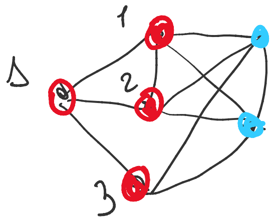
Lezione 11 maggio

Monday, May 11, 2020 11:12 AM

BFS Visita in ampiezza



$G=(V,E)$

Lista di adiacenza

$x.d$ = distanza da s

$x.\pi$ = predecessore nella ...

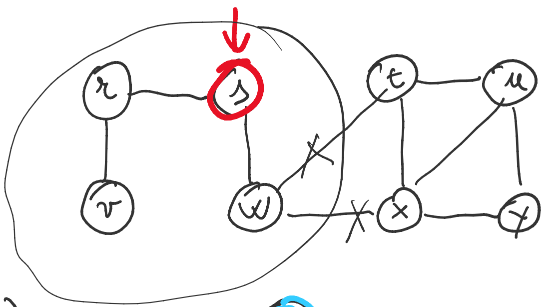
$\ell_1 + \ell_2 + \dots + \ell_n = 2m$ grafi non or.

$= m$ graf orient.

$\Theta(n+m)$ tempo $\quad \Theta(n)$ spazio $\downarrow$ per la coda Q

BFS su matrice di adiacenza?

induce un albero **BFS** (spanning tree)



grafo statico

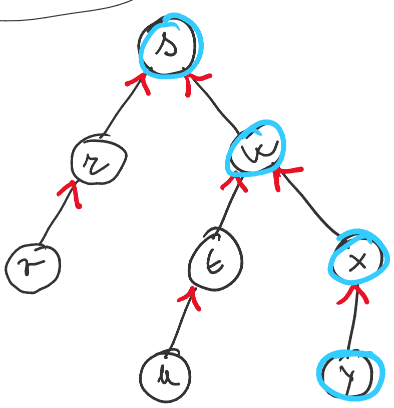
PP ($\cancel{G}$, s, y)

PP (G, $\cancel{s}$, x)

PP ($\cancel{G}$, s, w)

~~PP (G, s, s)~~

s, w, x, y



BFS

$\forall$ nodo dell'albero la distanza minima = prof. del nodo

raggiungibilità (connesso?)

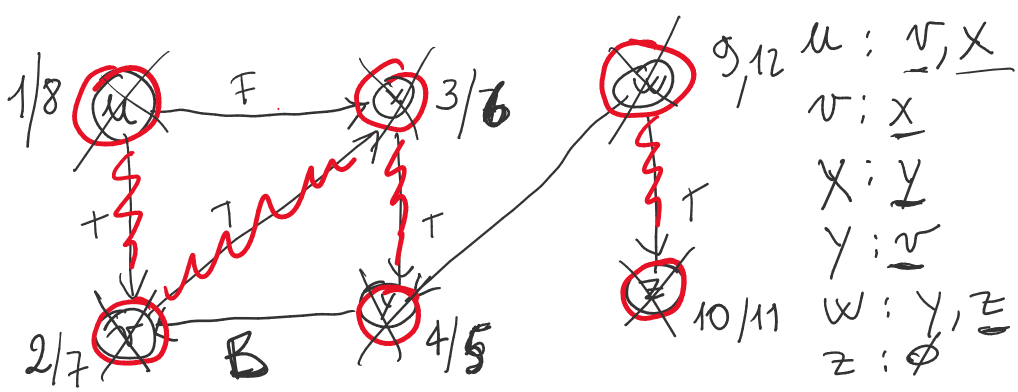
raggiungibilità (connesso ?)
percorsi minimi da ogni nodo a s
diametro BFS da ogni nodo

Stampa il percorso minimo
da s a qualsiasi nodo v
dopo BFS

```
PRINT_PATH(G, s, v):
if (v == s) print s;
   else if {(v.π == NIL) print "non ci sono cammini"
      else { PRINT_PATH(G, s, v.π);
             print v;
      }
}
```

DFS visita in profondità
(negli alberi = visita anticipata)

time 1
" 2
3
4
5
6
7
8
9
10
11
12



visita anche i nodi non raggiungibili da s.

u u.color iniziale: white
u π " : NIL

u   u.color   iniziale: white

u.π   " : NIL

u.d   tempo in cui u è stato scoperto

u.f   " è finita la visita di u

time   incrementato   ogni volta che inizia la visita di un nodo oppure quando termina la visita di un nodo

DFS è una visita ricorsiva
(usa la pila per la ricorsione)

```
DFS(G)
for ogni vertice u ∈ G.V {
    u.color = white;
    u.π = NIL;              inizializzazione
} time = 0;
Θ(n) for ogni vertice u ∈ G.V {
    if (u.color == white) DFS_Visit(G,u);
}

DFS_Visit(G,u);
time = time+1;     ← tempo inizio visita
u.d = time
u.color = gray;
for ogni v ∈ Adj[u] {
    if (v.color == white){
```

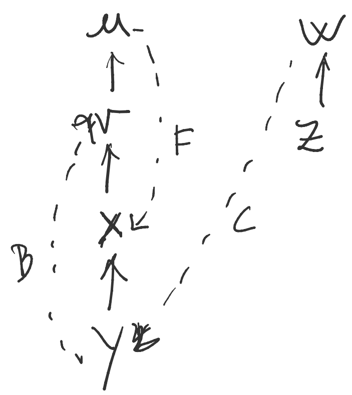
for ogni $v \in Adj[u]$ {
    if (v.color == white){
        $v.\pi = u$;
        DFS_Visit(G, v);
    }
} u.color = black;
time = time + 1;
$u.f$ = time; ← tempo fine visita

$$\ell_1 + \ell_2 + \dots \ell_m = \Theta(m)$$

$\Theta(n+m)$ tempo

$\Theta(n)$ spazio (pila)

Induce una foresta DFS



T archi dell'albero : (u,v) u è pred v nella DFS

B archi all'indietro : (u,v) v è antenato di u nella foresta

F archi in avanti : (u,v) v è discendente di u nella foresta

C archi trasversali : (u,v) u non è né antenato né discendente

Problema:

antenato né discendente

Problema:

Estendere la procedura DFS in modo tale da classificare tutti gli archi

1) Test arco dell'albero: (u, v)

se v = white

2) Test arco all'indietro (u, v)

v è grigio

3) Test arco in avanti (u, v)

v è nero $\quad u.d < v.d$

4) Test arco di attraversamento

v è nero $\quad u.d > v.d$