DFS(Graph G, Vertex *root)
{
    for each (v in G) {
        Encountered(v) = false;
        Finished(v) = false;
    }
    int num = 1;
    RecursiveDFS(root, &num);
}

RecursiveDFS(Vertex *v, int *pn)
{
    Encountered(v) = true;
    for each (w in v->Neighbors())
        if (!Encountered(w))
            RecursiveDFS(w, pn);
    Finished(v) = true;
    for each (w in v->Neighbors())
        if (Encountered(w) && !Finished(w))
            have_cycle = true;
}

TopSort(Graph G)
{
    next_num = 1;
    for (each vertex v of G)
        in_deg(v) = v->InDegree();
    num(v) = -1;
    for (each vertex v of G)
        if (in_deg(v) == 0)
            Recurse(v);
}

Recurse(Vertex v)
{
    num(v) = next_num++;
    for (each nbr w of v)
        if (in_deg(v) == 0)
            Recurse(v);
TopSort(Graph G)
{
    next_num = G.NumVtcs;
    for (each vertex v of G)
        encountered(v) = false;
    num(v) = -1;
    for (each vertex v of G)
        if (!encountered(v))
            Recurse(v);
}

Recurse(Vertex v)
{
    encountered(v) = true;
    for (each nbr w of v)
        if (!encountered(w))
            Recurse(v);
    num(v) = next_num--;