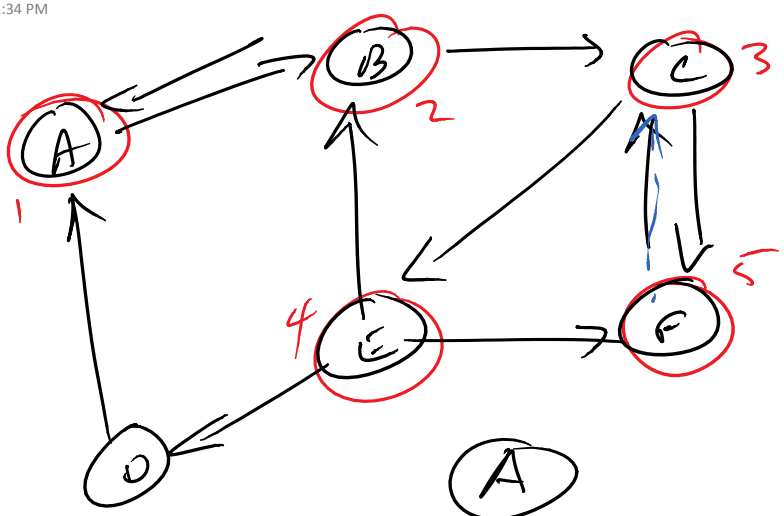
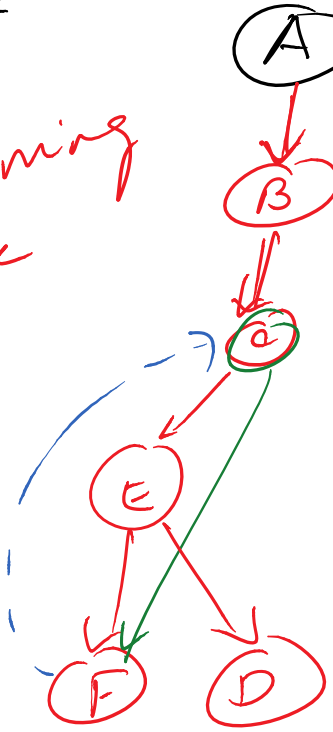


DFS spanning tree

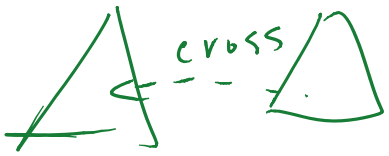


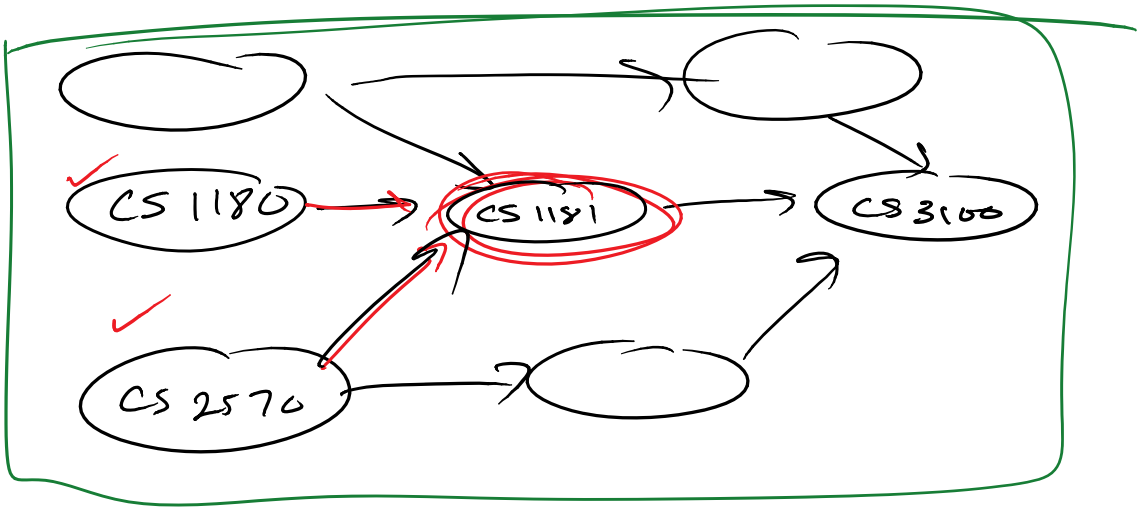
tree edges

back edge

forward edge

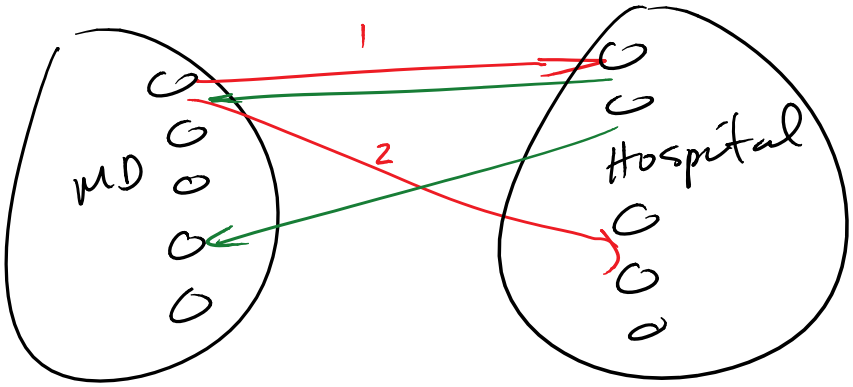
cross edge



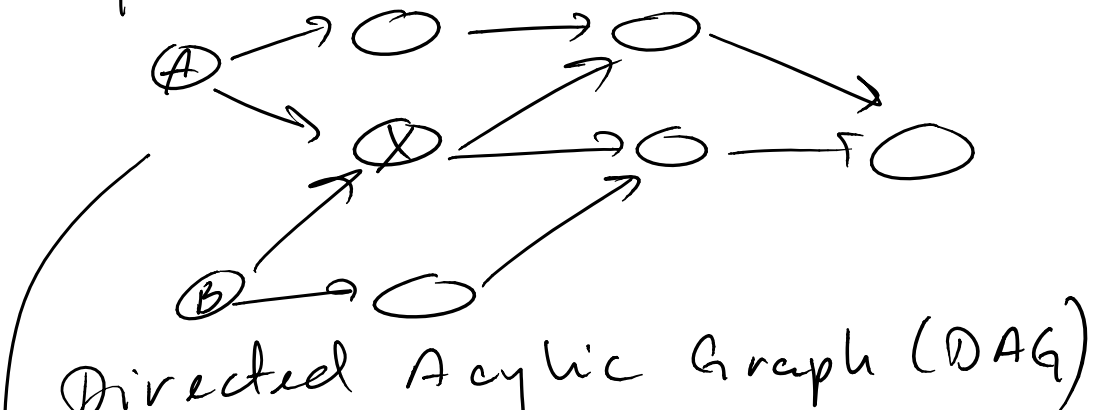


Med school

stable marriage

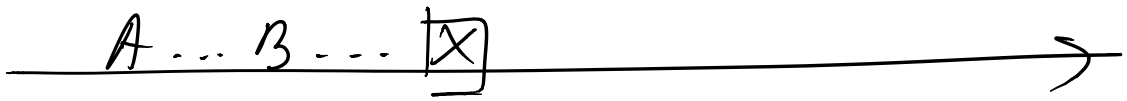


Topological Sort



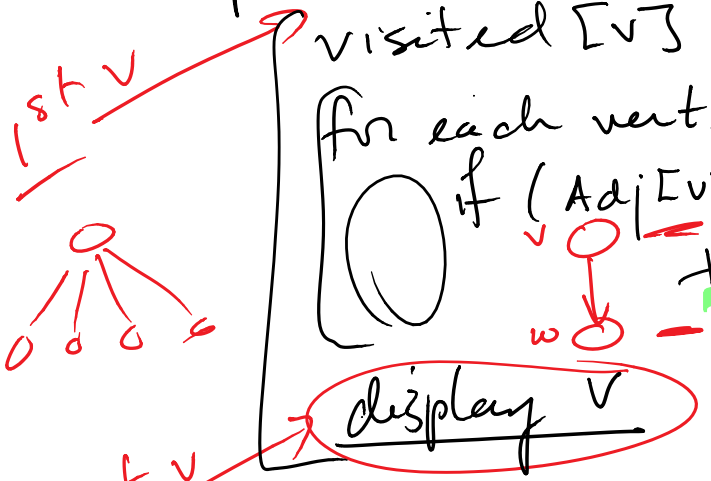
Directed Acylic Graph (DAG)

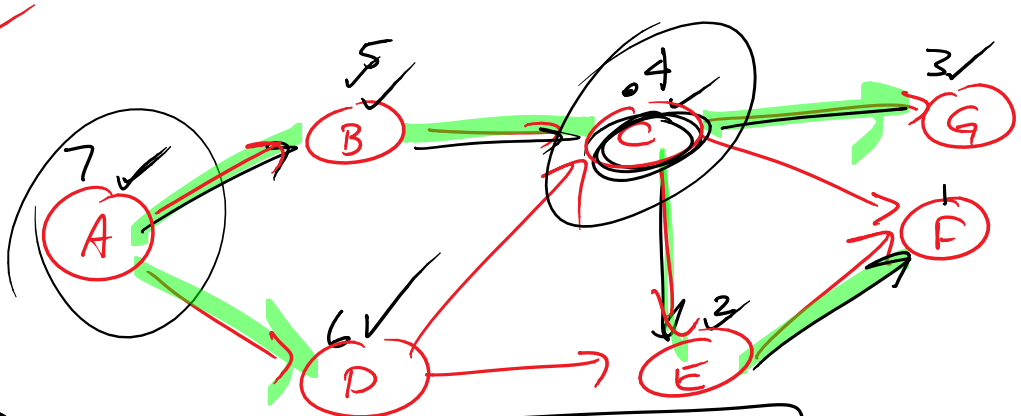
Directed Acylic Graph (DAG)



Topological Sort

```
topsort (vertex v)
visited[v] = true
for each vertex w
    if (Adj[v][w] && !visited[w])
        topsort(w)
display v
```





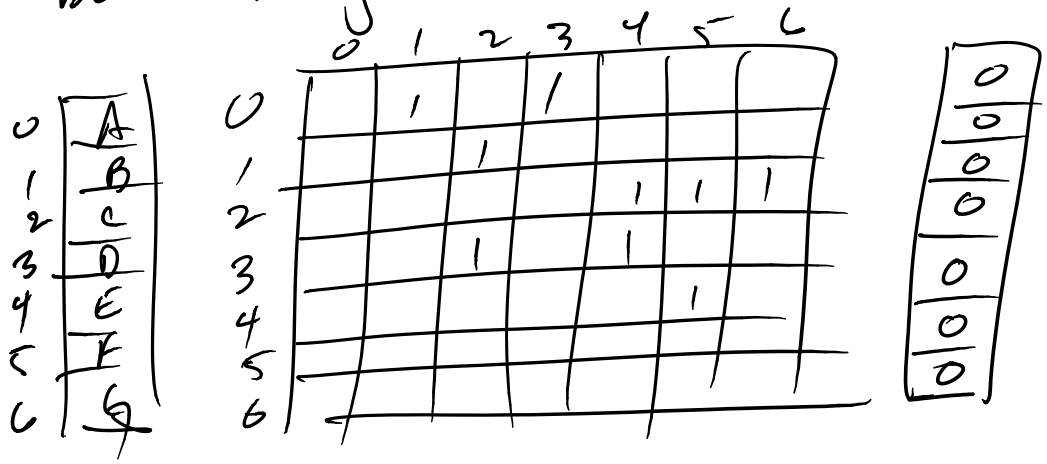
reverse order

F E G C B D A

A D B C G E F

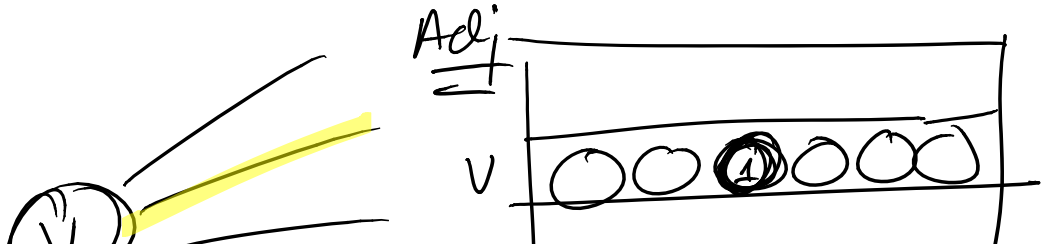
bool Adj [7] [7];

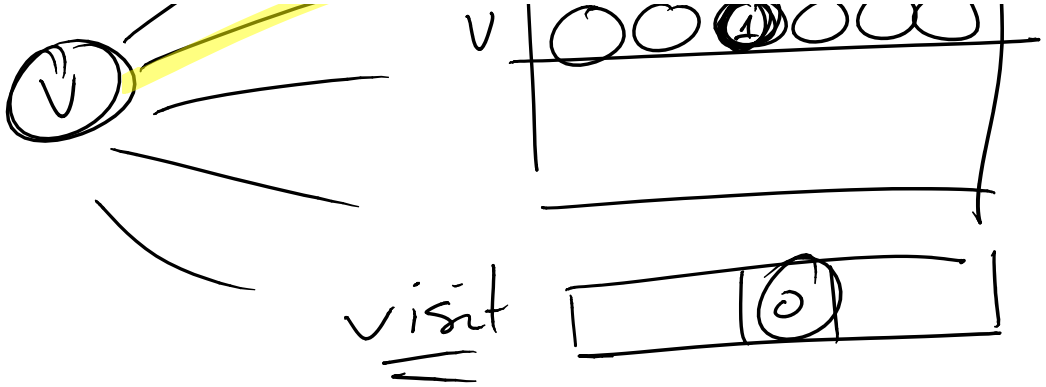
visited [7]




```
void DFS (bool Adj[][],
          bool visited[],
          int v, int numV)
{
  visited[v] = true;
  for (w = 0; w < numV; w++)
  {  if (Adj[v][w] && !visited[w])
       DFS(Adj, visited, w, numV);
  }
  cout << v; // top sort
}
```

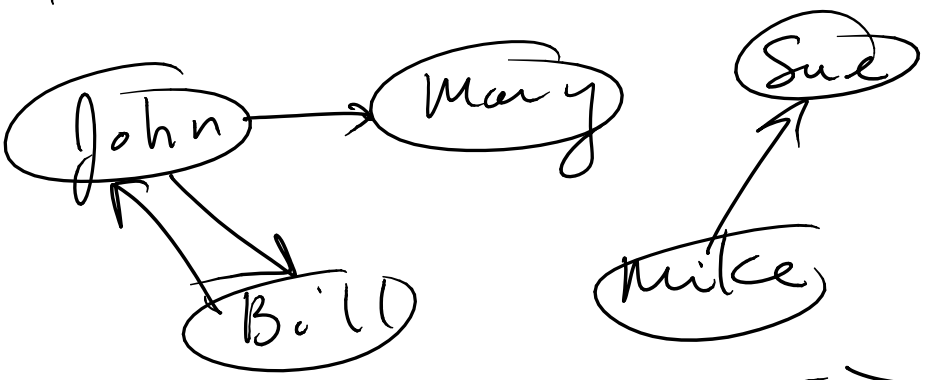
w=0 w=1 .....





Properties Directed Graphs

brother of



$DG = \langle V, E \rangle$

$=$

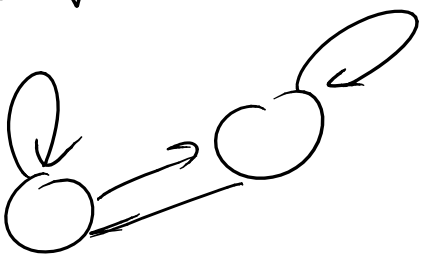
$\forall$ for all

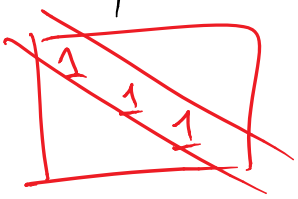
$\in$ element of

$\notin$ not element of

Reflexive

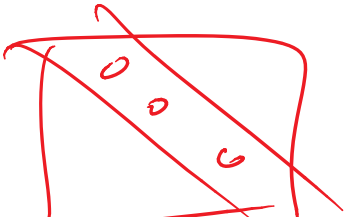
$\forall x \in V \quad \langle x, x \rangle \in E$



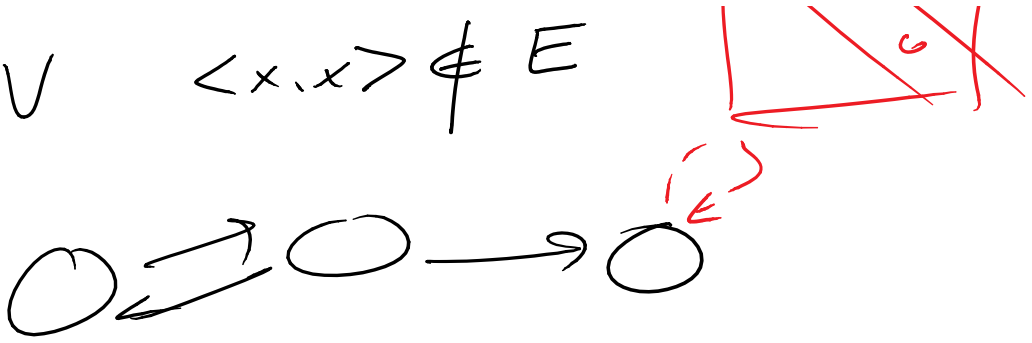


Irreflexive

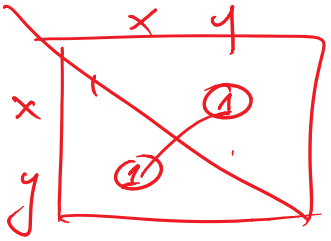
$\forall$ x all $\quad \langle x, x \rangle \notin E$



$\forall x \in V \quad \langle x, x \rangle \notin E$

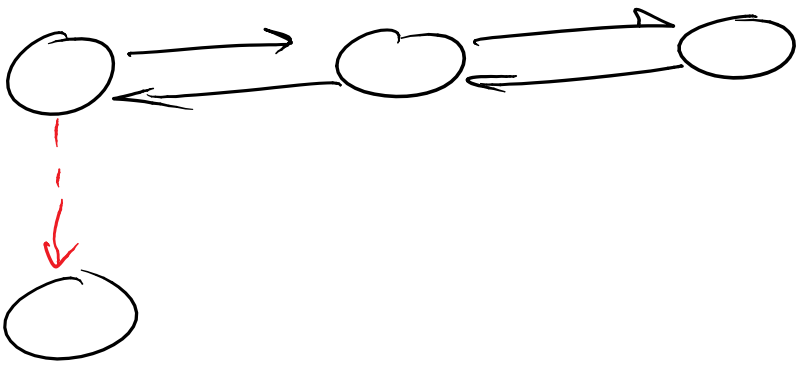


Symmetry



$\forall x \in V$ and $\forall y \in V$

if $\langle x, y \rangle \in E$ then $\langle y, x \rangle \in E$



Asymmetric

$\forall x \in V$ and $\forall y \in V$

if $\langle x, y \rangle \in E$ then $\langle y, x \rangle \notin E$