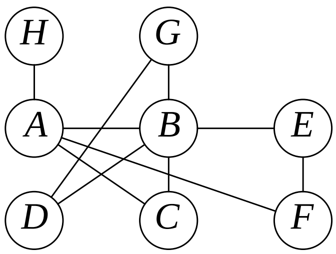
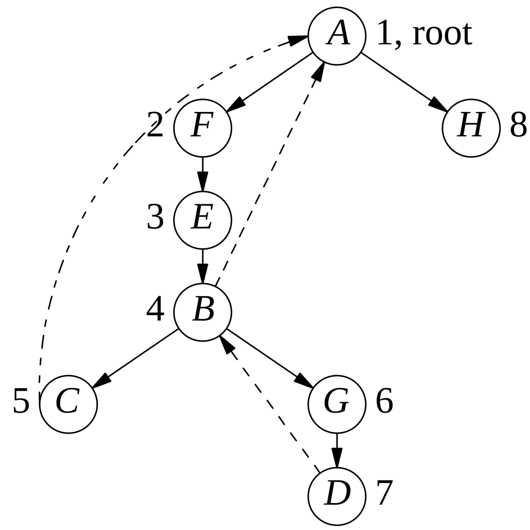
SOLUTIONS



adjList(A) = ⟨F, B, C, H⟩
adjList(B) = ⟨A, C, E, G, D⟩
adjList(C) = ⟨A, B⟩, adjList(D) = ⟨B, G⟩
adjList(E) = ⟨B, F⟩, adjList(F) = ⟨A, E⟩
adjList(G) = ⟨B, D⟩, adjList(H) = ⟨A⟩

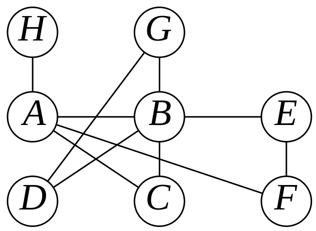


(ii) The dfTree, dfLabels, and back-edges.

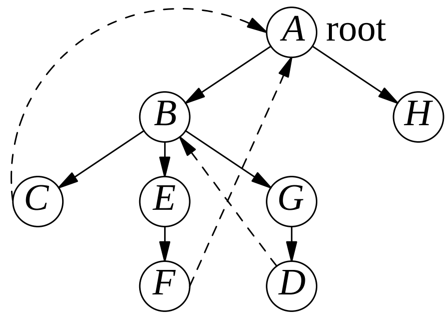
Order of Processing Links and Backtracking: StartNode = A.

(A, F)	tree-edge	(D, G)	2nd visit
(F, A)	2nd visit	backtrack	$D \to G$
(F, E)	tree-edge	backtrack	$G \to B$
(E, B)	tree-edge	(B, D)	2nd visit
(B, A)	back-edge	backtrack	$B \to E$
(B, C)	tree-edge	(E, F)	2nd visit
(C, A)	back-edge	backtrack	$E \to F$
(C, B)	2nd visit	backtrack	$F \to A$
backtrack	$C \to B$	(A, B)	2nd visit
(B, E)	2nd visit	(A, C)	2nd visit
(B, G)	tree-edge	(A, H)	tree-edge
(G, B)	2nd visit	(H, A)	2nd visit
(G, D)	tree-edge	backtrack	$H \to A$
(D, B)	back-edge	backtrack	$A \to$

SOLUTION



(i) A graph.



(ii) A depth-first tree.

Possible orderings of adjList(A), etc without changing the df-tree;

- We assume below that the order of the children of each node remain the same (for example, B = firstChild(A) and H = lastChild(A)).
- If we ignore the order, there are many more orderings; we show below two such cases for adjList(A) with a "*" next to them.

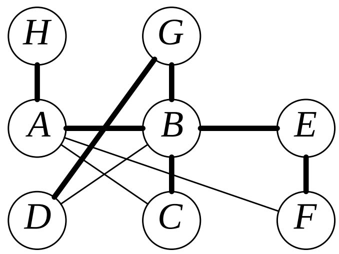
adList(A): The only restriction: C and F comes after B.
$\langle B, C, F, H\rangle$, $\langle B, F, C, H\rangle$, $\langle B, H, C, F\rangle$,
$\langle B, C, H, F\rangle$, $\langle B, F, H, C\rangle$, $\langle B, H, F, C\rangle$,
$\langle H, B, C, F\rangle$*, $\langle H, B, F, C\rangle$*

adjList(B): The only restriction: D comes after G
$\langle A, C, E, G, D\rangle$, $\langle C, A, E, G, D\rangle$, $\langle C, E, A, G, D\rangle$,
$\langle C, E, G, A, D\rangle$, $\langle C, E, G, D, A\rangle$,
(55 more if we ignore the order of C, E, G.)

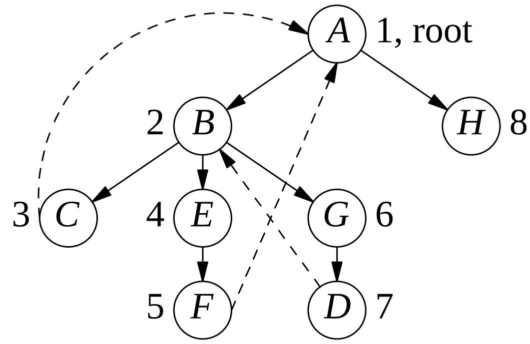
- Keeping the order of children at each node unchanged

Total = Multiply the possible number of orderings of each adjList
= 6×5×2×2×2×2×2×1 = 960.

SOLUTION



adjList(A) = $\langle B, C, F, H\rangle$
adjList(B) = $\langle A, C, E, G, D\rangle$
adjList(C) = $\langle A, B\rangle$, adjList(D) = $\langle B, G\rangle$
adjList(E) = $\langle B, F$ (ra, adjList(F) = $\langle A, E\rangle$
adjList(G) = $\langle B, D\rangle$, adjList(H) = $\langle A\rangle$



(ii) The dfTree, dfLabels, and back-edges.

Step of the algorithm used in processing each edge

(A, B)	tree-edge	2d(i)	(G, B)	2nd visit	···
(B, A)	2nd visit	2d(ii)	(G, D)	tree-edge	···
(B, C)	tree-edge	2d(i)	(D, B)	back-edge	
(C, A)	back-edge	2d(ii)	(D, G)	2nd visit	
(C, B)	2nd visit	2d(ii)	backtrack	$D \to G$	
backtrack	$C \to B$	2b	backtrack	$G \to B$	
(B, E)	tree-edge	···	(B, D)	2nd visit	
(E, B)	2nd visit	···	backtrack	$B \to A$	
(E, F)	tree-edge		(A, C)	2nd visit	
(F, A)	back-edge		(A, F)	2nd visit	
(F, E)	2nd visit		(A, H)	tree-edge	
backtrack	$F \to E$		(H, A)	2nd visit	
bacltrack	$E \to B$		backtrack	$H \to A$	
(B, G)	tree-edge		backtrack	$A \to$	