CS473-Algorithm I

Lecture 16

Strongly Connected Components
Strongly Connected Components

Definition: a strongly connected component (SCC) of a directed graph $G= (V,E)$ is a maximal set of vertices $U \subseteq V$ such that

- For each $u,v \in U$ we have both $u \rightarrow v$ and $v \rightarrow u$
  i.e., $u$ and $v$ are mutually reachable from each other ($u \xrightarrow{\mathcal{G}} v$)

Let $G^T= (V,E^T)$ be the transpose of $G= (V,E)$ where

$$E^T = \{(u,v): (v,u) \in E\}$$

- i.e., $E^T$ consists of edges of $G$ with their directions reversed

Constructing $G^T$ from $G$ takes $O(V+E)$ time (adjacency list rep)

Note: $G$ and $G^T$ have the same SCCs ($u \xrightarrow{\mathcal{G}} v$ in $G \iff u \xrightarrow{\mathcal{G}} v$ in $G^T$)
Strongly Connected Components

Algorithm

(1) Run \textbf{DFS}(G) to compute finishing times for all \( u \in V \)

(2) Compute \( G^T \)

(3) Call \textbf{DFS}(G^T) processing vertices in main loop in decreasing \( f[u] \) computed in Step (1)

(4) Output vertices of each \textbf{DFT} in \textbf{DFF} of Step (3) as a separate \textbf{SCC}
**Strongly Connected Components**

**Lemma 1**: no path between a pair of vertices in the same SCC, ever leaves the SCC

**Proof**: let $u$ and $v$ be in the same SCC $\Rightarrow u \leftrightarrow v$

let $w$ be on some path $u \rightarrow w \rightarrow v \Rightarrow u \rightarrow w$

but $v \rightarrow u \Rightarrow \exists$ a path $w \rightarrow v \rightarrow u \Rightarrow w \rightarrow u$

therefore $u$ and $w$ are in the same SCC

QED
SCC: Example
SCC: Example

(1) Run $\text{DFS}(G)$ to compute finishing times for all $u \in V$
SCC: Example

(1) Run $\text{DFS}(G)$ to compute finishing times for all $u \in V$
(1) Run $\text{DFS}(G)$ to compute finishing times for all $u \in V$. 

SCC: Example
SCC: Example

Vertices sorted according to the finishing times:

\[ \langle b, e, a, c, d, g, h, f \rangle \]
SCC: Example

(2) Compute $G^T$
SCC: Example

(3) Call $\text{DFS}(G^T)$ processing vertices in main loop in decreasing $f[u]$ order: $\langle b, e, a, c, d, g, h, f \rangle$
(3) Call \textbf{DFS}(G^T) processing vertices in main loop in decreasing $f[u]$ order: $\langle b, e, a, c, d, g, h, f \rangle$
(3) Call $\text{DFS}(G^T)$ processing vertices in main loop in decreasing $f[u]$ order: $\langle b, e, a, c, d, g, h, f \rangle$
(3) Call $\text{DFS}(G^T)$ processing vertices in main loop in decreasing $f[u]$ order: $\langle b, e, a, c, d, g, h, f \rangle$
SCC: Example

(3) Call $\text{DFS}(G^T)$ processing vertices in main loop in decreasing $f[u]$ order: $\langle b, e, a, c, d, g, h, f \rangle$
(3) Call $\text{DFS}(G^T)$ processing vertices in main loop in decreasing $f[u]$ order: $\langle b, e, a, c, d, g, h, f \rangle$
SCC: Example

(3) Call $\text{DFS}(G^T)$ processing vertices in main loop in decreasing $f[u]$ order: $(b, e, a, c, d, g, h, f)$
(3) Call $\text{DFS}(G^T)$ processing vertices in main loop in decreasing $f[u]$ order: $\langle b, e, a, c, d, g, h, f \rangle$
SCC: Example

(4) Output vertices of each DFT in DFF as a separate SCC

\[ C_b = \{b, a, e\} \]
\[ C_g = \{g, f\} \]
\[ C_h = \{h\} \]

\[ C_c = \{c, d\} \]
SCC: Example

Acyclic component graph

$C_b$

$C_g$

$C_h$

$C_c$
Strongly Connected Components

Thrm 1: in any DFS, all vertices in the same SCC are placed in the same DFT

Proof: let \( r \) be the first vertex discovered in SCC \( S_r \), because \( r \) is first, \( \text{color}[x] = \text{WHITE} \ \forall x \in S_r \setminus \{r\} \) at time \( d[r] \)

So all vertices are \text{WHITE} on each \( r \rightarrow x \) path \( \forall x \in S_r \setminus \{r\} \)

– since these paths never leave \( S_r \)

Hence each vertex in \( S_r \setminus \{r\} \) becomes a descendent of \( r \)
(White-path Thrm)

QED
Notation for the Rest of This Lecture

- $d[u]$ and $f[u]$ refer to those values computed by $\text{DFS}(G)$ at step (1)
- $u \rightarrow v$ refers to $G$ not $G^T$

**Definition:** forefather $\phi(u)$ of vertex $u$

1. $\phi(u) = \text{That vertex } w \text{ such that } u \rightarrow w \text{ and } f[u] \text{ is maximized}$
2. $\phi(u) = u$ possible because $u \rightarrow u \implies f[u] \leq f[\phi(u)]$
Lemma 2: $\phi(\phi(u)) = \phi(u)$

Proof try to show that $f[\phi(\phi(u))] = f[\phi(u)]$:

For any $u, v \in V; \ u \mapsto v \Rightarrow R_v \subseteq R_u \Rightarrow f[\phi(v)] \leq f[\phi(u)]$

So, $u \mapsto \phi(u) \Rightarrow f[\phi(\phi(u))] \leq f[\phi(u)]$

Due to definition of $\phi(u)$ we have $f[\phi(\phi(u))] \geq f[\phi(u)]$

Therefore $f[\phi(\phi(u))] = f[\phi(u)]$

QED

Note:

$f[x] = f[y] \Rightarrow x = y$

(same vertex)
Strongly Connected Components

Properties of forefather:

- Every vertex in an SCC has the same forefather which is in the SCC
- Forefather of an SCC is the representative vertex of the SCC
- In the DFS of $G$, forefather of an SCC is the
  - first vertex discovered in the SCC
  - last vertex finished in the SCC
Strongly Connected Components

THM2: $\phi(u)$ of any $u \in V$ in any DFS of $G$ is an ancestor of $u$

PROOF: Trivial if $\phi(u) = u$.

If $\phi(u) \neq u$, consider color of $\phi(u)$ at time $d[u]$

- $\phi(u)$ is GRAY: $\phi(u)$ is an ancestor of $u$ \(\Rightarrow\) proving the theorem
- $\phi(u)$ is BLACK: $f[\phi(u)] < f[u] \Rightarrow$ contradiction to def. of $\phi(u)$
- $\phi(u)$ is WHITE: $\exists$ 2 cases according to colors of intermediate vertices on $p(u, \phi(u))$

Path $p(u, \phi(u))$ at time $d[u]$: 

```
G  ?  ?  ?  \phi(u)
  \uparrow   \uparrow   \uparrow
u  x_1  x_2  \cdots  \cdots  x_r  w
```
Strongly Connected Components

Case 1: every intermediate vertex $x_i \in p(u, \phi(u))$ is WHITE

\[ \Rightarrow \phi(u) \text{ becomes a descendant of } u \text{ (WP-THM)} \]
\[ \Rightarrow f[\phi(u)] < f[u] \]
\[ \Rightarrow \text{contradiction} \]

Case 2: $\exists$ some non-WHITE intermediate vertices on $p(u, \phi(u))$

- Let $x_t$ be the last non-WHITE vertex on $p(u, \phi(u)) = \langle u, x_1, x_2, \ldots, x_r, \phi(u) \rangle$
- Then, $x_t$ must be GRAY since BLACK-to-WHITE edge ($x_t, x_{t+1}$) cannot exist
- But then, $p(x_t, \phi(u)) = \langle x_{t+1}, x_{t+2}, \ldots, x_r, \phi(u) \rangle$ is a white path
  \[ \Rightarrow \phi(u) \text{ is a descendant of } x_t \text{ (by white-path theorem)} \]
  \[ \Rightarrow f[x_t] > f[\phi(u)] \]
  \[ \Rightarrow \text{contradicting our choice for } \phi(u) \text{ Q.E.D.} \]
Strongly Connected Components

C1: in any DFS of $G = (V, E)$ vertices $u$ and $\phi(u)$ lie in the same SCC, $\forall u \in V$

PROOF: $u \leftrightarrow \phi(u)$ (by definition) and $\phi(u) \mapsto u$ since $\phi(u)$ is an ancestor of $u$ (by THM2)

THM3: two vertices $u,v \in V$ lie in the same SCC $\leftrightarrow \phi(u) = \phi(v)$ in a DFS of $G = (V, E)$

PROOF: let $u$ and $v$ be in the same SCC $C_{uv} \Rightarrow u \leftrightarrow v$
Strongly Connected Components

\[ \forall w: v \mapsto w \Rightarrow u \mapsto w \text{ and } \forall w: u \mapsto w \Rightarrow v \mapsto w, \text{ i.e., every vertex reachable from } u \text{ is reachable from } v \text{ and vice-versa} \]

So, \( w = \phi(u) \Rightarrow w = \phi(v) \) and \( w = \phi(v) \Rightarrow w = \phi(u) \) by definition of forefather

**PROOF:** Let \( \phi(u) = \phi(v) = w \in C_w \Rightarrow u \in C_w \text{ by C1 and } v \in C_w \text{ by C1} \)

By THM3: SCCs are sets of vertices with the same forefather

By THM2 and parenthesis THM: A forefather is the first vertex discovered and the last vertex finished in its SCC
SCC: Why do we Run DFS on GT?

Consider $r \in V$ with largest finishing time computed by DFS on $G$. $r$ must be a forefather by definition since $r \rightarrow r$ and $f[r]$ is maximum in $V$.

$C_r = \{u \in V: u \rightarrow r \}$ where $R_u = \{v \in V: u \rightarrow v\}$.

$C_r = \{u \in V: u \rightarrow r\}$ since $f[r]$ is maximum.

$C_r = R_r^T = \{u \in V: r \rightarrow u \text{ in } G^T\} = \text{reachability set of } r \text{ in } G^T$. i.e., $C_r = \text{those vertices reachable from } r \text{ in } G^T$. Thus $\text{DFS-VISIT}(G^T, r)$ identifies all vertices in $C_r$ and blackens them.
SCC: Why do we Run DFS on GT?

BFS($G^T$, $r$) can also be used to identify $C_r$

Then, DFS on $G^T$ continues with \texttt{DFS-VISIT}($G^T$, $r'$) where $f[r'] > f[w] \forall w \in V - C_r$

$r$ must be a forefather by definition since $r' \mapsto r'$ and $f[r']$ is maximum in $V - C_r$
SCC: Why do we Run DFS on GT?

Hence by similar reasoning $\text{DFS-VISIT}(G^T, r')$ identifies $C_{r'}$.

Impossible since otherwise $r', w \in C_r \Rightarrow r', w$ would have been blackened.

Thus, each $\text{DFS-VISIT}(G^T, x)$ in $\text{DFS}(G^T)$ identifies an SCC $C_x$ with $\phi = x$. 