Cuvânt Înainte

Prezenta lucrare cuprinde prelegerile susţinute de autor în cadrul cursului de "Structuri de Date şi Algoritmi", în faţa anului IAA al Facultăţii de Automaticã şi Calculatoare a Universitãţii POLITEHNICA din Bucureşti. Lucrarea prezintã studenţilor câteva metode de dezvoltare a aplicaţiilor de mari dimensiuni cum ar fi: dezvoltarea modularã a aplicaţiilor, programarea genericã, mecanisme de abstractizare proceduralã sau de abstractizare a datelor. Sunt puse în evidenţã şi mecanisme puse la dispoziţia programatorului de limbajul C++ în scopul elaborãrii unor aplicaţii robuste, generale de mari dimensiuni.

Lucrarea prezintã principalele tipuri de date abstracte lista, stiva, coada, arborele, graful precum şi unele metode generale de rezolvare a problemelor cum ar fi: DIVIDE ET IMPERA, backtracking sau programarea dinamicã. În elaborarea algoritmilor de manipulare a acestor structuri de date trebuie avutã în vedere stabilirea unor justificãri asupra corectitudinii acestor algoritmi precum şi analiza eficienţei fiecãrui astfel de algoritm. Unele aspecte ale domeniului de studiu al structurilor de date şi algoritmilor, cum ar fi: studiul corectitudinii algoritmilor, demonstrarea riguroasã a corectitudinii bazatã pe tehnici de semanticã axiomaticã au fost tratate de savanţi iluştri ca Floyd, Hoare, Dijkstra sau Gries. Aceastã direcţie de studiu ar putea fi aprofundatã în continuare de cititor.

Exemplele prezentate în cadrul fiecãrui capitol au grade de dificultate diferite de la exemple simple, clasice şi pânã la probleme propuse la concursurile de programare; sunt prezentate şi câteva probleme pe care cititorul este invitat sã încerce sã le rezolve singur. O aceiaşi aplicaţie poate fi rezolvatã în moduri diferite, uneori diferenţele între cantitatea de resurse (spaţiu de memorie, timp de execuţie) necesare diverselor soluţii sunt uriaşe. Toate aceste aspecte trebuie însuşite în profunzime de studenţii de astãzi pentru cã numai astfel ei vor putea deveni specialiştii competenţi, profesionişti de mâine.

***

Capitolul 1 descrie principiile care stau la baza dezvoltãrii aplicaţiilor de mari dimensiuni. Este prezentat modul de a descompune în module o aplicaţie C precum şi o serie de mecanisme de abstractizare a datelor specifice limbajului C++.
Capitolul 2 se referã la analiza algoritmilor. Sunt descrise modele de calcul, notaţii referitoare la analiza comportãrii asimptotice a algoritmilor, analiza implementãrii tipurilor abstracte de date, analiza amortizatã şi sunt date exemple care sã ilustreze diverse procedee de analizã algoritmicã.
În capitolul 3 sunt prezentate operaţiile caracteristice tipului de date abstracte LISTA precum şi diverse implementãri ale acestui TDA: implementãri bazate pe diverse structuri de memorare, implementãri iterative sau recursive. Se descriu modalitãţi de implementare a listelor generice atât în limbajul C cât şi în limbajul C++ folosind şabloane.
Capitolele 4 şi 5 prezintã douã metode generale de rezolvare a problemelor: metoda *Divide et Impera* şi *metoda cãutãrii cu revenire în spaţiul soluţiilor* (*backtracking*).

Exemplele semnificative sunt însoţite de analiza detaliatã a algoritmilor şi sunt prezentate în ordinea crescãtoare a nivelului de dificultate.
În capitolul 6 sunt descrise tipurile de date stivã şi coadã împreunã cu douã aplicaţii semnificative care se referã la evaluarea expresiilor aritmetice şi respectiv la simularea funcţionării sistemelor cu evenimente discrete.
O structurã de date utilizatã frecvent într-o gamã largã de aplicaţii (datoritã faptului cã permite cãutarea foarte eficientã în masive de date) este tabela de dispersie; ea este descrisã în capitolul 7.
Capitolul 8 este (alãturi de capitolul referitor la grafuri) unul din cele mai extinse din prezenta lucrare. El se referã la arbori, structuri de date ce permit elaborarea unor algoritmi deosebit de eficienţi şi utilizaţi în cele mai diferite situaţii. Sunt prezentaţi arborii binari de cãutare, arborii AVL, arborii de cãutare în spaţii multidimensionale şi arborii parţial ordonaţi.
Capitolul 9 referitor la grafuri înfãţişeazã principalele modalitãţi de reprezentare precum şi câteva categorii de aplicaţii relevante cum sunt: sortarea topologicã, determinarea punctelor de articulaţie, metoda lui Kuhn pentru determinarea unei împerecheri optimale într-un graf bipartit. Soluţia lui Floyd (bazatã pe metoda programãrii dinamice) la problema determinării drumurilor minime între oricare douã vârfuri ale unui graf face trecerea cãtre metoda programãrii dinamice prezentatã în capitolul 10.

Parcurgerea lucrării necesitã cunoştinţe preliminare de programare în limbajul C. Lucrarea poate fi folositã de toţi cei interesaţi (elevi, studenţi, specialişti) de îmbunãtãţirea calitãţii programelor pe care le elaborează, de înţelegerea profundã a principalelor structuri de date şi tehnici de programare ce pot fi folosite în dezvoltarea aplicaţiilor complexe. Autorul recomandã cititorilor şi în special studenţilor – valoroşi – ai Facultãţii de Automaticã şi Calculatoare sã nu considere aceastã lucrare decât un început şi sã folosească bibliografia prezentatã, pentru aprofundarea cunoştinţelor în acest vast şi fundamental domeniu de studiu.

Marius Dorian ZAHARIA
marius.zaharia@cs.pub.ro

CUPRINS