L20 - Depth First Search/Traversal/Visit

Tuesday, June 2, 2020 9:01 AM

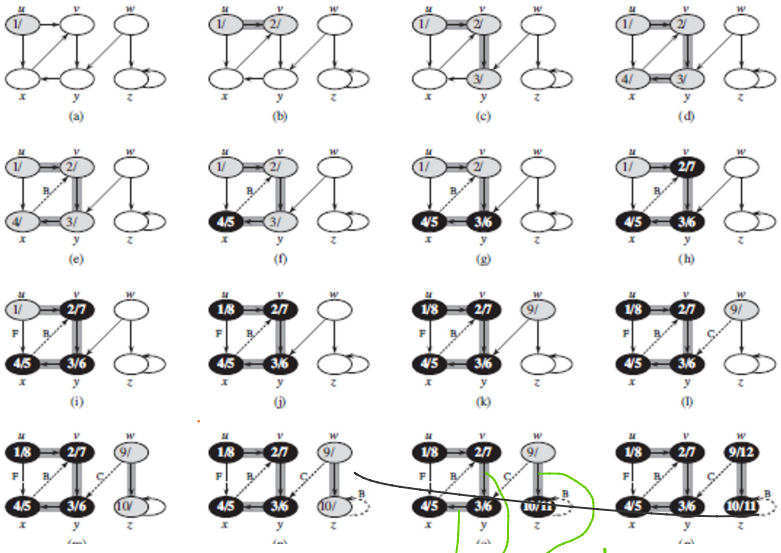
To search "deeper", at greater distance from a given node, whenever possible.

DFS(G)
1. for each vertex k ∈ G.V O(|V|)
2. u. *color* = WHITE
3. k.π = NIL
4. *time* = 0
5. for each vertex k ∈ G.V O(|V|) excluding time taken by DFS-VISIT
6. if k. *color* == WHITE
7. DFS-VISIT(G; k)

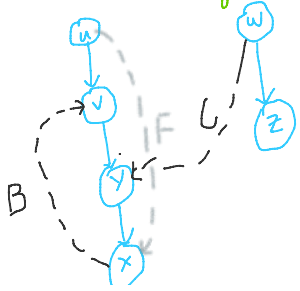
Predecessor: Nil u v y Nil w | Nil q s v q t x t
Nodes: u v y x w z | q s v w t x z y

DFS-VISIT(G; k) |V| O(|V|+|E|) Linear Time Complexity in the size of graph
1. *time* = *time* + 1 // white vertex u has just been discovered
2. k. *d* = *time* // k. d means the time when vertex k was first **discovered**.
3. k. *color* = GRAY
4. for each l ∈ G.Adj[k] // explore edge (k; l)
5. if l. *color* == WHITE
6. l.π = k
7. DFS-VISIT(G; l)
8. k. *color* = BLACK // blacken k; it is finished
9. *time* = *time* + 1
10. k. *f* = *time* //finish time

|V|+2×|E|
2×|E|
|E|
O(|E|)



Tree Edges



Predecessor: Nil u v y Nil w
Nodes: u v y x w z
Discover:
Finish

PRINT-PATH(G; s; v)
1. if v == s

DFS-VISIT(G; k) |V| O(|V|+|E|) Linear Time Complexity in the size of graph
1. *time* = *time* + 1 // white vertex u has just been discovered
2. k. *d* = *time* // k. d means the time when vertex k was first **discovered**.
3. k. *color* = GRAY
4. for each l ∈ G.Adj[k] // explore edge (k; l)
5. if l. *color* == WHITE
6. l.π = k
7. DFS-VISIT(G; l)
8. k. *color* = BLACK // blacken k; it is finished
9. *time* = *time* + 1
10. k. *f* = *time* //finish time

PRINT-PATH(G; s; v)
1. if v == s
2. print s
3. else if v.π == NIL
4. print “no path from” s “to” v “exists”
5. else PRINT-PATH(G; s; v.π)
6. print v

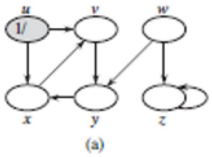
PRINT-PATH(G; u; x) ----> PRINT-PATH(G; u; y) ----> PRINT-PATH(G; u; v) -----> PRINT-PATH(G; u; u)
x y v u

Classification Of Edges
- Tree Edge: the edges of the Depth First tree
- Back Edges: The edges that go from a node to an ancestor in a Depth First Tree.
- Forward Edge: The edges that go from a vertex/node to a descendant in a Depth First Tree.
- Cross Edge: Any other edge is a Cross Edge.

(q, s) 0/ Tree
(s, v) 1/ Tree
(v, w) 2/ Tree
(w, s) Back
(q, w) Forward

q s t w
r u y
s v
t x y
u y
v w
w s
x z
y q
z x

Predecessor:
Node:
Discovery Time:
Finish Time:



(a)

DFS-VISIT(G; k) |V| O(|V|+|E|) Linear Time Complexity in the size of graph
1. *time* = *time* + 1 // white vertex u has just been discovered
2. k. *d* = *time* // k. d means the time when vertex k was first **discovered**.
3. k. *color* = GRAY
4. for each l ∈ G.Adj[k] // explore edge (k; l)
5. if l. *color* == WHITE
6. l.π = k
7. DFS-VISIT(G; l)
8. k. *color* = BLACK // blacken k; it is finished
9. *time* = *time* + 1
10. k. *f* = *time* //*finish time*