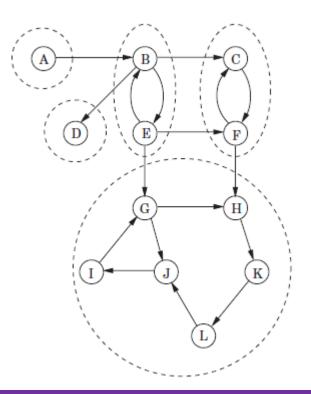
"본 강의 동영상 및 자료는 대한민국 저작권법을 준수합니다. 본 강의 동영상 및 자료는 상명대학교 재학생들의 수업목적으로 제작·배포되는 것이므로, 수업목적으로 내려받은 강의 동영상 및 자료는 수업목적 이외에 다른 용도로 사용할 수 없으며, 다른 장소 및 타인에게 복제, 전송하여 공유할 수 없습니다. 이를 위반해서 발생하는 모든 법적 책임은 행위 주체인 본인에게 있습니다."

Classification of graph algorithms graph U D U D **DFS BFS** Greedy (4.2 & (4.6 &4.3) 4.7) Previsit & postvisit Types of edges Directed acyclic graph Strongly All-pairs **Biconnected** Single-source **Spanning Connected** connected shortest Prim/ **Shortest path** Component component tree Component Kruskal path (4.5)(4.8)(4.4)U U U D DW/UW DW/UW DW/UW

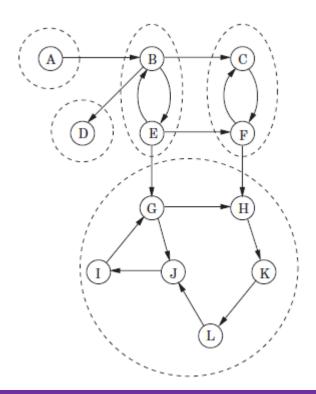
(1) Connectivity for directed graph

- Strongly connected
 - Two nodes u and v of a directed graph are connected if there is a path from u to v and vice versa
 - Example)
 - » B & E are connected
 - » G & H are connected
 - » J & L are connected
 - » E & G are not connected

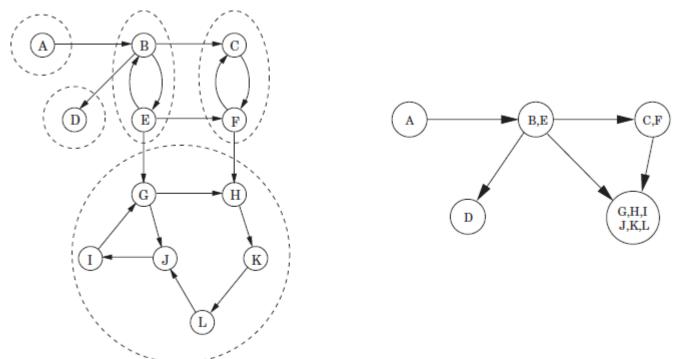


(1) Connectivity for directed graph

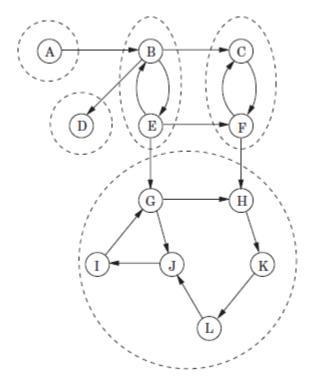
- Strongly connected component (SCC)
 - Partitioning of V into disjoint sets according to the definition of "strongly connected"
 - Example)
 - » B & E are SCC
 - » A is SCC
 - » G, H, I, J, K & L are SCC

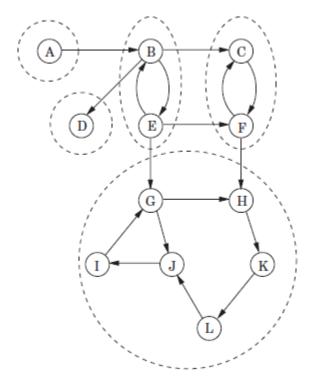


- (1) Connectivity for directed graph
 - Property
 - Every directed graph is a dag of its strongly connected components

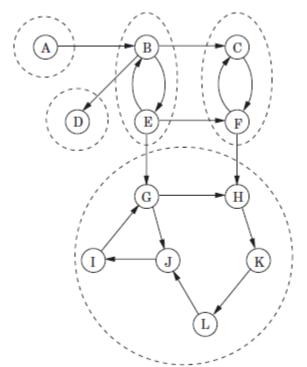


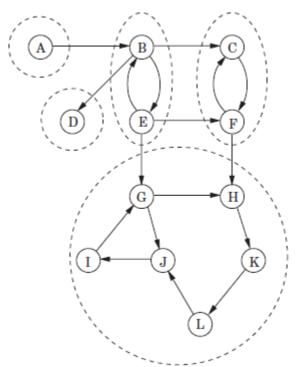
- Property1
 - If the dfs () subroutine is started at a node u, then it will terminate precisely when all nodes reachable from u have been visited



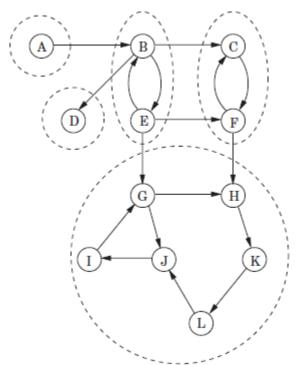


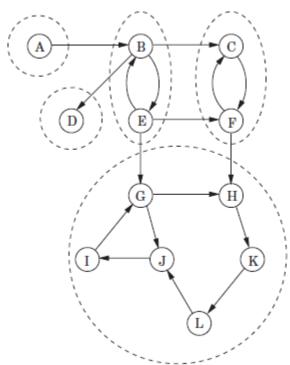
- Property2
 - The node that receives the highest post number in a depth-first search must lie in a source strongly connected components





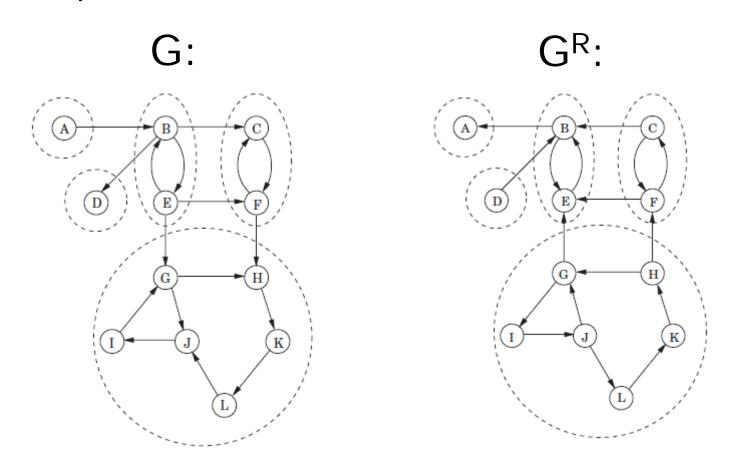
- Property3
 - If C and C' are strongly connected components, and there is an edge from a node in C to a node in C', then the highest post number in C is bigger than the highest post number in C'





- Strategy
 - Find a sink strongly connected component and remove it
 - Repeat this until we have only one strongly connected component
- Problem
 - How can we find a sink strongly connected component?
 - Motivation
 - » Use Property2
 - » Define G^R from G = (V, E)
 - » GR has same V, but reverse E
 - » Sink component in G = Source component in G^R

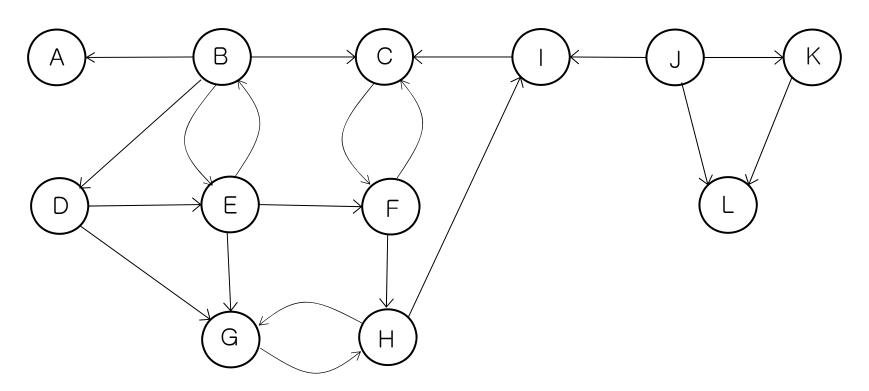
– Example



- Steps
 - Compute G^R from G
 - Run depth-first search on G^R
 - Run the undirected connected components algorithm
 - Process the vertices in decreasing order of their post numbers from the previous step

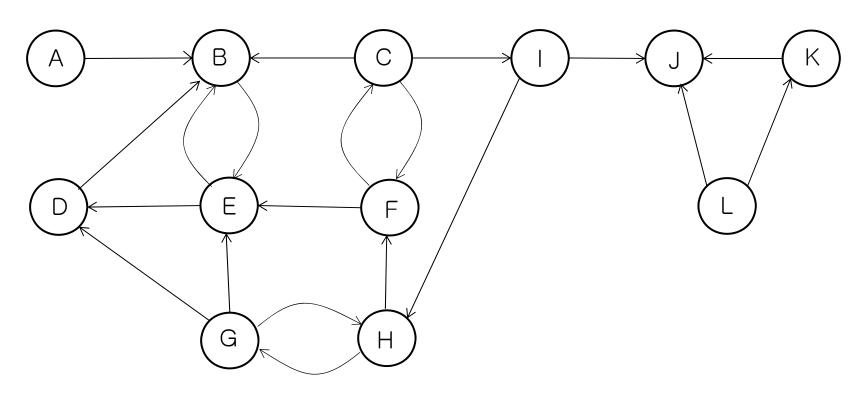
(2) Algorithm

- Test (알파벳 순으로)



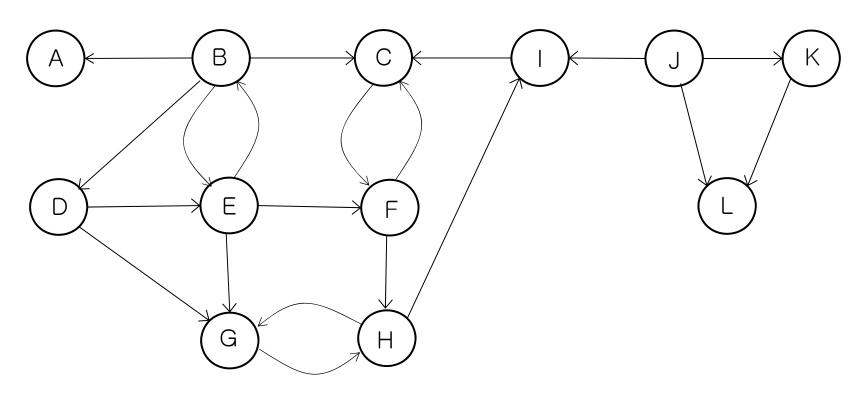
(2) Algorithm

- Test (알파벳 순으로)



(2) Algorithm

- Test (알파벳 역순으로)



All about graph

Туре	Purpose	Operations	Performance
DFS	Traverse all vertices	Visiting all vertices & visiting all edges	O(n) + O(m)
SCC	Finding SCC	DFS on G ^R and G	O(DFS)
ВСС			
BFS			
Dijkstra			
Floyd			
Kruskal (Greedy)			
Prim (Greedy)			
MultiStage (Dynamic)			

다음은 strongly connected component를 찾는 알고리즘에 대한 설명이다. 올바른 것을 모두 고르시오.

- (a) strongly connected component들은 dag를 형성한다.
- (b) strongly connected component에서 가장 높은 post num을 갖는 vertex를 가진 component는 항상 sink이다.
- (c) strongly connected component를 계산하는 연산 시간은 O(n + m)이다 (n은 vertex의 수, m은 edge의 수)
- (d) strongly connected component의 결과는 시작하는 vertex에 따라서 다르다.