

Tema de casa

Analiza Algoritmilor

Aplicația 8.1.

Se cere să se implementeze următorii operatori definiți peste **TDA Arbore binar ordonat**: *Creaza*, *Cauta*, *Insereaza*, *SupriMin* și *Suprimă*. Pentru implementare se va utiliza metoda bazată pe pointeri

Aplicația 8.2.

Se cere să se implementeze următoarele modalități de traversare ale unui arbore binar:

- Căutare în adâncime varianta preordine
- Căutare în adâncime varianta inordine
- Căutare în adâncime varianta postordine
- Căutare prin cuprindere, varianta cu evidențierea nivelurilor arborelui binar.

Utilizând implementarea de la aplicația 8.1 se cere în continuare:

- Să se construiască un arbore binar ordonat
- Să se afișeze pe ecran structura acestuia (vezi secvența [8.2.7.1.a])
- Să se parcurga arborele și să se afișeze succesiunea nodurilor, în fiecare dintre modalitățile de căutare mai sus precizate.

Aplicația 10.1

Se cere să se implementeze **TDA graf** (Varianta 1, Shiflet) utilizând matrici de adiacențe. Se vor implementa următorii operatori: *InitGraf*, *GrafVid*, *GrafPlin*, *InserNod*, *InserArc*, *SuprimNod*, *SuprimArc*. Tabloul nodurilor se consideră neordonat. Se cere să se precizeze performanțele operatorilor implementați în termenii funcției O.

Aplicația 10.2

Se cere să se implementeze TDA graf (varianta 1, Shiflet) utilizând structuri de adiacență reprezentate prin liste neordonate simplu înlănțuite. Se vor implementa următorii operatori: *InitGraf*, *GrafVid*, *InserNod*, *InserArc*, *SuprimNod*, *SuprimArc*. Se vor preciza performanțele operatorilor implementați în termenii funcției O.

Aplicația 10.3

Se cere să se redacteze un program care realizează următoarele activități:

- (1) Acceptă la intrare un graf precizat prin mulțimea nodurilor și arcelor sale;

- (2) Vizualizează graful afișând lista nodurilor și lista arcelor sale
- (3) Listează nodurile în ordinea parcurgerii lor pe baza căutării în adâncime;

Programul se va realiza în două variante:

- a) Graf reprezentat prin matrice de adiacențe;
- b) Graf reprezentat prin structuri de adiacențe.

Se vor utiliza reprezentările și operatorii de la problemele 10.1 respectiv 10.2

Aplicația 10.4

Se cere să se rezolve aplicația 10.3 utilizând în locul căutării în adâncime, căutarea prin curpindere.

Aplicația 11.1

Se cere să se redacteze un program care implementează algoritmul lui PRIM (secvența [11.2.1.1.a]). Programul primește la intrare în graf precizat prin mulțimea nodurilor și arcelor sale și vizualizează un arbore de acoperire minim al grafului.

Aplicația 12.1.

Se cere să se realizeze o implementare a algoritmului lui Dijkstra pentru determinarea drumurilor minime cu origine unica pentru un graf.

Aplicația 12.2

Se cere să se realizeze un program bazat pe algoritmul lui Floyd care primește la intrare un graf orientat ponderat precizat prin mulțimea nodurilor și arcelor sale și care determină lungimea drumurilor minime pentru toate perechile de noduri.

Precizari

1. Exercițiile se rezolvă în limbajul de programare C.
2. Fiecare exercițiu se redactează ca și un fișier separat de tip *.cpp.
3. Exercițiile vor fi precedate de un preambul de tip comentariu care descrie structurile de date folosite și maniera de abordare.
4. Fiecare funcție precum și programul principal sunt precedate de descrieri de tip comentariu care le explicitează
5. Liniile de program vor fi comentate la nivel de detaliu.