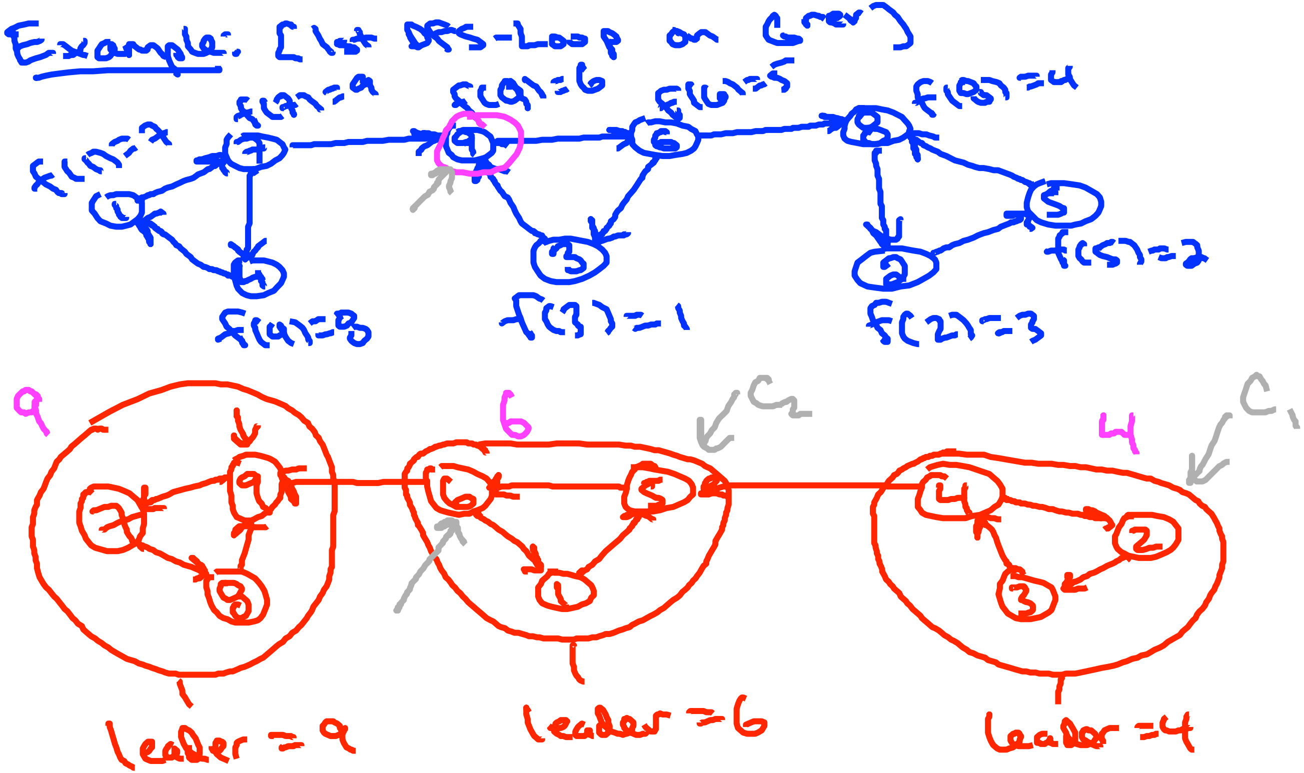
Design and Analysis of Algorithms I

Graph Primitives

Correctness of Kosaraju's Algorithm

Example Recap



Observation

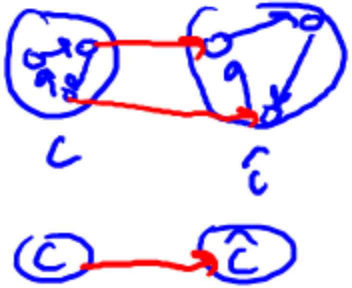
Claim : the SCCs of a directed graph G induce an acyclic "meta-graph":

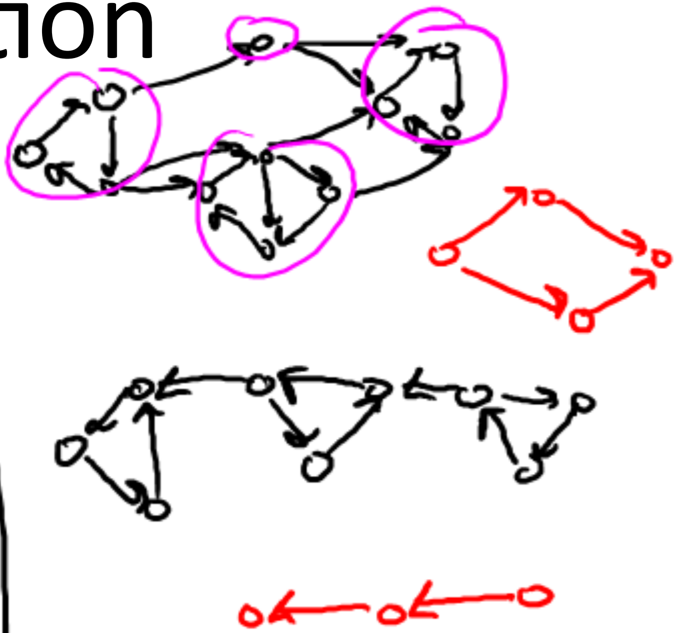
-- meta-nodes = the SCCs $C_1,..,C_k$ of G

-- $\exists\ arc\ C \rightarrow \hat{C} <=> \exists\ arc\ \square\ (i,j) \in G$

$with\ i \in C,\ \ j \in \hat{C}$

Why acyclic ? : a cycle of SCCs would collapse into one.



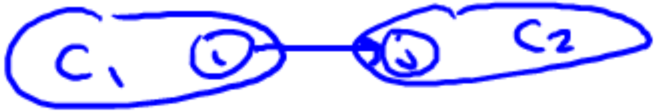


What how are the SCC of the original graph G and its reversal $G\uparrow rev$ related?

- ○ In general, they are unrelated.
- ○ Every SCC of G is contained in an SCC of $G\uparrow rev$, but the converse need not hold.
- ○ Every *SCC of* $G\uparrow rev$ *is contained in an SCC of* G*, but the converse need not hold.*
- ○ [✓ They are exactly the same.]

Key Lemma

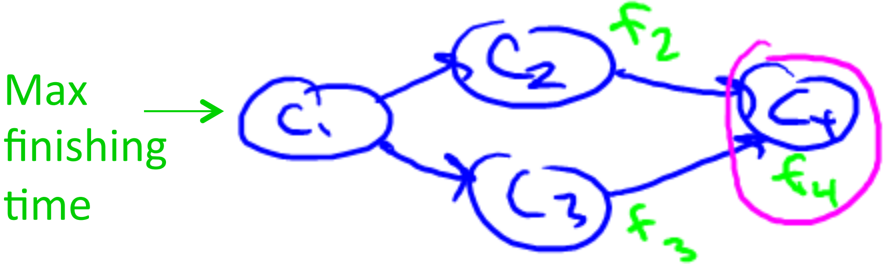
Lemma : consider two "adjacent" SCCs in G:



Let f(v) = finishing times of DFS-Loop in Grev

Then : $max_{v \in C_1} f(v) < max_{v \in C_2} f(v)$

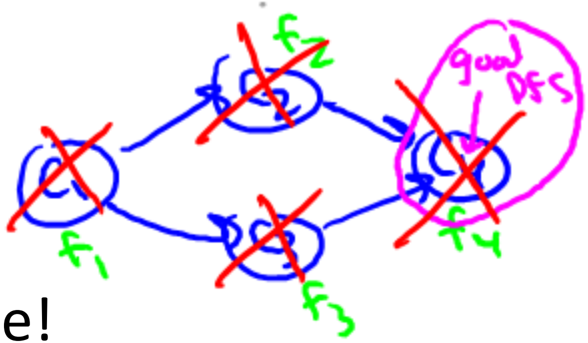
Corollary : maximum f-value of G must lie in a "sink SCC"



$f_1 < f_2, f_3 < f_4$

Correctness Intuition

(see notes for formal proof)



By Corollary : 2nd pass of DFS-Loop begins somewhere in a sink SCC C*.

⟹First call to DFS discovers C* and nothing else!

⟹Rest of DFS-Loop like recursing on G with C* deleted

[starts in a sink node of G-C*]

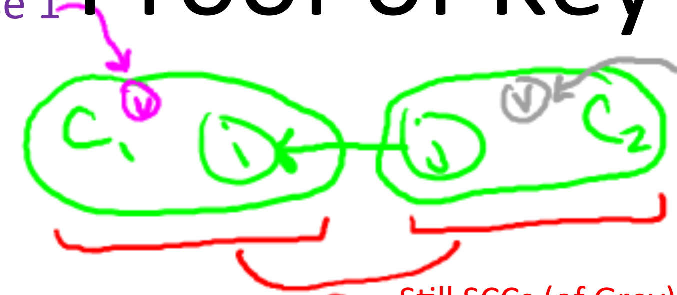
⟹ successive calls to DFS(G,i) "peel off" the SCCs one by one

[in reverse topological order of the "meta-graph" of SCCs]

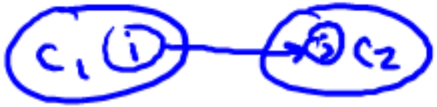
Proof of Key Lemma

<u>In Grev :</u>

Case 1



Case 1



$max_{v \in C_1} f(v) < max_{v \in C_2} f(v)$

Still SCCs (of Grev)
[by Quiz]

Let v = 1st node of $C_1 \cup C_2$
reached by 1st pass of DFS-Loop (on Grev)

Case 1 [$v \in C_1$] : all of C_1 explored before C_2 ever reached.

Reason : no paths from C_1 to C_2 (since meta-graph is acyclic)
⇒All f-values in C_1 less than all f-values in C_2

case 2 is when DFS 'usefulness' plays real part.
Else we can swap BFS in multiple places in the algo

Case 2 [$v \in C_2$] : DFS(Grev, v) won't finish until all of $C_1 \cup C_2$
completely explored => f(v) > f(w) for all w in C_1

Tim Roughgarden