Review: Depth First Search

Yufei Tao

Department of Computer Science and Engineering
Chinese University of Hong Kong

This lecture will review the **depth first search** (DFS) algorithm (covered in CSCI2100). The algorithm is deceptively simple and has numerous non-trivial properties.

Our focus will be the **white path theorem**, which we will need to find **strongly connected components** in the next lecture.

DFS

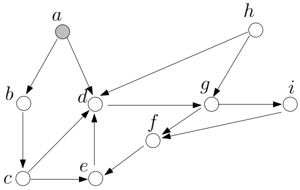
Let $G = (V, E)$ be a directed simple graph.

In the beginning, color all vertices in the graph **white** and create an empty DFS tree T.

Create a stack S. Pick an arbitrary vertex v. Push v into S, and color it **gray** (which means "in the stack"). Make v the root of T.

Example

Suppose that we start from a.



DFS tree

a

$S = (a)$.

DFS

Repeat the following until S is empty.

1. Let v be the vertex that currently tops the stack S (do not remove v from S).
2. Does v still have a white out-neighbor?

   2.1 If yes: let it be u.
   - Push u into S and color u **gray**.
   - Make u a child of v in the DFS-tree T.

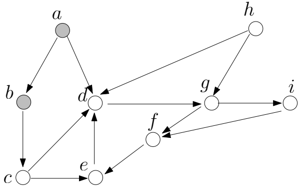
   2.2 If no: pop v from S and color v **black** (meaning v is done).

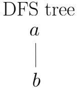
If there are still white vertices, repeat the above by restarting from an arbitrary white vertex v', creating a new DFS-tree rooted at v'.

> DFS finishes in $O(|V| + |E|)$ time.

Running Example

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

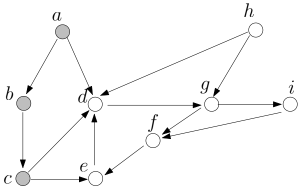


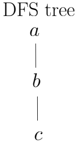


$S = (a, b)$.

Running Example

After pushing c into S:

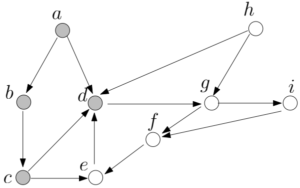


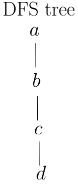


$S = (a, b, c)$.

Running Example

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

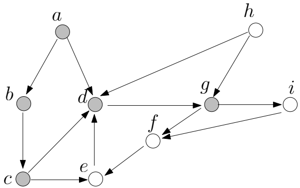


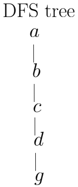


$S = (a, b, c, d)$.

Running Example

After pushing g into S:

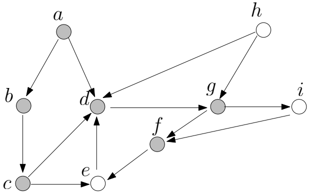


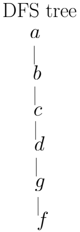


$S = (a, b, c, d, g)$.

Running Example

Suppose we visit white out-neighbor f of g first. Push f into S

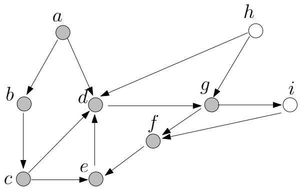


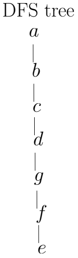


$S = (a, b, c, d, g, f)$.

Running Example

After pushing e into S:

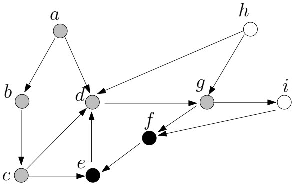


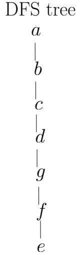


$S = (a, b, c, d, g, f, e)$.

Running Example

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

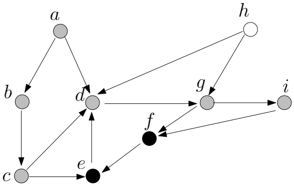


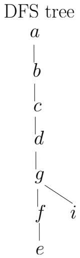


$S = (a, b, c, d, g)$.

Running Example

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

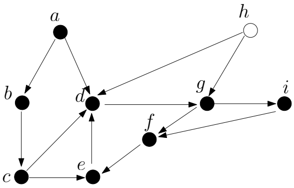


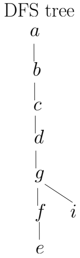


$S = (a, b, c, d, g, i)$.

Running Example

After popping i, g, d, c, b, a:

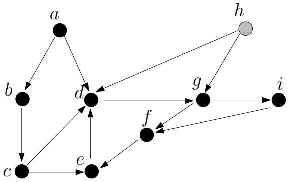


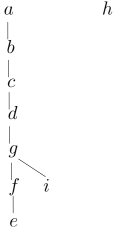


$S = ()$.

Running Example

Now there is still a white vertex h. So we perform another DFS starting from h.

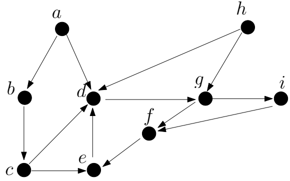


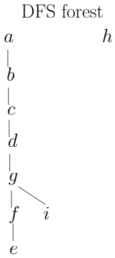


$S = (h)$.

Running Example

Pop h. The end.





$S = ()$.

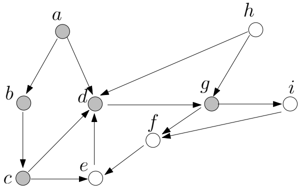
Note that we have created a DFS-forest, which consists of 2 DFS-trees.

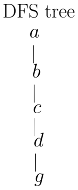
White Path Theorem

Theorem: Let u be a vertex in G. Consider the moment when u enters the stack. Then, a vertex v will become a proper descendant of u in the DFS-forest **if and only** if at the current moment we can go from u to v by traveling on white vertices only (i.e., there is a white path from u to v).

Example

Consider the moment in our previous example when g just entered the stack. $S = (a, b, c, d, g)$.





We can see that g can reach f, e, and i by hopping on only white vertices. Therefore, f, e, and i are proper descendants of g in the DFS-forest; and g has no other descendants.