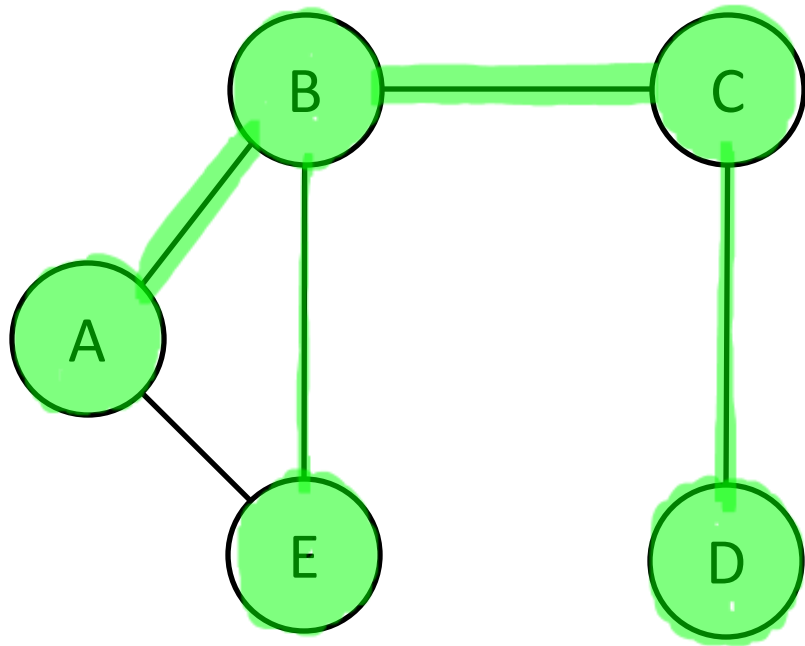


Graph Traversals

Depth First Traversal

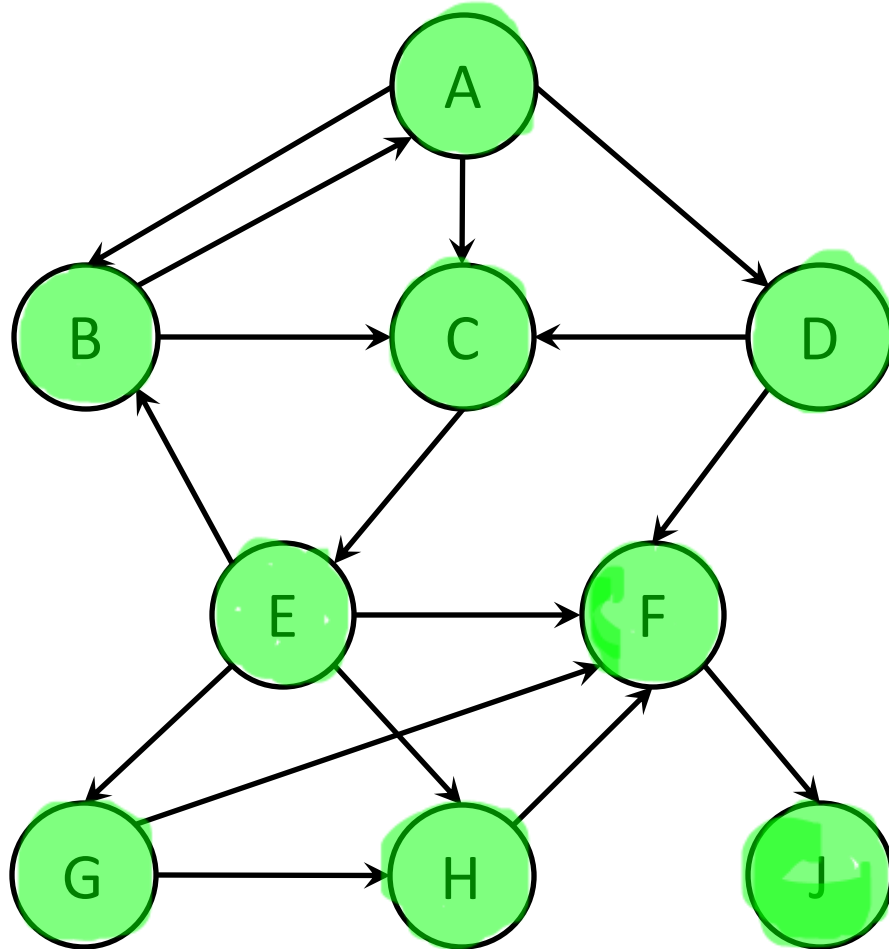
Assume: all vertices are marked unvisited at start $\Rightarrow V * V$

adj. matrix	adj. list
for each call to DFT: V for-loop iterations	across V calls to DFT: E for-loop iterations
$\Rightarrow V * V$	$\Rightarrow V + E$

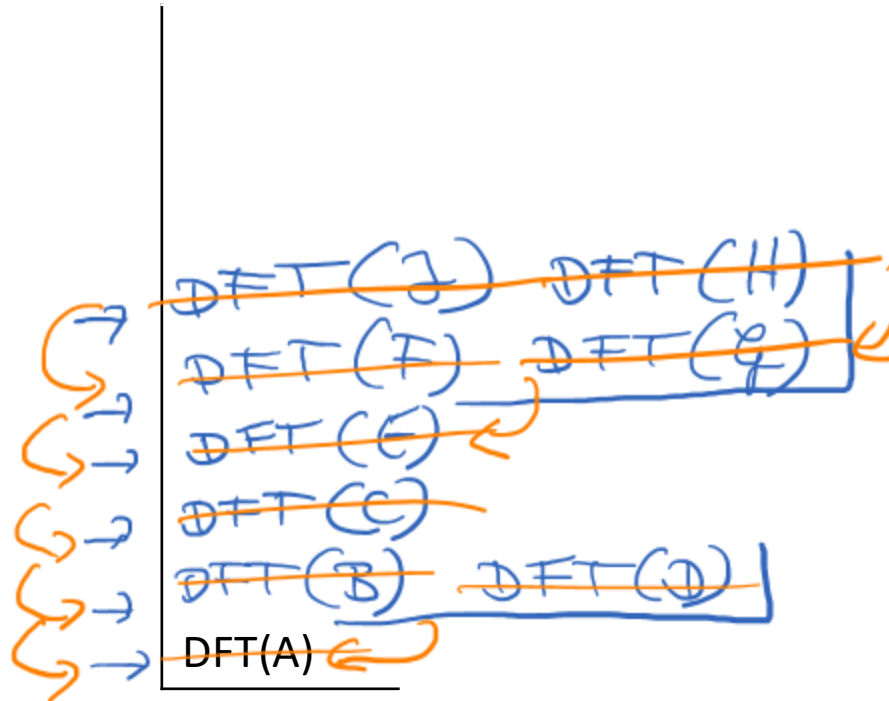


DFT(v): ↖ Start node
← V calls
↗ mark v as visited
↘ for each unvisited neighbor u of v:
DFT(u)

Depth First Example



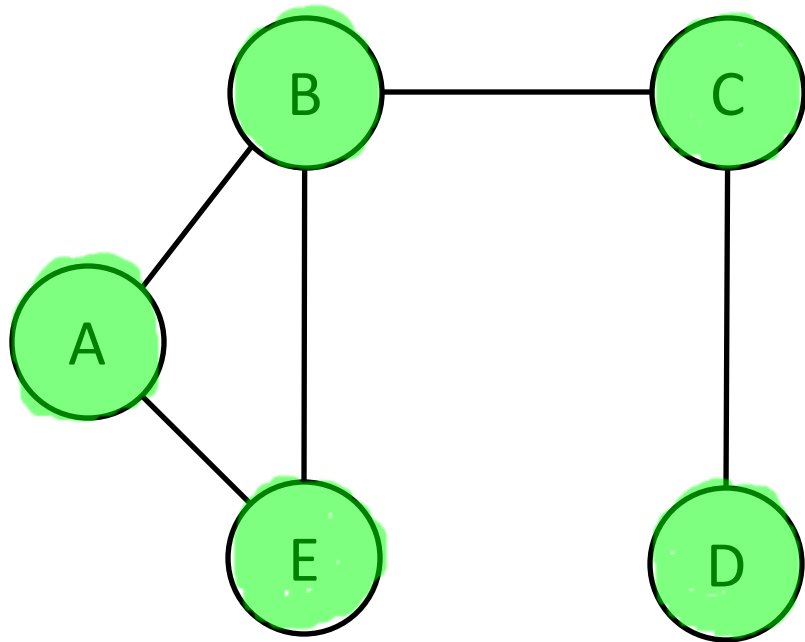
starting node: A



visit sequence: A, B, C, E, F, I, G, H, D

Breadth First Traversal

Assume: all vertices are marked unvisited at start



BFT(v):

q = new Queue()

mark v as visited

q.enqueue(v)

while (!q.isEmpty()):

c = q.dequeue()

for each unvisited neighbor u of c:

mark u as visited

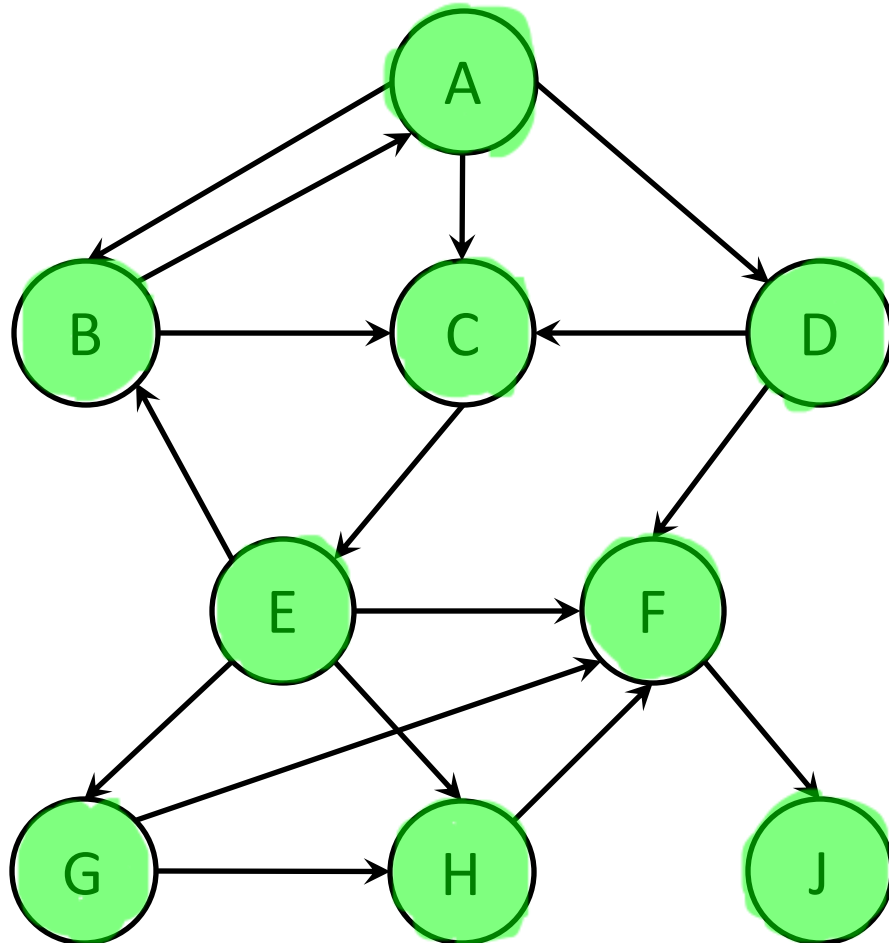
q.enqueue(u)

Start node

V iterations

adj. matrix	adj. list
for each while iter: V for-loop iter $\Rightarrow V * V$	across V while iter: E for-loop iter $\Rightarrow V + E$

Breadth First Example



starting node: A

queue:

<- out

<- in



visit sequence: A, B, C, D, E, F, G, H, J

Time Complexity

V : # of nodes in graph

E : # of edges in graph

Depth First		Breadth First	
adjacency matrix	adjacency list	adjacency matrix	adjacency list
$O(V^2)$	$O(V+E)$	$O(V^2)$	$O(V+E)$