

Se da un graf orientat $G=(V, E)$. Sa se determine componentele sale tare conexe: numarul lor, iar pentru fiecare componenta, sa se enumere varfurile.

Descriere implementare

Se citeste matricea de adiacenta a unui graf orientat.

→ alocare memorie graf

→ alocare memorie matrice si citire din fisier text

Rezolvarea problemei consta in parcurgerea DF in ambele directii, atat pe succesori, cat si pe predecesori. In acest sens, se creeaza 2 vectori de vizitare: pred si suc. Nodurile care au 1 in cazul ambelor parcurgeri, fac parte din aceeaasi componenta tare conexa.

Ex: pred: 1 1 0 0 1

suc : 1 0 0 0 1

Varfurile 0 si 4 fac parte din aceeaasi componenta tare conexa.

Pentru afisarea varfurilor care alcatuiesc componentele tare conexe, la fiecare parcurgere, atunci cand nodul este vizitat, vectorii suc si pred iau valoarea numarului componentei tare conexe din care fac parte.

Nodurile care au aceeaasi valoare si la parcurgerea predecesorilor, si la cea a succesorilor, fac parte din aceeaasi componenta tare conexa.

Nodul cu care incepe parcurgerea in adancime pentru o componenta tare conexa se cauta printre nodurile nevizitate inca. Pentru acest nod se determina multimea nodurilor care formeaza subgraful succesorilor si multimea nodurilor care formeaza subgraful predecesorilor. Prin intersectia celor 2 multimi de noduri se obtine componenta tare conexa. Identificarea celor 2 multimi de noduri se face parurgand in adancime graful pentru fiecare multime, pornind de la un nod initial. Un nod este vizitat numai daca a fost adaugat la o componenta tare conexa. Prelucrarea componentelor tare conexe prin parcurgerea in adancime a grafului se face pana cand vor fi vizitate toate varfurile.

Exemple de utilizare practica

Intr-un grup de n persoane intre care s-au stabilit relatii de cunoastina se poate stabili cel mai mare grup format din cele mai multe persoane care se cunosc reciproc.