

Proiect de lecție / unitate de învățare

Unitatea de învățământ: Colegiul Național Petru Rareș Suceava

Disciplina: Informatică

Clasa: a X-a intensiv informatică

Data¹: _____

Profesor: Tiberiu Socaciu

Unitatea de învățare: Vectori

Tema lecției: Vectori. Number of Visible People in a Queue

Tipul lecției: practică, de laborator

Scopul lecției: sistematizare și consolidare abilități și deprinderi practice

Competențe generale:

- a) **CG1:** Elaborarea algoritmilor de rezolvare a problemelor
- b) **CG2:** Implementarea algoritmilor într-un limbaj de programare

Competențe specifice:

- a) **CS1:** Analiza problemei în scopul identificării metodei de programare adecvate pentru rezolvarea problemei
- b) **CS2:** Aplicarea creativă a metodelor de programare pentru rezolvarea unor probleme intradisciplinare sau interdisciplinare sau a unor probleme cu aplicabilitate practică
- c) **CS3:** Utilizarea tehnicilor moderne în implementarea aplicațiilor

Obiective operaționale². La finalul lecției, elevul poate:

- a) **O1:** să demonstreze posibilitatea de a folosi corect noțiunile prezentate
- b) **O2:** să aplice noțiunile prezentate într-un caz concret
- c) **O3 :** să analizeze relațiile dintre funcțiile programului
- d) **O4:** să descrie în limbaj științific mecanismele de interdependență dintre modulele programului
- e) **O5:** să formuleze propria opțiune pentru o decizie optimă

Strategii didactice:

- a) **Mijloace de învățământ:** calculatoarele din laboratoarele de informatică, directorul FTP_Elevi -> Materiale_Profesorii prezent pe toate cele 30 stații de lucru, tableta grafică precum și resursele incluse în prezentul plan de lecție³ respectiv:

Link pentru enunțul și tema lecției	TBC ⁴
Link pentru mind map-ul lecției ⁵	TBC ⁶
Link pentru produsul final al lecției ⁷	TBC ⁸
Link și cod QR pentru activitatea independentă ⁹ și pentru tema acasă a elevilor	TBC ¹⁰
Generator de numere aleatoare	https://g.co/kgs/s3eGCQ4

- b) **Metode de învățământ:** expunerea, algoritimizarea, demonstrația, modelarea, problematizarea, conversația euristică, descoperirea sau brainstormingul.
- c) **Forme de activitate:** individuală, frontală, lucrul pe grupe pentru orele de laborator, lucrul pe grupe mici, lucrul în perechi.
- d) **Metode de evaluare:** evaluare orală, evaluare scrisă, observarea sistematică a elevilor sau prin teste și/sau fișe de evaluare.

Bibliografie:

- a) cursul *Make Technology your Friend*
- b) site-ul LeetCode, www.leetcode.ro
- c) Eusebiu Toader, Tiberiu Socaciu, Pregătirea interviului de angajare ca programator C++. 33 teme de interviu rezolvate, MatrixRom, 2023,

¹ Se va completa, după caz

² rezultă din derivarea competențelor specifice

³ link-ul lecției va fi disponibil elevilor și înafara orelor de curs pentru ca aceștia să poată accesa oricând materialele folosite

⁴ Se va completa ulterior

⁵ plan de lecție alternativ, necesar pentru susținerea activităților A și B din ”dirijarea învățării”

⁶ Se va completa ulterior

⁷ activitatea C

⁸ Se va completa ulterior

⁹ în clasă

¹⁰ Se va completa ulterior

<https://www.matrixrom.ro/produs/pregatirea-interviului-de-angajare-ca-programator-c-33-teme-de-interviu-rezolvate/>.

Succesiunea activităților de instruire

Nr. crt.	Evenimentul didactic	Durata (min.)	Obiective operaționale	Activitatea de instruire		Strategia didactică			
				Activitatea profesorului	Activitatea elevilor	Mijloace de învățământ	Metode și procedee	Forme de organizare	Forme de evaluare
1.	Moment organizatoric Prezentarea temei abordate și a obiectivelor lecției Captarea atenției	2		Salutul, efectuarea prezenței, verificarea funcționării rețelei de calculatoare, a FTP-ului. Prezentarea temei și a obiectivelor lecției.		AdServio	Conversația euristică	Frontal	Observarea sistematică a elevilor
2.	Dirijarea învățării	10	O1	Activitatea A	Notează lecția pe caiet sau calculator, adresează întrebări	Videoproiector	Învățare activă, prin cooperare	Frontal	
		10	O2	Activitatea B	Notează lecția pe caiet sau calculator, adresează întrebări	Videoproiector	Învățare activă, prin cooperare	Individual Frontal	Observarea sistematică a elevilor
		10	O3	Activitatea C	Notează lecția pe caiet sau calculator, adresează întrebări	Videoproiector	Învățare activă, prin cooperare	Frontal	
3	Transferul cunoștințelor/ consolidarea cunoștințelor.	10	O4	Problematizare legată de relația dintre funcțiile programului. Se invită un elev la tablă pentru a realiza un modul necesar rezolvării problemei	Implementează funcția pe calculator	Rețeaua de calculatoare Videoproiector	Algoritmizarea	Individual Frontal	Orala
4	Fixarea cunoștințelor și asigurarea feedback-ului	7	O5				Problematizare a	Frontal	Scrisa
5	Tema pentru acasă	1	O1, O2, O3, O4, O5	Solicită elevilor să completeze ca temă pentru acasă o problema de pe LeetCode, aleasa aleator, impreuna Genereaza sarcina de lucru pentru acasa in Classroom	Noteaza tema de casa pe caiet au calculator	Generator de numere aleatoare Google Classrom			Scrisa

Enunt

Sunt n persoane care stau la coadă și au numărat de la 0 la $n - 1$ în ordine **de la stânga la dreapta**. Vi se oferă o înălțimi matrice de numere întregi distincte în cazul în care înălțimi[i] reprezintă înălțimea persoanei V_i se oferă o înălțimi matrice de numere întregi **distincte** în cazul în care înălțimi[i] reprezintă înălțimea persoanei i^{th} .

O persoană poate vedea o altă persoană la dreapta lor în coada de așteptare în cazul în care toată lumea în între este mai scurtă decât ambele dintre ele. Mai formal, persoana i^{th} poate vedea persoana j^{th} dacă $i < j$ și $\min(\text{înălțimi}[i], \text{înălțimi}[j]) > \max(\text{înălțimi}[i+1], \text{înălțimi}[i+2], \dots, \text{înălțimi}[j-1])$.

Returnați un răspuns matrice de lungime n unde răspunsul[i] este numărul de persoane pe care persoana i^{th} le poate vedea în dreapta lor în coadă.

Exemplu 1:



Intrare: înălțimi = [10,6,8,5,11,9]

Ieșire: [3,1,2,1,1,0]

Explicație:

Persoana 0 poate vedea persoana 1, 2 și 4.

Persoana 1 poate vedea persoana 2.

Persoana 2 poate vedea persoana 3 și 4.

Persoana 3 poate vedea persoana 4.

Persoana 4 poate vedea persoana 5.

Persoana 5 nu poate vedea pe nimeni, deoarece nimeni nu este la dreapta lor.

Exemplu 2:

Intrare: înălțime = [5,1,2,3,10]

Ieșire: [4,1,1,1,0]

Constrângeri:

- $n == \text{înălțime.lungime}$
- $1 \leq n \leq 10^5$
- $1 \leq \text{înălțime}[i] \leq 10^5$

Toate valorile înălțimilor sunt unice.

Rezolvare

```
# include <iostream>
using namespace std;

int heights[] = {10, 6, 8, 5, 11, 9};
int n = 6;

int persoana_unu_o_vede_pe_persoana_doi(
    int indice_prima_persoana, int indice_a_doua_persoana)
{
    int i = indice_prima_persoana;
    int j = indice_a_doua_persoana;
    /*
    am un trapez dreptunghic de latura de la i la j
    cu baza heights[i] si cealalta baza heights[j]
    unghiul laturii oblice este alfa. daca alfa > 0
    primul e mai scund ca al doilea
    dar daca alfa < 0 primul e mai inalt
```

```

daca alfa = 0 au aceasi inaltime
x indicele unei persoane care eventual obtureaza
(for x de la i+1 pana la j-1)
    beta unghiul de la primul pana la x

alfa = arctg ( ( heights[j] - heights[i] ) / ( j - i ) )
beta = arctg ( ( heights[j] - heights[x] ) / ( j - x ) )

    x ma obtureaza <=> beta >= alfa <=> arctg(...) >= arctg(...)
    <=> ( heights[j] - heights[i] ) / ( j - i ) >= ( heights[j] - heights[x] ) /
( j - x )

concluzie: i il vede pe j <=> nu exista x, pentru care x sa obtureze privirea
de la i la j <=> nu exista x= i+1,j-1:
( heights[j] - heights[i] ) / ( j - i ) >=
( heights[j] - heights[x] ) / ( j - x )

daca exista x=i+1, j-1 pentru care (for + if)
( heights[j] - heights[i] ) / ( j - i ) >=
( heights[j] - heights[x] ) / ( j - x )
atunci raspunsul e NU (fals)
altfel e DA (adevarat)
*/
for (int x = i + 1; x <= j - 1; x++)
    if (( heights[j] - heights[i] ) / ( j - i ) >=
        ( heights[j] - heights[x] ) / ( j - x ))
        return 0;
return 1;
}

int main()
{
for (int i = 0; i < n - 1; i++)
{
    int contor = 0;
    for (int j = i + 1; j < n; j++)
        if (persoana_unu_o_vede_pe_persoana_doi(i, j))
            contor++;
    cout << contor << " ";
}
cout << 0; // pentru ultimul, ca nu e prins in for-ul cu i!!!
return 0;
}

```