Anexa nr. ……la ordinul ministrului educației, cercetării şi inovării nr. ……/……..

MINISTERUL EDUCAȚIEI, CERCETĂRII ŞI INOVĂRII

PROGRAME ŞCOLARE

INFORMATICĂ

CLASA A XI-A

CICLUL SUPERIOR AL LICEULUI

Filiera teoretică, profil real, specializarea: Matematică-informatică

Filiera vocațională, profil militar, specializarea: Matematică-informatică

Aprobată prin ordin al ministrului
Nr. _____/_____

Bucureşti, 2009

NOTA DE PREZENTARE

Prezentul document cuprinde programa şcolară de *Informatică*, clasa a XI-a, pentru specializarea matematică-informatică din cadrul filierelor teoretică, profilul real, şi vocațională, profil militar MApN.

În conformitate cu planurile-cadru de învățământ pentru ciclul superior al liceului, aprobate prin ordinul ministrului educației şi cercetării nr. 3410/ 16.03.2009, la această specializare se alocă **4 ore pe săptămână** pentru studierea disciplinei *Informatică*. Cele 4 ore vor fi distribuite astfel: **o oră de teorie şi 3 ore activități practice desfăşurate în laborator.**

Programa şcolară este parte componentă a curriculumului național. Aceasta reprezintă documentul şcolar de tip reglator – instrument de lucru al profesorului – care stabileşte, pentru fiecare disciplină, oferta educațională care urmează să fie realizată în bugetul de timp alocat pentru un parcurs şcolar determinat, în conformitate cu statutul şi locul disciplinei în planul-cadru de învățământ.

Programa şcolară pentru învățământul liceal are următoarele componente:

- notă de prezentare
- competențe cheie europene vizate prin studiul disciplinei
- competențe generale
- valori şi atitudini
- competențe specifice şi conținuturi
- sugestii metodologice.

Competențele generale se definesc pentru fiecare disciplină de studiu şi au un grad ridicat de generalitate şi complexitate.

Valorile şi atitudinile orientează dimensiunile axiologică şi afectiv-atitudinală aferente formării personalității elevului din perspectiva fiecărei discipline. Realizarea lor concretă derivă din activitatea didactică permanentă a profesorului, constituind un element implicit al acesteia.

Competențele specifice se formează pe parcursul unui an de studiu, sunt deduse din competențele generale şi sunt etape în formarea acestora. ***Conținuturile învățării*** sunt mijloace prin care se urmăreşte formarea competențelor specifice şi implicit a competențelor generale propuse. Unitățile de conținut sunt organizate tematic.

Sugestiile metodologice propun modalități de organizare a procesului de predare-învățare-evaluare. Pentru formarea competențelor specifice pot fi organizate diferite tipuri de activități de învățare. Exemplele de activități de învățare sunt construite astfel încât să pornească de la experiența concretă a elevului şi să se integreze unor strategii didactice adecvate contextelor variate de învățare.

În elaborarea prezentei programei şcolare au fost respectate principiile de proiectare curriculară, specifice curriculumului național, valorificându-se în acelaşi timp tendințele domeniului pe plan internațional şi opinii ale unor profesori cu o bogată experiență didactică.

În procesul de predare-învățare, activitatea va fi orientată pe probleme: analiza unor situații practice (generale sau specifice unui anumit domeniu), identificarea fluxului informațional, elaborarea unui model algoritmic de rezolvare, implementarea algoritmilor într-un limbaj de programare.

Exemplele utilizate la predare vor fi preponderent alese de la disciplinele din aria curriculară *Matematică şi ştiințe ale naturii*, în colaborare cu profesorii de la aceste discipline.

COMPETENȚELE CHEIE EUROPENE VIZATE PRIN STUDIUL DISCIPLINEI

Pe baza rezultatelor studiilor efectuate la nivelul Comisiei Europene au fost stabilite opt competențe cheie, fiind precizate, pentru fiecare competență cheie, cunoştințele, deprinderile şi atitudinile care trebuie dobândite, respectiv formate elevilor în procesul educațional.

Aceste competențe cheie răspund obiectivelor asumate pentru dezvoltarea sistemelor educaționale şi de formare profesională în Uniunea Europeană şi, ca urmare, stau la baza stabilirii curriculumului pentru educația de bază.

Principalele competențele cheie europene vizate prin studiul disciplinei sunt:

Competențe în matematică şi competențe de bază în ştiințe şi tehnologie Competențe digitale

COMPETENȚE GENERALE

1. Identificarea datelor care intervin într-o problemă şi aplicarea algoritmilor fundamentali de prelucrare a acestora
2. Elaborarea algoritmilor de rezolvare a problemelor
3. Implementarea algoritmilor într-un limbaj de programare

VALORI ŞI ATITUDINI

1. Exprimarea unui mod de gândire creativ, în structurarea şi rezolvarea problemelor
2. Conştientizarea impactului social, economic şi moral al informaticii
3. Formarea obişnuințelor de a recurge la concepte şi metode informatice de tip algoritmic specifice în abordarea unei varietăți de probleme
4. Manifestarea unor atitudini favorabile față de ştiință şi de cunoaştere în general
5. Manifestarea inițiativei şi disponibilității de a aborda sarcini variate

COMPETENȚE SPECIFICE ŞI CONȚINUTURI

1. Identificarea datelor care intervin într-o problemă şi aplicarea algoritmilor fundamentali de prelucrare a acestora

Competențe specifice	Conținuturi [1]
1.1. Analizarea unei probleme în scopul identificării datelor necesare şi alegerea modalităților adecvate de structurare a datelor care intervin într-o problemă 1.2. Identificarea avantajelor utilizării diferitelor metode de structurare a datelor necesare pentru rezolvarea unei probleme 1.3. Utilizarea şirurilor de caractere şi a structurilor de date neomogene în modelarea unor situații problemă 1.4. Implementarea unor algoritmi de prelucrare a a şirurilor de caractere şi a structurilor neomogene 1.5. Transpunerea unei probleme din limbaj natural în limbaj de grafuri, folosind corect terminologia specifică 1.6. Descrierea unor algoritmi simpli de verificare a unor proprietăți specifice grafurilor	**Tablouri bidimensionale*** **Şiruri de caractere** • Particularități de memorare a şirurilor de caractere • Subprograme predefinite de prelucrare a şirurilor de caractere **Structuri de date neomogene (struct/record)** • Rezolvarea unor probleme cu caracter practic. **Liste** • Reprezentarea grafică a structurilor de tip listă • Operații specifice • Stiva şi coada • Aplicații cu implementare statică **Grafuri orientate şi neorientate** • Terminologie şi proprietăți (graf orientat şi neorientat, adiacență, incidență, grad; lanț, lanț elementar, drum, drum elementar, ciclu,ciclu elementar, circuit, circuit elementar; subgraf, graf parțial; conexitate, tare conexitate, arbore, arbore parțial, etc) • Reprezentarea în memorie a grafurilor (matrice de adiacență, liste de adiacență, lista muchiilor/arcelor) • Parcurgerea grafurilor. Aplicații.

2. Elaborarea algoritmilor de rezolvare a problemlor

Competențe specifice	Conținuturi
2.1 Analiza problemei în scopul identificării metodei de programare adecvate pentru rezolvarea problemei 2.2 Construirea unor soluții pentru probleme simple care se rezolvă cu ajutorul metodelor de programare 2.3 Aplicarea creativă a metodelor de programare pentru rezolvarea unor probleme intradisciplinare sau interdisciplinare, sau a unor probleme cu aplicabilitate practică	**Metoda de programare *Divide et Impera*** • Prezentare generală • Aplicații **Metoda de programare *Backtracking*** • Prezentare generală • Implementarea iterativă sau recursivă a algoritmilor de generare a produsului cartezian, permutărilor, combinărilor, aranjamentelor, submulțimilor unei mulțimi

[1] Conținuturile sunt prezentate în tabele, grupate pe competențe şi asocierea acestora este obligatorie. Este la decizia cadrului didactic/ a autorului de manual şcolar ordinea abordării conținuturilor, cu respectarea logicii interne a domeniului.

3. Implementarea algoritmilor într-un limbaj de programare

Competențe specifice	Conținuturi
3.1 Recunoaşterea situațiilor în care este necesară utilizarea unor subprograme 3.2 Analiza problemei în scopul identificării subproblemelor acesteia 3.3 Utilizarea corectă a subprogramelor predefinite şi a celor definite de utilizator 3.4 Descrierea şi aplicarea mecanismului recursivității 3.5 Identificarea avantajelor şi a dezavantajelor aplicării tehnicii recursive în implementarea unor rezolvări	**Subprograme** • Structura şi a modul de definire al subprogramelor • Declararea şi apelul subprogramelor • Transferul parametrilor la apel (prin valoare şi referință) • Returnarea valorilor de către subprograme • Variabile locale şi globale • Aplicații folosind subprograme **Recursivitate** • Definire. Exemplificare • Mecanisme de implementare • Aplicații cu subprograme recursive

* Se studiază în vederea corelării programei curente cu programa anterioară.

SUGESTII METODOLOGICE

Predarea informaticii va fi orientată pe *rezolvarea de probleme*, utilizându-se preponderent metode activ-participative şi punându-se accent pe *analiza problemei*. Pentru buna desfăşurare a orelor şi aplicarea programei se sugerează următoarele activități de învățare:

- discuții despre activități cotidiene şi modelarea acestora în limbaj informatic;
- activități de dezvoltare a deprinderilor de organizare a informației în diferite structuri de date (şiruri de caractere, articole, liste alocate static, stive şi cozi);
- identificarea modalităților eficiente de reprezentare a datelor necesare pentru rezolvarea unei probleme
- descompunerea rezolvării unei probleme în subprobleme;
- prezentarea unor situații practice familiare elevilor care pot fi modelate în termenii teoriei grafurilor
- reprezentarea grafică a grafurilor, listelor şi ilustrarea prin exemple reprezentate grafic a diferitelor noțiuni şi proprietăți specifice
- demonstrarea modului de realizare a operațiilor elementare specifice diferitelor structuri de date pe exemple reprezentate grafic.
- aplicarea, pe exemple relevante, a algoritmilor fundamentali din teoria grafurilor
- adaptarea creativă a algoritmilor fundamentali de prelucrare a datelor pentru rezolvarea unei probleme
- identificarea unor situații în care alegerea unui algoritm prezintă avantaje în raport cu altul;
- exersarea creării şi aplicării programelor pentru rezolvarea unor probleme întâlnite de elevi în studiul altor discipline şcolare;
- evidențierea greşelilor tipice în elaborarea algoritmilor;
- proiectarea/modelarea unor algoritmi şi implementarea acestora;
- testarea şi analizarea comportamentului programelor pentru diferite date de intrare;
- încurajarea discuțiilor purtate între elevi, exprimarea şi ascultarea părerilor fiecăruia.

Conținuturile din prezenta programă vor fi susținute prin rezolvarea unor probleme intradisciplinare sau interdisciplinare, respectiv probleme cu aplicabilitate practică în viața cotidiană.

Elevii care manifestă interes pentru grafică pe calculator pot fi încurajați să realizeze prezentări în PowerPoint, Flash sau alte programe, care să ilustreze teoria şi aplicații ale acesteia.

Exemple de aplicații recomandate pentru unele conținuturi:

Grafuri orientate şi grafuri neorientate

1. Algoritmi simpli de verificare a însuşirii terminologiei sau de verificare a unor proprietăți specifice grafurilor (de exemplu, calcularea gradelor vârfurilor unui graf, verificarea faptului că o succesiune de vârfuri reprezintă un lanț, drum, ciclu sau circuit în graf etc.)

Probleme practice, care solicită aplicarea algoritmilor de parcurgere a grafurilor (conexitatea unei structuri relaționale între mai multe persoane, puncte accesibile ale unei rețele de comunicație etc.)

Recursivitate

1. Exemplificarea recursivității prin recurențe matematice şi modele intuitive
2. Compararea implementarii recursive cu cea iterativă în alegerea algoritmului de rezolvare a unei probleme

Metoda de programare *Divide et Impera*

1. Sortarea eficientă a unei mulțimi de valori aplicând metoda *Divide et Impera* (sortarea rapidă sau sortarea prin interclasare)
2. Căutarea eficientă a unui element într-o mulțime ordonată aplicând metoda *Divide et Impera* (căutarea binară)

Metoda de programare *Backtracking*

1. Colorarea unei hărți
2. Problema celor n dame
3. Plata unei sume folosind monede de valori date

Anexa nr. ……la ordinul ministrului educației, cercetării şi inovării nr. ……/……..

MINISTERUL EDUCAȚIEI, CERCETĂRII ŞI INOVĂRII

PROGRAME ŞCOLARE

INFORMATICĂ

CLASA A XI-A

CICLUL SUPERIOR AL LICEULUI

Filiera teoretică, profil real, specializarea: Matematică-informatică intensiv informatică

Filiera vocațională, profil militar, specializarea: Matematică-informatică intensiv informatică

Aprobată prin ordin al ministrului
Nr. _____/_____

Bucureşti, 2009

NOTA DE PREZENTARE

Prezentul document conține programa școlară pentru disciplina *Informatică*, pentru clasa a XI-a, ciclul superior al liceului, studiată în filiera teoretică, la profilul real, specializarea matematică-informatică, intensiv informatică, precum şi la filiera vocațională, profil militar, specializarea matematică-informatică intensiv informatică.

În baza planurilor-cadru pentru ciclul superior al liceului, aprobate prin **Ordinul ministrului educației, cercetării şi inovării nr. 3410/16.03.2009**, la clasele de matematică-informatică – intensiv informatică, acestei discipline i se alocă în total **7 ore** / săptămână, distribuite astfel**: trei ore pentru studiu teoretic şi 4 ore pentru activități practice, dintre care 3 se vor defăşura** cu **colectivul de elevi organizat pe grupe**.

Activitățile practice vor fi desfăşurate **obligatoriu în laborator.**

Programa şcolară este parte componentă a curriculumului național. Aceasta reprezintă documentul şcolar de tip reglator – instrument de lucru al profesorului – care stabileşte, pentru fiecare disciplină, oferta educațională care urmează să fie realizată în bugetul de timp alocat pentru un parcurs şcolar determinat, în conformitate cu statutul şi locul disciplinei în planul-cadru de învățământ.

Programa şcolară pentru învățământul liceal are următoarele componente:

- notă de prezentare
- competențe cheie europene vizate prin studiul disciplinei
- competențe generale
- valori şi atitudini
- competențe specifice şi conținuturi
- sugestii metodologice.

Competențele generale se definesc pentru fiecare disciplină de studiu şi au un grad ridicat de generalitate şi complexitate.

Valorile şi atitudinile orientează dimensiunile axiologică şi afectiv-atitudinală aferente formării personalității elevului din perspectiva fiecărei discipline. Realizarea lor concretă derivă din activitatea didactică permanentă a profesorului, constituind un element implicit al acesteia.

Competențele specifice se formează pe parcursul unui an de studiu, sunt deduse din competențele generale şi sunt etape în formarea acestora. ***Conținuturile învățării*** sunt mijloace prin care se urmăreşte formarea competențelor specifice şi implicit a competențelor generale propuse. Unitățile de conținut sunt organizate tematic.

Sugestiile metodologice propun modalități de organizare a procesului de predare-învățare-evaluare. Pentru formarea competențelor specifice pot fi organizate diferite tipuri de activități de învățare. Exemplele de activități de învățare sunt construite astfel încât să pornească de la experiența concretă a elevului şi să se integreze unor strategii didactice adecvate contextelor variate de învățare.

În elaborarea prezentei programei şcolare au fost respectate principiile de proiectare curriculară, specifice curriculumului național, valorificându-se în acelaşi timp tendințele domeniului pe plan internațional şi opinii ale unor profesori cu o bogată experiență didactică.

COMPETENȚELE CHEIE EUROPENE VIZATE PRIN STUDIUL DISCIPLINEI

Pe baza rezultatelor studiilor efectuate la nivelul Comisiei Europene au fost stabilite opt competențe cheie, fiind precizate, pentru fiecare competență cheie, cunoştinţele, deprinderile şi atitudinile care trebuie dobândite, respectiv formate elevilor în procesul educațional.

Aceste competențe cheie răspund obiectivelor asumate pentru dezvoltarea sistemelor educaționale şi de formare profesională în Uniunea Europeană şi, ca urmare, stau la baza stabilirii curriculumului pentru educația de bază.

Principalele competențele cheie europene vizate prin studiul disciplinei sunt:

Competențe în matematică şi competențe de bază în ştiințe şi tehnologie Competențe digitale

COMPETENȚE GENERALE

1. Identificarea datelor care intervin într-o problemă şi aplicarea algoritmilor fundamentali de prelucrare a acestora 2. Elaborarea algoritmilor de rezolvare a problemelor 3. Implementarea algoritmilor într-un limbaj de programare

VALORI ŞI ATITUDINI

1. Exprimarea unui mod de gândire creativ, în structurarea şi rezolvarea problemelor

2. Conştientizarea impactului social, economic şi moral al informaticii

3. Formarea obişnuinţelor de a recurge la concepte şi metode informatice de tip algoritmic specifice în abordarea unei varietăți de probleme

4. Manifestarea unor atitudini favorabile față de ştiinţă şi de cunoaştere în general

5. Manifestarea inițiativei şi disponibilității de a aborda sarcini variate

COMPETENȚE SPECIFICE ŞI CONȚINUTURI

1. Identificarea datelor care intervin într-o problemă şi aplicarea algoritmilor fundamentali de prelucrare a acestora

Competențe specifice	Conținuturi [1]
1.1. Transpunerea unei probleme din limbaj natural în limbaj de grafuri, folosind corect terminologia specifică 1.2. Analizarea unei probleme în scopul identificării datelor necesare şi alegerea modalităților adecvate de structurare a datelor care intervin într-o problemă 1.3. Descrierea unor algoritmi simpli de verificare a unor proprietăți specifice grafurilor 1.4. Descrierea algoritmilor fundamentali de prelucrare a grafurilor şi implementarea acestora într-un limbaj de programare 1.5. Descrierea operațiilor specifice listelor simplu înlănțuite şi elaborarea unor subprograme care să implementeze aceste operații 1.6. Descrierea operațiilor specifice structurilor arborescente şi elaborarea unor subprograme care să implementeze aceste operații 1.7. Analizarea în mod comparativ a avantajelor utilizării diferitelor metode de structurare a datelor necesare pentru rezolvarea unei probleme 1.8. Aplicarea în mod creativ a algoritmilor fundamentali în rezolvarea unor probleme concrete	**Structuri de date alocate dinamic*** **Grafuri neorientate şi grafuri orientate** • Terminologie (graf neorientat, graf orientat, lanț, lanț elementar, drum, drum elementar, ciclu, ciclu elementar, circuit, circuit elementar, grad, graf parțial, subgraf, conexitate, tare conexitate, arbore, graf ponderat, arbore parțial, arbore parțial de cost minim) • Tipuri speciale de grafuri (graf complet, graf hamiltonian, graf eulerian, graf bipartit, graf turneu) • Reprezentarea grafurilor (matrice de adiacență, liste de adiacență, lista muchiilor, matricea costurilor) • Algoritmi de prelucrare a grafurilor - Parcurgerea grafurilor în lățime şi în adâncime - Determinarea componentelor conexe ale unui graf neorientat - Determinarea componentelor tare conexe ale unui graf orientat - Determinarea matricei lanțurilor/drumurilor - Determinarea drumurilor de cost minim într-un graf (algoritmul lui Dijkstra, algoritmul Roy-Floyd) - Arbori parțiali de cost minim (algoritmul lui Kruskal sau algoritmul lui Prim) **Structuri de date arborescente** • Arbori cu rădăcină (definiție, proprietăți, reprezentare cu referințe ascendente, reprezentare cu referințe descendente) • Arbori binari (definiție, proprietăți specifice; reprezentarea arborilor binari cu referințe descendente; operații specifice) • Tipuri speciale de arbori binari - Arbore binar complet – definiție, proprietăți, reprezentare secvențială - Arbore binar de căutare – definiție, proprietăți, operații specifice (inserare nod, ştergere nod, căutare element) - Heap-uri – definiție, proprietăți, operații specifice (inserare nod, extragerea nodului cu cheie maximă/minimă)

[1] Conținuturile sunt prezentate în tabele, grupate pe competențe şi asocierea acestora este obligatorie. Este la decizia cadrului didactic/ a autorului de manual şcolar ordinea abordării conținuturilor, cu respectarea logicii interne a domeniului.

2. Elaborarea algoritmilor de rezolvare a problemelor

Competențe specifice	Conținuturi
2.1 Analiza problemei în scopul identificării metodei de programare adecvate pentru rezolvarea problemei 2.2 Aplicarea creativă a metodelor de programare pentru rezolvarea unor probleme intradisciplinare sau interdisciplinare, sau a unor probleme cu aplicabilitate practică 2.3 Analiza comparativă a eficienței diferitelor metode de rezolvare a aceleiași probleme și alegerea unui algoritm eficient de rezolvare a unei probleme	**Metode de programare** • Metoda de programare *Greedy* (descrierea generală a metodei, utilitate, aplicații) • Metoda de programare *Backtracking* (descrierea generală a metodei, utilitate, aplicații) • Metoda de programare ***Divide et Impera**** • Metoda programării dinamice (descrierea generală a metodei, utilitate, aplicații) **Analiza eficienței unui algoritm**

3. Implementarea algoritmilor într-un limbaj de programare

Competențe specifice	Conținuturi
3.1 Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării 3.2 Utilizarea tehnicilor moderne în implementarea aplicațiilor	**Rezolvarea unor probleme cu caracter practic** pentru grafuri, arbori, metode de programare **Elemente de programare orientată pe obiecte** • Principiile programării orientate pe obiecte • Clase și obiecte (definire, utilizare, operații specifice)

* Se studiază în vederea corelării programei curente cu programa anterioară

SUGESTII METODOLOGICE

Predarea informaticii va fi orientată pe *rezolvarea de probleme*, utilizându-se preponderent metode activ-participative și punându-se accent pe *analiza problemei*: analiza unor situații practice (generale sau specifice unui anumit domeniu), identificarea fluxului informațional, elaborarea unui model algoritmic de rezolvare, implementarea algoritmilor într-un limbaj de programare.

Pentru buna desfășurare a orelor și aplicarea programei se sugerează următoarele activități de învățare:

- discuții despre activități cotidiene și modelarea acestora în limbaj algoritmic;
- activități de dezvoltare a deprinderilor de organizare a informației în diferite structuri de date;
- identificarea modalităților eficiente de reprezentare a datelor necesare pentru rezolvarea unei probleme
- descompunerea rezolvării unei probleme în subprobleme;
- prezentarea unor situații practice familiare elevilor care pot fi modelate în termenii teoriei grafurilor;
- reprezentarea grafică a grafurilor, listelor, arborilor și ilustrarea prin exemple reprezentate grafic a diferitelor noțiuni și proprietăți specifice;
- demonstrarea modului de realizare a operațiilor elementare specifice diferitelor structuri de date pe exemple reprezentate grafic;
- aplicarea algoritmilor fundamentali din teoria grafurilor pe exemple relevante;
- adaptarea creativă a algoritmilor fundamentali de prelucrare a datelor pentru rezolvarea unei probleme;
- identificarea unor situații în care alegerea unui algoritm prezintă avantaje în raport cu altul;
- exersarea creării și aplicării programelor pentru rezolvarea unor probleme întâlnite de elevi în studiul altor discipline școlare;

- evidențierea greşelilor tipice în elaborarea algoritmilor;
- proiectarea/modelarea unor algoritmi şi implementarea acestora;
- implementarea structurilor de date alocate dinamic;
- testarea şi analizarea comportamentului programelor pentru diferite date de intrare;
- încurajarea discuțiilor purtate între elevi, exprimarea şi ascultarea părerilor fiecăruia.

Conținuturile din prezenta programă vor fi susținute prin rezolvarea unor probleme intradisciplinare sau interdisciplinare, respectiv probleme cu aplicabilitate practică în viața cotidiană.

Exemple de aplicații recomandate

Exemplele utilizate la predare vor fi preponderent alese din aria curriculară "Matematică şi ştiințe ale naturii", în colaborare cu profesorii de *Matematică*, *Fizică*, *Chimie* şi *Biologie*.

I. Grafuri orientate şi grafuri neorientate

1. Algoritmi simpli de verificare a însuşirii terminologiei sau de verificare a unor proprietăți specifice grafurilor (de exemplu, calcularea gradelor vârfurilor unui graf, verificarea faptului că o succesiune de vârfuri reprezintă lanț, drum, ciclu sau circuit în graf, identificarea tuturor ciclurilor de lungime 3 într-un graf, verificarea proprietății de graf complet sau graf turneu, etc.)
2. Probleme practice, care solicită aplicarea creativă a algoritmilor din teoria grafurilor, cum ar fi:
   - Determinarea unei modalități de conectare a unor calculatoare în rețea astfel încât costurile de conectare să fie minime
   - Determinarea unui traseu de lungime minimă între două localități a căror poziție pe hartă este specificată
   - Determinarea unei modalități de transmitere a unui mesaj într-o interrețea astfel încât numărul total de servere prin intermediul cărora este transmis mesajul să fie minim.
   - Determinarea structurii relaționale a unui grup de persoane

II. Structuri de date arborescente

1. Aplicații simple care să necesite implementarea operațiilor elementare pe structuri arborescente cum ar fi: parcurgerea unui arbore în scopul identificării tuturor nodurilor cu o anumită proprietate, determinarea înălțimii unui arbore, copierea unui arbore, etc.
2. Probleme mai complexe, în care elevii să identifice eficiența utilizării structurilor de date arborescente, cum ar fi: realizarea eficientă a unui dicționar, sortarea unei secvențe de valori cu ajutorul *heap*-urilor (*heapsort*), optimizarea algoritmului lui *Kruskal* prin organizarea muchiilor ca *heap*.

III. Metoda de programare *Greedy*

1. Problema rucsacului în variantă continuă
2. Determinarea arborelui parțial de cost minim (algoritmul lui Kruskal, algoritmul lui Prim)

IV. Metoda de programare *Backtracking*

1. Generarea permutărilor, combinărilor, aranjamentelor, funcțiilor surjective, partițiilor unui număr, partițiilor unei mulțimi
2. Generarea tuturor posibilităților de a ieşi dintr-un labirint
3. Generarea tuturor grafurilor parțiale ale unui graf
4. Determinarea tuturor ciclurilor hamiltoniene într-un graf

V. Metoda programării dinamice

1. Determinarea unui subşir crescător de lungime maximă
2. Înmulțirea optimală a unui şir de matrice.
3. Problema rucsacului în varianta discretă
4. Algoritmul Roy-Floyd de determinare a drumurilor de cost minim între oricare două vârfuri ale grafului

VI. Analiza comparativă a rezolvării unei probleme prin diferite metode de programare. De exemplu, problema determinării unui traseu de la vârful unui triunghi către baza acestuia, astfel încât suma elementelor care aparțin traseului să fie minimă; deplasările posibile sunt din elementul curent la unul dintre elementele situate sub el, în stânga sau în dreapta.