F17

① Strongly Connected Components

15-210
Oct 23, 2017

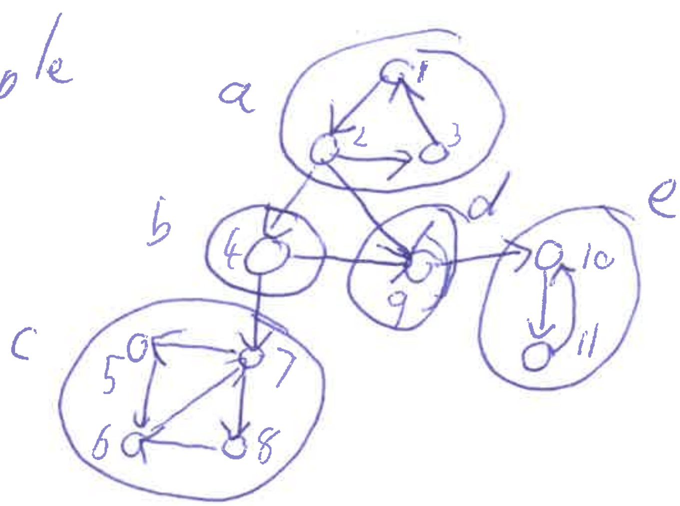
## Definitions

1) A set of vertices $U \subset V$ is <u>strongly connected</u> (SC) in a directed graph $G=(V,A)$ if every vertex in $U$ can reach every other vertex in $U$

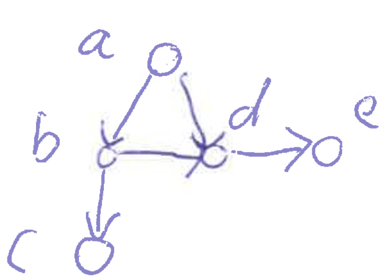
2) A SC set is <u>maximal</u> if we cannot add any more vertices and it remains SC

3) The <u>strongly connected components</u> of a directed graph are the maximal SC sets.

Example




Note: if we contract each SC into a vertex we are left with a DAG




← DAG

② Definition: The <u>SCC problem</u> is to find the SCCs and return them in topological order.

e.g. $\langle\langle 1,2,3\rangle, \langle 4\rangle, \langle 9\rangle, \langle 10,11\rangle, \langle 5,6,7,8\rangle\rangle$

Lemma I: For a directed graph $G$, if $u$ is the first vertex visited by DFS in a component then ~~for~~ all $v$ reachable from $u$ $F_u > F_v$ (recall $F_u$ is the finish time)

Proof: Cases

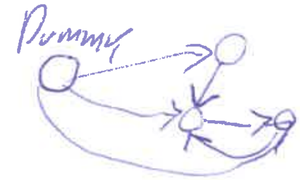
$u$ visited before $v$: In this case $u$ is an anscestor of $v$, again 2 cases

- if in same (SCC ~~component~~), since $u$ is visited first it will visit everything before finishing
- if in different SCC then none between $u$ and $v$ so will fully visit (discover ~~+ finish~~) $v$ before finishing $u$

$v$ visited before $u$: must be in a different component since $u$ is first in its component, so $u$ not reachable from $v$. $u$ will therefore be fully visited (discover + finish) before $v$ is discovered hence $F_v < S_u < F_u$

③ Algorithm for SCC

0) create a dummy vertex with arcs to all other vertices
(does not affect SCCs)




1) Order vertices by reverse finish times. Call this F

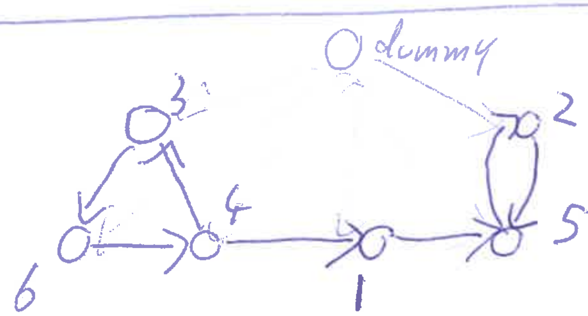
2) Let $G^T$ ← transpose be G with all edges reversed

3) ~~In order of step 1~~ For each v in F in order
- output unvisited vertices reachable from v, and mark all as visited

---

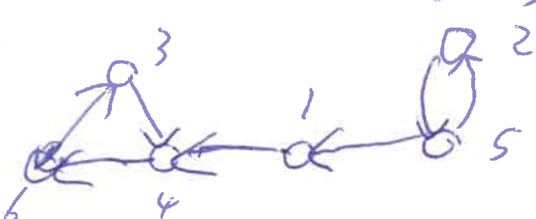
e.g

G




Reverse Finish times: 3, 6, 4, 1, 5, 2

$G^T =$




Reach(3) → ⟨3, 4, 6⟩
reach(6) → ⟨⟩ (3,4,6 already visited
reach(4) → ⟨⟩
reach(1) → ⟨1⟩
reach(5) → ⟨5, 2⟩
reach(2) → ⟨⟩

throw out empty sets

Final Output
⟨⟨3, 4, 6⟩, ⟨1⟩, ⟨5, 2⟩⟩

## Why it works (Proof)

1) after step I the first vertex in each component in F will appear before all vertices it can reach (By Lemma I)

2) In step 3, when visiting the first ~~component~~ vertex in component X:

   1) all components that can reach X have been ~~visited~~ fully visited (because of property I) and ~~because~~ since edges reversed they have been fully visited)

   2) all components that can be reached from X are "invisible" since the edges are backwards.

   3) We will fully visit X since all reachable

Hence we will visit exactly all of X, no more, no less

3) Components are visited in topological order since first vertices of each in F are in topological order

Q. E. D.

Code

BCC$((G = (V, A))$ = let

(0) $G'$ = add dummy node $S$

(1) $DFS_F((X, F), v)$ =

if $(v \in X)$ then $(X, F)$

else let $(X', F')$ = iterate $DFS_F$ $(X \cup \{v\}, F)$ $(N^+_G(v))$

in $(X', cons(v, F))$ end

appending in reverse finish times

(2) $G^T$ = transpose $G$

(3) ~~$DFS_X((X, L), v)$ = let $(X', F) = DFS_F(X', [], v)$ in if $|F| = 0$ then $(X', L)$ else $(X', cons(F, L))$~~

Reachability$((X, L), v)$ =

if $(v \in X)$ then $(X, L)$

else iterate Reachability $(X \cup \{v\}, L \cup \{v\})$ $(N^+_{G^T}(v))$

Helper$((X, L), v)$ =

let $(X', A)$ = reachability $((X, []), v)$

in if $(|A| = 0)$ then $(X', L)$ else $(X', cons(F, L))$ end

$F = DFS_F((\{\}, []), s)$

in

~~Helper$(\{\}, \langle\rangle)$~~

iterate Helper $(\{\}, \langle\rangle)$ $F$

end