Computer Algorithms I

Spring 2020

Depth-first search

start with a vertex s

for every undiscovered neighbor, discover that node, and proceed recursively

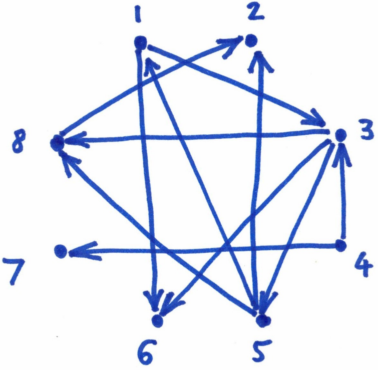
time: global variable, a counter increased at the beginning and end of each recursive call

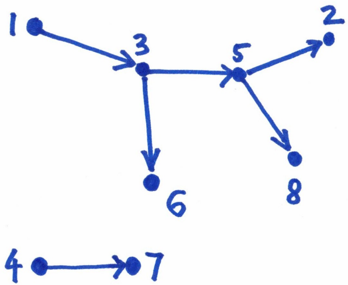
v.d, v.f: discovery and finishing times of vertex v

Depth-first search algorithm

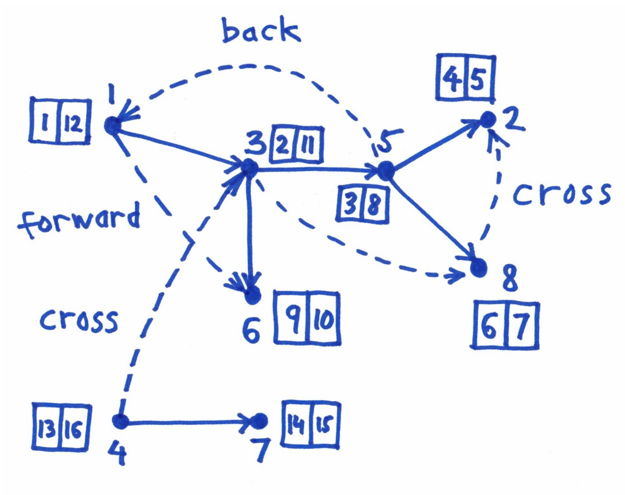
```
DFS(G)
for each vertex u ∈ G.V
    u.color = WHITE
    u.π = NIL
time = 0
for each vertex u ∈ G.V
    if u.color == WHITE
        DFS-VISIT(G, u)

DFS-VISIT(G, u)
time = time + 1              // white vertex u has just been discovered
u.d = time
u.color = GRAY
for each v ∈ G.Adj[u]        // explore edge (u, v)
    if v.color == WHITE
        v.π = u
        DFS-VISIT(G, v)
u.color = BLACK              // blacken u; it is finished
time = time + 1
u.f = time
```

Depth-first forest example

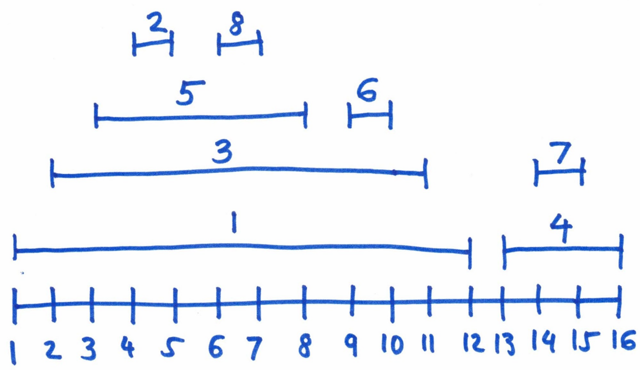




Discovery and finishing times, edge types



Parenthesis theorem



Parenthesis theorem

The time intervals $[u.d, u.f]$ and $[v.d, v.f]$ are either disjoint or one is contained in the other.

If the intervals are disjoint then neither vertex is a descendant of the other.

If $[u.d, u.f]$ contains $[v.d, v.f]$ then v is a descendant of u (and vice versa).

Proof outline

Assume $u.d < v.d$

Case 1: $u.d < v.d < u.f$

Then v was discovered inside $DFS - VISIT(G, u)$, and so it is a descendant of u and it finishes before u. So $[v.d, v.f] \subset [u.d, u.f]$.

Case 2:: $u.d < u.f < v.d$

Then v was discovered after u was finished, so neither is a descendant of the other, and the two time intervals are disjoint.

White path theorem

v is a descendant of u iff at time $u.d$ there is a path of white vertices from u to v

Edge types for directed graphs

tree edges: (u, v), such that $v.\pi = u$

back edges: (u, v), such that v is ancestor of u

forward edges: (u, v), such that v is a descendant of u, but $v.\pi \neq u$

cross edges: (u, v), such that neither is an ancestor of the other (in the same tree, or in different trees).

if (u, v) is a cross edge then $u.d > v.d$ (why?)

Edge types for undirected graphs

tree edges: (u, v), such that $\quad v.\pi = u$

back edges: (u, v), such that v is ancestor of u

no cross edges! (why?)

Remarks on depth-first search

stack (LIFO - last-in, first-out)

complexity $O(|V| + |E|)$, argument similar to BFS