Graph

$G = (V, E)$

vertex set $\quad E \subseteq V \times V$.

m edges.

n vertexes

- undirected:

$(u,v) \in E \iff (v,u) \in E$

- directed:

- adjacency matrix

$$M_G = \begin{array}{c|c} & v \\ \hline u & \begin{pmatrix} & 1 & \\ 0 & & 1 \\ & 0 & \end{pmatrix} \end{array}$$

- adjacency list: 1: 3, 7, 10
  2: 1, 6, 9
  3

	matrix	list
$(u,v) \overset{?}{\in} E$	$O(1)$	$\cancel{O(n)}$ $O(d)$
neigh(v)	$O(n)$	$\cancel{O(n)}$ $O(d)$
space.	$O(n^2)$	$O(m)$

Connectivity.

- depth-first search (DFS)

DFS(G):

Explore(G, v) — mark vertices reachable from v.

visited[v] = true.

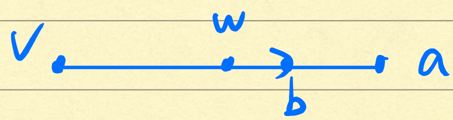
for each $(v, u) \in E$;

- $\forall v \in V$, visited[v] = false.
- $\forall v \in V$; if not visited[v]
  explore(G, v).

if not visited[u]:
    explore(G, u)

claim: explore(G, v) visits every node reachable from v:

Proof: suppose not. let a be a vertex that is not visited and is reachable



Run time of DFS: $O(n+m)$

Connectivity in undirected graph.

G $\rightarrow$ connected components

$u, v \in V$ are connected.

$$\begin{pmatrix} u \rightarrow v \\ v \rightarrow u \end{pmatrix}$$

CC(G):
- $\forall v \in V$, visited[v] = false.
  ccnum[v] = null.
- cc := 0.

Explore(G, u)
- visited[u] = true.
- $\boxed{\text{ccnum[u] = cc.}}$
- $\forall (v,u) \in E$:

- $\forall v \in V$:
  - if not visited
    - cc++
    - explore(G, v)

if not visited[u]:
Explore (G, u)

Maintain a global clock for time on stack.

DFS(G):

- $\forall v \in V$, visited[v] = false.
- $\boxed{\begin{array}{l}\text{pre[v] = null.}\\ \text{post[v] = null.}\end{array}}$
- clock = 0.
- $\forall v \in V$, if not visited[v]:
  explore(G, v)

Explore (G, v) :

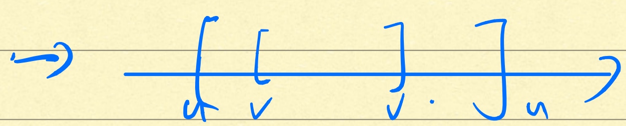
compute how long v is on the call stack

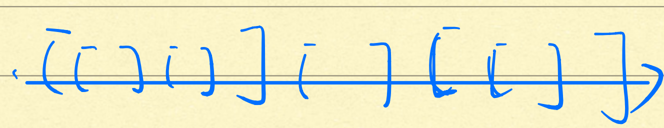
- visited[v] = true.
- $\boxed{\begin{array}{l}\bullet\ \text{pre[v] = clock.}\\ \bullet\ \text{clock++.}\end{array}}$
- $\forall (v, u) \in E$, if not visited[u]:
  explore(G, u)
- $\boxed{\begin{array}{l}\bullet\ \text{post[v] = clock}\\ \bullet\ \text{clock++.}\end{array}}$

Read off cc:

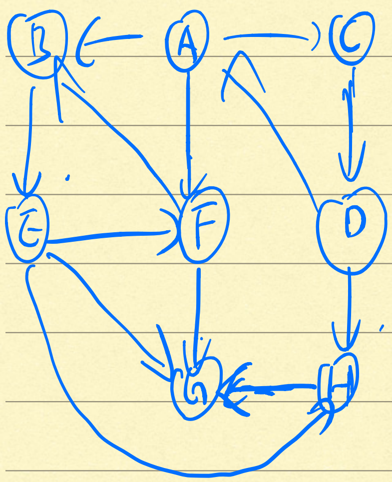
Type of intervals:

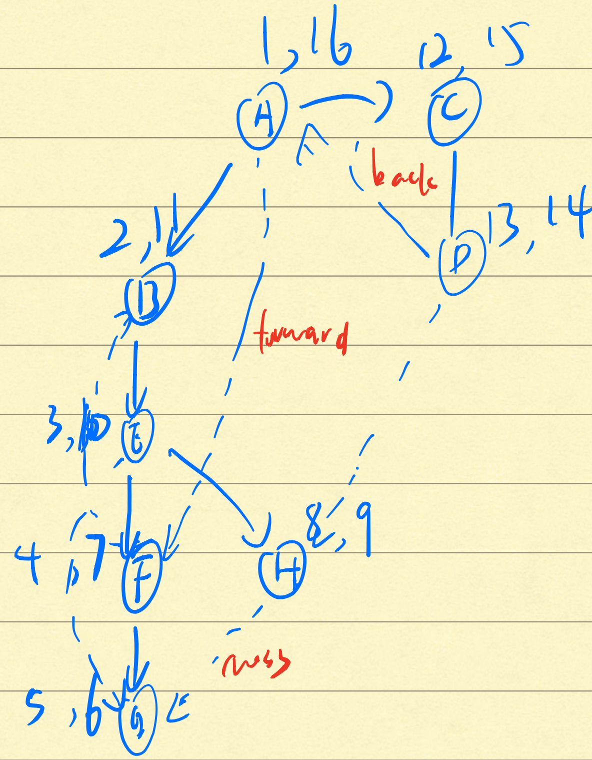
(u, v) $\in E$ →





↳ connected components.

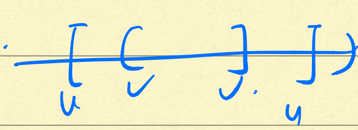




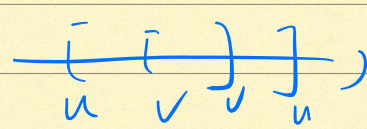
Type of edges:

$(u,v) \in E$

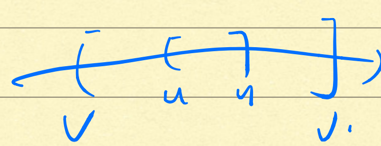
- tree edge (solid).



- forward edge :



- back edge



- cross edge.