

Cuprins

8.Arbori

- 8.1. Arbori generalizați
 - 8.1.1. Definiții
 - 8.1.2. Tipul de date abstract arbore generalizat
 - 8.1.3. Traversarea arborilor generalizați
 - 8.1.3.1. Traversarea arborilor generalizați prin tehnici bazate pe căutarea în adâncime: preordine, inordine și postordine
 - 8.1.3.2. Traversarea arborilor generalizați prin tehnica căutării prin cuprindere
 - 8.1.4. Tehnici de implementare a TDA arbore generalizat
 - 8.1.4.1. Implementarea arborilor generalizați cu ajutorul tablourilor
 - 8.1.4.2. Implementarea arborilor generalizați cu ajutorul listelor
 - 8.1.4.3. Implementarea structurii arbore generalizat pe baza relațiilor “primul-fiu” și “frate-dreapta”
- 8.2. Arbori binari
 - 8.2.1. Definiții
 - 8.2.2. Tehnica transformării unei structuri de arbore generalizat într-o structură de arbore binar
 - 8.2.3. TDA Arbore binar
 - 8.2.4. Tehnici de implementare a arborilor binari
 - 8.2.4.1. Implementarea arborilor binari cu ajutorul structurii tablou
 - 8.2.4.2. Implementarea arborilor binari cu ajutorul pointerilor
 - 8.2.5. Traversarea arborilor binari
 - 8.2.5.1. Traversarea arborilor binari prin tehnici bazate pe căutarea în adâncime
 - 8.2.5.2. Traversarea arborilor binari prin tehnica căutări prin cuprindere
 - 8.2.6. Arbori binari cu legaturi ("Threaded Trees")
 - 8.2.6.1. Arbori binari cu legaturi plini ("Fully Threaded Trees")
 - 8.2.6.2. Arbori binari cu legaturi parțiali
 - 8.2.7. Aplicații ale arborilor binari
 - 8.2.7.1. Construcția și reprezentarea grafică a unui arbore binar de înălțime minimă
 - 8.2.7.2. Generarea unui arbore binar pornind de la specificarea sa în preordine
 - 8.2.7.3. Construcția unui arbore de parcurgere ("Parse Tree")
- 8.3. Arbori binari ordonați
 - 8.3.1. Definiții
 - 8.3.2. Tipul de date abstract arbore binar ordonat
 - 8.3.3. Tehnici de căutare în arbori binari ordonați
 - 8.3.4. Inserția nodurilor în ABO. Crearea arborilor binari ordonați
 - 8.3.5. Suprimarea nodurilor în ABO
 - 8.3.6. Analiza căutării în ABO
 - 8.3.7. Arbori binari parțial ordonați
 - 8.3.8. Aplicații ale ABO.
 - 8.3.8.1. Problema concordanței
 - 8.3.8.2. Generator de referințe încrucișate
- 8.4. Arbori de regasire (“Trie Trees”)
 - 8.4.1. Definire

- 8.4.2. Structura de date "Nod arbore de regăsire"
 - 8.4.2.1. Implementare bazată pe tablouri.
 - 8.4.2.2. Implementare bazată pe liste înlănțuite
- 8.4.3 . Structura de date "Arbore de regasire"
- 8.4.4. Evaluarea performanței structurii Arbore de regasire
- 8.5 Arbori binari echilibrați AVL
 - 8.5.1. Definirea arborilor echilibrați AVL
 - 8.5.2. Arbori Fibonacci
 - 8.5.3. Inserția nodurilor în arbori echilibrați AVL
 - 8.5.4. Suprimarea nodurilor în arbori echilibrați AVL
- 8.6 Arbori binari optimi
 - 8.6.1. Definirea arborilor binari optimi
 - 8.6.2. Construcția arborilor binari optimi
- 8.7 Arbori Huffman
 - 8.7.1. Coduri prefix. Algoritmul lui Huffman
 - 8.7.2. Arbori Huffman. Implementarea algoritmului lui Huffman
- 8.8. Reprezentarea grafică a structurii arbore
- 8.9. Arbori multicăi
 - 8.9.1. Generalități
 - 8.9.2. Arbori-B
 - 8.9.2.1. Definire
 - 8.9.2.2. Căutarea cheilor în arbori-B
 - 8.9.2.3. Inserția nodurilor în arbori-B
 - 8.9.2.4. Suprimarea nodurilor în arbori-B
 - 8.9.3 Arbori-B binari
 - 8.9.3.1. Arbori-B binari simetrici
 - 8.9.4 Arbori 2-3
 - 8.9.4.1. Definire
 - 8.9.4.2. Inserția nodurilor în arbori 2-3
 - 8.9.4.3. Suprimarea nodurilor din arbori 2-3

9. Structura de date mulțime

- 9.1. Introducere
- 9.2. Tipul de date abstract mulțime (varianta 2)
- 9.3. Implementarea structurii mulțime utilizând structuri de date fundamentale
 - 9.3.1. Implementarea structurii mulțime cu ajutorul vectorilor binari
 - 9.3.2. Implementarea structurii mulțime cu ajutorul listelor înlănțuite
- 9.4. Structuri de date derivate din structura multime
 - 9.4.1. Structura dicționar
 - 9.4.1.1. Implementarea structurii dicționar cu ajutorul tablourilor liniare
 - 9.4.1.2. Implementarea structurii dicționar prin tehnica dispersiei
 - 9.4.2. Structuri de date complexe bazate pe structura multime
 - 9.4.2.1. Relația bazată pe corespondențe multiple.
 - 9.4.2.2. Implementare bazată pe structura tablou a relației bazate pe corespondențe multiple
 - 9.4.2.3. Implementarea relației bazate pe corespondențe multiple cu ajutorul mulțimilor
 - 9.4.2.4. Structuri de date combinate
- 9.5. Implementarea structurii mulțime cu ajutorul structurilor de date avansate
 - 9.5.1. Implementarea structurii mulțime cu ajutorul arborilor binari

- ordonați
- 9.5.1.1. Considerații referitoare la performanțele implementării structurii mulțime cu ajutorul arborilor binari ordonați
- 9.5.2. Implementarea structurii mulțime cu ajutorul arborilor de regăsire
- 9.5.3. Implementarea structurii mulțime cu ajutorul arborilor binari echilibrați
- 9.6. Mulțimi pe care sunt definiți operatorii UNIUNE și CAUTĂ
 - 9.6.1. Implementare bazată pe tablouri a mulțimii pe care sunt definiți operatorii UNIUNE și CAUTĂ
 - 9.6.2. Implementare bazată pe liste înlanțuite a mulțimii pe care sunt definiți operatorii UNIUNE și CAUTĂ
 - 9.6.3. Implementare bazată pe arbori a mulțimii pe care sunt definiți operatorii UNIUNE și CAUTĂ
 - 9.6.3.1. Comprimarea drumului
- 9.7. Mulțimi pe care sunt definiți operatorii UNIUNE, CAUTĂ și PARTIȚIONARE
 - 9.7.1. Problema celei mai lungi subsecvențe comune (LCS)
 - 9.7.2. Determinarea LCS utilizând mulțimi pe care sunt definiți operatorii UNIUNE, CAUTĂ și PARTIȚIONARE
 - 9.7.3. Analiza performanței algoritmului LCS
- 9.8. Aplicații propuse

10. Structura de date graf

- 10.1. Definiții
- 10.2. Tipul de date abstract graf
 - 10.2.1. TDA graf. Varianta 1 (Shiflet)
 - 10.2.2. TDA graf. Varianta 2 (Decker)
- 10.3. Tehnici de implementare a tipului de date abstract graf
 - 10.3.1. Implementarea grafurilor cu ajutorul matricilor de adiacențe
 - 10.3.1.1. Studiu de caz 1
 - 10.3.1.2. Studiu de caz 2
 - 10.3.2. Implementarea grafurilor cu ajutorul structurilor de adiacențe
 - 10.3.2.1. Studiu de caz 1
 - 10.3.2.2. Studiu de caz 2
 - 10.3.2.3. Studiu de caz 3
 - 10.3.3. Considerente referitoare la stabilirea modului de reprezentare a unui TDA graf.
- 10.4. Tehnici fundamentale de traversare a grafurilor
 - 10.4.1. Traversarea grafurilor prin tehnica căutării ”în adâncime” (”Depth-First Search”)
 - 10.4.1.1. Căutarea ”în adâncime”, varianta CLR
 - 10.4.1.2. Căutare ”în adâncime” în grafuri reprezentate prin structuri de adiacențe
 - 10.4.1.3. Căutare ”în adâncime” în grafuri reprezentate prin matrici de adiacențe
 - 10.4.1.4. Căutare ”în adâncime” nerecursivă
 - 10.4.1.5. Analiza căutării ”în adâncime”
 - 10.4.2. Traversarea grafurilor prin tehnica căutării ”prin cuprindere” (”Breadth-First Search”)
 - 10.4.2.1. Căutarea ”prin cuprindere”, varianta CLR
 - 10.4.2.2. Căutare ”prin cuprindere” în grafuri reprezentate prin

- structuri de adiacențe
- 10.4.2.3. Analiza căutării "prin cuprindere"
- 10.4.3. Comparatie între tehnicile fundamentale de traversare a grafurilor
- 10.5. Aplicații ale traversării grafurilor
 - 10.5.1. Arbori de acoperire ("Spanning Trees"). Tehnici de determinare a arborilor de acoperire
 - 10.5.1.1. Determinarea unui arbore de căutare "în adâncime"
 - 10.5.1.2. Determinarea unui arbore de căutare "prin cuprindere"
 - 10.5.2. Grafuri și conexiuni
 - 10.5.2.1. Determinarea componentelor conexe ale unui graf
 - 10.5.2.2. Puncte de articulație și componente biconexe
 - 10.5.2.3. Determinarea punctelor de articulație ale unui graf
- 10.6. Aplicații

11. Grafuri ponderate ("Weighted Graphs")

- 11.1. Arbori de acoperire minimi ("Minimum-Cost Spanning Trees")
 - 11.1.1. Proprietatea arborilor de acoperire minimi
- 11.2. Determinarea arborilor de acoperire minimi
 - 11.2.1. Algoritmul lui Prim
 - 11.2.1.1. Exemplu de implementare a algoritmului lui Prim
 - 11.2.2. Metoda căutării "bazate pe prioritate" ("Priority-First Search")
 - 11.2.2.1. Considerații referitoare la metoda căutării "bazate pe prioritate"
 - 11.2.3. Algoritmul lui Kruskal
 - 11.2.3.1. Exemplu de implementare a algoritmului lui Kruskal
- 11.3. Drumul minim ("Shortest Path")
 - 11.3.1. Determinarea drumurilor minime cu origine unică corespunzătoare unui nod al unui graf prin tehnica căutării "bazate pe prioritate"
- 11.4. Arbori de acoperire și drumuri minime în grafuri dense
- 11.5. Considerente referitoare la performanțele comparate ale algoritmilor de determinare a arborilor de acoperire minimi
- 11.6. Aplicații

12. Grafuri orientate

- 12.1. Problema drumurilor minime cu origine unică ("Single-Source Shortest Path Problem")
 - 12.1.1. Algoritmul lui Dijkstra
 - 12.1.2. Demonstrarea funcționalității algoritmului lui Dijkstra
 - 12.1.3. Analiza performanței algoritmului lui Dijkstra
- 12.2 Problema drumurilor minime corespunzătoare tuturor perechilor de noduri ("All-Pairs Shortest Path Problem")
 - 12.2.1. Algoritmul lui Floyd
 - 12.2.2. Comparatie între algoritmul lui Floyd și algoritmul lui Dijkstra
 - 12.2.3. Determinarea traseelor drumurilor minime
 - 12.2.4. Aplicație. Determinarea centrului unui graf orientat ponderat
- 12.3. Închiderea tranzitivă
 - 12.3.1. Algoritmul lui Warshal
 - 12.3.2. Analiza performanței algoritmului lui Warshal
- 12.4. Traversarea grafurilor orientate
 - 12.4.1. Traversarea grafurilor orientate prin tehnica căutării "în

adâncime"

12.4.2. Păduri de arbori de căutare în adâncime pentru grafuri orientate

12.5. Grafuri orientate aciclice

12.5.1. Determinarea aciclității unui graf orientat

12.5.2. Aplicație. Sortarea topologică

12.6. Componente puternic conectate

12.6.1. Algoritmul lui Kosaraju-Sharir

12.6.2. Algoritmul lui Tarjan

12.7. Rețele de curgere ("Network-Flow")

12.7.1. Problema rețelelor de curgere

12.7.2. Metoda Ford-Fulkerson

12.7.3. Implementarea metodei Ford-Fulkerson. Căutarea în rețea

12.8. Problema potrivirilor ("Matching")

12.8.1. Grafuri bipartite

12.8.2. Determinarea potrivirii maxime prin tehnica drumurilor augmentate

12.8.3. Determinarea potrivirii maxime prin metoda căutării în rețele de curgere

12.9. Aplicații