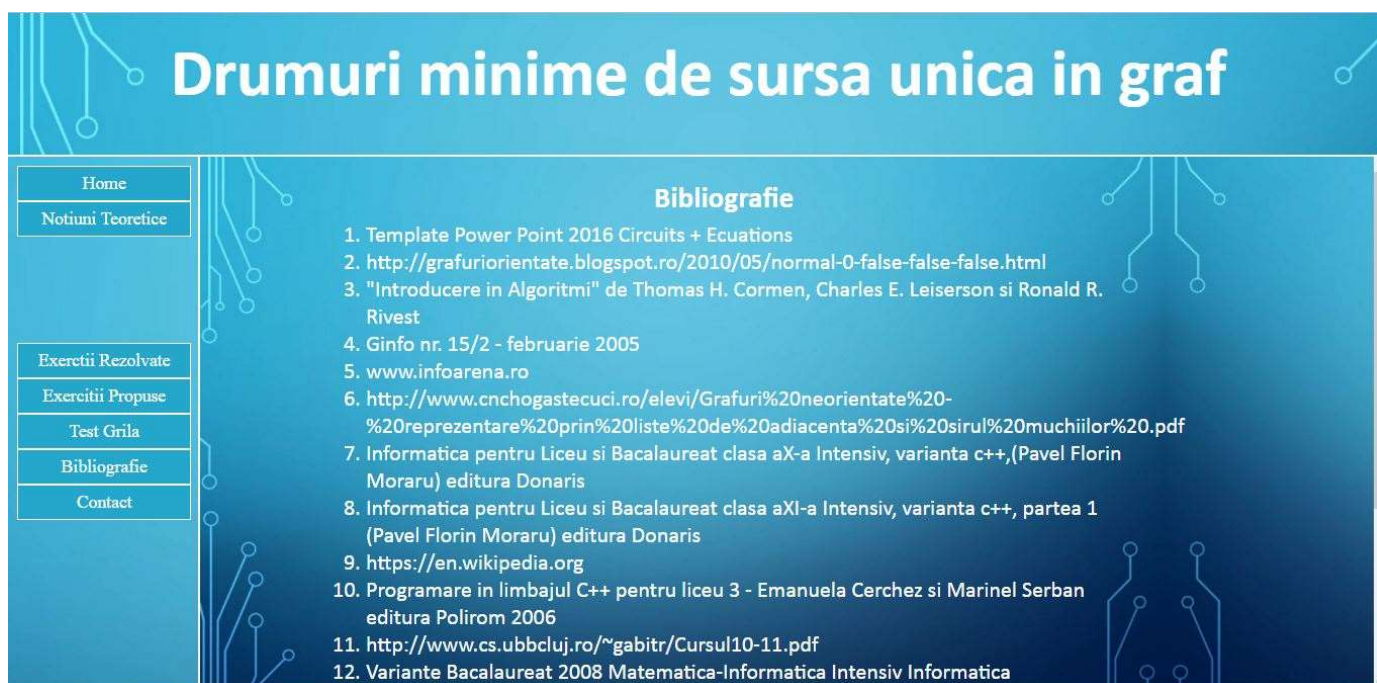
Clasa :

1. Intr-un graf ponderat, care are circuite de cost negativ, se poate determina un drum minim dintre 2 noduri?
 - ☒ a. Da
 - ☐ b. Nu
2. Cate circuite disjuncte (care difera prin cel putin un arc) de cost negativ exista in graful orientat din figura?

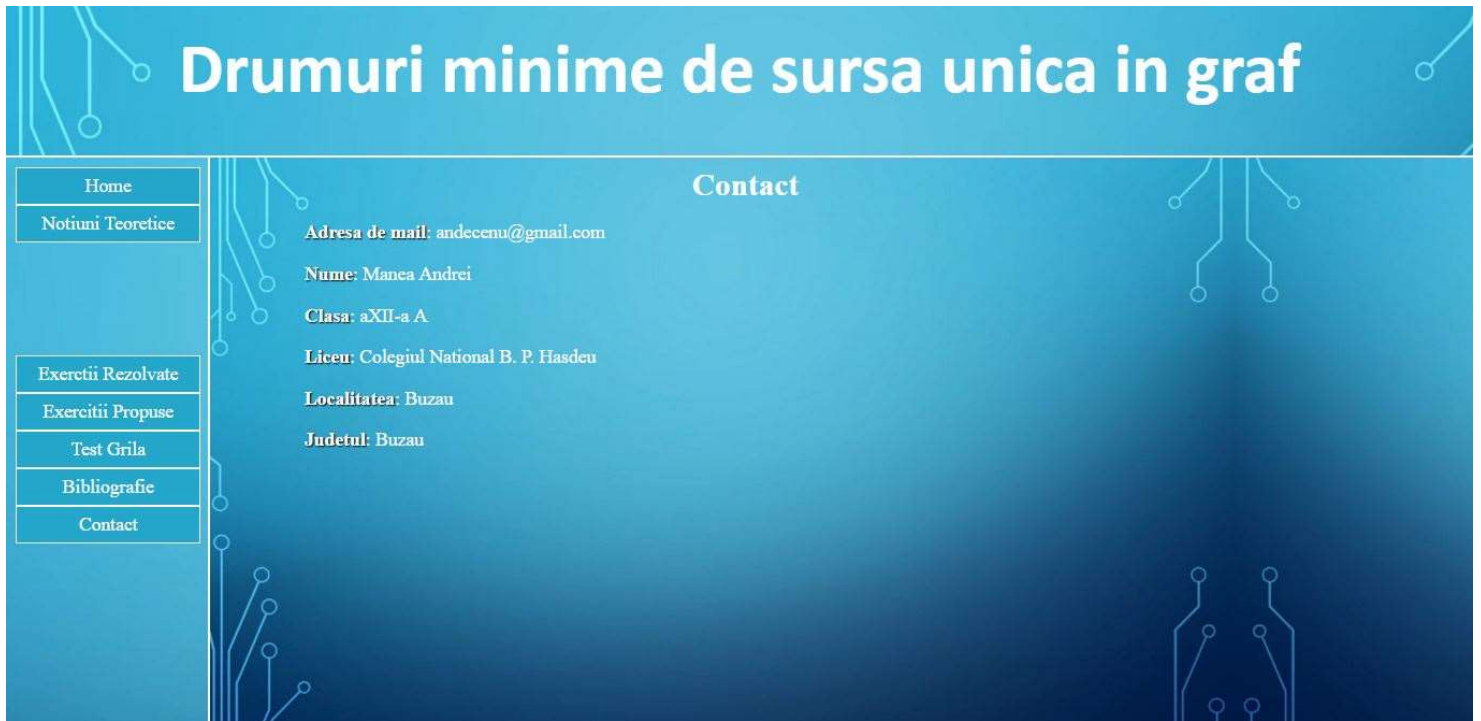
Nota se va calcula în limbajul **php**(în variabila \$nota). Rezultatele se vor depune într-o bază de date(**elevi**), într-un tabel(**rezultate**) care poate fi accesat din link-ul: <http://localhost/phpmyadmin/>. Tabelul va apărea la apăsarea pe „rezultate” la fel ca în poza de mai jos.



În documentul **bibliografie.html** sunt enumerate sursele folosite pentru a realiza proiectul de atestare. Bibliografia de la sfârșitul acestei documentații conține această bibliografie plus alte elemente folosite în această documentație.



În documentul **contact.html** sunt scrise câteva date personale.



Capitolul 4 - Instalarea aplicației

4.1 Resurse minimale

Programul a fost realizat în așa fel încât să ruleze cel mai bine pe o rezoluție de 1360x768. De asemenea, sistemul va trebui să aibă cel puțin 2 GB RAM și 500 MHz și o placă video cu cel puțin 16 MB Ram. Sistemul de operare va trebui să fie pe minim 32 de biți. Programul mai are nevoie de un spațiu pe hard de aproximativ 28 MB.

Mai este necesar pe hard un spațiu liber de cel puțin 50 MB.

Dacă aceste cerințe nu sunt respectate, aplicația nu va putea fi copiată.

De asemenea, pentru funcționarea deplină a testului grilă, este necesară instalarea prealabilă a unui server WEB de tip APACHE, împreună cu programul PHP, sau a pachetului WAMP server, care conține toate programele necesare rulării optime a aplicației. Se poate descărca de aici: <http://www.filehorse.com/download-wampserver-32/> .

Se recomandă instalarea aplicației Code Blocks 16.01 atât pentru rularea programelor deja scrise (în secțiunea Exerciții rezolvate) cât și pentru rezolvarea problemelor propuse: <https://sourceforge.net/projects/codeblocks/files/Binaries/16.01/Windows/codeblocks-16.01mingw-setup.exe/download> .

4.2 Instalare – dezinstalare

Lucrarea este prezentată pe suport magnetic CD. Pentru a putea fi vizualizată trebuie copiată în folderul www al serverului local de php - de exemplu wamp - și rulată într-un browser web prin tastarea căii complete prin <http://localhost>.

Dezinstalarea se realizează foarte ușor prin ștergerea efectivă a folder-ului „proiect”.

Capitolul 5 - Bibliografie

- Emanuela Cerchez, Marinel Șerban, "Informatica, manual pentru clasa a IX-a, profilul real"
- Burta Alin, "Informatică - manual pentru clasa a XII-a"
- <http://www.ro.wikipedia.org>
- Template Power Point 2016 Circuits + Ecuations
- <http://grafuriorientate.blogspot.ro/2010/05/normal-0-false-false-false.html>
- "Introducere in Algoritmi" de Thomas H. Cormen, Charles E. Leiserson si Ronald R. Rivest
- Ginfo nr. 15/2 - februarie 2005
- www.infoarena.ro
- <http://www.cnchogastecuci.ro/elevi/Grafuri%20neorientate%20-%20reprezentare%20prin%20liste%20de%20adiacente%20si%20sirul%20muchiiilor%20.pdf>
- Informatica pentru Liceu si Bacalaureat clasa a-X-a Intensiv, varianta c++, (Pavel Florin Moraru) editura Donaris
- Informatica pentru Liceu si Bacalaureat clasa a-XI-a Intensiv, varianta c++, partea 1 (Pavel Florin Moraru) editura Donaris
- Programare in limbajul C++ pentru liceu 3 - Emanuela Cerchez si Marinel Șerban editura Polirom 2006
- <http://www.cs.ubbcluj.ro/~gabitr/Cursul10-11.pdf>
- Variante Bacalaureat 2008 Matematica-Informatica Intensiv Informatica
- <https://convertio.co/mp4-ogv/>
- <http://pastebin.com>
- <https://www.wolframalpha.com>
- Adobe Dreamweaver CS6
- Notepad++
- Adobe Photoshop CS6
- <https://images.google.com>
- <https://www.youtube.com/watch?v=iCE2Js3KPRA>
- <https://www.w3schools.com>
- <http://www.yellowpipe.com/yis/tools/hex-to-rgb/color-converter.php>
- Aplicația Solo-Learning
- Manual de Informatică Intensiv pentru clasa a XII-a Vlad - Tudor Huțanu și Carmen Popescu editura L&S info-mat

Capitolul 6 – Anexe

- Fragment din pagina „Home”:

```
<!DOCTYPE HTML>
<html>
  <head>
    <style type="text/css">
      #apDiv1 {
        position: absolute;
        width: 729px;
        height: 366px;
        z-index: 1;
        left: 90px;
        top: 263px;
      }
      #umbra {
        position: absolute;
        width: 745px;
        height: 211px;
        z-index: 2;
        left: 73px;
        top: 47px;
      }
      #apDiv2 {
        position: absolute;
        width: 742px;
        height: 64px;
        z-index: 3;
        left: 91px;
        top: 991px;
      }
      #apDiv3 {
        position: absolute;
        width: 200px;
        height: 396px;
        z-index: 4;
        left: 165px;
        top: 588px;
      }
      #apDiv4 {
        position: absolute;
        width: 153px;
```

```

        height: 159px;
        z-index: 5;
        left: 864px;
        top: 130px;
    }
</style>
</head>
<body BackGround="f22.png">
    <link href="base.css" rel="stylesheet" type="text/css">
    <span style="font-family:calibri; color:white; font-size:125%;">
    <div id="apDiv1">
        <p> <tab1>Sa presupunem ca un automobilist doreste sa gaseasca cel mai scurt drum dintre doua orase,
        utilizand o harta rutiera pe care sunt indicate distantele dintre fiecare doua intersectii adiacente.</p>
        <p> <tab1>Un mod posibil de rezolvare a acestei probleme este de a enumera toate drumurile
        dintre aceste 2 orase si pe baza lungimilor drumurilor, de a selecta unul din cele de lungime minima.Este
        usor de vazut, char si in cazul in care sunt considerate numai drumurile care nu contin cicluri, ca numarul
        variantelor este enorm. </p>
        <p> <tab1> De exemplu, un drum de la Chicago la Boston care trece prin Houston este evident
        o solutie de neacceptat deoarece Houston se afla la distanta de 1000 de mile de cele doua orase</p>

    </div>
    <div id="apDiv3">
    
    </div>

<div id="apDiv2">

    <p><tab1>In acest soft educational, vom arata cum poate fi rezolvata aceasta problema in mod
    eficient.</p>
</div>
    </span>
    <div id="umbra">
        <p align="center"><b>
        <span style=" font-family:calibri; font-size:38pt;color:white; ">
            Lucrare practica de atestare a cunostintelor la informatica, Promotia 2017
        </span>
        </b></p>
    </div>
    <div id="apDiv4"></div>
</body>
</html>

```

- Fragment din pagina „Test”:

```

<html>
    <head>
    <style type="text/css">
    #apDiv1 {
    position: absolute;
    width: 353px;

```

```

        height: 282px;
        z-index: 1;
        left: 236px;
        top: 302px;
    }
    #apDiv2 {
        position: absolute;
        width: 649px;
        height: 57px;
        z-index: 2;
        left: 181px;
        top: 588px;
    }
    #apDiv3 {
        position: absolute;
        width: 750px;
        height: 115px;
        z-index: 3;
        left: 49px;
        top: 179px;
    }
    #apDiv4 {
        position: absolute;
        width: 715px;
        height: 358px;
        z-index: 4;
        left: 103px;
        top: 656px;
    }
    #apDiv5 {
        position: absolute;
        width: 530px;
        height: 247px;
        z-index: 5;
        left: 185px;
        top: 1024px;
    }
    #apDiv5 table tr td {
        text-align: center;
        color: #FFF;
        font-weight: bolder;
        font-family: Georgia, "Times New Roman", Times, serif;
    }
    #apDiv6 {
        position: absolute;
        width: 728px;
        height: 212px;
        z-index: 6;
        left: 84px;
        top: 1268px;
    }
}

```

```

#apDiv7 {
    position: absolute;
    width: 264px;
    height: 256px;
    z-index: 7;
    left: 134px;
    top: 1476px;
}
#apDiv8 {
    position: absolute;
    width: 260px;
    height: 256px;
    z-index: 8;
    left: 497px;
    top: 1484px;
}
#apDiv9 {
    position: absolute;
    width: 265px;
    height: 269px;
    z-index: 9;
    left: 132px;
    top: 1736px;
}
#apDiv10 {
    position: absolute;
    width: 263px;
    height: 254px;
    z-index: 10;
    left: 493px;
    top: 1739px;
}
#apDiv11 {
    position: absolute;
    width: 675px;
    height: 115px;
    z-index: 11;
    left: 108px;
    top: 2009px;
}
#apDiv12 {
    position: absolute;
    width: 729px;
    height: 162px;
    z-index: 12;
    left: 90px;
    top: 2398px;
}
#apDiv13 {
    position: absolute;
    width: 414px;

```

```

        height: 224px;
        z-index: 13;
        left: 226px;
        top: 2162px;
    }
    #apDiv14 {
        position: absolute;
        width: 465px;
        height: 105px;
        z-index: 14;
        left: 346px;
        top: 5px;
        font-family: calibri;
    }
    #apDiv15 {
        position: absolute;
        width: 760px;
        height: 115px;
        z-index: 15;
        left: 93px;
        top: 54px;
        font-family: calibri;
    }
</style>
</head>
<body Background="f22.png">
    <span style="font-family:calibri; color:white; font-size:125%;">
<div id="apDiv1"></div>
    <link href="base.css" rel="stylesheet" type="text/css">
    <div id="apDiv14">
        <tab2>
    <p>
        <?php
            include 'connect2.php';
            include 'tabel.php';

            $mesaj="";
            if (isset($_POST['submit'])) $mesaj = 'Va rugam sa va autentificati !';

            if (isset($_POST['nume']) && isset($_POST['prenume']) && isset($_POST['clasa']))
            {

                // Se verifica daca au fost completate corect toate campurile
                // Daca au fost completate le preia in variabile, in caz contrar seteaza o variabila tip Array
cu mesaj de eroare
                $mesaj='1';
                // Verifica daca Numele are cel putin 3 caractere si maxim 40
                if(strlen($_POST['nume'])>2 && strlen($_POST['nume'])<41) $nume =
$_POST['nume'];
                else $eroare[] = 'Caseta Nume trebuie sa contina intre 3 si 40 caractere';

```

```

// Verifica daca Numele are cel putin 3 caractere si maxim 40
if(strlen($_POST['prenume'])>2 && strlen($_POST['nume'])<41) $prenume =
$_POST['prenume'];
else $eroare[] = 'Caseta Prenume trebuie sa contina intre 3 si 40 caractere';

// Verifica daca a fost selectata o valoare pt. 'clasa'
if(strlen($_POST['clasa'])>0) $clasa = $_POST['clasa'];
else $eroare[] = 'Selectati "clasa"';
$greseli=array();
$rez=1;
if (isset($_POST['q1']))
{
    $q1=$_POST['q1'];
    if ($q1=="b") $rez=$rez+1;
    else $greseli[]=1;
}
else $err[] = '1';

if (isset($_POST['q2']))
{
    $q2=$_POST['q2'];
    if ($q2=="c") $rez=$rez+1;
    else $greseli[]=2;
}
else $err[] = '2';

if (isset($_POST['q3']))
{
    $q3=$_POST['q3'];
    if ($q3=="a") $rez=$rez+1;
    else $greseli[]=3;
}
else $err[] = '3';

if (isset($_POST['q4']))
{
    $q4=$_POST['q4'];
    if ($q4=="d") $rez=$rez+1;
    else $greseli[]=4;
}
else $err[] = '4';

if (isset($_POST['q5']))
{
    $q5=$_POST['q5'];
    if ($q5=="c") $rez=$rez+1;
    else $greseli[]=5;
}
else $err[] = '5';

if (isset($_POST['q6']))

```

```

        {
            $q6=explode("",$ _POST['q6']);
            if (strpos($q6,'a')) $q6a=TRUE;
            if (strpos($q6,'b')) $q6b=TRUE;
            if (strpos($q6,'c')) $q6c=TRUE;
            if (strpos($q6,'d')) $q6d=TRUE;
            if ($q6=="bd") $rez=$rez+1;
            else $greseli[]=6;
        }
        else $err[] = '6';

        if (isset($_POST['q7']))
        {
            $q7=$_POST['q7'];
            if ($q7=="a") $rez=$rez+1;
            else $greseli[]=7;
        }
        else $err[] = '7';

        if (isset($_POST['q8']))
        {
            $q8=$_POST['q8'];
            if ($q8=="c") $rez=$rez+1;
            else $greseli[]=8;
        }
        else $err[] = '8';

        if (isset($_POST['q9']))
        {
            $q9=$_POST['q9'];
            if ($q9=="d") $rez=$rez+1;
            else $greseli[]=9;
        }
        else $err[] = '9';

        if (isset($err))
        {
            if (count($err)>1) $eroare[]="Nu raspundeti la intrebarile ".implode("
", $err)."?";
            else $eroare[]="Nu raspundeti la intrebarea ".implode("
", $err)."?";
        }

        // Daca nu exista nici un mesaj de eroare, filtreaza datele cu mysql_real_escape_string()
        si le adauga in baza de date
        // Altfel, in cazul vreunei erori, adauga in variabila $mesaj
        if (!isset($eroare))
        {
            // Acum se adauga aceste date in tabelul rezultate
            $sql = "INSERT INTO elevi.rezultate (`Nume`, `Prenume`, `Clasa`, `Punctaj`)
VALUES ('".$nume."', '".$prenume."', '".$clasa."', '".$rez."');";

```



```

        if ($conn->query($sql) == TRUE)
        {
            $mesaj = "Datele au fost adaugate<br/>Felicitari! Ati obtinut nota
$.rez." !";
            if (count($greseli)>=1)
                $mesaj=$mesaj."<br/>Raspunsuri gresite: ".implode(",
$,greseli)." .";
        }
        else $mesaj = 'Datele nu au putut fi adaugate '. mysql_error();
    }
    else $mesaj = implode('<br />', $eroare);
}
echo $mesaj.<br/>;

?></p>
</div>
<form method="post" action="">

    <ptit> Test </ptit>
    <div id="apDiv3">
<ol>
    <ol>
        <li> Intr-un graf ponderat, care are circuite de cost negativ, se poate determina un drum minim dintre 2
noduri? <br/>
            <tab4><input type="radio" name="q1" <?php if (isset($q1) && $q1=="a") echo "checked";?>
value="a">a. Da
            <tab5><input type="radio" name="q1" <?php if (isset($q1) && $q1=="b") echo "checked";?>
value="b">b. Nu

        </li>
        <li>
            Cate circuite disjuncte (care difera prin cel putin un arc) de cost negativ exista in graful
orientat din figura?<br/>
            <tab1>

        </li>

    </ol>
    </ol>
</div>
<div id="apDiv4">
    <ol start="3">
        <li> Relaxarea drumurilor este:<br/>
            <input type="radio" name="q3" <?php if (isset($q3) && $q3=="a") echo "checked";?> value="a">a. O
tehnica care consta sta la baza determinarii unui drum minim de sursa unica intr-un graf;<br/>
            <input type="radio" name="q3" <?php if (isset($q3) && $q3=="b") echo "checked";?> value="b">b.
Un algoritm scris de Richard Bellman;<br/>
            <input type="radio" name="q3" <?php if (isset($q3) && $q3=="c") echo "checked";?>
value="c">c. O activitate sportiva;<br/>
            <input type="radio" name="q3" <?php if (isset($q3) && $q3=="d") echo "checked";?>
value="d">d. Metoda de parcurgere a unui graf.</li>

```

- Graful neorientat cu 8 noduri, numerotate de la 1 la 8, este reprezentat cu ajutorul matricei de adiacenta de mai jos. Pentru acest graf este adevarata afirmatia:

☐

☐

☐

☐

- Intr-un arbore cu radacina cu 10 noduri, numerotate de la 1 la 10, nodul 10 este radacina, iar intre celelalte noduri exista relatia: nodul cu numărul $i+1$ este tatăl celui cu numărul i , pentru $i \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Vectorul de tati al arborelui astfel definit, este:

☐

☐

☐

☐

- Care dintre urmatorii arbori reprezinta un max-heap:

- Sa consideram urmatorul vector $H = (2, 5, 3, 9, 8, 4, 10, 20, 10, 9)$. Poate fi considerat reprezentarea secventiala a unui min-heap?

☐

☐

- Pentru graful orientat din figura, cel mai scurt drum posibil din varful 1 la varful 2 are costul:

☐

☐

☐

☐

```

<input type="radio" name="q8" <?php if (isset($q8) && $q8=="e") echo "checked";?> value="e">e.
9<br/>
<ol start="9">
<li>Se considera un graf neorientat cu nodurile numerotate de la 1 la 5 si muchiile (1, 2), (1, 3),
(3, 2), (4, 2), (3, 4), (3, 5), (4, 5). Care este numarul maxim de muchii ce pot fi eliminate din graf
astfel incat graful partial rezultat sa fie conex?<br/>
<tab2>
<input type="radio" name="q9" <?php if (isset($q9) && $q9=="a") echo "checked";?>
value="a">a. 0<tab2>
<input type="radio" name="q9" <?php if (isset($q9) && $q9=="b") echo "checked";?> value="b">b.
1<tab2>
<input type="radio" name="q9" <?php if (isset($q9) && $q9=="c") echo "checked";?> value="c">c.
2<tab2>
<input type="radio" name="q9" <?php if (isset($q9) && $q9=="d") echo "checked";?> value="d">d.
3<tab2>
<input type="radio" name="q9" <?php if (isset($q9) && $q9=="e") echo "checked";?> value="e">e.
4<br/>
</li>
</ol>
</div>
<div id="apDiv15">
Nume :
<input type="text" name="nume" id="nume" value="<?php if(isset($nume)) echo $nume; ?>" /><br />
Prenume: <input type="text" name="prenume" id="prenume" value="<?php if(isset($prenume)) echo
$prenume; ?>" /><br />
Clasa :
<select name="clasa">
<option value=""></option>
<option <?php if( isset($clasa) && $clasa=="a5-a" ) echo "selected"; ?> value="a5-a">a5-a</option>
<option <?php if( isset($clasa) && $clasa=="a6-a" ) echo "selected"; ?> value="a6-a">a6-a</option>
<option <?php if( isset($clasa) && $clasa=="a7-a" ) echo "selected"; ?> value="a7-a">a7-a</option>
<option <?php if( isset($clasa) && $clasa=="a8-a" ) echo "selected"; ?> value="a8-a">a8-a</option>
<option <?php if( isset($clasa) && $clasa=="a9-a" ) echo "selected"; ?> value="a9-a">a9-a</option>
<option <?php if( isset($clasa) && $clasa=="a10-a" ) echo "selected"; ?> value="a10-a">a10-
a</option>
<option <?php if( isset($clasa) && $clasa=="a11-a" ) echo "selected"; ?> value="a11-a">a11-
a</option>
<option <?php if( isset($clasa) && $clasa=="a12-a" ) echo "selected"; ?> value="a12-a">a12-
a</option>
</select><br />
<br/>
<tab4>
<input type="submit" value="Trimite Raspuns">
<tab4>
<a href="test.php" target="desk"><input type="button" value="Mai incearca o data"></a>
</div>
<div id="apDiv7">
<input type="checkbox" name="q6[]" <?php if ( isset($q6) && strpos($q6,'a') ) echo 'checked';?>
value="a">a.<br/>
</div>
<div id="apDiv8">

```

```

<input type="checkbox" name="q6[]" <?php if ( isset($q6) && strpos($q6,'b') ) echo 'checked';?>
value="b">b.<br/>
    
</div>
<div id="apDiv9">
    <input type="checkbox" name="q6[]" <?php if ( isset($q6) && strpos($q6,'c') ) echo 'checked';?>
value="c">c.
    <br/>
</div>
<div id="apDiv10">
    <input type="checkbox" name="q6[]" <?php if ( isset($q6) && strpos($q6,'d') ) echo 'checked';?>
value="d">d.
    <br/>
</div>
<div id="apDiv2">
    <input type="radio" name="q2" <?php if (isset($q2) && $q2=="a") echo "checked";?> value="a">
    a. 1
    <table>
    <input type="radio" name="q2" <?php if (isset($q2) && $q2=="b") echo "checked";?> value="b">
    b. 2
    <table>
    <input type="radio" name="q2" <?php if (isset($q2) && $q2=="c") echo "checked";?> value="c">
    c. 3
    <table>
    <input type="radio" name="q2" <?php if (isset($q2) && $q2=="d") echo "checked";?> value="d">
    d. 4 </div>
    <div id="apDiv5" align="center">
    <table width="523" height="234" border="1" align="center" cellpadding="1" cellspacing="1">
        <tr>
<td width="501">&nbsp;</td>
<td width="501">1</td>
<td width="501">2</td>
<td width="501">3</td>
<td width="501">4</td>
<td width="501">5</td>
<td width="501">6</td>
<td width="501">7</td>
<td width="501">8</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>

```

```

<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>

```

```

<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>

</table>
</div>

<div id="apDiv13"></div>
</form>
</span>
</body>
</html>

```