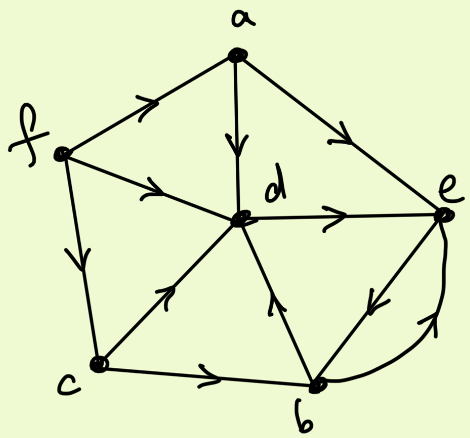
COL 106 Lecture 38

Topic : Directed Graphs and their Depth - First Traversal

Directed Graph: $G = (V, E)$, where V is a finite set, and $E \subseteq V \times V$ (ie edges are directed)

(Directed paths, walks, cycles are defined in the obvious manner.)

u is called the <u>tail</u> and v is called the <u>head</u> of edge (u,v).



eg. a-d-e-b is a directed path

There is no directed path from b to a

is the only directed cycle

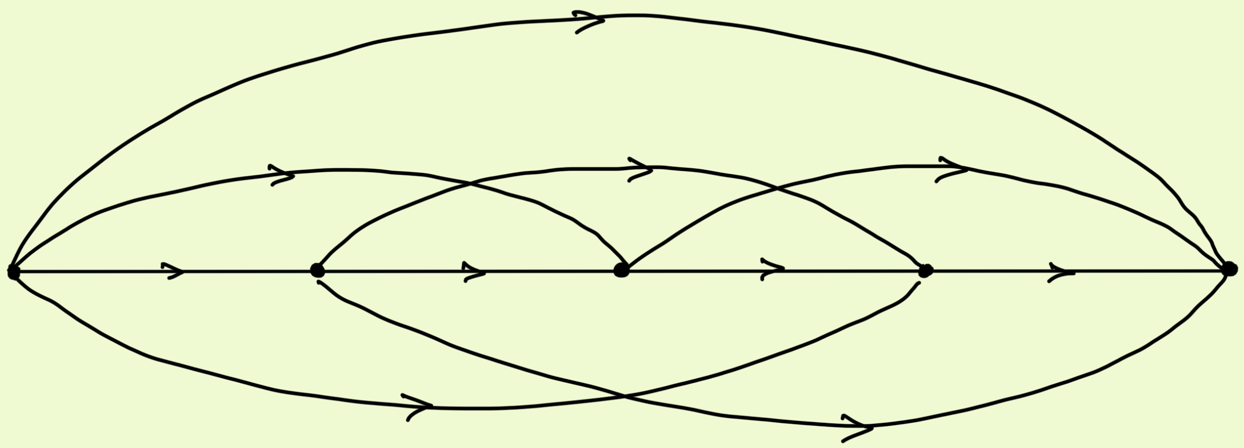
Question : What does it mean for a directed graph to be connected ?

A directed graph $G = (V, E)$ is called <u>strongly connected</u> if $\forall\, a, b \in V$, G has a directed path from a to b.

A directed acyclic graph that doesn't have a directed cycle is called a <u>Directed Acyclic Graph (DAG)</u>.

Observation: A DAG on n vertices can have as many as $n(n-1)/2$ edges.

eg. $V = \{1, \dots, n\}$ $E = \{(u,v) \mid u,v \in V, u<v\}$.



Representations of Directed Graphs

Adjacency matrix representation of $G=(V,E)$:

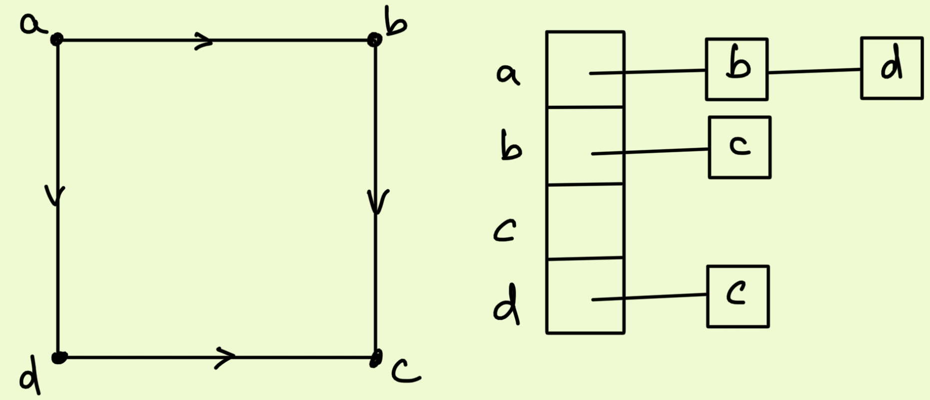
$|V| \times |V|$ matrix, say A, with rows and cols indexed by V.

$$A[u,v] = \begin{cases} 1 & \text{if } (u,v) \in E \\ 0 & \text{if } (u,v) \notin E \end{cases}$$

Note: Adjacency matrix of a directed graph is not necessarily symmetric.

Adjacency List Representation of Directed Graphs

- Array / Linked List indexed by vertices.
- Entry of vertex v contains reference of linked list containing out-neighbors of v.



Depth-First Traversal of Directed Graphs

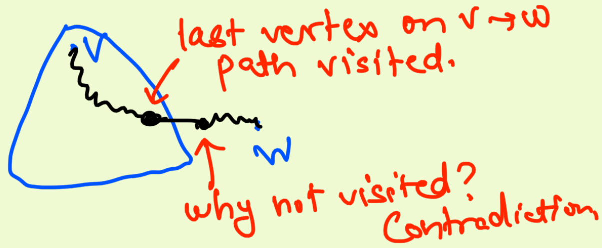
Recursive Subroutine:

```
DFT_rec (v):
    arr [v] ← count
    count ← count +1.
    For each out-neighbor u of v:
        If arr [u] = -1:
            prev [u] ← v
            DFT_rec (u)
    dep [v] ← count
    count ← count + 1
```

Claim: If arr [u] is initialised to −1 ∀ u and DFT_rec(v) is performed, then it visits a vertex w iff there is a v to w path

Proof: w visited ⇒ v to w path exists: follow prev[], then reverse.

v to w path exists ⇒ w visited:



(Global) Depth-First Traversal

DFT()

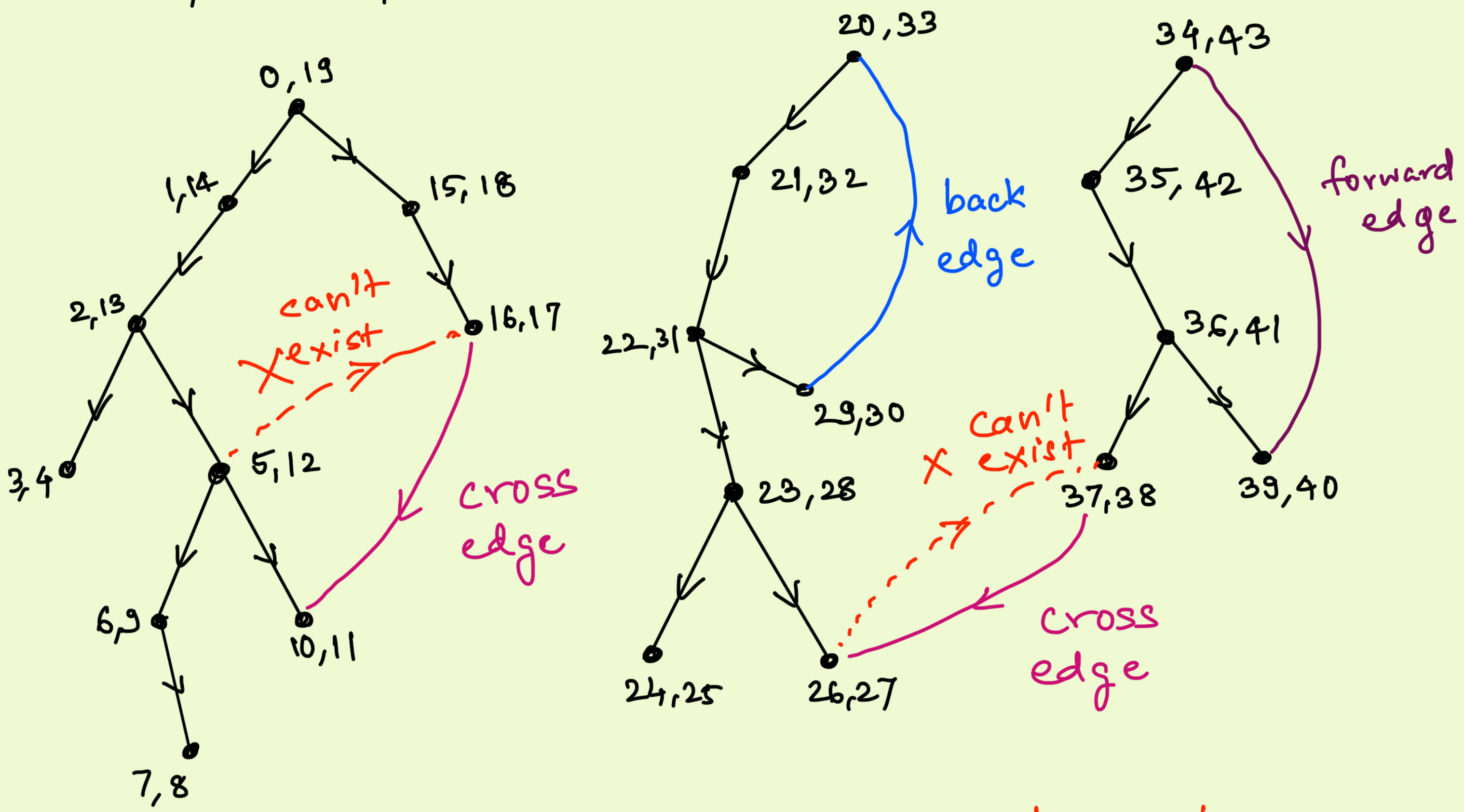
arr[v] ← −1, dep[v] ← −1 ∀ vertices v

For each vertex v:

If arr[v] = −1 then

DFT_rec(v)

Example output:



A DFT classifies edges of a directed graph into the following 4 types.

1. Tree edges: (u,v) where prev[v] = u
2. Forward edges: (u,v) where v is a descendant of u but prev[v] ≠ u.
3. Back edges: (u,v) where v is an ancestor of u
4. Cross edges: (u,v) where v is neither ancestor nor descendant of u. (This implies v must be visited before u.)

Claim: Consider a DFT of a directed graph and the resulting arrival and departure times of vertices. Suppose (u,v) is an edge.

1. If (u,v) is a tree edge
2. If (u,v) is a forward edge

$\Big\}$ then $arr[u] < arr[v] < dep[v] < dep[u]$.

3. If (u,v) is a back edge, then $arr[v] < arr[u] < dep[u] < dep[v]$
4. If (u,v) is a cross edge, then $arr[v] < dep[v] < arr[u] < dep[u]$

Testing Acyclicity of Directed Graphs

isAcyclic (G):

1. Perform global DFT of G.
2. If G has a back edge:

    return FALSE → Back edge + some tree edges form a directed cycle

   Else:

Can we say that G is necessarily acyclic? Proof?