

Announcement

- **Quiz 6** , Tuesday, Nov 24 - BFS, DFS

Graphs: Strong Connectivity

- **Strong Connectivity** A digraph $G = (V, E)$ is *strongly connected* if every two vertices u and v are reachable from each other, i.e. there is a path from u to v and a path from v to u .
- **Strongly connected vertices u and v** Two vertices u and v are strongly connected if they are reachable from each other.
- **A Strongly Connected Component (SCC)** of a digraph $G = (V, E)$ is a maximal subset of strongly connected vertices (Maximal in the sense that, if you add one more vertex from V to the SCC, the new larger set is not strongly connected anymore).
- **Strongly connected components of a graph**
If a digraph is not strongly connected, it can be decomposed into disjoint strongly connected components.

■ Example

A

B

C

D

E

F

G

H

■ SCC Problem

- *Input*: digraph $G = (V, E)$
- *Output*: lists the vertices of the different SCCs
- The algorithm uses 2 DFS searches, one of G and one of the transpose $G^T = (V, E^T)$ of G .

G^T has the same set of vertices as G , but the direction of the edges is reversed, i.e., $(u, v) \in E$ iff $(v, u) \in E^T$.

The set of vertices of the depth-first trees resulting from the $DFS(G^T)$ are the SCCs (one component from each tree).

- *SCC Algorithm*:
 1. Call $DFS(G)$ to get finishing times $f[u], u \in V$
 2. Construct the transpose graph G^T
 3. Call $DFS(G^T)$ with the condition that: new roots of DFS trees are selected in decreasing order of their finishing times, and that the vertices adjacent to each vertex are considered by DFS in decreasing order of their finishing times.
 4. Output the set of vertices of each DFS tree as separate SCCs

■ **Correctness** Why are the sets of vertices of DFS trees of G^T produced by the algorithm the SCC of G ? Notice that:

- **If u and v are in the same SCC in a graph, they are in the same DFS tree**

Let assume r_u and r_v are roots of *two different DFS trees containing u and v* .

If $f[r_u] < f[r_v]$, while DFS tree rooted in r_u is being explored, all that is in DFS tree rooted in r_v stays white, in particular,

$$f[u] < f[r_u] < f[v] < f[r_v]$$

But this is impossible, since, after u is discovered, DFS would have explored deep all reachable from u , and v is reachable from u , thus $f[v] < f[u]$.

Similarly we obtain a contradiction if $f[r_u] > f[r_v]$.

Thus u and v are in the same DFS tree.

- **If u and v are in the same SCC of G , they are in the same SCC of G^T .**

This is because there is a path which goes from u to v , and another from v to u in G , in G^T this paths will be “reversed”, and u and v strongly connected in G^T .

- If u and v are in the same SSC in G , then they are in the same DFS tree of G^T (this follows directly from the previous two observations).

A: Thus, each SSC of G is a subset of a DFS tree for G^T .

- Reverse: *Can we show that the set of vertices of a DFS tree of G^T is a subset of the same SSC of G , i.e. that if u and v are in the same DFS tree of G^T , they are in the same SSC of G ?*

Let r be the root in the common DFS tree (of G^T) contains u and v .

Thus u and v are both descendants of r in that DFS tree for G^T .

Thus *there are paths in G from u to r and from v to r .*

Since $f[r] > f[u]$ and $f[r] > f[v]$ (because of the way we run the DFS on G^T), and since r is reachable from v and u , the only way v and u could have been finished before r in DFS of G , is if u and v were descendants of r in G .

Thus *there are paths from r to u and to v , in G .*

Thus: u and v are strongly connected in G .

Thus:

B: The set of vertices of a DFS tree of G^T is a subset of the same SSC of G .

Recall:

A: Each SSC of G is a subset of a DFS tree for G^T .

- Since the strongly connected components are disjoint, and the DFS trees produced by DFS are disjoint, from **A** and **B** it follows that:

The sets of vertices of the different DFS trees of G^T produced by the algorithm are the different strongly connected components of G .

■ **Time complexity** $O(n + m)$

Constructing G^T is $O(n + m)$ and so are the two DFS'es.

■ Example