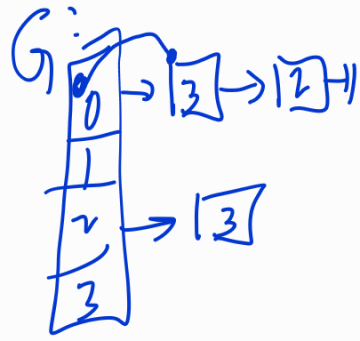
A


A'

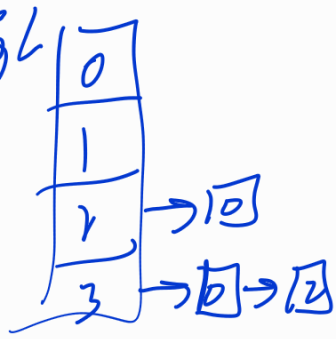

```
Graph GRAPHreverse(Graph G)
{
    Graph GL = GRAPHinit(G->V);
    for(int v=0; v<G->V; v++)
    {
        for(link l = G->adj[v]->next; l!=NULL; l=l->next)
        {
            GRAPHinsertE(GL, EDGE(l->v, v));
        }
    }
    return GL;
}
```

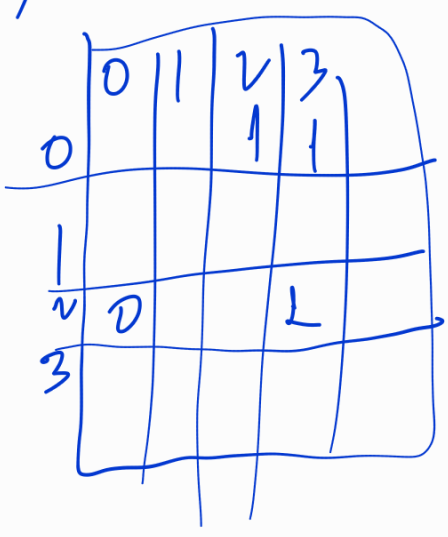
G:

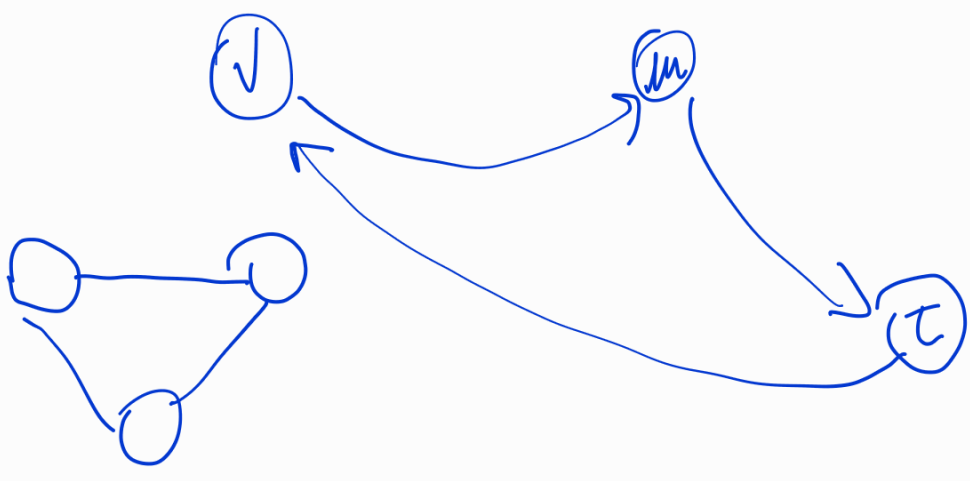



$(V+E)$

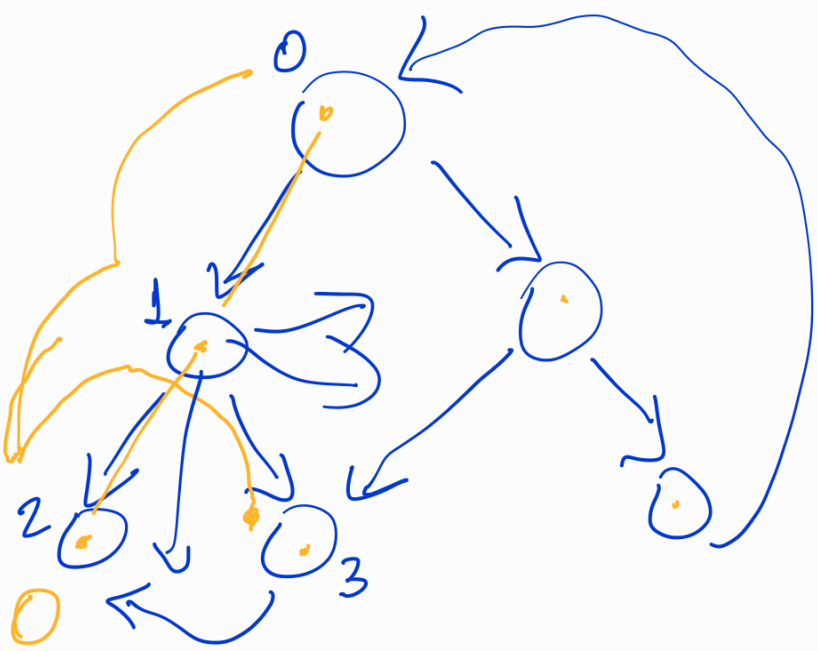
GL

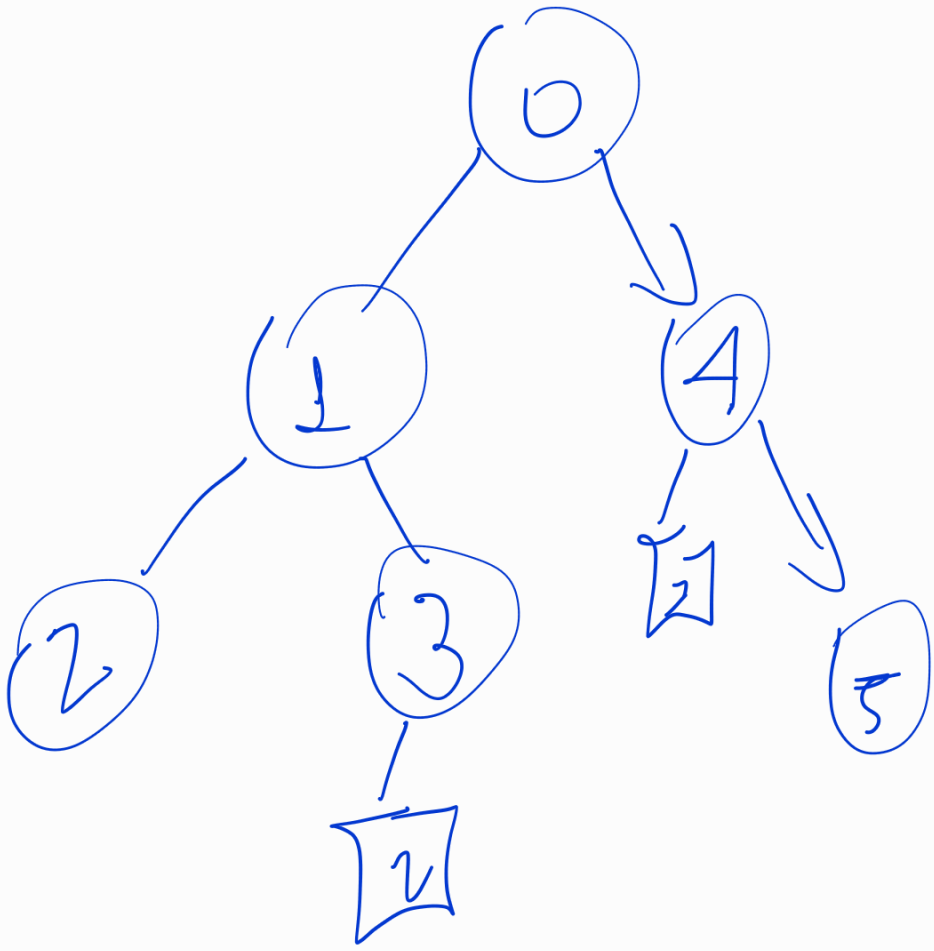


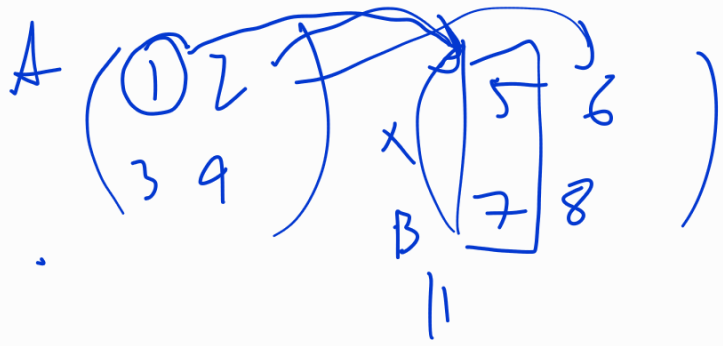



```
void dfsRd(Graph G, Edge e)
{
  int u = e.u;
  pre[u] = cnt++;
  for(link l = G->adj[u]->next; l != NULL; l = l->next)
    if(pre[l->v] == -1)
      dfsRd(G, EDGE(u, l->v));
    else
    {  if(pos[l->v] == -1)
         VOLTEI();
       else if(pre[l->v] < pre[u])
         ALOGO();
       else OUTROALGO()
    }
  pos[u] = cntP++;
```




0
1
4
2
3
1
5
2

$C = A \times B$

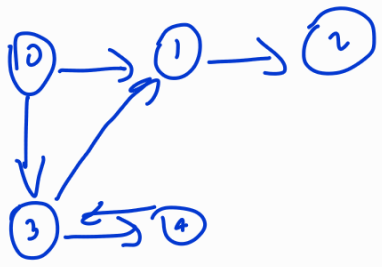



```
for(i=0;i<N;i++)
  for(j=0;j<N;j++)
    for(k=0;k<N;k++)
      C[i][j] += A[i][k] x B[k][j]
```

$\begin{pmatrix} 1\times5+2\cdot7 & 1\cdot6+2\cdot8 \\ 3\cdot5+4\cdot7 & 3\cdot6+4\cdot8 \end{pmatrix}$

$\begin{pmatrix} 1\times5+2\cdot7 & 1\times6+2\cdot8 \\ 3\times5+4\times7 & 3\times6+4\times8 \end{pmatrix}$

```
for(i=0;i<N;i++)
  for(j=0;j<N;j++)
    for(k=0;k<N;k++)
      if(A[i][k] && A[k][j])
        C[i][j] = 1;
```




A)

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 1 | 1 | | 1 | |
| 1 | | 1 | 1 | | |
| 2 | | | 1 | | |
| 3 | | 1 | | 1 | 1 |
| 4 | | | | 1 | 1 |

B)

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 1 | 1 | | 1 | |
| 1 | | 1 | 1 | | |
| 2 | | | 1 | | |
| 3 | | 1 | | 1 | 1 |
| 4 | | 1 | | 1 | 1 |

C)

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | | 1 | 1 | | 1 |
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | 1 | | | |

$i = 1$

$j = 2$

$k =$

```
for(r=0;r<V;r++)
  multiply()
```

$A^V = A \times A \times A \cdots A_V$

$V \cdot V^3$

$O(V^4)$

```
for(i=0;i<N;i++)
  for(j=0;j<N;j++)
    if(A[i][k])
      for(k=0;k<N;k++)
        if(A[i][k] && A[k][j])
          C[i][j] = 1;
```

Dfs $O(V+E)$

$V \cdot (V+E) \sim V^2$

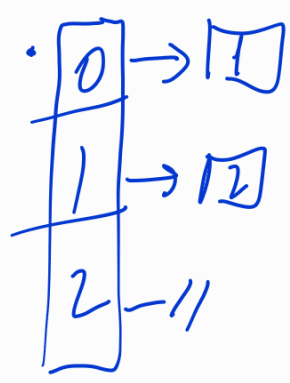
$E \to V^2 \therefore V(V+V^2) \sim V^3$

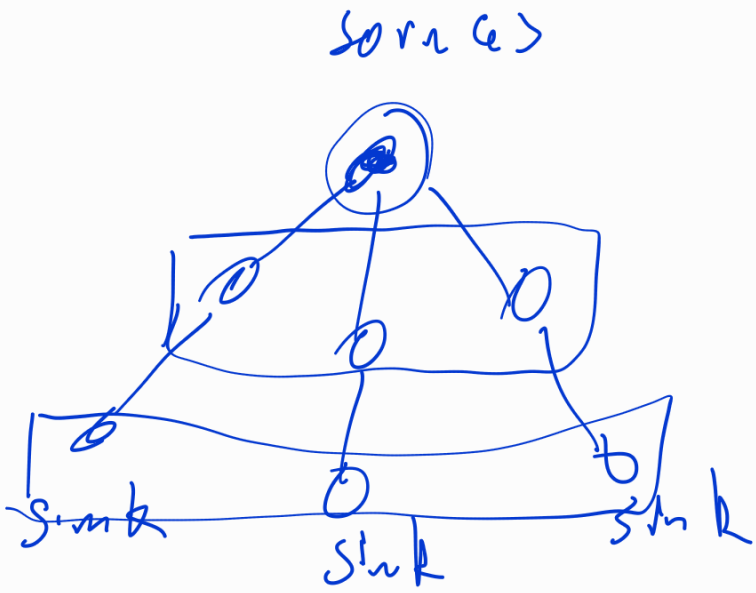
```
Void GraphTC(Graph G)        V*(V+E)
{
  G->tc = MATRIXinit(G->V, G->V, 0);

  for(int i=0; i<G->V; i++)
      dfsTCR(G, EDGE(i,i), 1)

}

Void dfsTCR(Graph G, Edge e, int depth)
{ d
  • G->tc[e.v][e.w] = 1;                    1)

  for(link l = G->adj[e.w]; l!=NULL; l=l->next)
  {
  • if(G->tc[e.v][l->v] == 0)
      • dfsTCR(G, EDGE(e.v, l->v), depth+1)
```

G->tc

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 1 | 1 |
| 1 | | 1 | 1 |
| 2 | | | 1 |

(1,1)
(1,2)
(2,2)




Source
Sink
Sink
Sink