



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

Proiectarea Algoritmilor

14. Parcurgere în lățime (BFS)

Bibliografie

- Giumale – Introducere în Analiza Algoritmilor cap 5 si 5.1
- Cormen – Introducere în Algoritmi cap 22, 22.1, 22.2, 22.3 si 22.4
- <http://ist.marshall.edu/ist362/pics/OSPF.gif>
- <http://ashitani.jp/gv/>
- <http://en.wikipedia.org/wiki/PageRank>

Algoritmi de parcurgere – Notatii (1)

- $G = (V, E)$;
- V – mulțimea de **noduri**;
- E – mulțimea de **muchii / arce**;
- (u, v) – arcul / muchia u, v ;
- $u..v$ – drum de la u la v ; dacă există **mai multe variante** notăm $u..x..v$, $u..y..v$;
- $R(u)$ - **reachable**(u) = mulțimea nodurilor ce pot fi atinse pe căi ce pleacă din u ;



Algoritmi de parcurgere – Notații (2)

- $\text{succs}(u)$ – mulțimea **succesorilor** lui u (graf **orientat**) sau mulțimea nodurilor **adiacente** lui u (graf **neorientat**);
- $c(u)$ – **culoarea nodului** – specifică **starea nodului la un anumit moment al parcurgerii**:
 - **Alb** – nedescoperit;
 - **Gri** – descoperit, în curs de prelucrare;
 - **Negru** – descoperit și terminat (cu semnificații diferite pentru BFS și DFS).
- $p(u)$ ($\pi(u)$) – “**părintele lui u** ” – identificator al nodului din care s-a ajuns în nodul u prima oară.

Parcurgere în lățime (BFS)

- Nod de start (sursă): s .
- Determină numărul minim de arce / muchii între s și $\forall u \in V =$ numărul de pași între sursă și orice alt nod din graf (acesta este cel mai scurt drum în condițiile în care nu există o funcție de cost asociată grafului).
- $\delta(s,u)$ – costul optim al $s..u$; $\delta(s,u) = \infty \Rightarrow u \notin R(s)$.
- $\text{Dist}(s,u)$ – costul drumului descoperit $s..u$.
- Ex: Politehnica \rightarrow restul stațiilor de autobuz (de câte bilete am nevoie?)

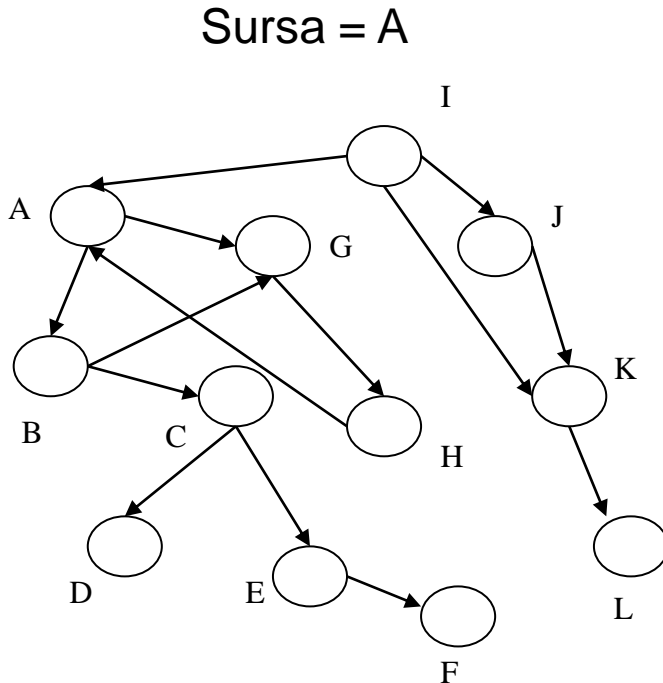
BFS – Structura de date

- Folosește o **coadă (FIFO)** pentru a reține nodurile ce trebuie prelucrate.
- Pentru fiecare nod se rețin:
 - **Părintele** – $\pi(u)$ ($p(u)$);
 - **Dist(s,u)** – distanța până la nodul sursă;
 - **Culoarea nodului.**

BFS – Algoritm

- **BFS(s,G)**
 - **Pentru fiecare** nod u ($u \in V$)
 - $p(u) = \text{null}$; $\text{dist}(s,u) = \text{inf}$; $c(u) = \text{alb}$; // inițializări
 - $Q = ()$; // se folosește o coadă în care reținem nodurile de prelucrat
 - $\text{dist}(s,s) = 0$; // actualizări: distanța de la sursă până la sursă este 0
 - $Q \leftarrow Q + s$; // adăugăm sursa în coadă \rightarrow începem prelucrarea lui s
 - $c(s) = \text{gri}$; // și atunci culoarea lui devine gri
 - **Cât timp** ($! \text{empty}(Q)$) // cât timp mai am noduri de prelucrat
 - $u = \text{top}(Q)$; // se determină nodul din vârful cozii
 - **Pentru fiecare** nod $v \in \text{succs}(u)$ // pentru toți vecinii
 - **Dacă** $c(v)$ este alb // nodul nu a mai fost găsit, nu e în coadă
 - **Atunci** $\{ \text{dist}(s,v) = \text{dist}(s,u) + 1$; $p(v) = u$; $c(v) = \text{gri}$; $Q = Q + v$; }
// actualizăm structura date
 - $c(u) = \text{negru}$; // am terminat de prelucrat nodul curent
 - $Q = Q - u$; // nodul este eliminat din coadă

BFS – Exemplu

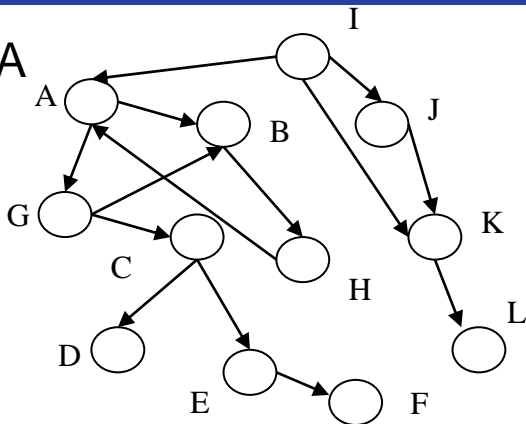


● BFS(s,G)

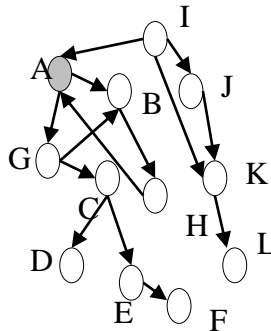
- **Pentru fiecare** nod u ($u \in V$)
 - $p(u) = \text{null}$; $\text{dist}(s,u) = \text{inf}$; $c(u) = \text{alb}$; // inițializări
- $Q = ()$; // se folosește o coadă pentru nodurile de prelucrat
- $\text{dist}(s,s) = 0$; // actualizări: distanța de la s la s e 0
- $Q \leftarrow Q + s$; // adăugăm sursa în coadă → începem cu s
- $c(s) = \text{gri}$; // și atunci culoarea lui devine gri
- **Cât timp** (!empty(Q)) // cât timp am noduri de prelucrat
 - $u = \text{top}(Q)$; // se determină nodul din vârful cozii
 - **Pentru fiecare** nod $v \in \text{succs}(u)$ // pentru toți vecinii
 - Dacă $c(v)$ este alb // nodul nu a mai fost găsit, nu e în Q
 - **Atunci** { $\text{dist}(s,v) = \text{dist}(s,u) + 1$; $p(v) = u$; $c(v) = \text{gri}$; $Q = Q + v$; } // actualizăm structura date
 - $c(u) = \text{negru}$; // am terminat de prelucrat nodul curent
 - $Q = Q - u$; // nodul este eliminat din coadă

BFS – Evoluția explorării

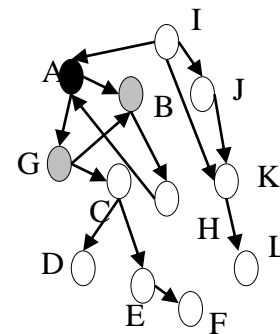
Sursa = A



$Q = A; d(A) = 0$
 $p(A) = \text{null}$

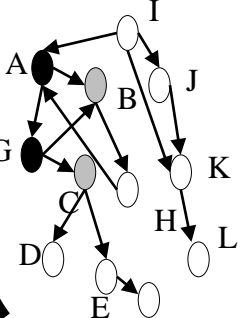


$Q = B, G$
 $d(B) = d(G) = 1$

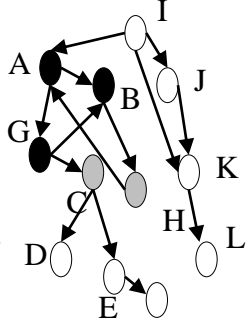


$p(B) = A$
 $p(G) = A$

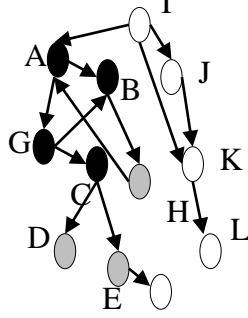
$Q = G, C$
 $d(C) = 2$



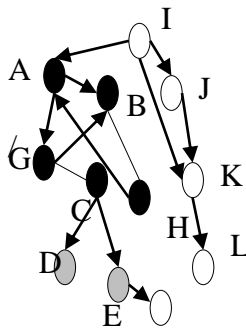
$Q = C, H$
 $d(H) = 2$



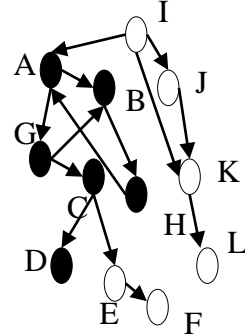
$Q = H, D, E$
 $d(D) = d(E) = 3$



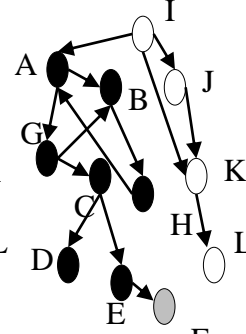
$Q = D, E$



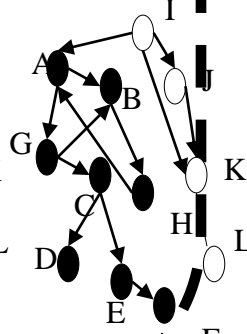
$Q = E$



$Q = F$
 $d(F) = 4$



$Q = \emptyset$



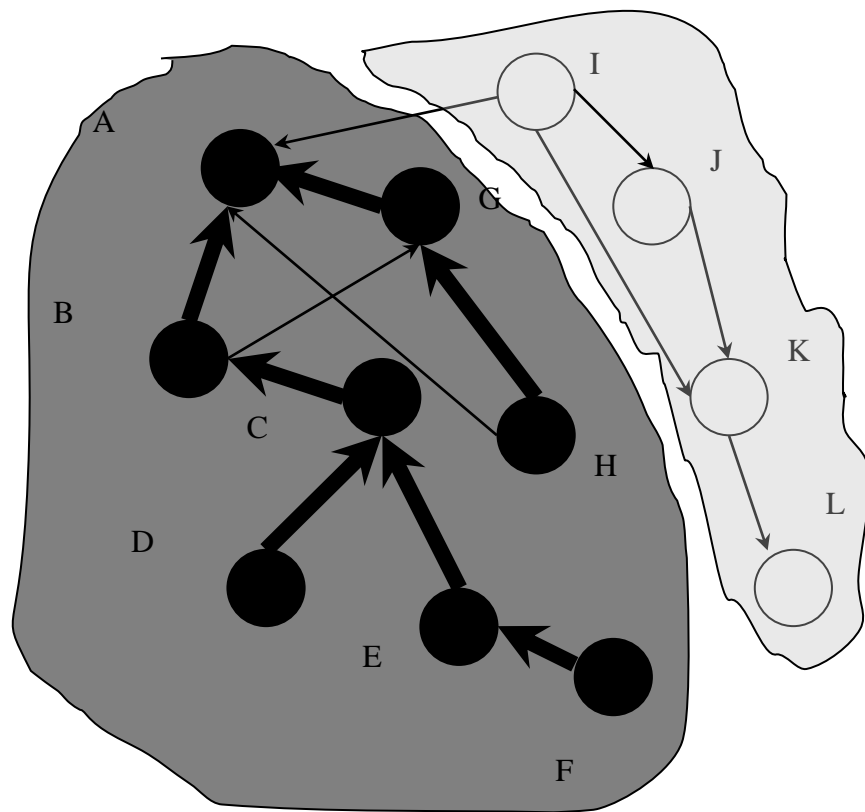
$p(C) = B$

$p(H) = G$

$p(D) = p(E) = C$

$p(F) = E$

BFS – Zona de explorare



BFS – Proprietăți (I)

- **Lema 5.1.** În cursul execuției BFS(s, G) $v \in Q \Leftrightarrow v \in R(s)$.
 - $\rightarrow Q$ conține exclusiv noduri din $R(s)$ (BFS parcurge toate nodurile ce pot fi atinse din s);
 - \leftarrow toate nodurile ce pot fi atinse din s vor fi introduse cândva în coadă. (drum $s = v_0 v_1 \dots v_p = v$)
 - Dem prin inducție!
- **Lema 5.2.** $\forall (u, v) \in E, \delta(s, v) \leq \delta(s, u) + 1$
 - $\delta(s, v) \leq \delta(s, u) + 1$ în general sunt = când v este descoperit din u ; < când v deja a fost descoperit înainte să se ajungă în u .
 - Dem prin reducere prin absurd!

BFS – Proprietăți (II)

- **Lema 5.3.** La terminarea BFS(s, G) există proprietatea $\text{dist}(s, u) \geq \delta(s, u)$.
 - Dem prin inducție folosind Lema 5.1 și Lema 5.2!
- **Lema 5.4.** După orice execuție a ciclului principal al BFS, Q conține v_1, v_2, \dots, v_p ai:
 - $\text{Prop}(Q) = \text{dist}(s, v_1) \leq \text{dist}(s, v_2) \leq \dots \leq \text{dist}(s, v_p) \leq \text{dist}(s, v_1) + 1$
 - \Rightarrow la un moment dat în coadă sunt elemente de pe același nivel din arborele generat de BFS (sau maxim 1 nivel diferență).
 - Dem prin inducție după numărul de elemente din Q ! (demonstrăm invarianța $\text{Prop}(Q)$ la inserare și eliminare de elemente în/din Q .)

BFS – Proprietăți (III)

● Corolar

- $d(u)$ = momentul in care nodul u este inserat in coada Q .
Atunci:

$$d(u) < d(v) \Rightarrow \text{dist}(s,u) \leq \text{dist}(s,v).$$

- **Teorema 5.1. BFS este corect** si după terminare $\delta(s,u) = \text{dist}(s,u)$, $\forall u$ din V .

- Utilizăm notația $V_k = \{ v \in V \mid \delta(s,u) = k \}$
- Dem prin inducție $P(k) = \{ u \in V_k \mid \delta(s,u) = \text{dist}(s,u) \ \&\& \ (k > 0 \Rightarrow \pi(u) \in V_{k-1}) \ \&\& \ (k = 0 \Rightarrow \pi(u) = \text{null}) \}$

Complexitate? Optimalitate? Completitudine?



BFS – Complexitate și Optimalitate

Complexitate:

$$O(n+m)$$

n = număr noduri

m = număr muchii

Optimalitate: DA

Parcurge tot graful? NU