

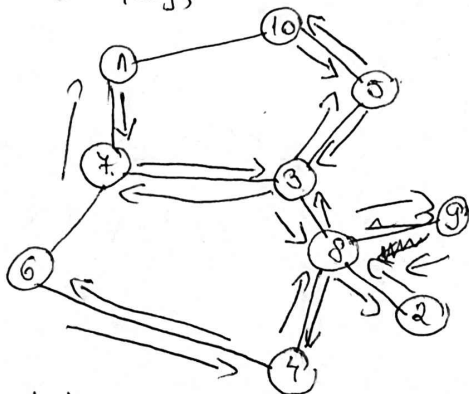
DFS(v)

$vis(v) = 1$;
 $print(v)$;
 pentru toate nodurile u vecine cu v
 dacă $vis(u) = 0$
 $pred(u) = v$
 DFS(u)

BFS(v)

creada C ;
 $C \leftarrow \{v\}$;
 $vis[v] = 1$;
 cât timp C e nevidă
 $x \leftarrow C$;
 pt. fiecare u vecin al lui x
 dacă $vis[u] = 0$,
 $vis[u] = 1$;
 $pred[u] = x$;
 $C \leftarrow \{u\}$;

Exemplu:



DFS(1): 1, 7, 3, 5, 10, 8, 2, 4, 6, 9

1	2	3	4	5	6	7	8	9	10
8	7	8	3	4	1	3	2	5	

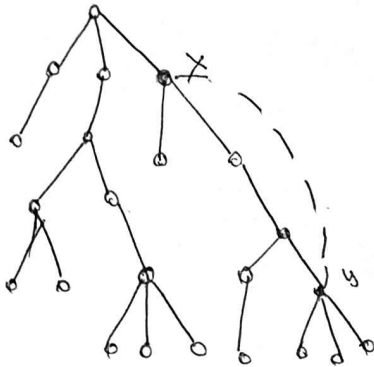
BFS(1): 1 7 10 3 6 5 8 4 2 9

1	2	3	4	5	6	7	8	9	10
8	7	6	10	7	1	3	8	1	

COADĂ: ~~1~~ ~~7~~ ~~10~~ ~~3~~ ~~6~~ ~~5~~ ~~8~~ ~~4~~ ~~2~~ ~~9~~

DFS Tree

- muchiile care se află în graf, dar nu sunt în arborele DFS se numesc muchii de întoarcere.
- muchiile din arbore sunt muchii de înaintare.



Status[mod] = { white, dacă nu a fost vizitat
gray, dacă a fost vizitat,
dar mă pot întoarce la el prin recursiv
black, modul a fost părăsit definitiv

DFS(u)

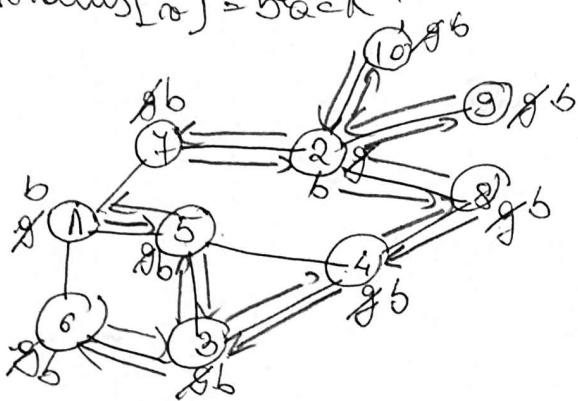
status[u] = gray.
print [u].

pt. toate modulele v , reame care
dacă status[v] = white

pred(u) = v

DFS(v)

status[u] = black;



1, 5, 3, 4, 8, 2, 7, 9, 10, 6

Tot v e în stânga modului
curent e black,

Tot v e dinasupra lui e gray,
tot v e sub el poate fi black-v.
Tot v e în dreapta e n/.