DFS and the White Path Theorem

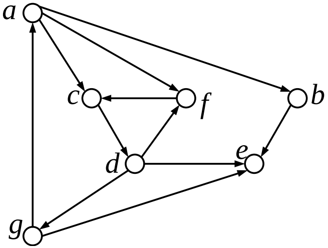
CSCI 2100 Teaching Team

Department of Computer Science and Engineering
Chinese University of Hong Kong

In this tutorial, we will first review DFS and then prove the white path theorem.

Let us first go over the DFS algorithm through an example.

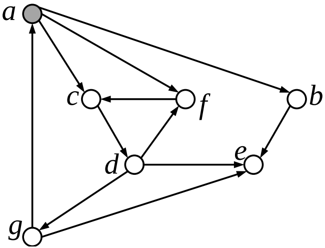
Input



Suppose we start from the vertex a, namely a is the root of DFS tree.

DFS

First, color all the vertices white. Then, create a **stack** S, push the starting vertex a into S and color it gray. Create a DFS tree with a as the root and set its time interval to $I(a) = [1, -]$.

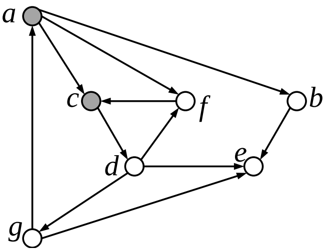


DFS Tree	Time Interval
a	$I(a) = [1, \ ]$

$S = (a)$.

DFS

Top of stack: a, which has white out-neighbors b, c, f. Suppose we access c first. Push c into S.



DFS Tree

a
|
c

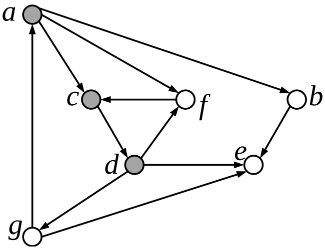
Time Interval

$I(a) = [1, \ ]$
$I(c) = [2, \ ]$

$S = (a, c)$.

DFS

After pushing d into S:

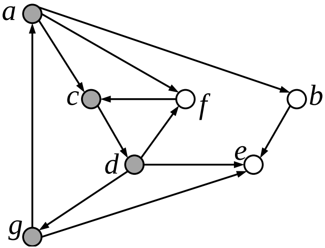


DFS Tree	Time Interval
a	$I(a) = [1, \ ]$
c	$I(c) = [2, \ ]$
d	$I(d) = [3, \ ]$

$S = (a, c, d)$.

DFS

Now d tops the stack. It has white out-neighbors e, f and g. Suppose we visit g first. Push g into S.

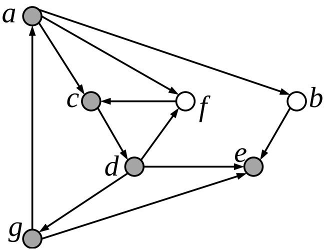


DFS Tree	Time Interval
a	$I(a) = [1,\ ]$
c	$I(c) = [2,\ ]$
d	$I(d) = [3,\ ]$
g	$I(g) = [4,\ ]$

$S = (a, c, d, g)$.

DFS

After pushing e into S:

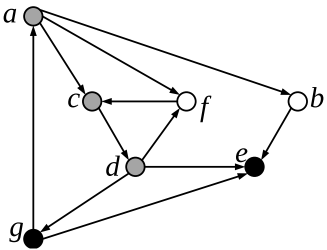


DFS Tree	Time Interval
a	$I(a) = [1,\ ]$
c	$I(c) = [2,\ ]$
d	$I(d) = [3,\ ]$
g	$I(g) = [4,\ ]$
e	$I(e) = [5,\ ]$

$S = (a, c, d, g, e)$.

DFS

e has no white out-neighbors. So pop it from S and color it black. Similarly, g has no white out-neighbors. Pop it from S and color it black.

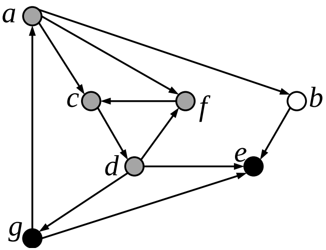


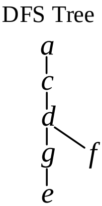
DFS Tree	Time Interval
a	$I(a) = [1,\ ]$
c	$I(c) = [2,\ ]$
d	$I(d) = [3,\ ]$
g	$I(g) = [4, 7]$
e	$I(e) = [5, 6]$

$S = (a, c, d)$.

DFS

Now d tops the stack again. It still has a white out-neighbor f. So, push f into S.





Time Interval

$I(a) = [1,\ ]$

$I(c) = [2,\ ]$

$I(d) = [3,\ ]$

$I(g) = [4, 7]$

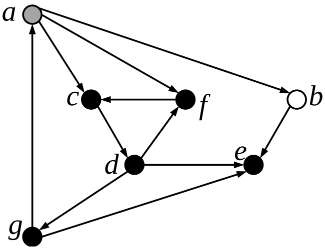
$I(e) = [5, 6]$

$I(f) = [8,\ ]$

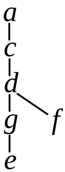
$S = (a, c, d, f)$.

DFS

After popping f, d, c:



DFS Tree



Time Interval

$I(a) = [1, \ ]$

$I(c) = [2, 11]$

$I(d) = [3, 10]$

$I(g) = [4, 7]$

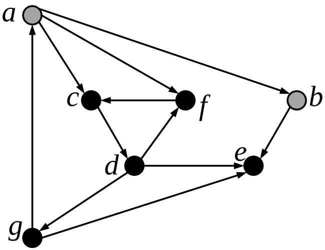
$I(e) = [5, 6]$

$I(f) = [8, 9]$

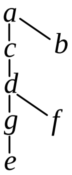
$S = (a)$.

DFS

Now a tops the stack again. It still has a white out-neighbor b. So, push b into S.



DFS Tree



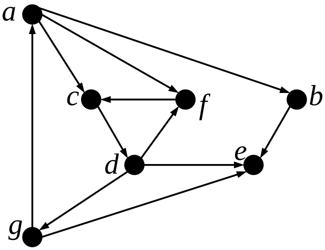
Time Interval

$I(a) = [1, \ ]$
$I(c) = [2, 11]$
$I(d) = [3, 10]$
$I(g) = [4, 7]$
$I(e) = [5, 6]$
$I(f) = [8, 9]$
$I(b) = [12, \ ]$

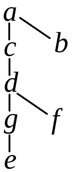
$S = (a, b)$.

DFS

After popping b and a:



DFS Tree



Time Interval

$I(a) = [1, 14]$
$I(c) = [2, 11]$
$I(d) = [3, 10]$
$I(g) = [4, 7]$
$I(e) = [5, 6]$
$I(f) = [8, 9]$
$I(b) = [12, 13]$

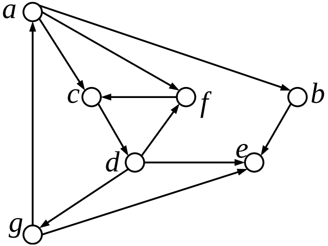
$S = ()$.

There are no more white vertices. The algorithm terminates.

Cycle Detection

Problem Input:

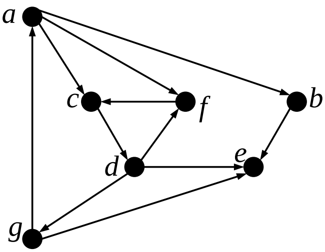
A directed graph.



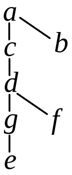
Problem Output:

A boolean value indicating whether the graph contains a cycle.

First Step: DFS



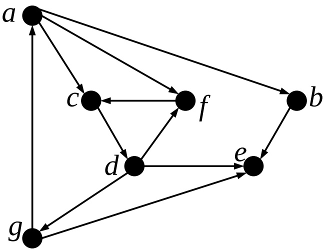
DFS Tree



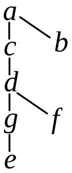
Time Interval

$I(a) = [1, 14]$
$I(c) = [2, 11]$
$I(d) = [3, 10]$
$I(g) = [4, 7]$
$I(e) = [5, 6]$
$I(f) = [8, 9]$
$I(b) = [12, 13]$

Second Step: Find Back Edges



DFS Tree

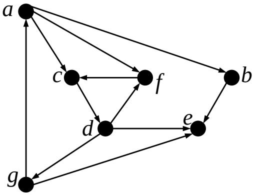


Time Interval

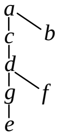
$I(a) = [1, 14]$
$I(c) = [2, 11]$
$I(d) = [3, 10]$
$I(g) = [4, 7]$
$I(e) = [5, 6]$
$I(f) = [8, 9]$
$I(b) = [12, 13]$

Cycle Theorem: Let T be an **arbitrary** DFS-forest. G contains a cycle **if and only if** there is a back edge with respect to T.

Second Step: Find Back Edges



DFS Tree



Time Interval

$I(a) = [1, 14]$
$I(c) = [2, 11]$
$I(d) = [3, 10]$
$I(g) = [4, 7]$
$I(e) = [5, 6]$
$I(f) = [8, 9]$
$I(b) = [12, 13]$

Theorem (the Parenthesis Theorem): Let u and v be two distinct vertices in G. Then:

- $I(u)$ contains $I(v)$ **if and only if** u is an ancestor of v in the DFS-forest.
- $I(v)$ contains $I(u)$ **if and only if** v is an ancestor of u in the DFS-forest.
- $I(u)$ and $I(v)$ are disjoint **if and only if** neither u nor v is an ancestor of the other.

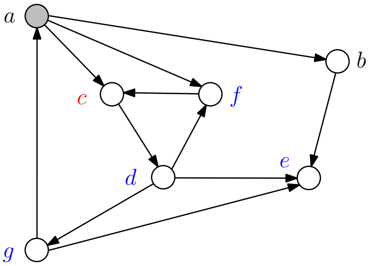
Recall that our proof of the cycle theorem (presented in the lecture) relied on the **white path theorem**. We will prove the latter theorem in the rest of the tutorial.

Theorem: Let u be a vertex in G. Consider the moment right before u enters the stack in the DFS algorithm. Then, a vertex v becomes a proper descendant of u in the DFS-forest **if and only** if the following is true at this moment:

- there is a path from u to v including only white vertices.

Example

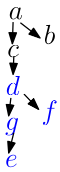
When c is pushed into the stack, we have:



DFS Tree

a

Final DFS Tree

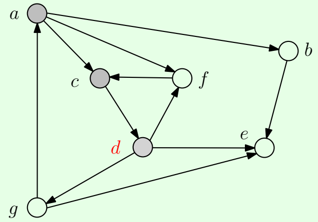


$$S = \boxed{a\quad\quad\quad} \leftarrow c$$

Recall:

Lemma (the Ancestor-Descendent Lemma): Let u and v be two distinct vertices in G. Then, u is an ancestor of v in the DFS-forest **if and only if** the following holds: u is already in the stack when v enters the stack.

Example: When d enters the stack, a and c are in the stack. d is a descendant of a and c in the DFS-tree.



DFS Tree

$$a \to c \to d$$

Time Interval

$I(a) = [1, \ ]$

$I(c) = [2, \ ]$

$I(d) = [3, \ ]$

$S = \boxed{a \ c \ d}$

Proof of White Path Theorem

Theorem: Let u be a vertex in G. Consider the moment right before u enters the stack in the DFS algorithm. Then, a vertex v becomes a proper descendant of u in the DFS-forest **if and only** if the following is true at this moment:

- there is a path from u to v including only white vertices.

Proof: The "only-if direction" ($\Rightarrow$): Let

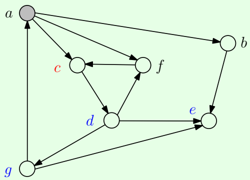
- v be a descendant of u in the DFS tree;
- π be the path from u to v in the tree.

Clearly, π is also a path from u to v in G.

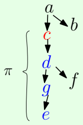
Proof of White Path Theorem

All the nodes on π except for u are proper descendants of u. By the ancestor-descendant lemma, those nodes must enter the stack after u. Hence, π must be white at the moment right before u enters the stack.

Example: $u = c$, $v = e$, $\pi = (c, d, g, e)$.



Final DFS Tree



Final Time Interval

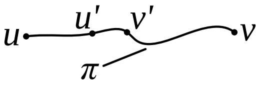
$I(a) = [1, 14]$
$I(c) = [2, 11]$
$I(d) = [3, 10]$
$I(g) = [4, 7]$
$I(e) = [5, 6]$
$I(f) = [8, 9]$
$I(b) = [12, 13]$

$S = \boxed{a} \leftarrow c$

Proof of White Path Theorem

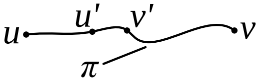
The "if direction" ($\Leftarrow$): Right before u enters the stack, a white path π exists from u to v. We will prove that all the vertices on π must be descendants of u in the DFS-forest.

Suppose that this is not true. Let v' be the first vertex on π — in the order from u to v — that is **not** a descendant of u in the DFS-forest. Clearly $v' \neq u$. Let u' be the vertex preceding v' on π (note: u' may be u).



By the way u' is defined, it must be a descendant of u in the DFS-forest. By the ancestor-descendant lemma, if $u \neq u'$, then u must already be in the stack when u' enters the stack.

Proof of White Path Theorem



Consider the moment when u' turns **black** (i.e., u' leaving the stack).

1. The color of v' cannot be white.

   Otherwise, v' is a white out-neighbor of u', in which case u' cannot be turning black.

2. Hence, the color of v' must be gray or black.

   Recall that when u entered stack, v' was white. Therefore, v' must have been pushed into the stack while u was still in the stack. By the ancestor-descendant lemma, v' must be a descendant of u in the DFS-forest. This, however, contradicts the definition of v'.