

# DRUMURI MINIME ÎN GRAF

## MANUALUL PROFESORULUI



### ALGORITMUL ROY FLOYD - PREZENTAREA PROBLEMEI

**UNELTE**

ADAUGĂ AEROPORT

ADAUGĂ DRUM

ȘTERGE AEROPORT

ȘTERGE DRUM

ȘTERGE HARTĂ

GRAF ASOCIAT

Gigel este angajatul unui mare aeroport din România. El se ocupă de relațiile cu publicul. Zilnic sute de clienți îl sună solicitând informații referitoare la cele mai mici costuri între oricare două aeroporturi. Salariul lui Gigel depinde de numărul de clienți corect informați, de aceea Gigel, cum ajunge la serviciu începe să calculeze rute. De curând aeroportul l-a dotat cu un calculator.

Ți se pune la dispoziție o hartă pe care sunt marcate aeroporturile din orașe. De asemenea pe hartă sunt marcate prin săgeți cursele aeriene existente între orașe. Sensul săgeții indică sensul de zbor, iar numărul specificat pe săgeată reprezintă costul zborului.

Arată pe Gigel să obțină un salariu cât mai mare, scriind un algoritm de găsire a drumurilor de cost minim între oricare două aeroporturi.

Construiește o hartă folosind bara de unelte și indicațiile din fereastra de informații. (butonul **i** din bara de sus).

Folosește butonul **Arată drum minim** pentru a observa drumuri minime găsite de algoritmul Roy Floyd.

**Obiective**

---

### ALGORITMUL LUI DIJKSTRA - IMPLEMENTARE

```

void dijkstra(int x0)
{
    int i, j, min, k, ok;
    int viz[NMAX], d[NMAX], tata[NMAX];
    for (i = 1; i <= n; i++) {
        d[i] = C[x0][i];
        tata[i] = x0;
        viz[i] = 0;
    }
    tata[x0] = 0;
    viz[x0] = 1; ok = 1;
    while (ok) {
        min = INFINIT;
        for (i = 1; i <= n; i++)
            if (!viz[i] && min > d[i]) {
                min = d[i];
                k = i;
            }
        if (min != INFINIT) {
            viz[k] = 1;
            for (i = 1; i <= n; i++)
                if (!viz[i] && d[i] > d[k] + C[k][i]) {
                    d[i] = d[k] + C[k][i];
                    tata[i] = k;
                }
            else ok = 0;
        }
    }
}
        
```

CONTROL ANIMAȚIE UNELTE

viz:	1	1	1	1	0	0	1	1	0	
tata:	1	2	3	0	1	8	3	4	7	8
d:	3	1	0	4	9	∞	6	7	9	
	1	2	3	4	5	6	7	8	9	

i: 
k:

min: 
ok:

Control hartă
Obiective

INFORMATICA  
CLASA A XI-A



# Cuprins

## 1. Terminologie

## 2. Structură generală

- 2.1. Obiective didactice
- 2.2. Conținut
- 2.3. Recomandări de structurare și predare

## 3. Obiecte de conținut - detalieri


- 3.1. **M**<sub>1.1</sub> – Algoritmul lui Dijkstra – Prezentarea problemei
- 3.2. **M**<sub>1.2</sub> – Algoritmul lui Dijkstra – Reprezentarea informațiilor
- 3.3. **M**<sub>1.3</sub> – Algoritmul lui Dijkstra – Implementare
- 3.4. **M**<sub>1.4</sub> – Algoritmul lui Dijkstra – Evaluare
- 3.5. **M**<sub>2.1</sub> – Algoritmul Roy-Floyd – Prezentarea problemei
- 3.6. **M**<sub>2.2</sub> – Algoritmul Roy-Floyd – Reprezentarea informațiilor
- 3.7. **M**<sub>2.3</sub> – Algoritmul Roy-Floyd – Implementare
- 3.8. **M**<sub>2.4</sub> – Algoritmul Roy-Floyd – Evaluare
- 3.9. **M**<sub>3.1</sub> – Algoritmul Belman-Ford – Prezentarea problemei
- 3.10. **M**<sub>3.2</sub> – Algoritmul Belman-Ford – Reprezentarea informațiilor
- 3.11. **M**<sub>3.3</sub> – Algoritmul Belman-Ford – Implementare
- 3.12. **M**<sub>3.4</sub> – Algoritmul Belman-Ford – Evaluare


## 4. Elemente de implementare a aplicației

## 5. Bibliografie








## 1. Terminologie


**Butoane definiție** –  – sunt amplasate totdeauna în partea din dreapta jos a ecranului și, atunci când sunt accesate, prezintă într-o fereastră de detaliu, definițiile termenilor necesari momentului respectiv.


**Butoane care indică obiectivele lecției respective** -  - sunt amplasate totdeauna în partea din dreapta jos a ecranului. Prin apăsarea lor, într-o fereastră detaliu se prezintă obiectivele lecției.


**Butoane de control a animației** -  - prin apăsarea butoanelor corespunzătoare:

- se execută animația “pas cu pas” înainte – 
- se execută animația “pas cu pas” înapoi – 
- se rulează animația în mod continuu – 
- se face pauză în executarea animației – 
- se oprește animația – 



**Buton care permite deschiderea/închiderea barei de unelte** –  – atunci când este accesat deschide/închide bara de unelte.

**Buton care permite deschiderea/închiderea ferestrei de control a hărților** –  – atunci când este accesat deschide/închide fereastra de control a hărților.

**Butoane cuprinse în bara de unelte care indică anumite acțiuni care trebuie executate asupra hărții** –  – atunci când sunt accesate realizează operația indicată.

**Butoane selectare item** –  – atunci când sunt accesate se realizează operația de selectare și afișare a item-ului corespunzător din test.

**Butoane de indicare a corectitudinii răspunsului**

- răspuns corect – 
- răspuns eronat – 

**Ferestre detaliu** – sunt ferestre care oferă informații suplimentare despre o anumită noțiune. Asupra unei ferestre detaliu se poate face "drag\_and\_drop" acționând asupra barei de titlu a ferestrei.

Exemplu :



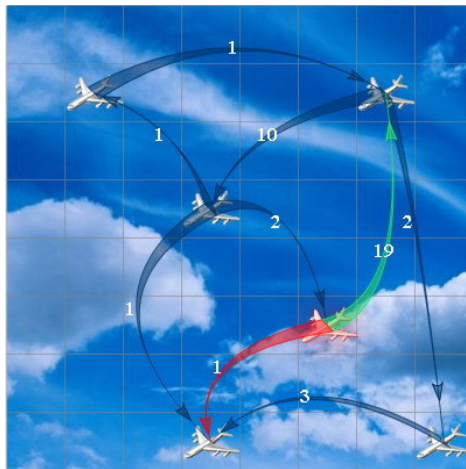
**Obiective** INCHIDE X

După parcurgerea lecției, elevul va fi capabil

- să construiască o configurație corectă pentru problema enunțată
- să urmărească pas cu pas modul în care sunt determinate drumurile de cost minim în configurația construită
- să identifice semnificația săgeților verzi și a celor roșii și rolul acestora pe parcursul determinării drumurilor de cost minim
- să descrie situații practice în care este necesară determinarea drumurilor de cost minim de la un vârf fixat la alte puncte

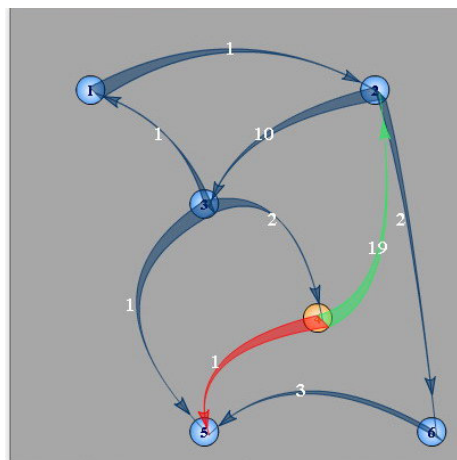
**Ferestre de animație** – sunt ferestre în care se prezintă o animație ce indică în mod sugestiv modul în care se desfășoară algoritmul

Exemplu :




**Ferestre afișare graf asociat** – sunt ferestre în care la acționarea butonului corespunzător din bara de unelte este afișat grafurile asociate hărții respective.

Exemplu :






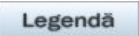
**Ferestre de control a hărților** – fereastră care conține butoane care permit încărcări de hărți predefinite, respectiv salvări/încărcări de hărți construite online.

**Butoane de selectare a modului de afișare al algoritmului** –  – atunci când sunt accesate realizează afișarea algoritmului în pseudocod sau în limbajul de programare indicat.

**Butoane pentru închis ferestre detalii** –  – sunt amplasate în colțul dreapta sus a ferestrelor de detalii iar acționarea lor duce la închiderea ferestrei.

**Ferestre eroare** – sunt ferestre care oferă informații despre încercări de încărcare a unor hărți cu un număr de noduri nepermis.


**Buton de informații** –  – la acționarea lui se deschide o fereastră de detalii care conține informații despre modul de utilizare a barei de unelte, precum și diferite avertismente sau atenționări.


**Buton legendă** –  – la acționarea lui se deschide o fereastră de detalii care conține descrierea elementelor grafice utilizate la momentul respectiv și semnificația lor.

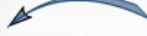



**Ferestre legendă** – sunt ferestre care oferă informații despre elementele grafice utilizate la momentul respectiv și semnificația lor:


**Legendă** INCHIDE X


 sunt arce selectate în drumul de cost minim de la sediul central (vârful de start) la celelalte aeroporturi (vârfuri).


 sunt arce prin care, pentru un anumit vârf (indicat de sensul arcului) se obține un cost minim la momentul respectiv.


 sunt arce normale

 reprezintă sediul central

 reprezintă vârful de start

 reprezintă un aeroport

 reprezintă un vârf

 reprezintă costul arcului pe care este amplasat



## 2. Structura generală

În acest capitol sunt prezentate obiectivele didactice care pot fi atinse utilizând acest material. În finalul prezentării sunt incluse câteva recomandări privind unele moduri în care ar putea fi combinate aceste momente pentru a obține o lecție.

### 2.1. Obiective didactice

Obiectiv	Detaliere
<b>Obiective de referință</b>	
<b>R1</b>	Analizarea modului de determinare a drumurilor minime în graf
<b>R2</b>	Realizarea aplicațiilor utilizând algoritmi specifici
<b>R3</b>	Urmărirea etapelor de realizare a unei aplicații
<b>Obiective operaționale</b>	
<b>OP1</b>	Definirea corectă a noțiunilor de graf neorientat, noduri, muchii,
<b>OP2</b>	Definirea corectă a noțiunilor de graf orientat, arce, drum, cost minim
<b>OP3</b>	Descrierea unor situații practice care să necesite determinarea unui drum minim de la un punct fixat la alte puncte
<b>OP4</b>	Identificarea semnificației săgeților (verzi/roșii) și rolul acestora pe parcursul determinării drumului e cost minim
<b>OP5</b>	Modelarea cu ajutorul teoriei grafurilor a situației practice prezentate
<b>OP6</b>	Alegerea reprezentării adecvate a grafului în memoria calculatorului
<b>OP7</b>	Descrierea structurilor de date necesare implementării algoritmului
<b>OP8</b>	Descrierea algoritmului Dijkstra
<b>OP9</b>	Implementarea algoritmului lui Dijkstra într-un limbaj de programare
<b>OP10</b>	Urmărirea execuției “pas cu pas” a algoritmului lui Dijkstra
<b>OP11</b>	Urmărirea valorilor variabilelor care intervin în desfășurarea algoritmului lui Dijkstra pe un exemplu
<b>OP12</b>	Determinarea drumului de cost minim folosind vectorul $tata$
<b>OP13</b>	Descrierea algoritmului Roy-Floyd
<b>OP14</b>	Implementarea algoritmului lui Roy-Floyd într-un limbaj de programare
<b>OP15</b>	Urmărirea execuției “pas cu pas” a algoritmului lui Roy-Floyd
<b>OP16</b>	Urmărirea valorilor variabilelor care intervin în desfășurarea algoritmului lui Roy-Floyd pe un exemplu
<b>OP17</b>	Descrierea algoritmului Bellman-Ford
<b>OP18</b>	Implementarea algoritmului lui Bellman-Ford într-un limbaj de programare
<b>OP19</b>	Urmărirea execuției “pas cu pas” a algoritmului lui Bellman-Ford
<b>OP20</b>	Urmărirea valorilor variabilelor care intervin în desfășurarea algoritmului lui Bellman-Ford pe un exemplu
<b>OP21</b>	Determinarea complexității timp a unui algoritm
<b>OP22</b>	Dezvoltarea gândirii algoritmice, logice, flexibile, creatoare
<b>OP23</b>	Dezvoltarea atenției concentrate și spiritului de observație



## 2.2 Conținut

Se prezintă lista obiectelor de conținut (notate cu M) și caracteristicile lor generale.

<b>M<sub>1.1</sub> – Algoritmul Dijkstra – prezentarea problemei</b>	
Obiective didactice	OP1, OP2, OP3, OP4, OP22, OP23
Timp de predare	20 min
Tip de interacțiune cu elevii	<ul style="list-style-type: none"><li>• <b>metode de comunicare orală:</b> expunere, conversație, algoritmizare, studiu de caz</li><li>• <b>metode de acțiune:</b> exercițiul, învățarea prin descoperire</li><li>• <b>procedee de instruire:</b> explicația în etapa de comunicare; învățarea prin descoperire dirijată, inductivă, experimentală, exercițiul de consolidare</li></ul>
Descriere	<ul style="list-style-type: none"><li>• prezentarea problemei</li><li>• exemplificarea problemei pe o hartă – predefinită sau construită pe loc</li><li>• construcția unor hărți utilizând bara de unelte</li><li>• urmărirea determinării drumului minim de la punctul de start la toate celelalte puncte de pe hartă</li></ul>
Cuvinte cheie	<ul style="list-style-type: none"><li>• hartă</li><li>• graf asociat</li><li>• graf neorientat, graf orientat</li><li>• muchie, arc</li><li>• cost, cost minim</li></ul>

<b>M<sub>1.2</sub> – Algoritmul Dijkstra – reprezentarea informațiilor</b>	
Obiective didactice	OP1, OP2, OP3, OP4, OP5, OP6, OP7, OP22, OP23
Timp de predare	30 min
Tip de interacțiune cu elevii	<ul style="list-style-type: none"><li>• <b>metode de comunicare orală:</b> expunere, conversație, studiu de caz</li><li>• <b>metode de acțiune:</b> exercițiul, învățarea prin descoperire</li><li>• <b>procedee de instruire:</b> explicația în etapa de comunicare; învățarea prin descoperire dirijată, inductivă, experimentală, exercițiul de consolidare</li></ul>
Descriere	<ul style="list-style-type: none"><li>• prezentarea noțiunii de graf asociat</li><li>• prezentarea noțiunii de matrice a costurilor</li><li>• prezentarea grafului asociat unei hărți</li><li>• prezentarea structurilor de date necesare</li></ul>
Cuvinte cheie	<ul style="list-style-type: none"><li>• graf asociat</li><li>• matricea costurilor</li></ul>





<b>M<sub>1.3</sub> – Algoritmul Dijkstra – implementare</b>	
Obiective didactice	OP1, OP2, OP3, OP4, OP7, OP8, OP9, OP10, OP11, OP22, OP23
Timp de predare	50 min + 20 min
Tip de interacțiune cu elevii	<ul style="list-style-type: none"><li>• <b>metode de comunicare orală:</b> expunere, conversație, studiu de caz</li><li>• <b>metode de acțiune:</b> exercițiul, învățarea prin descoperire</li><li>• <b>procedee de instruire:</b> explicația în etapa de comunicare; învățarea prin descoperire dirijată, inductivă, experimentală, exercițiul de consolidare</li></ul>
Descriere	<ul style="list-style-type: none"><li>• prezentarea algoritmului în limbaj natural</li><li>• prezentarea algoritmului în pseudocod/C/Pascal</li><li>• prezentarea evoluției variabilelor de memorie</li><li>• prezentarea în paralel a execuției programului și a determinării drumurilor de cost minim pe harta selectată sau pe graful asociat</li><li>• determinarea complexității algoritmului</li><li>• descrierea metodei de reconstituire a drumului utilizând vectorul <math>tata</math></li></ul>
Cuvinte cheie	<ul style="list-style-type: none"><li>• pseudocod</li><li>• complexitate</li></ul>

<b>M<sub>1.4</sub> – Algoritmul Dijkstra – evaluare</b>	
Obiective didactice	OP12, OP21, OP22, OP23
Timp de lucru	30 min
Tip de interacțiune cu elevii	<ul style="list-style-type: none"><li>• evaluare în formă scrisă prin intermediul calculatorului</li></ul>
Descriere	<ul style="list-style-type: none"><li>• test grilă</li></ul>
Cuvinte cheie	<ul style="list-style-type: none"><li>• hartă</li><li>• cost minim</li><li>• drum de cost minim</li><li>• matricea costurilor</li></ul>



<b>M<sub>2.1</sub> – Algoritmul Roy Floyd – prezentarea problemei</b>	
Obiective didactice	OP1, OP2, OP3, OP4, OP22, OP23
Timp de predare	20 min
Tip de interacțiune cu elevii	<ul style="list-style-type: none"><li>• <b>metode de comunicare orală:</b> expunere, conversație, algoritmizare, studiu de caz</li><li>• <b>metode de acțiune:</b> exercițiul, învățarea prin descoperire</li><li>• <b>procedee de instruire:</b> explicația în etapa de comunicare; învățarea prin descoperire dirijată, inductivă, experimentală, exercițiul de consolidare</li></ul>
Descriere	<ul style="list-style-type: none"><li>• prezentarea problemei</li><li>• exemplificarea problemei pe o hartă – predefinită sau construită pe loc</li><li>• construcția unor hărți utilizând bara de unelte</li><li>• urmărirea determinării drumului minim de la punctul de start la toate celelalte puncte de pe hartă</li></ul>
Cuvinte cheie	<ul style="list-style-type: none"><li>• hartă</li><li>• graf asociat</li><li>• graf neorientat</li><li>• graf orientat</li><li>• muchie</li><li>• arc</li><li>• cost</li><li>• cost minim</li></ul>

<b>M<sub>2.2</sub> – Algoritmul Roy Floyd – reprezentarea informațiilor</b>	
Obiective didactice	OP1, OP2, OP3, OP4, OP5, OP6, OP7, OP22, OP23
Timp de predare	30 min
Tip de interacțiune cu elevii	<ul style="list-style-type: none"><li>• <b>metode de comunicare orală:</b> expunere, conversație, studiu de caz</li><li>• <b>metode de acțiune:</b> exercițiul, învățarea prin descoperire</li><li>• <b>procedee de instruire:</b> explicația în etapa de comunicare; învățarea prin descoperire dirijată, inductivă, experimentală, exercițiul de consolidare</li></ul>
Descriere	<ul style="list-style-type: none"><li>• prezentarea noțiunii de graf asociat</li><li>• prezentarea noțiunii de matrice a costurilor</li><li>• prezentarea grafului asociat unei hărți</li><li>• prezentarea structurilor de date necesare</li></ul>
Cuvinte cheie	<ul style="list-style-type: none"><li>• graf asociat</li><li>• matricea costurilor</li></ul>



<b>M<sub>2.3</sub> – Algoritmul Roy Floyd – implementare</b>	
Obiective didactice	OP1, OP2, OP3, OP4, OP7, OP13, OP14, OP15, OP16, OP22, OP23
Timp de predare	30 min
Tip de interacțiune cu elevii	<ul style="list-style-type: none"><li>• <b>metode de comunicare orală:</b> expunere, conversație, studiu de caz</li><li>• <b>metode de acțiune:</b> exercițiul, învățarea prin descoperire</li><li>• <b>procedee de instruire:</b> explicația în etapa de comunicare; învățarea prin descoperire dirijată, inductivă, experimentală, exercițiul de consolidare</li></ul>
Descriere	<ul style="list-style-type: none"><li>• prezentarea algoritmului în limbaj natural</li><li>• prezentarea algoritmului în pseudocod/C/Pascal</li><li>• prezentarea evoluției variabilelor de memorie</li><li>• prezentarea în paralel a execuției programului și a determinării drumurilor de cost minim pe harta selectată sau pe graful asociat</li><li>• determinarea complexității algoritmului</li><li>• descrierea metodei de reconstituire a drumului utilizând vectorul <math>tata</math></li></ul>
Cuvinte cheie	<ul style="list-style-type: none"><li>• pseudocod</li><li>• complexitate</li></ul>

<b>M<sub>2.4</sub> – Algoritmul Roy Floyd – evaluare</b>	
Obiective didactice	OP12, OP21, OP22, OP23
Timp de lucru	20 min
Tip de interacțiune cu elevii	<ul style="list-style-type: none"><li>• evaluare în formă scrisă prin intermediul calculatorului</li></ul>
Descriere	<ul style="list-style-type: none"><li>• test grilă</li></ul>
Cuvinte cheie	<ul style="list-style-type: none"><li>• hartă</li><li>• cost minim</li><li>• drum de cost minim</li><li>• matricea costurilor</li></ul>



<b>M<sub>3.1</sub> – Algoritmul Belmann Ford – prezentarea problemei</b>	
Obiective didactice	OP1, OP2, OP3, OP4, OP22, OP23
Timp de predare	20 min
Tip de interacțiune cu elevii	<ul style="list-style-type: none"><li>• <b>metode de comunicare orală:</b> expunere, conversație, algoritmizare, studiu de caz</li><li>• <b>metode de acțiune:</b> exercițiul, învățarea prin descoperire</li><li>• <b>procedee de instruire:</b> explicația în etapa de comunicare; învățarea prin descoperire dirijată, inductivă, experimentală, exercițiul de consolidare</li></ul>
Descriere	<ul style="list-style-type: none"><li>• prezentarea problemei</li><li>• exemplificarea problemei pe o hartă – predefinită sau construită pe loc</li><li>• construcția unor hărți utilizând bara de unelte</li><li>• urmărirea determinării drumului minim de la punctul de start la toate celelalte puncte de pe hartă</li></ul>
Cuvinte cheie	<ul style="list-style-type: none"><li>• hartă</li><li>• graf asociat</li><li>• graf neorientat</li><li>• graf orientat</li><li>• muchie</li><li>• arc</li><li>• cost</li><li>• cost minim</li></ul>

<b>M<sub>3.2</sub> – Algoritmul Belmann Ford – reprezentarea informațiilor</b>	
Obiective didactice	OP1, OP2, OP3, OP4, OP5, OP6, OP7, OP22, OP23
Timp de predare	30 min
Tip de interacțiune cu elevii	<ul style="list-style-type: none"><li>• <b>metode de comunicare orală:</b> expunere, conversație, studiu de caz</li><li>• <b>metode de acțiune:</b> exercițiul, învățarea prin descoperire</li><li>• <b>procedee de instruire:</b> explicația în etapa de comunicare; învățarea prin descoperire dirijată, inductivă, experimentală, exercițiul de consolidare</li></ul>
Descriere	<ul style="list-style-type: none"><li>• prezentarea noțiunii de graf asociat</li><li>• prezentarea noțiunii de matrice a costurilor</li><li>• prezentarea grafului asociat unei hărți</li><li>• prezentarea structurilor de date necesare</li></ul>
Cuvinte cheie	<ul style="list-style-type: none"><li>• graf asociat</li><li>• matricea costurilor</li></ul>



<b>M<sub>3.3</sub> – Algoritmul Belmann Ford – implementare</b>	
Obiective didactice	OP1, OP2, OP3, OP4, OP7, OP17, OP18, OP19, OP20, OP22, OP23
Timp de predare	30 min
Tip de interacțiune cu elevii	<ul style="list-style-type: none"><li>• <b>metode de comunicare orală:</b> expunere, conversație, studiu de caz</li><li>• <b>metode de acțiune:</b> exercițiul, învățarea prin descoperire</li><li>• <b>procedee de instruire:</b> explicația în etapa de comunicare; învățarea prin descoperire dirijată, inductivă, experimentală, exercițiul de consolidare</li></ul>
Descriere	<ul style="list-style-type: none"><li>• prezentarea algoritmului în limbaj natural</li><li>• prezentarea algoritmului în pseudocod/C/Pascal</li><li>• prezentarea evoluției variabilelor de memorie</li><li>• prezentarea în paralel a execuției programului și a determinării drumurilor de cost minim pe harta selectată sau pe graful asociat</li><li>• determinarea complexității algoritmului</li><li>• descrierea metodei de reconstituire a drumului utilizând vectorul <math>tata</math></li></ul>
Cuvinte cheie	<ul style="list-style-type: none"><li>• pseudocod</li><li>• complexitate</li></ul>

<b>M<sub>3.4</sub> – Algoritmul Belmann Ford – evaluare</b>	
Obiective didactice	OP12, OP21, OP22, OP23
Timp de lucru	20 min
Tip de interacțiune cu elevii	<ul style="list-style-type: none"><li>• evaluare în formă scrisă prin intermediul calculatorului</li></ul>
Descriere	<ul style="list-style-type: none"><li>• test grilă</li></ul>
Cuvinte cheie	<ul style="list-style-type: none"><li>• hartă</li><li>• cost minim</li><li>• drum de cost minim</li><li>• matricea costurilor</li></ul>



### 2.3. Recomandări de structurare și predare

- **Planul unității de învățare 1** **Timp: 1 oră**

Obiect de conținut	Timp (min)
M <sub>1,1</sub>	20
M <sub>1,2</sub>	30

- **Planul unității de învățare 2** **Timp: 1 oră**

Obiect de conținut	Timp (min)
M <sub>1,3</sub>	50

- **Planul unității de învățare 3** **Timp: 1 oră**

Obiect de conținut	Timp (min)
M <sub>1,3</sub>	20
M <sub>1,4</sub>	30

- **Planul unității de învățare 4** **Timp: 1 oră**

Obiect de conținut	Timp (min)
M <sub>2,1</sub>	20
M <sub>2,2</sub>	30

- **Planul unității de învățare 5** **Timp: 1 oră**

Obiect de conținut	Timp (min)
M <sub>2,3</sub>	30
M <sub>2,4</sub>	20

- **Planul unității de învățare 6** **Timp: 1 oră**

Obiect de conținut	Timp (min)
M <sub>3,1</sub>	20
M <sub>3,2</sub>	30

- **Planul unității de învățare 7** **Timp: 1 oră**

Obiect de conținut	Timp (min)
M <sub>3,3</sub>	30
M <sub>3,4</sub>	20



### 3. Obiecte de conținut - detalieri

În continuare vom prezenta în detaliu modul de utilizare a elementelor din ferestrele lecției (navigare, elemente specifice, funcționarea aplicațiilor, etc.). Subliniem că navigarea elementară se face cu ajutorul butoanelor descrise în Capitolul 1 – Terminologie, al acestui manual. Nu ne vom referi la acestea decât spicuitiv.

#### 3.1. Algoritmul Dijkstra – prezentarea problemei

În acest obiect de conținut este prezentată o problemă practică care necesită utilizarea unui algoritm de determinare a drumurilor de cost minim. Tot aici este posibilă descrierea acestei situații pe o hartă și determinarea drumurilor de cost minim de la un aeroport la toate celelalte.

### ALGORITMUL LUI DIJKSTRA – PREZENTAREA PROBLEMEI

**UNELTE**

- ADAUGĂ AEROPORT
- ADAUGĂ DRUM
- ȘTERGE AEROPORT
- ȘTERGE DRUM
- ȘTERGE HARTĂ
- SETEAZĂ START
- GRAF ASOCIIAT

Domnul Ionescu are una dintre cele mai mari companii de transport aerian pentru mărfuri din lume. În multe orașe ale lumii are și aeroporturi proprii. Zilnic din orașul în care se află sediul central pleacă avioane cu marfă în toată lumea.

Pentru a reduce din cheltuieli, el angajează un expert (pe tine) care să găsească cele mai economice rute din orașul în care se află sediul central către fiecare dintre celelalte orașe.

Ți se pune la dispoziție o hartă pe care sunt marcate aeroporturile din orașe. De asemenea pe hartă sunt marcate prin săgeți cursele aeriene existente între orașe. Sensul săgeții indică sensul de zbor, iar numărul specificat pe săgeată reprezintă costul zborului.

Scrie un algoritm care să determine pentru fiecare oraș  $x$  care este costul minim pentru a transporta mărfuri de la sediul central la orașul  $x$ , precum și o rută de transport pentru care se obține costul minim.

Construiește o hartă folosind bara de unelte și indicațiile din fereastra de informații (butonul **i** din bara de sus).

Observă cum rulează algoritmul lui Dijkstra pe harta construită folosind controlul animației.

CONTROL ANIMAȚIE

Control hartă

Obiective

Ecranul este împărțit în cinci zone :

- în stânga sus zona unde se construiește/încarcă o hartă
- în dreapta sus zona unde este descrisă situația practică
- între cele două zone se găsește bara de unelte prin intermediul căreia poate fi editată orice hartă – numărul maxim permis de aeroporturi este 10



- în stânga jos se găsește fereastra în care sunt afișate mesaje cu privire la acțiunile care au fost executate prin intermediul barei de unelte
  - în dreapta jos se găsește fereastra care conține sarcinile de lucru
- Pe lângă aceste zone în fereastră mai apar câteva butoane și controale:




Control hartă

Obiective

controlul animației – cele cinci butoane au respectiv efectul: execuție, pauză, oprire, înainte un pas, înapoi un pas care acționat deschide o fereastră detaliu care conține informații cu privire la utilizarea editorului de hărți

la acționarea acestui buton se deschide fereastra care permite accesul la bibliotecile de hărți

la acționarea acestui buton se deschide fereastra detaliu pentru descrierea obiectivelor

Obiectul de conținut cuprinde descrierea problemei practice în limbaj natural, precum și sarcinile de lucru. Tot aici sunt date informații despre modalitățile de utilizare a editorului de hărți. Aceste informații se obțin acționând butonul . La acționarea lui (din oricare din pozițiile unde apare) se deschide o fereastră detaliu care conține help-ul corespunzător:

**Informații**
ÎNCHIDE X

**Control Hartă** - deschide o fereastră din care se pot efectua diferite operații cu harta, cum ar fi salvarea acesteia în memorie, încărcare, etc.

**Unelte**

**Adaugă aeroport** – adaugă un aeroport pe harta curentă

**Cum se face:** apasă pe zona din hartă unde vrei să plasezi un aeroport.

**Atenție:** Numărul maxim de aeroporturi ce se pot afla pe hartă este 10.

**Adaugă drum** – adaugă un drum între două aeroporturi de pe hartă

**Cum se face:** selectează un aeroport sursă apoi un aeroport destinație. În acest moment ar trebui să îți apară un drum între cele două aeroporturi. Prin deplasarea mouse-ului peste hartă se stabilește forma arcului. La un nou click se poate introduce costul arcului.

**Șterge aeroport** – șterge un aeroport de pe harta curentă

**Cum se face:** execută un click pe aeroportul pe care dorești să îl ștergi

**Șterge drum** – șterge drumul dintre două aeroporturi

**Cum se face:** execută click întâi pe aeroportul sursă, apoi pe aeroportul destinație.

**Șterge hartă** – șterge toate aeroporturile și liniile de zbor de pe hartă.

**Setează start** – setează vârful din care se începe aplicarea algoritmului lui Dijkstra

**Cum se face:** execută un click pe aeroportul de pornire.

**Graf asociat** – transformă harta într-un graf orientat

**Control animație**

**Rulează** - pornește aplicarea algoritmului lui Dijkstra pe exemplul creat de tine.

**Pauză** - oprește temporar aplicația. La un al doilea click continuă rularea aplicației.

**Stop** – oprește definitiv aplicația, resetând drumurile

**Pas înapoi** – execută un singur pas înapoi în aplicarea algoritmului lui Dijkstra

**Pas înainte** - execută un singur pas înainte în aplicarea algoritmului lui Dijkstra

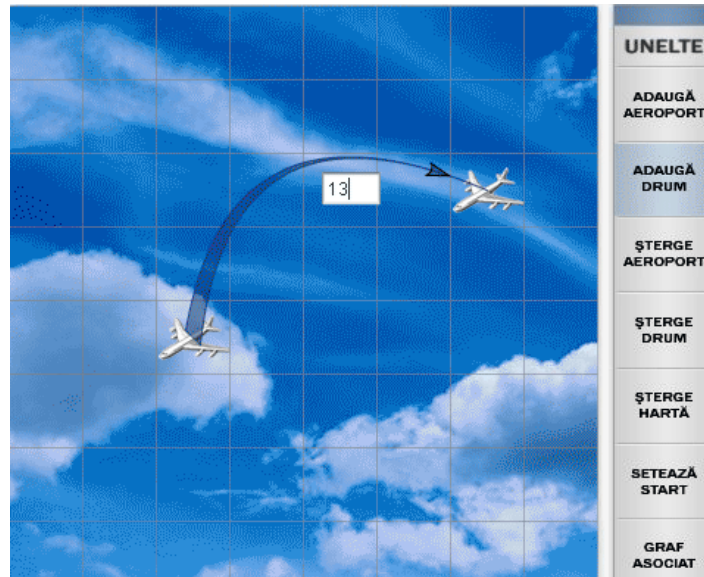
**Avertisment!** În partea de jos a paginii apar indicații și erori, în funcție de acțiunile tale. De exemplu, dacă apeși **Rulează** înainte de a seta aeroportul de start primești un avertisment. Un alt exemplu ar fi construirea drumului. Dacă ai început un drum, acesta trebuie finalizat, altfel apare un mesaj de eroare: "Trebuie să termini acțiunea începută!"

Pentru a crea o hartă proprie, operațiile indicate trebuie executate într-o anumită ordine și trebuie finalizate. Astfel, prima operație care se face este operația de **adăugare de aeroporturi** (maxim 10) pe hartă. După ce aeroporturile au fost adăugate se pot **adăuga drumuri** între diferite aeroporturi. Pentru aceasta se





selectează din bara de unelte operația corespunzătoare apoi, pe hartă, se selectează (click) cele două aeroporturi între care se dorește adăugarea unui drum – primul aeroport selectat este aeroportul de plecare (sursa). În momentul în care a fost selectat și al doilea aeroport (de sosire, destinația), între cele două aeroporturi apare o săgeată (de la sursă la destinație) care poate fi modificată executând operația *drag&drop* cu mouse-ul fără nici un buton apăsat. În momentul în care se execută un nou click, lângă săgeată apare o casetă text în care se introduce distanța dintre cele două aeroporturi ( $0 < d < 256$ ). Încheierea operației este semnalată prin apăsarea tastei ENTER.



Pe parcursul editării hărții se pot executa oricare dintre operațiile indicate în bara de unelte: se selectează operația dorită din bara de unelte apoi se execută operația/operațiile.


După finalizarea hărții, pentru a continua desfășurarea momentului de lecție, pot fi apelate operațiile:



Printr-un click pe unul din aeroporturile hărții se setează orașul în care este aeroportul în care se află sediul central al companiei. Acesta devine roșu. Acest aeroport poate fi schimbat pentru a determina o nouă configurație.



Acționarea acestui buton provoacă apariția grafului asociat hărții.

După indicarea aeroportului de plecare poate fi derulată animația care determină drumurile minime de la aeroportul de plecare la celelalte aeroporturi. Pentru aceasta avem la dispoziție controlul animației  cu cele cinci butoane descrise mai sus. Pe parcursul animației, colorarea săgeților în verde indică un arc luat în considerare pentru determinarea unui drum minim, iar un arc colorat în roșu indică faptul că arcul respectiv face parte din drumul minim de la aeroportul de start până la alt aeroport.



Butonul **Control hartă** permite executarea operațiilor de gestiune a hărților:

- încărcarea unei hărți dintr-o bibliotecă de hărți predefinite oferite de soft
- salvarea unei hărți construite de utilizator
- încărcarea unei hărți construite într-un moment anterior
- eliminarea unei hărți din memorie

După salvarea unei hărți în memorie aceasta devine accesibilă în orice moment al derulării aplicației.

Acționarea butonului **Obiective** permite apariția unei ferestre detaliu în care sunt indicate obiectivele pe care trebuie să le atingă elevul după parcurgerea acestui obiect de conținut.



### 3.2. Algoritmul Dijkstra – reprezentarea informațiilor

În acest obiect de conținut sunt prezentate noțiunile necesare implementării algoritmului: datele implicate, organizarea lor, semnificația notațiilor folosite. Fereastra principală este organizată astfel:

#### ALGORITMUL LUI DIJKSTRA – REPREZENTAREA INFORMAȚIILOR



Putem asocia problemei un graf orientat astfel:

- vârfurile grafului sunt aeroporturile; acestea vor fi numerotate de la 1 la  $N$  (unde  $N$  reprezintă numărul de aeroporturi);
- există arc de la vârful  $x$  la vârful  $y$ , dacă de la aeroportul corespunzător vârfului  $x$  există un zbor (arc) având ca destinație aeroportul corespunzător vârfului  $y$ .

Graful orientat asociat problemei este ponderat, deoarece putem asocia o funcție de cost, definită pe mulțimea arcelor cu valori în mulțimea numerelor reale strict pozitive, costul unui arc fiind egal cu costul zborului modelat de arcul respectiv.

Aeroportul de start este un vârf special al grafului, denumit vârf de start.

În limbaj de grafuri, problema poate fi formulată astfel:

Fie  $G = (X, V)$  un graf orientat,  $c: V \rightarrow \mathbb{R}_+$  o funcție de cost, prin care se asociază fiecărui arc o valoare reală strict pozitivă și  $x_0$  un vârf din graf. Să se determine câte un drum de cost minim de la  $x_0$  la fiecare dintre celelalte vârfuri ale grafului.

Costul unui drum este egal cu suma costurilor arcelor din care este constituit drumul.

Unelte      Matricea costurilor      Graful asociat      **Reprezentarea informațiilor**

Control hartă      Definiții      Obiective

- în stânga sus zona unde se construiește/încarcă o hartă și unde apare graful asociat hărții respective
- în dreapta sus zona unde sunt descrise noțiunile teoretice necesare asocierii unui graf precum și modalitatea în care se realizează aceasta; tot aici, la acționarea butonului **Reprezentarea informațiilor** se vor afișa datele necesare și modalitățile de organizare a lor
- în stânga jos se găsește fereastra în care sunt afișate mesaje cu privire la acțiunile care au fost executate prin intermediul barei de unelte

În cadrul acestui obiect de conținut bara de unelte nu mai este prezentă pe ecran tot timpul; ea poate fi adusă aici prin acționarea butonului **Unelte**.

Pentru derularea firească a obiectului de conținut, în primul rând se încarcă una dintre hărțile din bibliotecă sau dintre cele salvate la momentul anterior. Aceasta se realizează utilizând butonul **Control hartă**. Evident, și în cadrul acestui obiect de conținut poate fi construită o hartă utilizând facilitățile oferite de bara de unelte. După încărcarea/construirea hărții, prin acționarea butonului **Graful asociat** în fereas-



tra din dreapta apar noțiunile teoretice cu privire la modalitatea de a asocia hărții respective un graf. Acționarea butonului **Reprezentarea informațiilor** produce apariția în aceeași fereastră a descrierii datelor și organizării lor în vederea prelucrării:

Reprezentarea informațiilor:

1. Vom reprezenta graful prin matricea costurilor. Matricea costurilor este o matrice pătratică cu  $N$  linii și  $N$  coloane (unde  $N$  reprezintă numărul de vârfuri din graf), definită astfel:

$$C[i][j] = \begin{cases} \infty, & \text{dacă nu } \exists \text{ arc de la } i \text{ la } j \\ c(i,j), & \text{unde } c(i,j) \text{ reprezintă costul arcului } (i,j) \\ 0, & \text{dacă } i = j \end{cases}$$

2. Mulțimea vârfurilor selectate (pentru care a fost deja calculat drumul de cost minim) va fi reprezentată prin vectorul caracteristic  $viz$  cu  $N$  componente:

$$viz[i] = \begin{cases} 1, & \text{dacă nodul } i \text{ a fost selectat} \\ 0, & \text{altfel} \end{cases}$$

3. Vom reține pentru fiecare vârf  $i$ , costul drumului de cost minim de la  $x_0$  la  $i$ , drum care trece numai prin vârfuri selectate.

$$d[i] = \begin{cases} \text{costul drumului de cost minim de la } x_0 \text{ la } i, \\ \text{ce trece numai prin vârfuri selectate, } i = 1, n \end{cases}$$

4. Pentru a reconstitui drumurile de cost minim de la vârful de start la fiecare dintre celelalte vârfuri, vom folosi un vector  $tata$ , cu  $N$  componente:

$$tata[i] = \begin{cases} 0, & \text{dacă } i = x_0 \\ \text{vârful ce îl precedă pe } i \text{ pe drumul de cost minim de la } x_0 \text{ la } i \end{cases}$$

Acționarea butonului **Matricea costurilor** produce apariția unei ferestre detaliu în care este descrisă modul în care sunt inițializate elementele matricii costurilor precum și matricea costurilor atașată grafului asociat hărții respective.

**Matricea costurilor** INCHIDE X

Fiecare element al matricii este inițializat astfel:

$$C[i][j] = \begin{cases} \infty, & \text{dacă nu } \exists \text{ arc de la } i \text{ la } j \\ c(i,j), & \text{unde } c(i,j) \text{ reprezintă costul arcului } (i,j) \\ 0, & \text{dacă } i = j \end{cases}$$

În practică  $\infty$  ia o valoare suficient de mare astfel încât să fie mai mare decât costul drumului de lungime maximă din graf.

	1	2	3	4
1	0	$\infty$	10	1
2	10	0	11	$\infty$
3	$\infty$	11	0	10
4	1	10	$\infty$	0

Deoarece în acest obiect de conținut apar noțiuni utilizate în teoria grafurilor, am considerat util ca anumite elemente utilizate să fie reamintite elevilor. Acest lucru este realizat prin intermediul noului buton care apare **Definiții**. Acționarea lui produce apariția unei ferestre detaliu în care sunt reamintite noțiunile necesare:

- graf neorientat, vârfuri, muchii
- graf orientat, arce
- drum în graf, drum elementar în graf



**Definiții** INCHIDE X


- Un **graf neorientat** este o pereche **ordonată**  $G = (X, V)$ , unde  $X$  este o mulțime finită și nevidă de elemente numite **vârfuri (noduri)**, iar  $V$  este o mulțime de perechi de elemente ale lui  $X$ , numite **muchii**. O muchie având vârfurile  $i$  și  $j$  (numite **extremitățile** sale) este notată  $(i, j)$ .
- Un **graf orientat** este o pereche **ordonată**  $G = (X, V)$ , deosebirea față de grafurile neorientate constând în faptul că elementele lui  $V$  sunt **perechi ordonate** de vârfuri numite **arce**. Orice arc  $(i, j)$  are stabilit un sens de parcurgere și anume de la extremitatea sa inițială  $i$ , la extremitatea sa finală  $j$ .
- Se numește **drum** o succesiune de arce de forma  $(i_1, i_2), (i_2, i_3), \dots, (i_{n-1}, i_n)$ , notată prescurtat sub forma  $(i_1, i_2, \dots, i_n)$ . Lungimea drumului este egală cu numărul arcelor pe care le conține. Dacă toate vârfurile drumului sunt distincte, atunci drumul se numește **elementar**.

Ca la fiecare moment de conținut și aici există butonul **Obiective** la acționarea căruia, într-o fereastră detaliu, sunt afișate obiectivele pe care trebuie să le atingă elevul după parcurgerea acestui obiect de conținut.

**Obiective** INCHIDE X

După parcurgerea lecției, elevul va fi capabil

- ☛ să modeleze cu ajutorul teoriei grafurilor situația practică prezentată
- ☛ să aleagă o reprezentare adecvată a grafului în memoria calculatorului
- ☛ să descrie structurile de date necesare pentru implementarea algoritmului

**Buton de informații**  care conține informații despre modul de utilizare a barei de unelte, precum și diferite avertismente sau atenționări, este în continuare disponibil.



### 3.3. Algoritmul Dijkstra – implementare

În acest obiect de conținut este prezentată implementarea algoritmului lui Dijkstra în pseudocol, C și Pascal. Fereastra principală este organizată astfel:

```
void dijkstra(int x0)
{
    int i, j, min, k, ok;
    int viz[NMAX], d[NMAX], tata[NMAX];
    for (i = 1; i<=n; i++) {
        d[i] = c[x0][i];
        tata[i] = x0;
        viz[i] = 0;
    }
    tata[x0] = 0;
    viz[x0] = 1; ok = 1;
    while (ok) {
        min = INFINIT;
        for (i = 1; i<=n; i++)
            if (!viz[i] && min>d[i]) {
                min = d[i];
                k = i;
            }
        if (min != INFINIT) {
            viz[k] = 1;
            for (i = 1; i<=n; i++)
                if (!viz[i] && d[i]>d[k]+c[k][i]) {
                    d[i] = d[k]+c[k][i];
                    tata[i] = k;
                }
        }
        else ok = 0;
    }
}
```

- în stânga sus zona unde se construiește/încarcă o hartă și unde apare, la cerere, graful asociat hărții respective
- în dreapta sus zona unde este descrisă implementarea algoritmului în C, pseudocod sau Pascal
- sub această zonă se găsesc trei butoane care permit selectarea modalității de afișare a implementării algoritmului
- în dreapta jos se găsește fereastra în care sunt afișate mesaje cu privire la acțiunile care au fost executate prin intermediul barei de unelte
- în partea din stânga jos a ferestrei principale se află fereastra în care vor fi afișate valorile tuturor variabilelor care apar în implementarea algoritmului
- sub zona care conține harta se găsesc butoanele pentru controlul animației descrise la început
- lângă controlul animației apare butonul care permite activarea / dezactivarea barei de unelte

Butonul permite, așa cum am arătat activarea ferestrei pentru lucrul cu hărți.



Pe lângă aceste elemente fereastra mai conține trei butoane care permit deschiderea unor ferestre detaliu care conțin:

**Algoritm**

Descrierea în limbaj natural a algoritmului, precum și semnificația colorării în roșu/verde a arcelor în timpul desfășurării animației: arcele roșii au fost selectate ca făcând parte din drumul minim, cele verzi vor participa la calculul la pasul următor.

**Drum**

Descrierea în limbaj natural a modului de reconstituire a drumului de cost minim în vederea afișării acestuia. Sunt date în pseudocod atât implementarea recursivă cât și implementarea nerecursivă a algoritmului de reconstituire a drumului.

**Complexitate**

Descrierea calculului ordinului de complexitate a algoritmului lui Dijkstra

După construirea/încărcarea unei hărți și stabilirea aeroportului de plecare (start), operații permise fie cu ajutorul barei de unelte, fie prin intermediul ferestrei de control a hărților, se poate trece la vizualizarea algoritmului în una dintre cele trei modalități permise (C/pseudocod/Pascal). Pentru a înțelege cât mai bine modul în care a fost implementat algoritmul, obiectul de conținut permite vizualizarea separată a zonelor implementării alese. Pentru aceasta este suficient să fie poziționat cursorul mouse-ului în fereastra care descrie codul. În acest mod pot fi puse în evidență cele patru zone ale codului:



- declarațiile de variabile
- inițializarea variabilelor
- căutarea distanței minime
- actualizarea distanțelor

prin colorarea în albastru a zonei respective:

```
void dijkstra(int x0)
{
    int i, j, min, k, ok;
    int viz[NMAX], d[NMAX], tata[NMAX];
    for (i = 1; i <= n; i++) {
        d[i] = C[x0][i];
        tata[i] = x0;
        viz[i] = 0;
    }
    tata[x0] = 0;
    viz[x0] = 1; ok = 1;
    while (ok) {
        min = INFINIT;
        for (i = 1; i <= n; i++) Căutare distanță minimă
            if (!viz[i] && min > d[i]) {
                min = d[i];
                k = i;
            }
        if (min != INFINIT) {
            viz[k] = 1;
            for (i = 1; i <= n; i++)
                if (!viz[i] && d[i] > d[k] + C[k][i]) {
                    d[i] = d[k] + C[k][i];
                    tata[i] = k;
                }
            else ok = 0;
        }
    }
}
```

Următorul moment în desfășurarea lecției este urmărirea desfășurării algoritmului cu ajutorul animației lansate prin intermediul butoanelor din controlul animației.



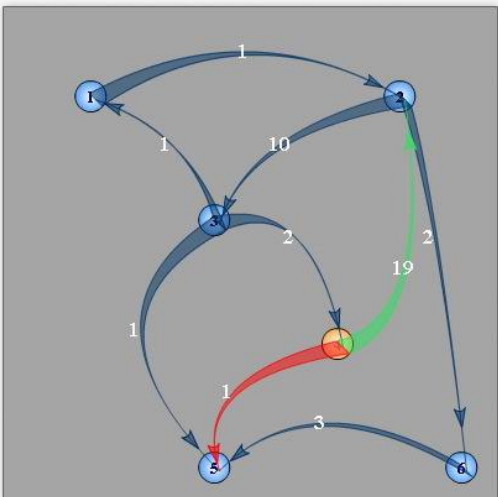
Animația poate fi lansată fie prin acționarea butonului  care declanșează execuția algoritmului, fie prin acționarea butonului  care permite execuția "pas cu pas" a algoritmului.






În momentul lansării în execuție în zona destinată variabilelor apar valorile variabilelor (neinițializate), apoi, pe măsura desfășurării algoritmului, aceste valori sunt actualizate în fiecare moment:

viz :	0	0	0	0	0	0				i :	6	
	1	2	3	4	5	6	7	8	9		k :	
tata :	3	3	3	3	3	3					min :	
	1	2	3	4	5	6	7	8	9		ok :	
d :	∞	1	0	∞	∞	∞						
	1	2	3	4	5	6	7	8	9			

În același timp în zona codului, o bandă portocalie indică în fiecare moment instrucțiunea care este executată:

**ALGORITMUL LUI DIJKSTRA - IMPLEMENTARE**



CONTROL ANIMAȚIE      UNELTE

viz :	0	0	0	1	1	0
	1	2	3	4	5	6
tata :	4	4	4	0	4	4
	1	2	3	4	5	6
d :	∞	19	∞	0	1	∞
	1	2	3	4	5	6

```

void dijkstra(int x0)
{
    int i, j, min, k, ok;
    int viz[NMAX], d[NMAX], tata[NMAX];
    for (i = 1; i<=n; i++) {
        d[i] = c[x0][i];
        tata[i] = x0;
        viz[i] = 0;
    }
    tata[x0] = 0;
    viz[x0] = 1; ok = 1;
    while (ok) {
        min = INFINIT;
        for (i = 1; i<=n; i++)
            if (!viz[i] && min>d[i]) {
                min = d[i];
                k = i;
            }
        if (min != INFINIT) {
            viz[k] = 1;
            for (i = 1; i<=n; i++)
                if (!viz[i] && d[i]>d[k]+c[k][i]) {
                    d[i] = d[k]+c[k][i];
                    tata[i] = k;
                }
            else ok = 0;
        }
    }
}

```

Control hartă
Obiective


În același timp, pe hartă sau pe graful asociat hărții (ca în imaginea anterioară) sunt indicate prin culori arcele care participă în acel moment la calcule – prin culoarea roșie sunt indicate arcele deja selectate pentru un drum minim, iar prin culoarea verde arcele care sunt luate în calcul pentru următoarele calcule.

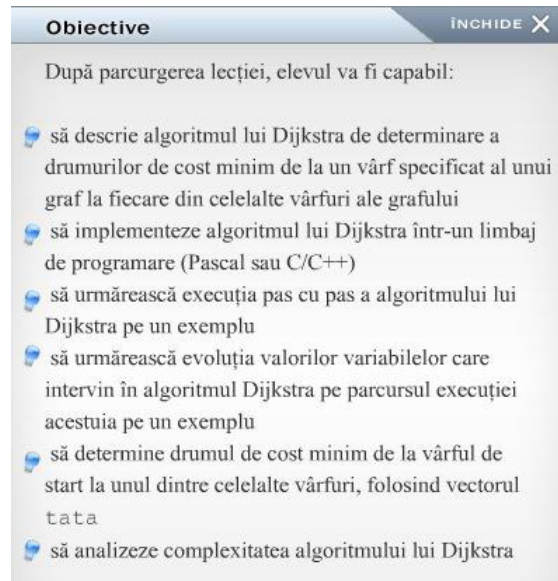
Oricare dintre butoanele descrise mai sus sunt accesibile în orice moment al animației, însă acționarea unor instrumente din bara de butoane poate duce la resetarea animației. Astfel, de exemplu, dacă în timpul derulării animației se






dorește adăugarea unui nou aeroport, operația duce la oprirea animației, deoarece operația dorită produce un alt graf.

Ca la fiecare moment de conținut și aici există butonul  la acționarea căruia, într-o fereastră detaliu, sunt afișate obiectivele pe care trebuie să le atingă elevul după parcurgerea acestui obiect de conținut.



**Buton de informații**  care conține informații despre modul de utilizare a barei de unelte, precum și diferite avertismente sau atenționări, este în continuare disponibil.



### 3.4. Algoritmul Dijkstra – evaluare

Acest obiect de conținut constă dintr-un set de exerciții recapitulative de diferite tipuri, numerotate de la 1 la 9 de forma **1**.

Exercițiile 1, 3, 4, 5, 7 sunt de tip interactiv, solicitând completarea unor câmpuri cu valori care se cer determinate pe baza informațiilor furnizate:

- exercițiul 1 cere determinarea drumului minim pe baza informațiilor furnizate de vectorul  $tata$
- exercițiul 3 cere determinarea matricii costurilor pe baza unei hărți
- exercițiul 4 cere determinarea a două drumuri distincte de cost minim de la vârful de start la unul din vârfuri
- exercițiul 5 indică o situație pe hartă din timpul desfășurării algoritmului și cere determinarea vectorului distanțelor  $d$  după următoarea selecție
- exercițiul 7 indică o situație pe hartă din timpul desfășurării algoritmului și cere determinarea următorului vârf care va fi selectat

Exercițiile 2, 6, și 8 sunt de tip test grilă cu răspunsuri de tip “complement simplu”, adică doar o variantă de răspuns corectă.

Exercițiile se selectează cu ajutorul mouse-ului.

**ALGORITMUL LUI DIJKSTRA – EVALUARE** i

1

2

3

4

5

6

7

8

Se consideră vectorul  $tata$  de mai jos, care reține pentru fiecare vârf informații despre vârful care îl precedă pe drumul de cost minim de la vârful de plecare la acesta. Reconstituie drumul de la vârful de plecare la vârful 13, indicând, în ordine vârfurile parcurse.

2	5	5	5	6	0	3	7	1	9	10	5	2	3
1	2	3	4	5	6	7	8	9	10	11	12	13	14

1	2	3	4	5	6	7	8	9	10	11	12	13	14

TEST NOU    EVALUARE

După rezolvarea celor opt exerciții, pentru a determina corectitudinea rezolvării se accesează butonul **EVALUARE**. În urma evaluării în stânga fiecărui exercițiu

- 26 -



apare semnul ✓ dacă exercițiul a fost corect rezolvat, respectiv semnul ✗ dacă exercițiul nu a fost rezolvat corect. În cazul în care exercițiul nu a fost rezolvat corect, la accesarea butonului care indică exercițiul incorect, în fereastra exercițiului apare și soluția corectă.

**ALGORITMUL LUI DIJKSTRA - EVALUARE** ⓘ

✓ 1  
✗ 2  
✓ 3  
✗ 4  
✗ 5  
✗ 6  
✗ 7  
✗ 8

Se consideră harta de mai jos pe care sunt figurate aeroporturi și costuri. Pe hartă este surprins un moment din timpul desfășurării algoritmului lui Dijkstra, prin săgeți roșii fiind indicate drumurile calculate, iar prin săgeți verzi drumuri care sunt luate în calcul pentru următorul pas. Completați vectorul  $d$  după selectarea vârfului 5.

$d$        
1 2 3 4 5

$d = \{0, 2, 6, 3, 4\}$

TEST NOU EVALUARE

După terminarea evaluării, pentru a se genera un nou test se accesează butonul **TEST NOU**. Pentru fiecare exercițiu există un număr de posibilități de selectare în mod aleator, astfel încât probabilitatea de a selecta același test de evaluare format din nouă exerciții pentru două calculatoare alăturate este extrem de mică.

Ambele butoane ( **EVALUARE** și **TEST NOU** ) pot fi accesate în orice moment al evaluării, însă după accesarea butonului de evaluare nu mai este permis accesul la nici unul din cele nouă exerciții și trebuie generat un nou test.

Și în cazul acestui obiect de conținut există **butonul de informații** ⓘ la accesarea căruia într-o fereastră detaliu sunt date informații cu privire la modul de lucru în timpul rezolvării celor opt exerciții de evaluare.



**3.5. Algoritmul Roy-Floyd – prezentarea problemei**

**3.6. Algoritmul Roy-Floyd – reprezentarea informațiilor**

**3.7. Algoritmul Roy-Floyd – implementare**

**3.8. Algoritmul Roy-Floyd – evaluare**

**3.9. Algoritmul Belmann-Ford – prezentarea problemei**

**3.10. Algoritmul Belmann-Ford – reprezentarea informațiilor**

**3.11. Algoritmul Belmann-Ford – implementare**

**3.12. Algoritmul Belmann-Ford – evaluare**

Pentru obiectele de conținut 3.5. – 3.8. elementele prezentate în obiectele de conținut 3.1. – 3.4. rămân valabile, singura deosebire fiind algoritmul prezentat, și anume algoritmul Roy-Floyd.

În mod analog, pentru obiectele de conținut 3.9. – 3.12. elementele prezentate în obiectele de conținut 3.1. – 3.4. rămân valabile, algoritmul prezentat fiind de data aceasta algoritmul Belmann-Ford.



## 4. Elemente de implementare a aplicației

Tehnologiile utilizate în acest proiect sunt:

- **HTML**
- **JavaScript**
- **Flash**
- **XML**
- **ActionScript**
- **Shared Objects**

Întreaga parte de interactivitate a fost realizată utilizând facilitățile puse la dispoziție de platforma **Flash Player 7**. Fiecare moment al lecției este creat ca fiind un fișier Flash distinct iar pentru biblioteca de hărți am folosit un fișier XML extern, al cărui format este ușor de înțeles și editat, fără a fi necesar un editor specializat.

Pentru parcurgerea pas cu pas a oricărui algoritm prezentat am folosit un vector de mișcări, pliate pe algoritm, ce simulează modul în care procedează mediul în care se rulează programul. Astfel, unui pas  $X$  îi corespunde o acțiune în vectorul de mișcări, acțiune ce poate fi o modificare a unei variabile sau o mutare simplă pe cod. Pentru a obține desfășurarea cursivă a animației pe cod a fost suficientă crearea unei funcții care este apelată din secundă în secundă și care incrementează numărul pasului ce trebuie afișat. Practic, am implementat fiecare algoritm în ActionScript.

Pentru operații ce au necesitat stocarea unor date pe harddisk și utilizarea lor în mai multe momente s-a folosit tehnologia **Shared Objects**. Pentru salvarea hărților în memorie pentru a putea fi folosite și în alte momente am folosit tehnologia Shared Objects în Flash. Fiecare hartă este transformată în graf iar fiecare graf este transformat într-un obiect care, după ce este automat serializat, este scris într-un fișier pe harddisk, fiind accesibil și la o altă rulare a aplicației. Pentru încărcarea hărților și transpunerea lor pe hartă am inversat procesul. Astfel, dintr-un fișier de pe harddisk, prin deserializare automată, se extrage un obiect care este transformat într-un graf. În cazul în care graful rezultat este compatibil cu momentul (are numărul de vârfuri mai mic sau egal cu numărul maxim de vârfuri pentru momentul respectiv și costurile din graf corespund costurilor acceptate de moment) acesta este pliat pe hartă.



## 4. Bibliografie

- **Cormen Th., Leiserson Ch., Rivest R.**, Introduction to Algorithms, MIT, 1990
- **Cerchez Em.**, Informatica, Culegere de probleme pentru liceu, Editura Polirom, Iași, 2002
- **Livoschi L., Georgescu H.**, Bazele informaticii. Algoritmi. Elaborare și complexitate. București, 1985
- **Lica D., Boriga R., Popescu-Anastasiu D., ș.a.**, Fundamentele programării (Culegere de probleme pentru clasele IX-XI). Editura L&S Infomat, București, 2003