Design and Analysis of Algorithms I

Graph Primitives

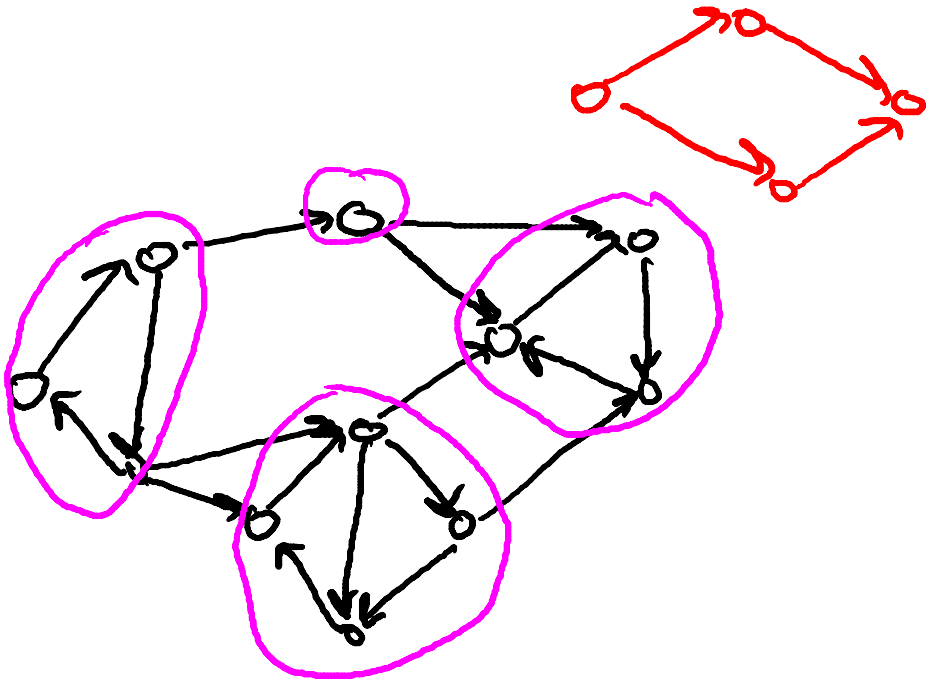
An O(m+n) Algorithm for Computing Strong Components

Strongly Connected Components

Formal Definition : the strongly connected components (SCCs) of a directed graph G are the equivalence classes of the relation

u<->v <==> there exists a path u->v and a path v->u in G

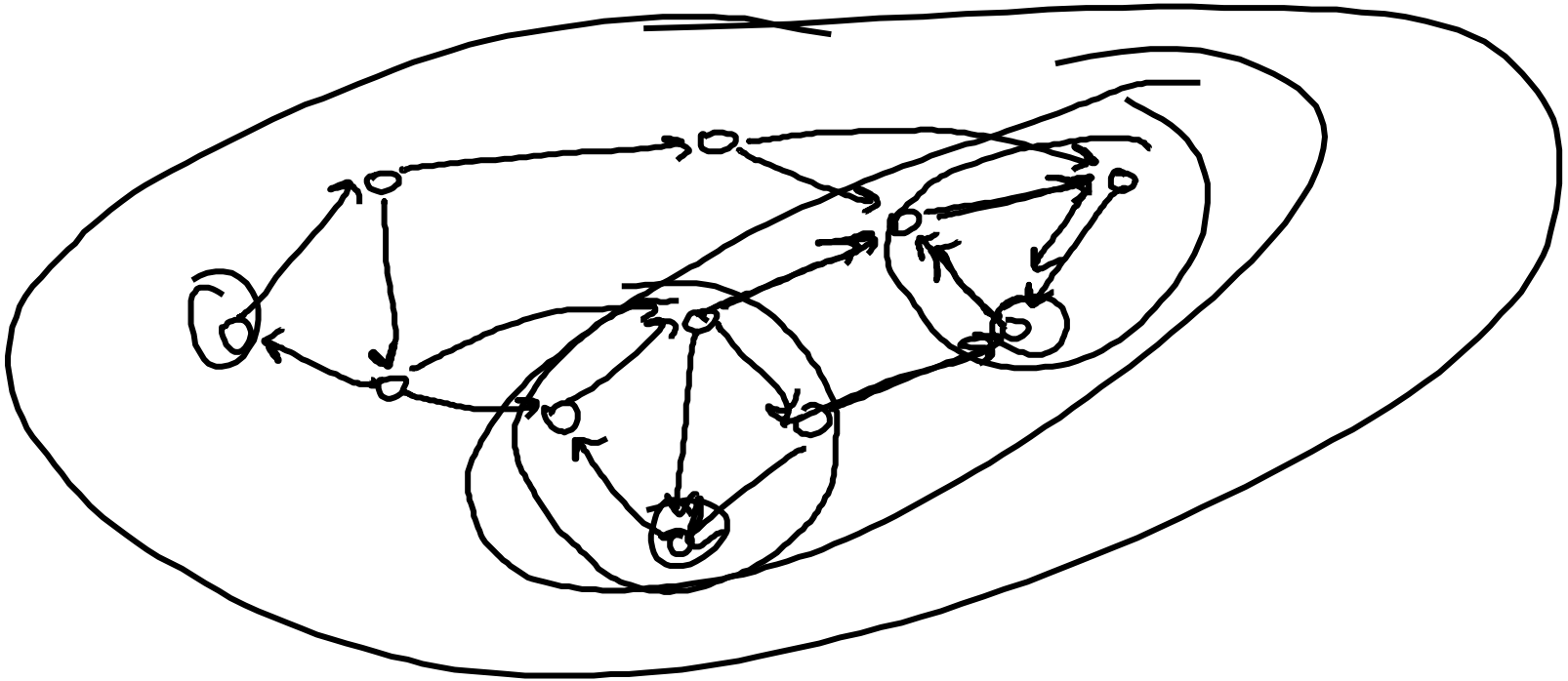
You check : <-> is an equivalence relation



小圖就和大圖很像，rightmost 都不能走到leftmost
只是大圖的四個nodes變成scc

Tim Roughgarden

Why Depth-First Search?



Tim Roughgarden

Kosaraju's Two-Pass Algorithm

Theorem : can compute SCCs in O(m+n) time.

Algorithm : (given directed graph G)

1. Let Grev = G with all arcs reversed
2. Run DFS-Loop on Grev

   Let f(v) = "finishing time" of each v in V

1. Run DFS-Loop on G

   processing nodes in decreasing order of finishing times

[SCCs = nodes with the same "leader"]

Goal : compute "magical ordering" of nodes

Goal : discover the SCCs one-by-one

t only has to show up in 1st DFS-Loop,
s is only used in 2nd DFS-Loop

DFS-Loop

DFS-Loop (graph G)

Global variable t = 0

magical order

[# of nodes processed so far]

For finishing times in 1st pass

Global variable s = NULL

For leaders in 2nd pass

[current source vertex]

Assume nodes labeled 1 to n

For i = n down to 1

    if i not yet explored

        s := i

        DFS(G,i)

DFS (graph G, node i)

i may be encountered more than once

For rest of DFS-Loop

-- mark i as explored

-- set leader(i) := node s

-- for each arc (i,j) in G :

        -- if j not yet explored

            -- DFS(G,j)

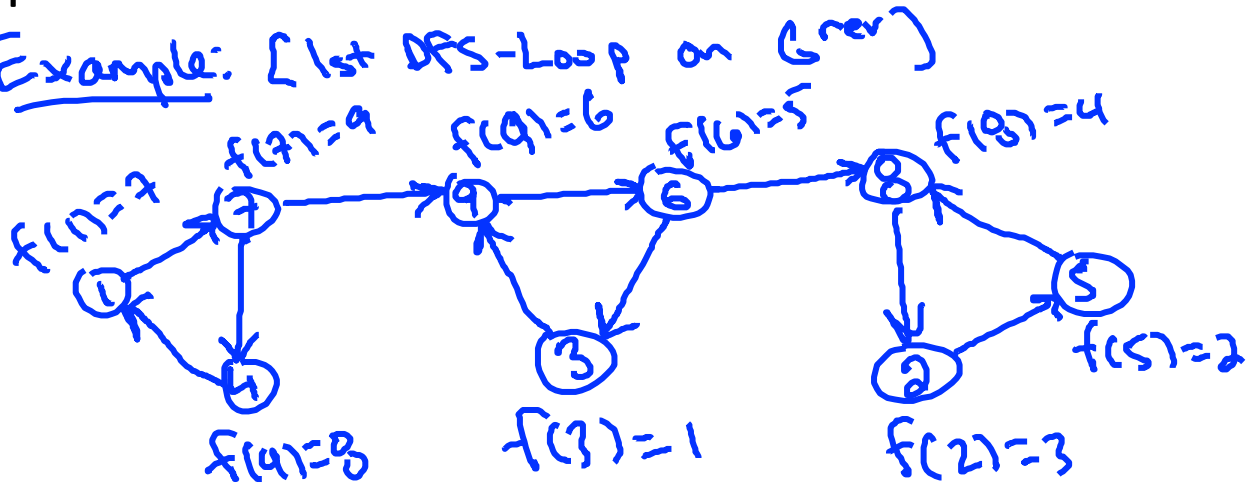
-- t++

-- set f(i) := t

i's finishing time

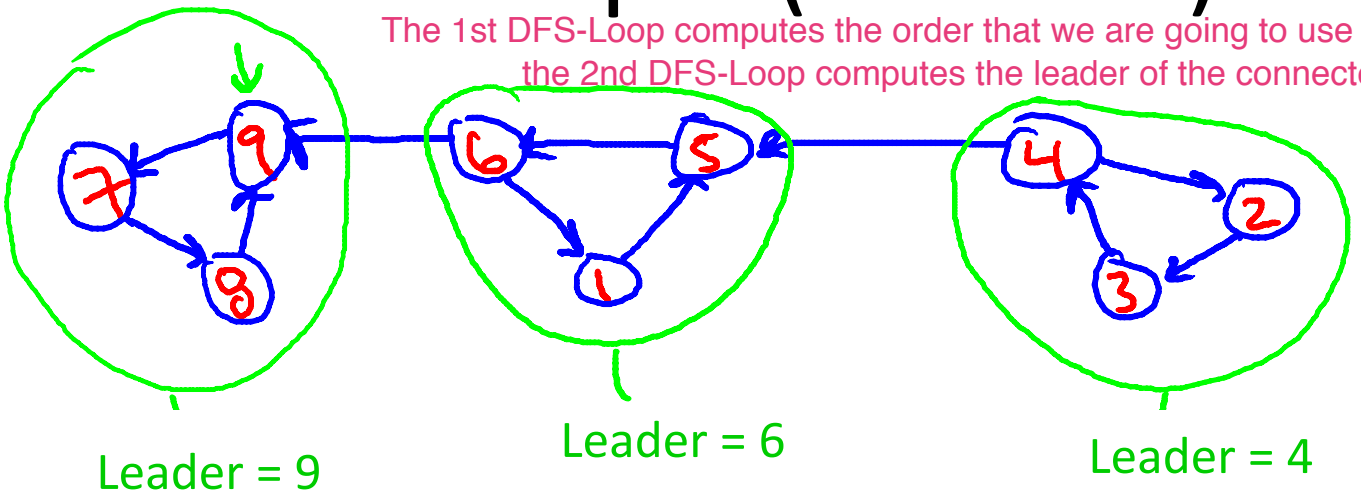
Only one of the following is a possible set of finishing times for the nodes 1,2,3,…,9, respectively, when the DFS-Loop subroutine is executed on the graph below. Which is it?

- 9,8,7,6,5,4,3,2,1
- 1,7,4,9,6,3,8,2,5
- 1,7,9,6,8,2,5,3,4
- 7,3,1,8,2,5,9,4,6



Example (2nd Pass)

The 1st DFS-Loop computes the order that we are going to use in 2nd DFS-Loop; the 2nd DFS-Loop computes the leader of the connected graph



Running Time : 2*DFS = O(m+n)