

Programa de diferențe/ evaluări Disciplina Informatică

Pentru clasa a X-a Științele naturii

Noțiunea de algoritm. Caracteristici. Exemple.

Date cu care lucrează algoritmi (constante, variabile, expresii).

Operații asupra datelor (aritmetice, logice, relaționale).

Reprezentarea algoritmilor. Pseudocod.

Principiile programării structurate. Structuri de bază:

- structura liniară
- structura alternativă
- structura repetitivă

Algoritmi elementari

1. Prelucrarea numerelor :

- prelucrarea cifrelor unui număr (de exemplu, suma cifrelor, testarea proprietății de palindrom, etc.)
- probleme de divizibilitate (de exemplu, determinarea divizorilor unui număr, determinarea c.m.m.d.c./c.m.m.m.c., testare primalitate, etc.)
- calculul unor expresii simple (sume, produse, etc.)

2. Prelucrarea unor secvențe de valori

- determinare minim/maxim
- verificarea unei proprietăți (de exemplu, dacă toate elementele din secvență sunt numere perfecte, etc.)
- calculul unor expresii în care intervin valori din secvență (de exemplu: numărarea elementelor pare/impare, etc.)
- generarea șirurilor recurente (de exemplu: șirul Fibonacci)

Manuale: Clasa a IX-a, varianta Pascal, intensiv (nr. 1) - valabilă și pt cls IX-X neintensiv, editura Donaris

Pentru clasa a X-a, Matematică- informatică, intensiv informatică

Noțiunea de algoritm. Caracteristici. Exemple.

Date cu care lucrează algoritmi (constante, variabile, expresii).

Operații asupra datelor (aritmetice, logice, relaționale).

Etapele rezolvării problemelor. Exemple.

Noțiunea de algoritm. Caracteristici. Exemple.

Reprezentarea algoritmilor. Pseudocod.

Principiile programării structurate. Structuri de bază: structura liniară, structura alternativă, structura repetitivă.

Algoritmi elementari.

1. Prelucrarea numerelor :

- prelucrarea cifrelor unui număr (de exemplu, suma cifrelor, testarea proprietății de palindrom, etc.)
- probleme de divizibilitate (de exemplu, determinarea divizorilor unui număr, determinarea c.m.m.d.c./c.m.m.m.c., testare primalitate, etc.)
- calculul unor expresii simple (sume, produse, etc.)

2. Prelucrarea unor secvențe de valori:

- determinare minim/maxim
- verificarea unei proprietăți (de exemplu, dacă toate elementele din secvență sunt numere perfecte, etc.)
- calculul unor expresii în care intervin valori din secvență (de exemplu: numărarea elementelor pare/impare, etc.)
- generarea șirurilor recurente (de exemplu: șirul Fibonacci)

Elementele de bază ale limbajului de programare

Noțiuni introductive

- Structura programelor
- Vocabularul limbajului
- Tipuri simple de date (standard)
- Constante, variabile, expresii
- Citirea/scrierea datelor

Structuri de control

- Structura liniară
- Structura alternativă

- Structuri repetitive

Tipuri structurate de date. Tipul tablou

- Tablouri unidimensionale
- Tablouri bidimensionale

Algoritmi fundamentali de prelucrare a datelor structurate în tablouri

- căutare secvențială, căutare binară
- sortare
- interclasare
- parcurgerea tablourilor bidimensionale pe linii/coloane

Fișiere text. Definiție, operații specifice Mediul limbajului de programare studiat

- Prezentare generală
- Editarea programelor sursă
- Compilare, rulare, depanare

Analiza eficienței unui algoritm.

Manuale: Clasa a IX-a, varianta Pascal, intensiv (nr. 1) - valabilă și pt cls IX-X neintensiv, editura Donaris

Pentru clasa a XI-a Științele naturii

Elementele de bază ale limbajului de programare

Noțiuni introductive

- Structura programelor
- Vocabularul limbajului
- Tipuri simple de date (standard)
- Constante, variabile, expresii
- Citirea/scrierea datelor

Structuri de control

- Structura liniară
- Structura alternativă
- Structuri repetitive

Mediul limbajului de programare studiat

- Prezentare generală
- Editarea programelor sursă
- Compilare, rulare, depanare

Implementarea unor algoritmi elementari cu aplicabilitate practică

Tipuri structurate de date.

Tipul tablou. Tablouri unidimensionale și bidimensionale.

Fișiere text.

- Definire.
- Operații specifice.

Algoritmi fundamentali de prelucrare a datelor structurate în tablouri

- căutare secvențială, căutare binară
- sortare
- interclasare
- prelucrări specifice tablourilor bidimensionale

Aplicații interdisciplinare

Exemple orientative:

- Prelucrări statistice ale unei serii de valori
- Calculul valorii unei expresii algebrice
- Calcule combinatoriale
- Determinarea unor mărimi fizice dintr-un circuit electric

- Aplicații din genetică (legea creșterilor organice, etc.)

Analiza eficienței unui algoritm

Manuale: Clasa a IX-a, varianta Pascal, intensiv (nr. 1) - valabilă și pt cls IX-X neintensiv, editura Donaris

Pentru clasa a XI-a, Matematică- informatică, intensiv informatică

Tipuri structurate de date

- Șir de caractere
- Înregistrare (structură)
- Listă, stivă, coadă

Subprograme

- Declararea, definirea și apelul subprogramelor
- Transferul parametrilor la apel
- Returnarea valorilor de către subprograme
- Variabile locale și globale

Subprograme recursive. Mecanismul de realizare a recursivității

Algoritmi fundamentali de prelucrare a datelor structurate

- prelucrarea unui șir de caractere la nivel de caracter și la nivel de structură, utilizând subprograme specifice
- prelucrarea unei înregistrări/structuri la nivel de câmp și la nivel de structură

Aplicații folosind subprograme

Metoda de programare *Divide et Impera*

- Sortarea eficientă a unei mulțimi de valori aplicând metoda Divide et Impera (sortarea rapidă, sortarea prin interclasare)
- Căutarea eficientă a unui element într-o mulțime ordonată aplicând metoda Divide et Impera (căutarea binară)
- Generarea unor modele fractale

Manual: Clasa a X-a, varianta Pascal, intensiv (nr. 3) editura Donaris

Grafuri orientate și grafuri neorientate

1. Algoritmi simpli de verificare a însușirii 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 ieși 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 subșir crescător de lungime maximă
2. Înmulțirea optimală a unui șir 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.

Manual: Clasa a XI-a, varianta Pascal, intensiv (nr. 8) editura Donaris