

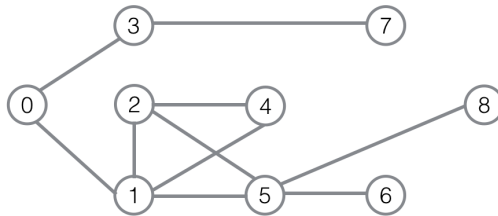
Probleme Curs 8 SDA

Probleme propuse pentru grafuri. Unele din problemele propuse sunt preluate de pe *GeeksForGeeks*.

Probleme trasare

Problema 1 Se da graful de mai jos.

- (a) Dati reprezentarea prin matrice de adiacenta a grafului.
- (b) Aplicati algoritmul $BFS(G, 1)$. Desenati arborele rezultat.
- (c) Desenati continutul cozii la fiecare pas al algoritmului de la punctul (b).

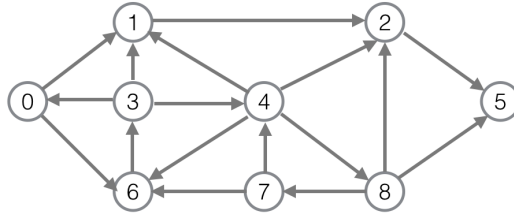


Problema 2 Pe graful de la problema 1:

- (a) Desenati arborele rezultat in urma aplicarii algoritmului $DFS_Visit(G, 0)$.
- (b) Marcati, pe arbore, timpii de descoperire si de finalizare ai fiecarui nod parcurs.
- (c) Se pot ordona nodurile grafului intr-o ordine topologica? Motivati.

Author(s): Raluca Brehar, Camelia Lemnaru

Problema 3 Se da urmatorul graf, $G = (V, E)$. Trasati algoritmul lui Kosaraju, care determina componentele tare conexe ale lui G , si desenati graful componentelor.



Problema 4 Pentru graful de la problema 3:

- Dati reprezentarea prin liste de adiacenta a grafului (nodurile sa apara in ordine lexicografica).
- Aplicati algoritmul $BFS(G, 0)$. Desenati arborele rezultat.
- Desenati continutul cozii la fiecare pas al algoritmului de la punctul (b).

Problema 5 Pentru graful de la problema 3:

- Desenati arborele rezultat in urma aplicarii algoritmului $DFS_Visit(G, 0)$.
- Marcati, pe arbore, timpii de descoperire si de finalizare ai fiecarui nod parcurs.
- Macati, pe graf, tipul fiecarei muchii, asa cum reiese din parcurgerea ceruta.

Probleme elaborare algoritmi

BFS

Problema 6 Se da un graf neorientat. Scrieti pseudocodul pentru a determina care sunt nodurile din drumul de lungime minima de la un varf s la un varf u si care este lungimea drumului. Analizati eficienta algoritmului propus.

Problema 7 Modificati algoritmul $BFS(G,s)$ astfel incat sa stocati pentru fiecare varf v nu doar distanta de la s la v , notata cu $d[v]$, cat si numarul drumurilor diferite de la s la v care au lungimea $d[v]$. Analizati eficienta algoritmului propus.

Problema 8 Se da un graf orientat si doua varfuri x si y din graf. Proiectati un algoritm care determina toate drumurile de la varful x la varful y . Analizati complexitatea algoritmului propus.

Problema 9 Se da un graf orientat si doua varfuri x si y din graf. Proiectati un algoritm care determina cate drumuri de lungime k exista între cele doua varfuri din graf. Analizati complexitatea algoritmului propus.

Problema 10 Se da un graf neorientat. Elaborati un algoritm care gaseste un ciclu de lungime impara in graf (daca exista). Daca nu exista, algoritmul va indica acest lucru. Analizati complexitatea algoritmului propus.

Problema 11 Se da un graf orientat. Modificati algoritmul de traversare in latime astfel incat sa determinati daca exista un ciclu in graf .

Explicatie. Pentru fiecare nod vizitat, v daca este un nod adiacent u astfel incat u a fost vizitat iar parintele lui u este diferit de v atunci in graf exista un ciclu. Daca pentru orice varf nu gasim un astfel de nod atunci nu exista cicluri.

Problema 12 Padurea de parcurgere in adancime clasifica muchiile unui graf in muchii de tip arbore, inainte, transversale si inapoi. BFS se poate folosi de asemenea pentru a clasifica muchiile grafului. Demonstrati ca, aplicand BFS pe un graf neorientat, urmatoarele proprietati sunt adevarate:

- (a) nu avem muchii inainte sau inapoi
- (b) pentru fiecare muchie (u,v) de tip arbore: $d[v] = d[u] + 1$.
- (c) pentru fiecare muchie (u,v) transversala: $d[v] = d[u]$ sau $d[v] = d[u] + 1$.

Demonstrati ca, aplicand BFS pe un graf orientat, urmatoarele proprietati sunt adevarate:

- (a) nu avem muchii inainte
 - (b) pentru fiecare muchie (u,v) de tip arbore: $d[v] = d[u] + 1$.
 - (c) pentru fiecare muchie (u,v) transversala: $d[v] \leq d[u] + 1$.
 - (d) pentru fiecare muchie (u,v) inapoi: $0 \leq d[v] \leq d[u]$.
-

DFS

Problema 13 Scrieti pseudocodul unui algoritm care determina daca un graf neorientat cu V varfuri are sau nu cicluri. Complexitatea algoritmului sa fie $O(|V|)$.

Explicatie. Un graf neorientat este fara cicluri daca DFS nu produce muchii inapoi (back edges).

Problema 14 Un ciclu Eulerian al unui graf $G=(V,E)$ – conex, orientat – este un ciclu ce traverseaza fiecare muchie din G exact o data (desi ar putea vizita un varf de mai multe ori).

- (a) Aratati ca G are ciclu Eulerian iff $\text{inDegree}(v) = \text{outDegree}(v)$, pentru orice varf $v \in V$.
 - (b) Descrieti un algoritm in timp $O(E)$ pentru a gasi un ciclu Eulerian, daca acesta exista (sugestie - combinarea ciclurilor disjuncte).
 - (c) Ce se modifica daca graful nu este orientat?
-