Anexa 2

DETERMINAREA STABILITĂŢII DOMENIULUI STRUCTURI DE DATE

Structurile de date reprezintă un domeniu de sine stătător, iar conţinutul său este structurat în numeroase lucrări pe capitole în care sunt abordate aspecte legate de fiecare tip de structură de date existent şi de operaţiile care se definesc pe acestea.

Pentru a evidenţia că acest domeniu are un conţinut bine definit au fost luate spre analiză următoarele lucrări fundamentale, realizate pe un interval de timp reprezentativ, identificate prin **K_1, K_2, K_3, K_4, K_5, K_6** şi respectiv, **K_7**:

- Saumyendra SENGUPTA, Carl Phillip KOROBKIN – *C++ Object-Oriented Data Structures,* Springer – Verlag, Berlin 1994 **(K_1)**;
- Aaron M. TENENBAUM, Yedidyah LANGSAM, Moshe J. AUGENSTEIN – *Data Structures Using C,* Prentice-Hall INC. Englewood Cliffs, New Jersey, 1990 **(K_2)**;
- Niklaus WIRTH – *Algorithms + Data Structures = Programs,* Prentice-Hall INC. Englewood Cliffs, New Jersey, 1976 **(K_3)**;
- Robert L. KRUSE – *Data Structures & Program Design,* Prentice-Hall INC. Englewood Cliffs, New Jersey, 1984 **(K_4)**;
- Joseph BERGIN – *Data Abstraction The object-Oriented approach using C++,* McGraw-Hill, Inc. 1994 **(K_5)**;
- Adam Drozdek – *Data Structures and Algorithms in C++*, Third Edition, Brooks/Cole Pacific Grove, California 2004 **(K_6)**;
- Erik Demaine – *Advanced Data Structures*, Lecture curricula Massachusetts Institute of Technology 2003, http://theory.csail.mit.edu/classes/6.897/spring03/scribe_notes/ **(K_7)**.

Definirea con□inutului domeniului de Structuri de Date

Nr. Crt.	Conţinut	K_1	K_2	K_3	K_4	K_5	K_6	K_7
	Tipuri de date	Da	Da		Da	Da		
	Structuri de date abstracte		Da		Da	Da		
	Modele de reprezentare							Da
	Modele tolerante la erori							Da
	Tipuri de date abstracte	Da	Da		Da	Da	Da	
	Implementarea structurilor de date în limbaje de programare				Da	Da		
	Prezentare limbaj de programare	C++	C	Pascal	Pascal	C++	C++	
	Concepte de bază ale limbajului	Da		Da		Da		
	Alfabet, caractere speciale, identificatori, operatori, cuvinte cheie, instrucţiuni			Da		Da		
	Tipuri de date fundamentale		Da	Da		Da	Da	
	Structuri de date		Da	Da		Da	Da	
	Expresii		Da	Da		Da	Da	
	Obiecte			Da		Da	Da	
	Proceduri, funcţii		Da	Da		Da	Da	
	Funcţii pe structuri de date	Da				Da	Da	
	Implementarea programării orientate obiect în limbajul de programare descris	Da				Da	Da	
	Concepte de bază ale programării orientate obiect	Da				Da	Da	
	Implementarea structurilor de control			Da		Da	Da	
	Supraîncărcarea operatorilor	Da				Da	Da	
	Stil de programare, alegere identificatori, autodocumentare , forme de prezentare, rafinare, modularitate				Da		Da	
	Mentenanţa programelor				Da		Da	

	Testarea programelor				Da			
	Sortare			Da				
	Concepte de bază		Da		Da	Da	Da	
	Metode de sortare	Da	Da	Da	Da	Da	Da	
	Heap sort		Da	Da		Da	Da	
	Sortare prin inserare	Da	Da	Da	Da	Da	Da	
	Sortare prin selecţie	Da	Da	Da	Da	Da	Da	
	Bubble sort	Da	Da			Da	Da	
	Quicksort	Da	Da		Da	Da	Da	
	Sortare shell	Da	Da		Da	Da	Da	
	Sortare prin calcularea adresei		Da					
	Merge Sort	Da	Da		Da	Da	Da	
	Sortare prin interschimb			Da				
	Sortare cu arbori	Da	Da	Da				
	Radix sort	Da	Da				Da	
	Radix exchange sort	Da						
	Sortarea fişierelor secvenţiale			Da				
	Sortare polifazică			Da				
	Algoritmi recursivi			Da		Da	Da	
	Regăsirea informaţei				Da		Da	
	Sortare în memoria externă							Da
	Analiza performanţei metodelor de sortare	Da	Da			Da	Da	
	Căutare							
	Tehnici de căutare		Da		Da		Da	
	Căutare prin interpolare	Da	Da					
	Căutare Fibonacci	Da						
	Căutare secvenţială	Da	Da		Da	Da	Da	
	Căutare secvenţială indexată		Da					
	Căutare binara		Da		Da	Da		
	Căutare în arbori		Da			Da	Da	
	Tehnici hashing	Da	Da				Da	
	Analiza performanţei metodelor de căutare	Da	Da			Da		
	Recursivitate							

	Recursivitatea – concepte de bază, definire şi procese		Da	Da	Da	Da	Da	
	Implementarea recursivităţii în limbajul de programare		Da		Da	Da	Da	
	Simularea recursivităţii		Da		Da	Da		
	Eficienţa recursivităţii		Da		Da	Da		
	Recursivitatea şi tehnica divede et impera	Da			Da	Da		
	Algoritmul backtracking bazat pe recursivitate		Da	Da	Da	Da	Da	
	Problema reginelor, folosind recursivitate			Da			Da	
	Problema mariajului stabil, folosind recursivitate			Da				
	Problema selecţiei optime, folosind recursivitate			Da				
	Recursivitate de nivel 1						Da	
	Recursivitate de nivel 2 Funcţia Akkermann , Turnurile din Hanoi						Da	
	Structuri de date dinamice			Da				
	Tipuri de date recursive			Da		Da	Da	
	Pointeri sau referinţe	Da	Da	Da		Da	Da	
	Operaţii aritmetice cu pointeri	Da	Da				Da	
	Apelul prin referinţe	Da	Da				Da	
	Returnarea valorii funcţiei ca pointer	Da	Da				Da	
	Pointer către structură de tip articol	Da	Da				Da	
	Definirea structurii de tip articol	Da	Da				Da	
	Referirea membrilor structurii de tip articol	Da	Da				Da	
	Structura de tip articol ca argument în funcţie sau procedură	Da	Da				Da	
	Structura de tip articol returnată de funcţie	Da	Da				Da	
	Pointer spre articol ca argument în funcţie	Da	Da				Da	
	Vector de structură	Da					Da	
	Matrice de structură	Da					Da	
	Uniune	Da	Da			Da	Da	
	Clase	Da						
	Definirea unei funcţii membre în afara clasei	Da				Da	Da	
	Definirea unui obiect al clasei	Da				Da	Da	
	Accesarea unui membru al clasei	Da				Da		
	Moştenire	Da				Da		

	Funcţii virtuale	Da				Da		
	Funcţii virtuale pure	Da				Da		
	Funcţii şi clase friend	Da				Da		
	Clase abstracte	Da				Da		
	Derivare şi moştenire multiplă	Da				Da		
	Clasă imbricată	Da				Da		
	Constructori şi destructori	Da				Da	Da	
	Polimorfism	Da				Da	Da	
	Supraîncărcarea operatorilor şi a funcţiilor	Da				Da		
	Scurgeri de memorie	Da						
	Masive	Da				Da		
	Masive unidimensionale	Da	Da			Da		
	Declarare, stocare, iniţializare şi regăsirea elementelor pentru masive unidimensionale	Da	Da			Da		
	Translatarea adresei masivelor unidimensionale	Da						
	Masive unidimensionale ca argmente în funcţii	Da				Da		
	Masive bidimensionale	Da	Da			Da		
	Declarare, stocare, iniţializare şi regăsirea elementelor pentru masive bidimensionale	Da	Da			Da		
	Translatarea adresei masivelor bidimensionale	Da						
	Masive bidimensionale ca argumente în funcţii	Da				Da		
	Şiruri de caractere	Da						
	Implementarea şirurilor de caractere folosind masive şi pointeri	Da						
	Masive de pointeri la şiruri de caractere	Da						
	Liste	Da	Da	Da	Da	Da		
	Liste liniare – concepte de bază		Da	Da	Da	Da		
	Sortarea listelor			Da	Da	Da		
	Reorganizarea listelor			Da		Da		
	Liste circulare	Da	Da	Da	Da	Da		
	Liste simplu înlănţuite	Da	Da		Da	Da	Da	
	Creare	Da	Da		Da	Da	Da	
	Traversare	Da	Da		Da	Da	Da	

	Inserare	Da	Da		Da	Da	Da	
	Interschimb	Da	Da		Da	Da		
	Ştergere	Da	Da		Da	Da	Da	
	Concatenare					Da		
	Managementul listelor externe							Da
	Procese de mentenanţă							Da
	Liste dublu înlănţuite	Da	Da		Da	Da	Da	
	Creare	Da	Da		Da	Da	Da	
	Traversare	Da	Da		Da	Da	Da	
	Inserare	Da	Da		Da	Da	Da	
	Interschimb	Da	Da		Da	Da		
	Ştergere	Da	Da		Da	Da	Da	
	Concatenare					Da		
	Liste generalizate		Da					
	Liste simulate cu masive unidimensionale	Da	Da		Da		Da	
	Liste realizate cu pointeri	Da	Da		Da	Da	Da	
	Managementul automatic al listelor		Da					
	Managementul memoriei dinamice	Da	Da				Da	
	Stive	Da	Da				Da	
	Definire	Da	Da		Da	Da	Da	
	Reprezentare în limbajul de programare descris	Da	Da		Da	Da	Da	
	Obiecte de stiva	Da					Da	
	Simularea stivei cu ajutorul listelor		Da				Da	
	Simularea stivei cu ajutorul unui masiv unidimensional						Da	
	Cozi	Da						
	Definire	Da	Da		Da	Da	Da	
	Reprezentare în limbajul de programare descris	Da	Da		Da	Da	Da	
	Simularea cozilor cu ajutorul masivelor unidimensionale	Da					Da	
	Simularea cozilor cu ajutorul listelor	Da	Da				Da	
	Cozi circulare	Da					Da	
	Simularea cozilor circulare cu ajutorul masivelor unidimensionale	Da	Da					
	Simularea cozilor circulare cu ajutorul listelor	Da						

	Cozi de prioritate	Da	Da				Da	
	Analiza performanţei operaţiilor pe cozi	Da						
	Arbori							
	Structuri arborescente – concepte de baza	Da	Da	Da	Da	Da	Da	
	Operaţii de bază pe structuri arborescente	Da	Da	Da	Da	Da	Da	
	Căutarea în arbori	Da	Da	Da	Da	Da	Da	
	Inserarea de noduri în arbori		Da	Da	Da	Da	Da	
	Ştergerea în arbore		Da	Da	Da	Da	Da	
	Reprezentarea listelor ca arbori		Da					
	Simularea arborilor cu ajutorul masivelor unidimensionale	Da	Da					
	Comparare arbori				Da			
	Arbori Binari	Da	Da	Da	Da	Da	Da	
	Creare	Da	Da	Da	Da	Da	Da	
	Traversare	Da	Da	Da	Da	Da	Da	
	Sortarea dupa chei	Da	Da	Da	Da	Da	Da	Da
	Căutari	Da	Da	Da	Da	Da	Da	Da
	Căutari optimale							Da
	Inserare	Da	Da	Da	Da	Da	Da	
	Ştergere	Da	Da	Da	Da	Da	Da	
	Fuziune							Da
	Utilizarea funcţiilor splay							Da
	Arbori de decizie							Da
	Optimizări bazate pe costuri							Da
	Arbori Binari Echilibrati	Da	Da	Da	Da	Da		
	Creare	Da		Da	Da	Da	Da	
	Traversare	Da		Da	Da	Da	Da	
	Inserare	Da		Da	Da	Da		
	Ştergere	Da		Da	Da	Da		
	Căutare optimală în arbori			Da	Da			
	Afişarea structurii arborescente	Da		Da	Da			
	Arbori AVL	Da	Da		Da	Da	Da	
	Creare	Da			Da	Da	Da	

	Traversare	Da			Da	Da	Da	
	Inserare	Da			Da	Da	Da	
	Ştergere	Da			Da	Da	Da	
	Tehnici de rotaţie pentru arboii AVL	Da			Da	Da	Da	
	Analiza eficienţei arborilor echilibrati	Da	Da		Da			
	Arbori multicăi	Da		Da	Da			
	Radix search trees	Da						
	Pădure de arbori		Da		Da			
	Arbori B	Da	Da	Da		Da	Da	
	Operaţii pe arbori B	Da	Da	Da		Da	Da	
	Arbori B binari			Da		Da	Da	
	Creare	Da	Da	Da		Da	Da	
	Traversare	Da	Da	Da		Da	Da	
	Căutare	Da	Da	Da	Da	Da	Da	Da
	Căutare prin metoda divide et impera							Da
	Inserare	Da	Da	Da		Da	Da	
	Ştergere	Da	Da	Da		Da	Da	
	Optimizare multicriterială							Da
	Arbori B+		Da				Da	
	Arbori B-		Da				Da	
	Arbori B*						Da	
	Arbori R						Da	
	Arbori 2-4						Da	
	Analiza performanţei pentru operaţiile ce privesc arborii binari	Da						
	Transformări hashing			Da			Da	
	Heapuri						Da	
	Concepte de bază						Da	
	Organizarea heap-ului ca o coadă de prioritate						Da	
	Organizarea heap-ului ca un masiv unidimensional						Da	
	Grafuri	Da						
	Concepte de bază	Da	Da			Da	Da	
	Problema fluxului	Da	Da					

	Reprezentarea grafului	Da	Da			Da	Da	
	Ca masiv bidimensional	Da	Da			Da	Da	
	Ca listă de liste	Da	Da			Da	Da	
	Ca listă de pointeri							
	Algoritmul Dijkstra						Da	
	Algoritmul lui Ford						Da	
	Traversarea grafului	Da	Da			Da	Da	
	Hashing	Da	Da		Da		Da	
	Rehashing	Da	Da					
	Hashing circular					Da		
	Crearea şi distrugerea tabelelor hash	Da	Da		Da	Da	Da	
	Ştergere, adăugare şi regăsire în tabele hash	Da	Da		Da	Da	Da	
	Eficienţa metodelor de rehashing		Da					
	Tabel hash cu liste înlănţuite	Da						
	Hashing pe arbori binari		Da				Da	
	Hashing pe medii de stocare externe		Da					
	Metoda separatorului		Da					
	Hashing dinamic	Da	Da					
	Hashing liniar	Da	Da				Da	
	Funcţii hash	Da	Da		Da		Da	
	Funcţii perfecte pentru hash		Da				Da	Da
	Analiza tehnicii hasing	Da	Da		Da			
	Iteratori							
	Pentru liste							
	Pentru arbori					Da		
	Pentru masive unidimensionale							
	Pentru masive bidimensionale							
	Notaţia poloneză inversă	Da			Da			
	Definirea problemei	Da			Da		Da	
	Evaluarea expresiilor	Da			Da		Da	
	Transformarea expresiilor în forma poloneză	Da			Da		Da	
	Compresia datelor	Da					Da	Da
	Baze matemetice				Da			

	Algoritmi	Da	Da	Da	Da	Da	Da	
	Permutări, combinări, factorial	Da	Da	Da	Da	Da		
	Numarul Fibonacci	Da	Da	Da	Da		Da	
	Notaţii asimptotice şi eficienţa algoritmilor	Da	Da		Da	Da	Da	
	Numere catalane				Da			
	Exemplele sunt prezentate exclusiv în limbajul ales de autor şi se rulează direct fară modificări			Da	Da		Da	
	Exemplele sunt prezentate numai într-un limbaj formal de descriere al algoritmilor şi pentru implementarea procedurilor trebuie scrise programe într-un limbaj, ca de exemplu Pascal, C, C++, Java, C#.							Da
	O parte din exemple sunt prezentate în limbajul de programare ales de autor şi se rulează direct fără modificări, iar o altă parte dintre exemple sunt prezentate numai într-un limbaj formal de descriere al algoritmilor, iar pentru implementare trebuie scrise proceduri într-un limbaj de programare, ca de exemplu Pascal, C, C++, Java, C#.	Da	Da			Da		

Din analiza comparată a unor lucrări de referinţă din literatura de specialitate se desprind următoarele idei:

- structurile de date se prezintă atât în limbaje de programare clasice cât şi în limbaje de programare evoluate, în ambele cazuri impunându-se necesitatea unei recapitulări a cunoştinţelor legate de limbaj în vederea obţinerii garanţiei că programele şi funcţiile create pentru a manipula structuri de date sunt corecte, eficiente şi complete;
- sortarea ca proces de transformare a unui şir de date oarecare într-un şir de date ordonat este o operaţie care se desfăşoară în masive unidimensionale, în liste simple sau liste duble şi în structuri arborescente; indiferent de metoda utilizată se urmăreşte minimizarea numărului de comparaţii şi de interschimburi de pointeri, fapt ce impune ca procesele de sortare să fie incluse în domeniul structurilor de date, ca operaţii pe structuri de date;
- listele simple sunt prezente în toate lucrările şi se abordează: definirea elementului de tip listă, crearea listei simple, inserarea de elemente în lista simplă, ştergerea de elemente din lista simplă, interschimbul de elemente din lista simplă, copierea unei liste simple, concatenarea listelor simple, numărarea elementelor dintr-o listă simpla, traversarea unei liste simple, sortarea listei simple cu memorarea vechilor poziţii a elementelor;
- listele duble sunt, de asemenea prezentate în toate lucrările şi se abordează: definirea elementului de tip listă, crearea listei duble, inserarea elementelor în listele duble,concatenarea listelor duble, numărarea elementelor într-o listă dublă, traversarea unei liste duble, sortarea listei duble cu memorarea vechilor poziţii a elementelor;
- stivele sunt prezentate doar în unele lucrări şi se urmaresc aspecte multiple ale utilizării lor: în gestionarea memoriei, folosirea lor pentru a evita metodele recursive în operaţii cu arbori;se evidenţiază şi metode de simulare cu ajutorul listelor simplu înlănţuite a conceptului de stiva
- cozile sunt prezentate în două lucrări şi se abordează : noţiunea de coadă circulară, noţiunea de coadă de prioritate, simularea cozilor utilizând masive unidimensionale şi liste simplu înlănţuite
- arborii binari sunt o parte componentă a fiecărei lucrări şi sunt prezente următoarele particularităţi: crearea arborilor binari, inserarea nodurilor în arborii binari, ştergerea nodurilor din arborii binari, căutarea în arborii binari, formarea arborilor binari de căutare, compararea arborilor, simularea arborilor cu ajutorul masivelor unidimensionale;
- arborii echilibraţi sunt prezentaţi în toate lucrările şi se urmăresc caracteristicile şi modurile de lucru: ştergere şi adăugare de noduri, echilibrare şi reechilibrare, analiza eficienţei;

- arborii B sunt prezentaţi urmărind: adăugarea, ştergerea şi spargerea nodurilor, menţinerea structurii arborescente echilibrate, caracteristici dominante, şi analiza eficienţei, prezentarea conceptelor de arbori B+, arbori B- , arbori R, arbori 2-4 şi arbori B binari
- grafurile sunt analizate, urmărind: modul de reprezentare, inserarea de noi noduri, ştergerea nodurilor, modificarea legăturilor, analiza eficienţei prelucrărilor pe grafuri
- exemplu reprezentativ cu compresie de date folosind arbori quattro
- exemplu reprezentativ folosind arbori B pentru elaborarea unui sistem de gestiune a bazelor de date

Există trei moduri de abordare practică a domeniului structuri de date şi anume:

- lucrările includ principii şi scheme de dezvoltare a algoritmilor; cititorul trebuie să definească variabile să elaboreze cod sursă într-un limbaj de programare şi să se preocupe de realizarea unei biblioteci de proceduri care lucrează cu structuri de date; în absenta unor exemple complexe, cititorul va structura produsul program şi numai prin încercări succesive va ajunge fie la utilizarea maximă a structurilor de date fie la alegerea adecvată a acestora;
- lucrările includ prezentări de structuri de date, de operaţii pe structuri de date şi proceduri scrise într-un limbaj de programare; cititorul are menirea de a utiliza procedurile, de a proiecta aplicaţii complexe care să folosească aceste proceduri; cititorul se autoinstruieşte în a integra proceduri disparate într-o construcţie unitară;
- lucrările conţin descrieri de structuri de date, proprietăţi, însoţite de secvenţe de cod sursă; de asemenea sunt prezentate operaţii pe structuri de date şi procedurile corespunzătoare, folosind definirile de operanzi prezentate anterior; se descriu clase de probleme reprezentative cu nivel de complexitate ridicat şi se dau programele care soluţionează aceste probleme; cititorul găseşte în acest tip de cărţi exemple reprezentative pe care le preia şi işi soluţionează direct propriile probleme, adaptând numai definiri de date, relizarea interfeţelor şi modul de stocare a informaţiilor utile din structuri dinamice în fişiere.

Cărţile de structuri de date trebuie să preia experienţa acumulată în timp în domeniul programării şi trebuie să includă capitole precum:

- conversii de structuri de date ;
- agregarea structurilor de date;
- analiza comparată a variantelor de program care folosesc structuri de date diferite;
- metrici ale structurilor de date;
- optimizarea programelor prin alegerea structurilor de date;
- creşterea calităţii programelor prin modul în care sunt utilizate structurile de date.

Rezultă că orice demers în abordarea domeniului structuri de date trebuie să fie bazat pe aceste tipuri de structuri, asigurând în acest fel

compatibilitatea cu toate lucrările de referinţă existente în literatura de specialitate.

Întrucât au apărut articole cu descrierea unor algoritmi performanţi de manipulare pe structuri de date şi au fost implementaţi în produse software din sisteme de gestiune a bazelor de date utilizate curent, cursurile moderne de structuri de date fac apel la aceşti algoritmi, referindu-i prin numele celor care i-au prezentat pentru prima dată. Prezentarea structurilor de date avansate este bazată pe o astfel de abordare şi este specifică numai în cazul unor studii aprofundate unde demersul este orientat spre detaliu şi spre urmărirea performanţei proceselor de stocare, căutare şi manipulare a elementelor din structurile de date.

În concluzie, înseamnă că orice carte de structuri de date trebuie să conţină:

- concepte de bază;
- masive;
- matrice rare;
- liste simple;
- liste duble;
- stive, cozi;
- arbori binari şi varietăţi;
- arbori B;
- grafuri;
- sortare şi căutare.

În anexe se prezintă:

- concepte de bază ale unui limbaj de programare
- cod sursă cu proceduri
- exemple complexe reprezentative pentru clase de probleme, utilizând structuri de date.

O astfel de structură crează premisele atingerii obiectivului de însuşire a tuturor elementelor necesare privind proiectarea şi realizarea de software performant, prin alocarea şi nivelarea de resurse, dintre care structurile de date reprezintă unele dintre resursele cele mai importante.