Note Title 04-06-2024

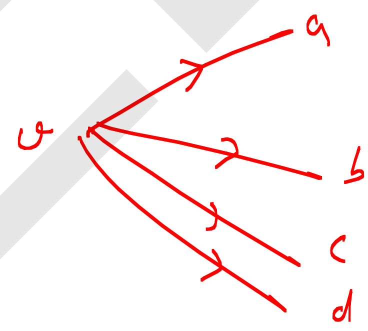
Iterative DFS (Non recursive DFS).

DFS (G, u)

For each $v \in Adj(u)$

if v is not visited

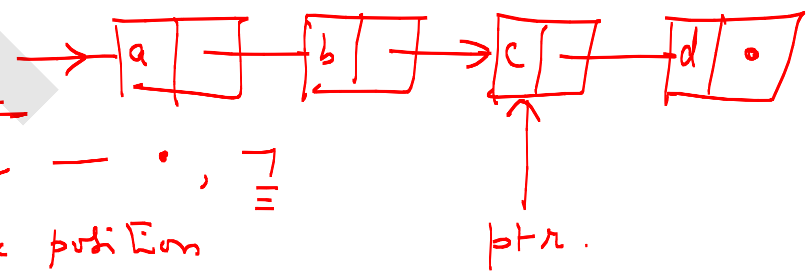
DFS (G, v).



G - represented as an Adj List.

$v \rightarrow Adj(v)$.

v. first

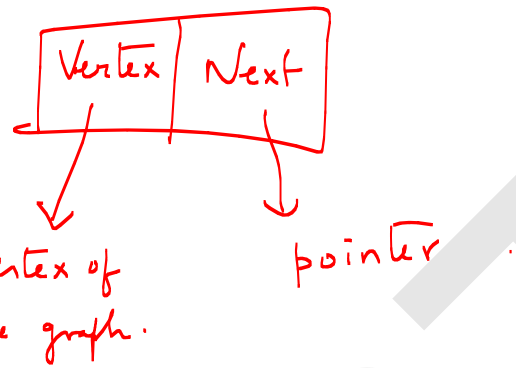


NULL — •, ⏚

ptr — ind pointer to indicate a specific position in Adj List.

(v, ptr) — position indicated by the pointer ptr in the Adj (v).

A box in the linked list



Non_Recursive_DFS (G, u)

$S = \phi$ // S is a stack containing pairs of the forms (v, ptr) that corresponds to the position of the neighbour of v to be explored.

Time = Time + 1

u.d = Time

u.color = Gray

Push[(u, u.first), S]

While ($S \neq \phi$)

(v, ptr) = POP(S)

if (ptr ≠ NULL) → Push((v, ptr.next), S)

w = ptr.vertex;

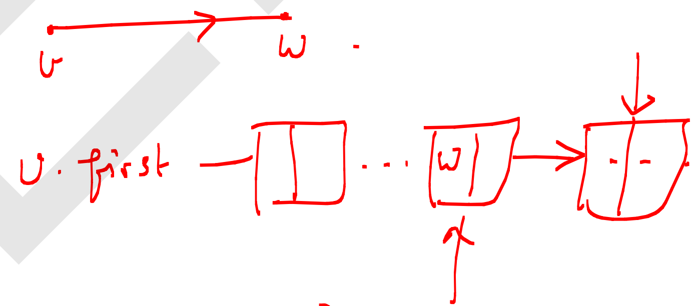
if (w.color == White)

w.p = v

time = time + 1;

w.color = Gray;

push[(w, w.first), S].



ptr

else : // (v, w) is a non tree edge.
// Do computations related to non - tree edge (v, w).

else // (ptr = NULL).

v. color = Black;
Time = Time +1;
v. f = Time.

(v, ptr) = (v, NULL) indicates that all neighbours of v were visited. Hence visit at v may be finished.

Keep pushing the positions for which explorations are still to be done into the stack.

After finishing the visit corresponding to the position, (v, ptr), we must continue with $(v, ptr.next)$. That is why, we push $(v, ptr.next)$ into the stack.

$n = |V|$

$m = |E|$.

The complexity is same as the recursive one $O(n+m)$.