Prefață

Cei care vor utiliza această carte mai întâi ar trebui să parcurgă cu atenție manualul destinat elevilor din clasa a 9-a. Totuşi, mai ales pentru cei care nu au urmat fidel conținutul programei din clasa a 9-a, am început cu o recapitulare a celor mai importante cunoştințe studiate în această clasă. După săptămânile afectate recapitulării, recomandăm rezolvarea problemelor de la sfârşitul acestui capitol pentru ca atât profesorii, dar mai ales elevii să poată să-şi dea seama de nivelul cunoştințelor pe care le au.

Este important ca de la început să clarificăm un aspect important: nu este vorba de un manual obişnuit care, pe baza unei programe şcolare, oferă cunoştințe acoperitoare, intră în detalii care sunt necesare elevilor pentru a înțelege şi a dobândi concepte fundamentale. Această carte se adresează acelor copii care sunt doritori de cunoştințe suplimentare, doresc să învețe mai repede şi să atingă performanțe în informatică, cu precădere în algoritmică şi programare. În consecință, cunoştințele teoretice sunt prezentate sumar, sunt „orientate" spre scopul clar de a crea deprinderile necesare pentru a „ataca" cât mai repede problemele propuse spre rezolvare din domeniu.

Elevii din România cărora le place programarea pot participa la o serie de concursuri, simpozioane, tabere unde pot învăța şi pot arăta ceea ce ştiu. Olimpiadele de informatică sunt cu precădere de algoritmică şi programare. Prima condiție pe care un elev trebuie s-o îndeplinească pentru a avea succes la aceste olimpiade este să fie în stare să se gândească singur, să se concentreze şi să aibă curaj pentru implementarea unei soluții originale. Fiecare proiect de algoritm, fiecare program este o creație nouă şi trebuie să îndeplinească toate cerințele şi restricțiile impuse în specificațiile problemelor de rezolvat. Pentru a ajunge în situația respectivă, este nevoie de multe exerciții, de multe antrenamente. Aceste abilități se pot forma doar rezolvând multe, foarte multe probleme.

În concluzie, cartea este mai mult o culegere de probleme decât un manual. În concepția autorilor, ea totuşi trebuie parcursă în succesiunea capitolelor, aşa cum sunt ele înşiruite în carte. Recomandăm acest lucru deoarece, cartea nefiind o versiune îmbogățită a unui manual elaborat conform programei şcolare, cunoştințele din carte depăşesc conținutul acestei programe, este nevoie de o abordare treptată a problemelor.

Profesorul este sfătuit să lucreze cu elevii ținând cont de întrebările şi de răspunsurile lor, să-şi modeleze activitatea în funcție de nivelul grupului. Dacă activitatea decurge fără probleme, evident se poate avansa repede şi se pot studia şi probleme din alte surse sau chiar din manualele care constituie continuarea celui de față. Se ştie că la olimpiade „nu prea" se ține cont de programa şcolară, iar la olimpiadele internaționale există o singură categorie de vârstă: 14-18 ani.

Rezolvările problemelor propuse sunt prezentate sub formă de explicații şi eventual pseudocodul algoritmului. Textele sursă le veți putea consulta în software-ul care însoțeşte aceste manuale. Nu avem pretenția că întotdeauna am propus cea mai elegantă şi cea mai eficientă rezolvare. Dacă ne contactați şi veți fi de acord ca soluțiile interesante şi originale ale dumneavoastră să devină publice, acestea vor fi înglobate în software şi vor fi adăugate noilor ediții ale manualelor.

Vă dorim succes!

Autorii

1. Generalităţi

- *Conţinutul următoarei programe (dezvoltate pentru doritorii de cunoştinţe solide în programare) nu precizează limbajul de programare. În funcţie de cerinţele elevilor şi disponibilităţile profesorilor toate temele se pot trata realizând aplicaţii în limbajul **Pascal** şi/sau **C++**. De asemenea, se vor face aplicaţii şi în **Free-Pascal**, respectiv **gnu C**. Evident, nu sunt excluse nici mediile vizuale (Delphi, Visual C++, VisualBasic sau Java).*
- *Conţinutul programei este grupat în 32 de săptămâni, pe parcursul cărora aproximativ din patru în patru săptămâni se vor organiza „miniconcursuri" din domeniul (domeniile) de cunoştinţe abordate deja. Elevii vor lucra în condiţii similare viitoarelor concursuri la care vor participa. Evaluarea se va face automat, folosind software-ul de evaluare care va fi utilizat şi în celelalte săptămâni pentru a verifica lucrările elevilor şi a evalua modul în care ei avansează.*

2. Obiective generale

Elevii participanţi vor fi capabili:

- să analizeze enunţuri de probleme care depăşesc nivelul programei şcolare;
- să stabilească metoda de rezolvare cea mai potrivită în cazul problemelor enunţate;
- să estimeze ordinul de complexitate al unui algoritm;
- să identifice problemele care se pot rezolva cu metoda *greedy*;
- să proiecteze şi să implementeze algoritmi de rezolvare bazaţi pe metoda *greedy*;
- să demonstreze că rezolvarea *greedy* conduce la optimul căutat;
- să identifice problemele care se rezolvă cu metoda *divide et impera*, respectiv cei doi paşi ai metodei (împărţirea în subprobleme şi compunerea rezultatelor subproblemelor);
- să proiecteze şi să implementeze algoritmi de rezolvare pentru probleme care se rezolvă cu metoda *divide et impera*;
- să identifice problemele care trebuie rezolvate cu metoda *programării dinamice*;
- să arate că problema are o substructură internă optimă şi să recunoască subproblemele suprapuse;
- să proiecteze şi să implementeze algoritmi de rezolvare pentru probleme care se rezolvă cu metoda *programării dinamice*;
- să proiecteze şi să implementeze algoritmi care se bazează pe metode *euristice*;
- să utilizeze variabile alocate *dinamic* în cazul în care memoria centrală nu este suficientă pentru stocarea datelor;
- să lucreze cu date organizate în structuri de date înlănţuite dinamic (să prelucreze *stive*, *cozi*, *liste ordonate*, *arbori binari*);
- să reprezinte şi să prelucreze *grafuri*;

– să rezolve probleme simple pe grafuri (determinarea arborelui parțial de cost minim şi a drumurilor minime) cu metodele de programare învățate;
– să rezolve probleme care necesită realizarea unor programe interactive.

3. Gruparea temelor pe săptămâni

Săptămânile 1-2: Recapitulare

Obiective specifice:

Recapitularea claselor de probleme, rezolvate în cadrul activităților realizate pe parcursul anului precedent.

Săptămâna 3: Recapitulare III

Obiective specifice:

Analiza problemelor propuse spre rezolvare la olimpiadele județene şi naționale (clasa a IX-a) din anul şcolar precedent (tabere de pregătire, concursuri).

Săptămâna 4-5: Metoda greedy

Obiective specifice:

Rezolvarea problemelor de optim necesită cunoaşterea mai multor metode de rezolvare. Elevii trebuie să fie capabili să stabilească metoda de rezolvare adecvată în cazul acestor probleme. Dacă restricțiile şi cerințele problemei permit utilizarea optimului local, se recomandă metoda greedy. În cazul unor elevi doritori de performanță este necesar ca ei să poată demonstra că metoda greedy conduce la determinarea optimului global. Ei trebuie să se obişnuiască cu instrumentele şi metodele care trebuie aplicate în aceste demonstrații. Elevii trebuie să aprofundeze cunoştințele necesare aplicării metodei rezolvând cât mai multe probleme şi trebuie să învețe metode de demonstrare prin care să justifice aplicarea ei.

- Identificarea clasei de probleme în cazul căreia se poate aplica metoda greedy;
- Algoritmul general care implementează metoda greedy;
- Demonstrarea corectitudinii alegerii metodei greedy pentru problema de rezolvat;
- Metoda inducției matematice
- Reducere la absurd

1. determinarea ordinii în care se vor servi persoanele care stau la o coadă astfel încât timpul mediu de aşteptare să fie minim;
2. determinarea submulțimii de sumă maximă a unei mulțimi de numere reale date;

3. eliminarea unor puncte date de-a lungul unei linii drepte astfel încât distanţa minimă dintre oricare două puncte succesive să fie cel puţin egală cu o distanţă dată;
4. problema benzii magnetice (Să se stabilească ordinea în care se vor înregistra *N* fişiere pe o bandă magnetică astfel încât timpul mediu de accesare a lor să fie minimă. Se ştie că pentru accesarea unui fişier, toate cele înregistrate înaintea lui trebuie citite.)
5. problema continuă a rucsacului (Într-un rucsac încap obiecte de greutate totală *G*. Cunoscând greutăţile şi valorile a *N* obiecte, să se stabilească care trebuie împachetate în rucsac pentru a obţine o valoare totală maximă.)
6. obţinerea unei configuraţii de *N* numere plasate pe *N*+1 poziţii (o poziţie fiind liberă) dintr-o configuraţie iniţială dată în care de asemenea există o poziţie neocupată de nici un număr; un număr poate fi mutat dintr-o anumită poziţie doar în poziţia liberă;
7. eliminarea de către doi jucători a unor numere dintr-un şir de numere, ştergând un număr de la careva din marginile şirului; se declară câştigător cel care şterge numere având suma maximă;
8. închirierea unei cabane pe anumite intervale de timp solicitate astfel încât numărul solicitanţilor serviţi să fie maxim;
9. problema spectacolelor (Într-o sală de spectacole trebuie planificate cât mai multe spectacole dintre cele *N* existente în repertoriu, astfel încât fiecare să respecte ora de începere şi de încheiere planificate.)
10. descompunerea unui şir de numere în număr minim de subşiruri descrescătoare; ordinea din subşiruri trebuie să fie aceeaşi cu ordinea din şirul dat;
11. analiza problemei plăţii sumei cu bancnote de valori date; găsirea unei mulţimi de monede pentru care algoritmul furnizează (respectiv nu furnizează) soluţie optimă.

Săptămâna 6: Metode greedy euristice

Obiective specifice:

Elevii familiarizaţi cu metoda greedy va trebui să ştie când anume această metodă, chiar dacă nu se poate demonstra că ar garanta soluţia optimă, merită aplicată în lipsa unei alte metode sigure, deoarece în cazul majorităţii datelor de intrare ea rezolvă problema.

- Identificarea clasei de probleme pentru care se poate aplica euristica greedy
- Demonstrarea faptului că în anumite cazuri metoda greedy nu conduce la optimul căutat, găsirea unor contraexemple (exemple pentru care metoda greedy nu furnizează rezultat optim)
- Aplicarea metodei

1. plata sumei cu număr minim de bancnote;

2. împachetarea unui rucsac astfel încât valoarea obiectelor împachetate să fie cât mai mare posibil (problema discretă a rucsacului);
3. ghicirea tuturor poziţiilor unui şir de numere 0 şi 1 pe care se află valoarea 0 ştiind că există exact *k* astfel de poziţii;
4. problema submarinului (determinarea numărului minim de submarine necesare pentru distrugerea a *N* vapoare);
5. determinarea ordinii în care *M* lucrări independente ale căror timpi de executare se cunosc, se vor executa de către *N* procesoare care pot lucra în paralel astfel încât durata totală să fie minimă;
6. repartiţia, pe baza unor liste de preferinţe, a *N* apartamente la *N* persoane astfel încât satisfacerea cerinţelor să fie maximă.

Săptămâna 7: Concurs

Obiective specifice:

Se va testa capacitatea elevilor de a aplica metoda greedy şi euristica greedy. Mai mult, se va testa existenţa abilităţii de a diferenţia cele două clase de probleme care se rezolvă cu acestea. Elevii vor rezolva trei probleme în două ore.

Săptămânile 8-9: Divide et Impera (I)

Obiective specifice:

Atunci când o problemă se poate despărţi în subprobleme astfel încât acestea să rezolve aceeaşi problemă pe o structură de date mai mică, se poate aplica metoda divide et impera. Elevii vor fi capabili să identifice această clasă de probleme şi vor învăţa implementarea algoritmilor de rezolvare în mod iterativ şi recursiv. Elevii familiarizaţi cu metoda divide et impera va trebui să ştie să aplice metoda şi în cazul în care descompunerea datelor nu se face în două, ci în mai multe subprobleme.

- Identificarea clasei de probleme care se rezolvă cu metoda *divide et impera*
- Algoritmul general care implementează metoda divide et impera
- Identificarea celor doi paşi ale metodei (descompunerea problemei iniţiale şi compunerea rezultatelor subproblemelor);
- Identificarea problemelor care prelucrează date bidimensionale şi care se rezolvă cu metoda *divide et impera*;
- Descompunerea problemelor în *n* subprobleme;

1. algoritmul căutării binare şi realizarea implementării iterative şi recursive;
2. calcularea produsului a *N* numere cu metoda *divide et impera*;
3. determinarea celui mai mare (celui mai mic) număr într-un şir dat cu metoda *divide et impera*;
4. plierea unui vector (Se consideră un vector de lungime *N*. Definim plierea vectorului prin suprapunerea unei jumătăţi, numite *donatoare* peste cealaltă jumătate, numită *receptoare*. Dacă numărul de elemente este impar, elementul

din mijloc este eliminat. În acest mod se ajunge la un subşir ale cărui elemente au numerotarea corespunzătoare jumătății receptoare. a) Fiind dat un indice *i*, să se determine dacă al *i*-lea element poate fi element final ca rezultat al unor plieri succesive. b) Să se precizeze toate elementele finale posibile. c) Fiind dat un element *i* final, să se determine o succesiune de plieri prin care se ajunge la el.)

5. ghicirea numărului ascuns;
6. raftul de cărți;
7. sortare prin interclasare (*merge-sort*);
8. sortare rapidă (*quick-sort*);
9. placa cu găuri;
10. turnurile din Hanoi.

Săptămâna 10: Concurs

Obiective specifice:

Se va testa capacitatea elevilor de a aplica metoda divide et impera. Se va testa existența abilității de a aplica metoda în rezolvarea diferitelor probleme şi însuşirea algoritmilor clasici. Elevii vor rezolva trei probleme în două ore.

Săptămâna 11: Grafuri (I)

Obiective specifice:

Foarte multe probleme de algoritmică se modelează pe grafuri. În concluzie elevii vor învăța cunoştințele fundamentale, introductive în teoria grafurilor. Ei se vor familiariza cu noțiunea de graf neorientat, adiacență, incidență, conexitate, componentă conexă, graf parțial, subgraf, lanț şi ciclu. Elevii vor învăța mai multe modalități de a reprezenta grafuri.

- Grafuri neorientate, concepte fundamentale
- Reprezentarea grafurilor

1. reprezentarea grafurilor prin matricea de adiacență;
2. reprezentarea grafurilor prin lista vârfurilor adiacente;
3. reprezentarea grafurilor prin lista extremităților muchiilor;
4. reprezentarea grafurilor prin matricea de incidență;
5. reprezentarea grafurilor prin matricea costurilor;
6. reprezentarea grafurilor prin matricea succesorilor vârfurilor;
7. reprezentarea grafurilor prin şirul succesorilor vârfurilor.

Săptămâna 12: Grafuri (II)

Obiective specifice:

După ce elevii s-au familiarizat cu diverse modalități de reprezentare a grafurilor, ei îşi vor forma deprinderi cu privire la traversarea grafurilor. Elevii vor crea structu-

rile de date folosind diverse reprezentări în implementările programelor care vor realiza traversarea acestora conform diverselor algoritmi.

- Traversarea în lățime;
- Traversarea în adâncime;
- Ciclu hamiltonian;
- Ciclu eulerian.

1. problema comis-voiajorului (Se consideră un graf neorientat complet cu *N* vârfuri. Fiecărei muchii i se atribuie un cost numit distanță. Să se determine un drum *hamiltonian* de cost cât mai mic. Se va aplica metoda euristică greedy.)
2. colorarea grafurilor (Se consideră un graf neorientat având *N* vârfuri. Să se determine numărul minim de culori necesare pentru a colora vârfurile grafului, astfel încât oricare două vârfuri legate printr-o muchie să fie colorate diferit. Se va aplica metoda euristică greedy.)
3. stabilirea componentelor conexe ale unui graf (prin traversare în lățime şi adâncime);
4. sortarea topologică;
5. determinarea unui ciclu *eulerian*.

Săptămâna 13: Grafuri (III)

Obiective specifice:

Graful conex care nu are cicluri se numeşte arbore. În multe probleme se cere stabilirea arborelui parțial de cost minim. Elevii vor fi capabili să recunoască printre cerințele problemei apariția acestei subprobleme şi vor fi capabili să implementeze algoritmi cunoscuți pentru rezolvarea acestora.

- Algoritmul *greedy* de determinare a arborelui parțial de cost minim al lui *Kruskal*;
- Algoritmul *greedy* de determinare a arborelui parțial de cost minim al lui *Prim*.

1. problema cu stațiile de benzină şi rafinăriile;
2. stabilirea componentelor conexe ale unui graf folosind algoritmul lui *Kruskal*.

Săptămâna 14: Grafuri (IV)

Obiective specifice:

Elevii vor învăța algoritmii clasici de determinare a drumurilor minime în grafuri neorientate şi orientate.

- Determinarea drumului critic;
- Determinarea tuturor drumurilor între două puncte date;
- Determinarea lungimii (numărului de muchii) a celui mai scurt drum între două puncte date;
- Determinarea celui mai scurt drum între două puncte date cu traversare în lățime;

- Determinarea celui mai scurt drum între două puncte date cu traversare în lățime într-un graf de costuri;
- Algoritmul lui *Dijkstra*;
- Algoritmul lui *Bellman-Ford*.

Săptămâna 15: Concurs

Obiective specifice:

Se va testa capacitatea elevilor de a reprezenta grafuri şi de a aplica metodele de programare învățate pentru a rezolva probleme modelate pe grafuri. Elevii vor rezolva trei probleme în două ore.

Săptămâna 16:-18 Metoda programării dinamice (I)

Obiective specifice:

Metoda programării dinamice se aplică în probleme de optim când se cere o singură soluție, precum şi modul în care este obținută aceasta. Problemele în cazul cărora se poate aplica această metodă trebuie să îndeplinească anumite codiții. În urma descompunerii în subprobleme a problemei date se obțin probleme dependente una de cealaltă. Rezolvarea optimă a acestor subprobleme duce la rezolvarea optimă a problemei. Elevii îşi vor forma deprinderea de a recunoaşte aceste probleme, precum de a identifica substructura optimă a acestor probleme.

Metoda programării dinamice poate fi folosită astfel încât să se poată afişa şi „termenii" din care optimul furnizat se compune. Pentru reconstituirea soluției, uneori este nevoie de structuri de date suplimentare. Elevii vor învăța detalii de implementare care să favorizeze obținerea acestor rezultate.

Frecvent, metoda programării dinamice se aplică pe date structurate în tablouri bidimensionale, sau rezolvarea impune păstrarea rezultatelor intermediare (a subproblemelor) în tablouri multidimensionale. Elevii vor exersa crearea şi prelucrarea acestor structuri de date.

- Identificarea clasei de probleme;
- Algoritmul general care implementează metoda programării dinamice;
- Demonstrarea optimalității problemei;
- Reconstituirea soluției pe baza deciziilor luate pe parcursul aplicării metodei;
- Structuri de date multidimensionale.

1. determinarea subşirului crescător de lungime maximă;
2. problema agricultorului (vezi metoda *greedy*);
3. problema dicționarului (Se consideră un dicționar şi un text. Să se descompună textul în cuvinte din dicționar.)

4. determinarea sumei maxime care poate fi obținută folosind elementele unui tablou bidimensional triunghiular situate în linii diferite şi aceeaşi coloană sau învecinate de-a lungul unei diagonale stânga dreapta.

Săptămâna 19: Concurs

Obiective specifice:

Se va testa capacitatea elevilor de a rezolva probleme care necesită cunoaşterea metodei programării dinamice. Elevii vor rezolva trei probleme în trei ore.

Săptămâna 20-21: Combinatorică

Obiective specifice:

Există o mulțime de probleme care necesită anumite cunoştințe de combinatorică. Elevii vor fi capabili să recunoască astfel de probleme şi să aplice tehnicile adecvate pentru rezolvarea lor.

O clasă de probleme este formată din cele în care se cere determinarea unei configurații având un număr de ordine dat sau determinarea numărului de ordine al unei configurații date. Elevii vor fi capabili să utilizeze cunoştințe de combinatorică pentru rezolvarea acestor probleme.

- Permutări
- Aranjamente
- Combinări

1. Calculul numărului de permutări / aranjamente / combinări;
2. Generarea permutărilor / aranjamentelor / combinărilor folosind metoda backtracking;
3. Numărul lui Catalan;
4. Determinarea numărului de ordine a unei permutări cu N elemente;
5. Determinarea permutării de N elemente care are numărul de ordine dat;
6. Determinarea celui de-al k-lea cuvânt (din punct de vedere lexicografic) format din anumite litere;
7. Determinarea numărului de ordine al unui cuvânt (din punct de vedere lexicografic) format din anumite litere.

Săptămâna 22: Principiul lui Dirichlet

Obiective specifice:

Principiul cutiei lui Dirichlet poate fi aplicat pentru rezolvarea unei clase de probleme în care mai multe obiecte sunt grupate în diverse moduri şi se cere verificarea unei proprietăți a unei submulțimi a acestor obiecte şi, eventual, obiectele care formează această submulțime. Elevii vor fi capabili să recunoască aceste probleme şi să aplice acest principiu pentru rezolvarea lor.

1. problema pantofilor desperechiați;

2. identificarea unei subsecvenţe a unui şir cu N elemente astfel încât suma elementelor din subsecvenţă să fie divizibilă cu N;
3. determinarea multiplului format din 0 şi 1;
4. problema celor $(M-1)\cdot(N-1)+1$ numere;
5. repartizarea imprimantelor.

Săptămâna 23: Concurs

Obiective specifice:

Se va testa capacitatea elevilor de a rezolva probleme de combinatorică şi de a aplica principiul cutiei lui Dirichlet. Elevii vor rezolva patru probleme în trei ore.

Săptămâna 24: Alocare dinamică (I)

Obiective specifice:

Elevii vor cunoaşte cele două modalităţi de alocare a memoriei, înţelegând diferenţele esenţiale între alocarea statică şi cea dinamică. Ei vor învăţa să aleagă dintre cele două modalităţi în funcţie de cerinţele de spaţiu, precum şi a operaţiilor specifice care se vor efectua pe o anumită structură.

Alocarea dinamică, de regulă, se recomandă pentru implementarea unor structuri de date speciale în care inserările şi ştergerile se fac pe baza anumitor restricţii. Elevii vor învăţa despre stivă şi coadă din punct de vedere logic. Ei au învăţat în clasa a 9-a să realizeze programe în care aceste structuri sunt alocate static, acum ei vor fi interesaţi să-şi formeze abilităţile necesare de a aloca dinamic aceste structuri. Elevii vor înţelege modul în care înlănţuirea se poate specifica similar cu indicii din şiruri, dar şi în câmpuri suplimentare ataşate unui element dintr-o structură de date înlănţuită.

- Principiile fundamentale ale alocării statice;
- Principiile fundamentale ale alocării dinamice;
- Diferenţele dintre cele moduri de alocare a memoriei;
- Diferenţele dintre modul în care se declară şi se prelucrează pointerii, respectiv variabilele dinamice;
- Stiva şi coada, (structuri de date, prezentate independent de implementare);
- Operaţii specifice (inserare, ştergere, verificarea dacă stiva este vidă, verificarea dacă stiva este plină);
- Implementarea statică a acestor operaţii;
- Implementarea dinamică a stivei şi a cozii.

1. exerciţii în care intervine alocarea dinamică a unei variabile simple;
2. exerciţii în care intervine alocarea dinamică a unei variabile de tip structurat;
3. determinarea tuturor numerelor prime mai mici decât 50.000;
4. determinarea tuturor numerelor prime mai mici decât 400.000;

5. determinarea tuturor numerelor prime mai mici decât 1.000.000;
6. determinarea tuturor numerelor prime mai mici decât 1.000.000.000;
7. scrierea unor subprograme care realizează operații specifice pentru prelucrarea stivelor şi cozilor.

Săptămâna 25: Alocare dinamică (II)

Obiective specifice:

Elevii vor înțelege modul în care înlănțuirea se poate utiliza pentru a specifica schimbarea succesiunii într-o listă şi astfel se vor familiariza cu structuri de date de tipul listelor ordonate. Ei vor exersa inserările şi ştergerile în toate variantele posibile (avem pointer la elementul în fața căruia se face inserarea, avem pointer la elementul după care se face inserarea, avem pointer la elementul care trebuie şters, sau la elementul din fața lui. Listele înlănțuite se vor traversa cu algoritmi iterativi şi recursivi.

Elevii vor face cunoştință cu liste dublu înlănțuite şi circulare. Vor fi capabili să le creeze şi să le prelucreze.

- Lista ordonată;
- Coada cu priorități;
- Operații specifice;
- Lista dublu înlănțuită;
- Lista circulară.

1. completarea unit-ului cu operații care prelucrează liste ordonate;
2. completarea unit-ului cu operații care prelucrează cozi cu priorități;
3. utilizarea unit-ului în rezolvarea problemei cu săritura cu schiurile;
4. completarea unit-ului cu operații care prelucrează liste dublu înlănțuite;
5. completarea unit-ului cu operații care prelucrează liste circulare;
6. utilizarea unit-ului în rezolvarea problemei care cere simularea unui joc de cărți;
7. problema lui Josef.

Săptămâna 26: Arbori (I)

Obiective specifice:

Elevii vor fi capabili să recunoască problemele în care ierarhia introdusă în structură nu este secvențială ci arborescentă. Ei se vor familiariza cu crearea şi alocarea dinamică a acestora, precum şi cu implementarea operațiilor specifice.

- Arborele binar
- Crearea unui arbore binar total echilibrat
- Traversarea arborilor binari

1. completarea unit-ului cu subprogramul care creează un arbore binar total echilibrat;

2. completarea unit-ului cu subprogramul care creează un arbore binar pe baza secvenței obținute în urma traversării în preordine a arborelui binar;
3. realizarea unui program care generează notația poloneză a unei expresii aritmetice.

Săptămâna 27: Arbori (II)

Obiective specifice:

Arborele binar se utilizează frecvent ca o structură de căutare. Elevii vor fi capabili să creeze arbori binari de căutare şi să efectueze toate operațiile specifice acestora.

- Arbori binari de căutare;
- Operații specifice (căutare, inserare, eliminare, traversare).

1. completarea unit-ului cu subprogramele care creează şi prelucrează un arbore binar de căutare;
2. realizarea unui program care utilizează un astfel de arbore şi îl afişează în preordine, inordine şi postordine.

Săptămâna 28: Concurs

Obiective specifice:

Se va testa capacitatea elevilor de a aloca dinamic diverse structuri de date şi de a aplica cele învățate în rezolvarea problemelor Elevii vor rezolva trei probleme în două ore.

Săptămâna 29-30: Proiect

Obiective specifice:

Se va testa capacitatea elevilor de a lucra în echipă la un proiect. Se va realiza o aplicație de tip soft educațional şi/sau o pagină Web. Se va lucra cu HTML şi/sau cu un program grafic.

- Realizarea unor unit-uri pentru prelucrarea structurilor de date alocate dinamic (stive, cozi, liste ordonate, liste dublu înlănțuite, arbori).
- Susținerea proiectului realizat.

Cuprins