Tutorials - Week 11 - Nov 21

Graphs

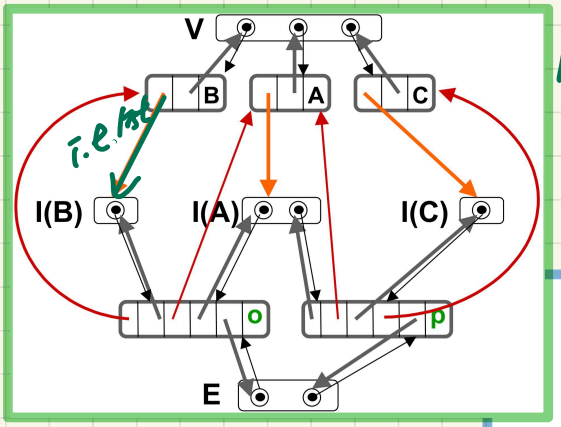
Assignment 2 Solution
Graph Implementation Strategies

Graphs in Java: Strategies

	Vertex	Edge	Graph
Adjacency List	incidentEdges	originIncidentListPos	isDirected vertices edges
		destIncidentListPos	
Edge List	vertexListPosition	edgeListPosition	
Adjacency Matrix			
	index		matrix

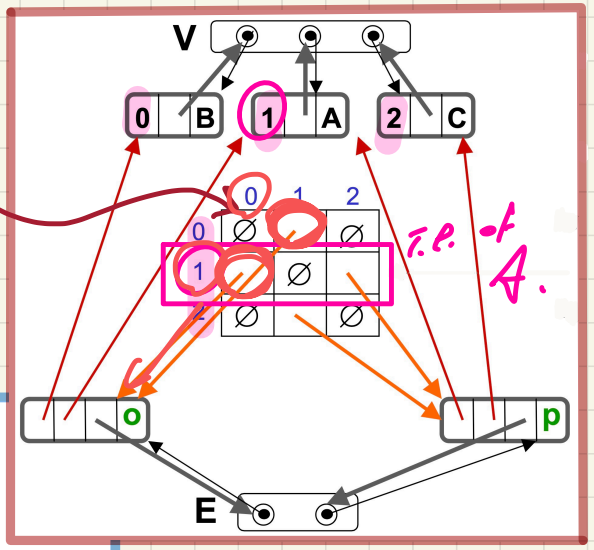
matrix[1][0] == matrix[0,1]

adjacency matrix

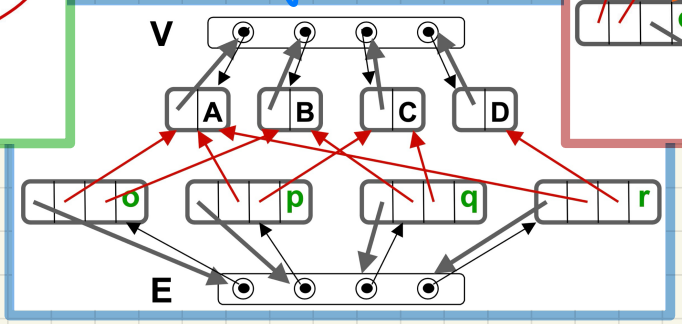


Adjacency List.

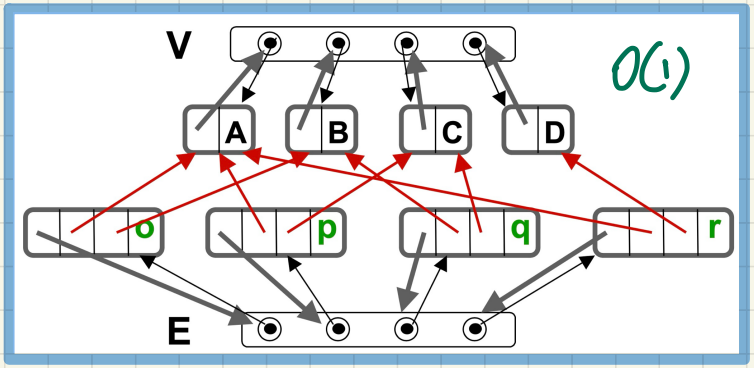
matrix (edges).



Edge List

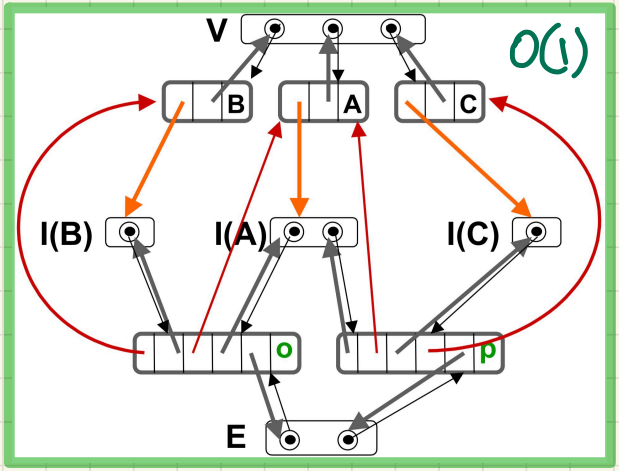


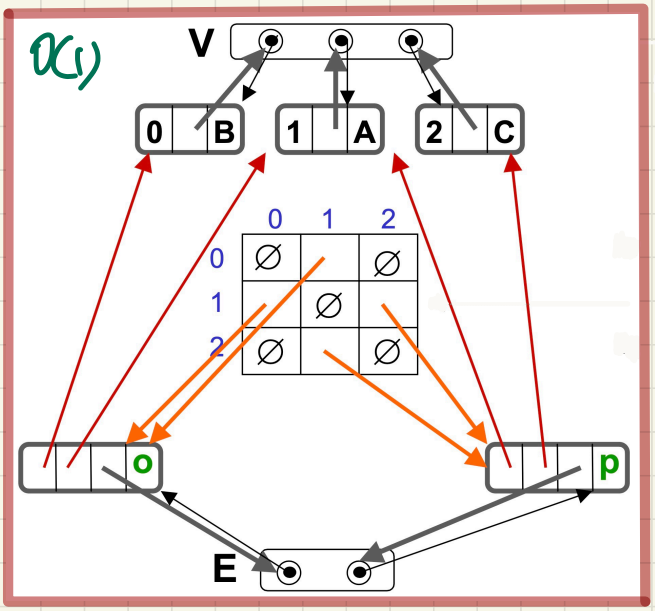
Graphs in Java: Time Complexities (1)



numVertices(), numEdges()

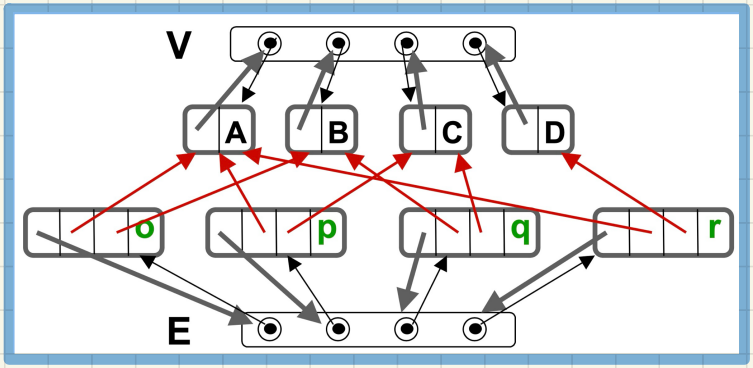
(size).
attributes of DLL.





Graphs in Java: Time Complexities (2)

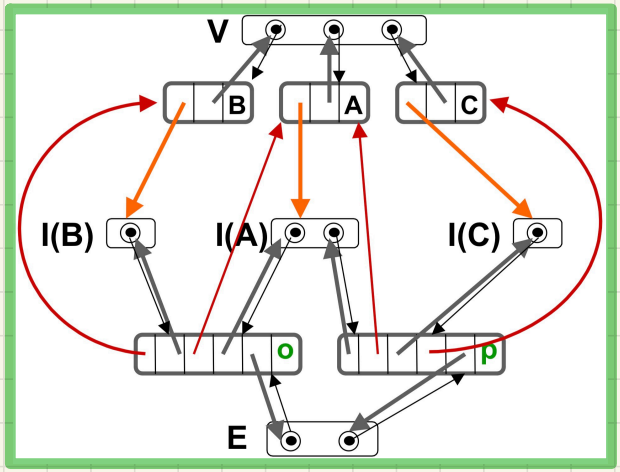
$|V| = n$
$|E| = m$

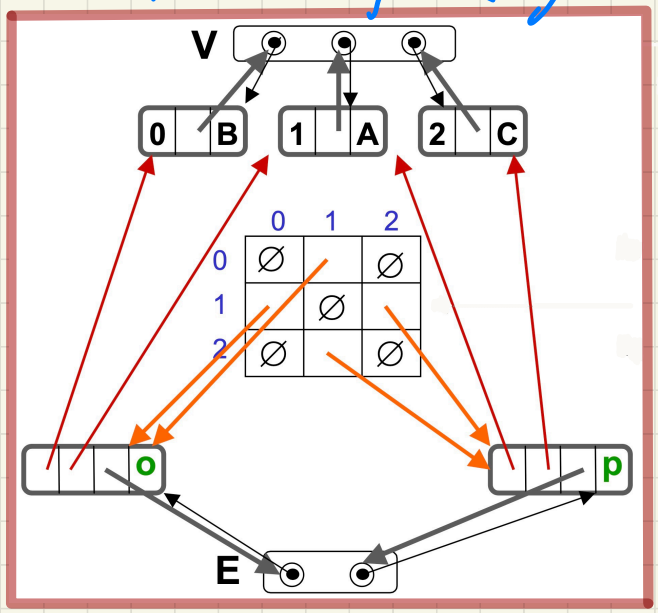


vertices(), edges()

$O(n)$ $O(m)$

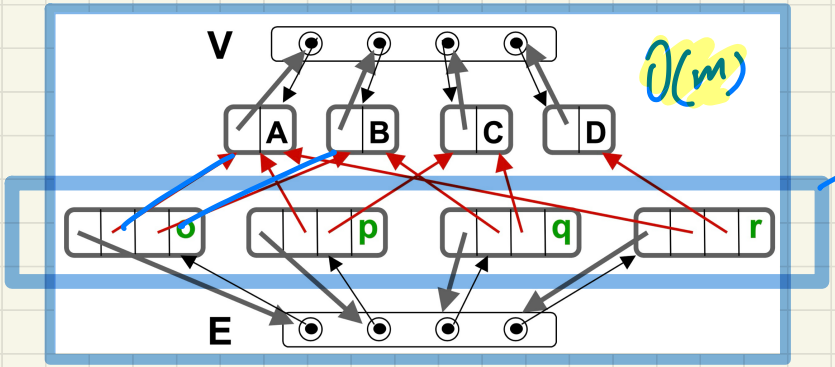
go over DLLs to create the iterable objects (e.g., ArrayList)





Graphs in Java: Time Complexities (3)

* A.getIndex()
** C.getIndex()

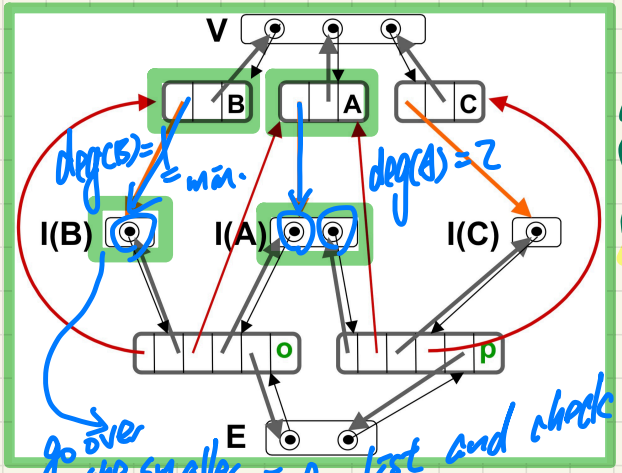


getEdge(u, v)

O(m)

→ go over m edges and check ori. & des. against u & v.

getEdge(A, C) → matrix[ ][ ]

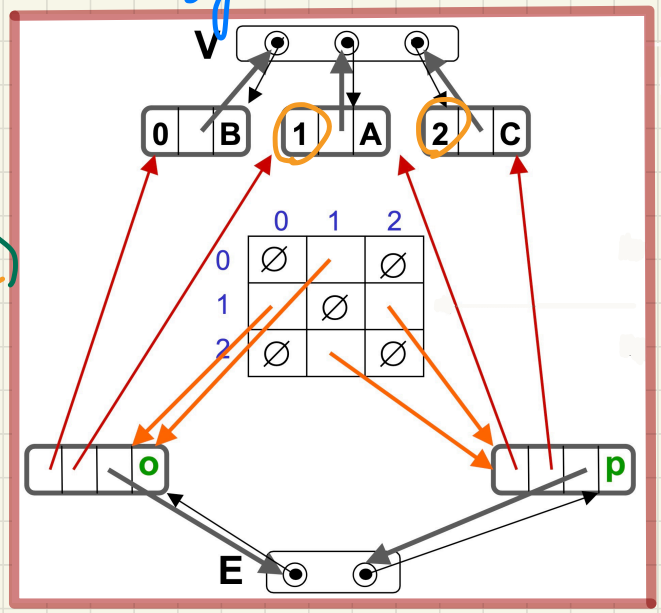


go over the smaller i.e. 1st and check ori. & des. against A & B.

getEdge(A, B)

O(min(d_u, d_v))

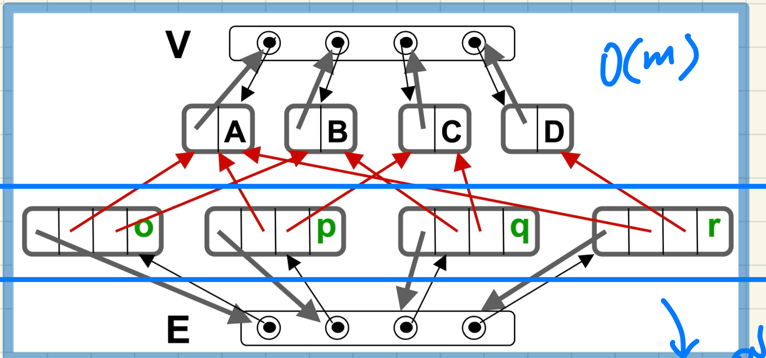
↳ possible n-1



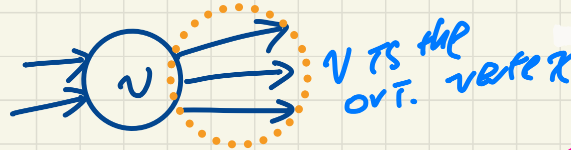
Graphs in Java: Time Complexities (4)

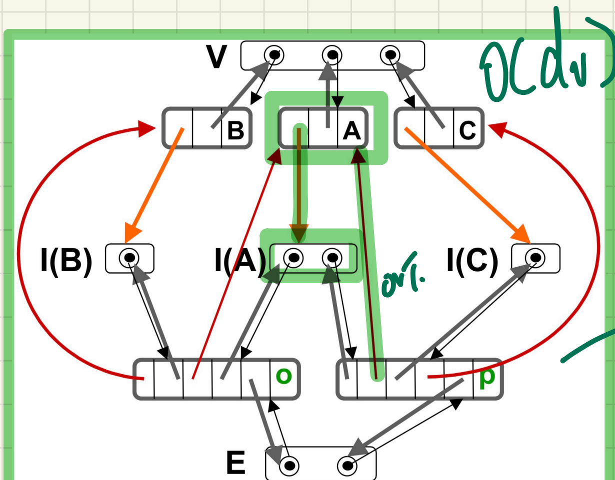
go over columns at the row indexed by v, count those whose ori. is v.

outDegree(u), inDegree(u)
inEdges(v), outEdges(v)

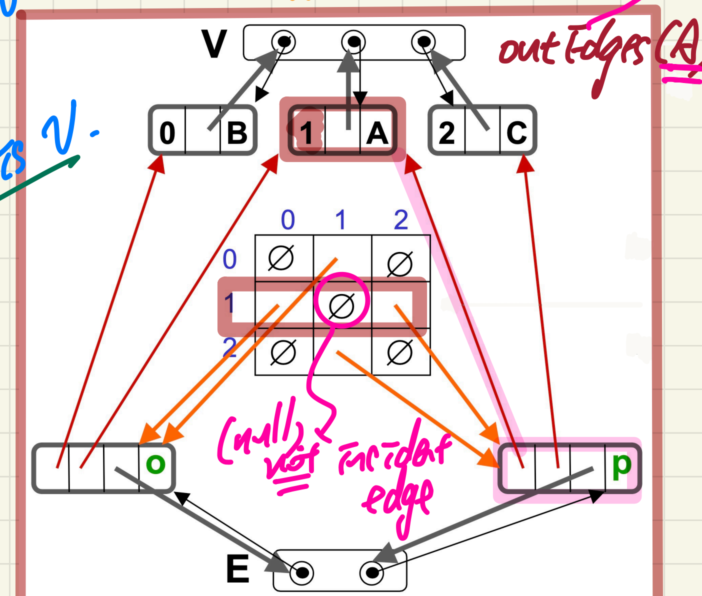


go over m edges count those whose origins is v.

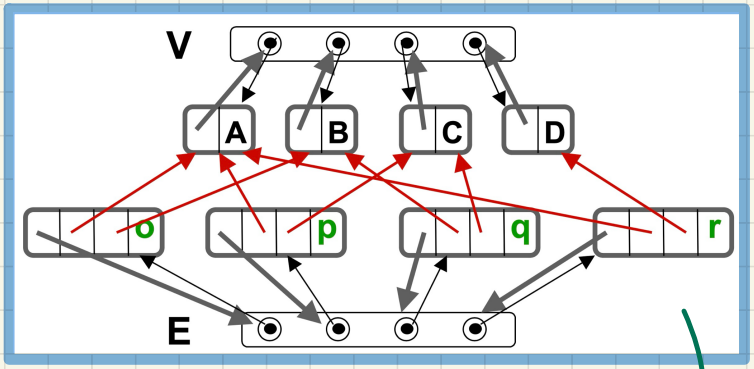


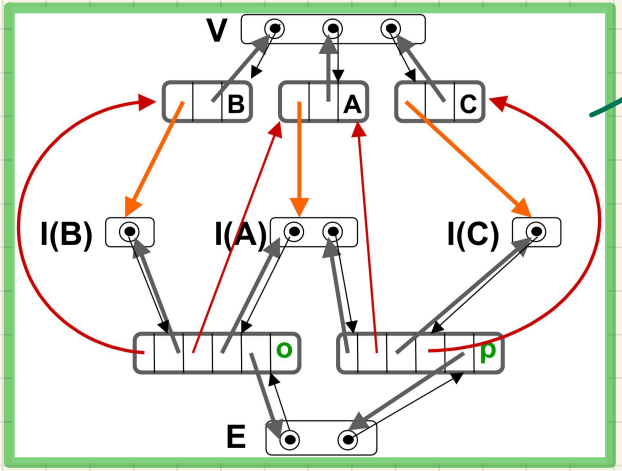


go over v's i.e. list count those whose origin is v.



Graphs in Java: Time Complexities (5)





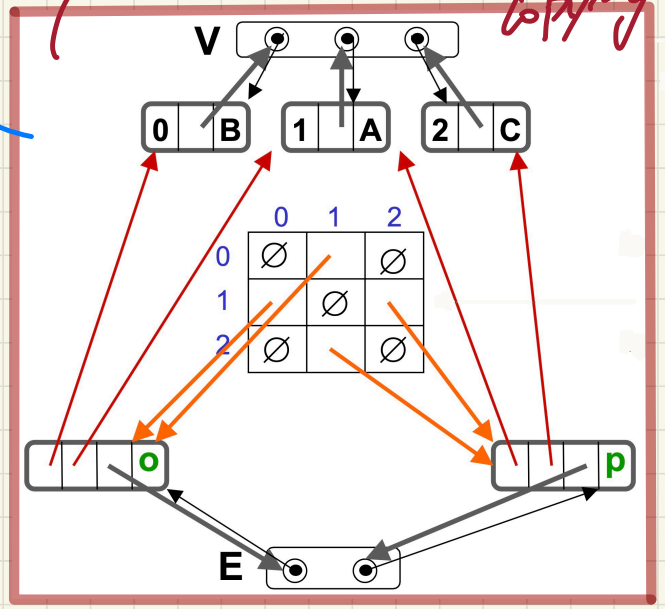
O(1)
insert last to DLL.

insertVertex(x)

what if AL<AL<E>> is used?

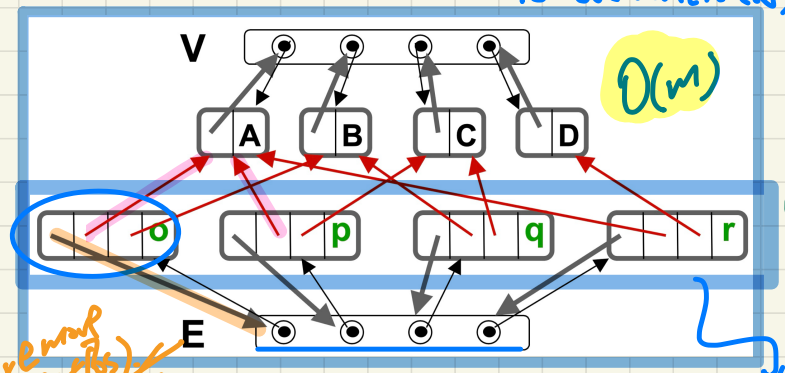
n x n
↳ (n+1) x (n+1) matrix

O(n²) copying.



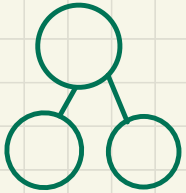
Graphs in Java: Time Complexities (6)

removeVertex(A).

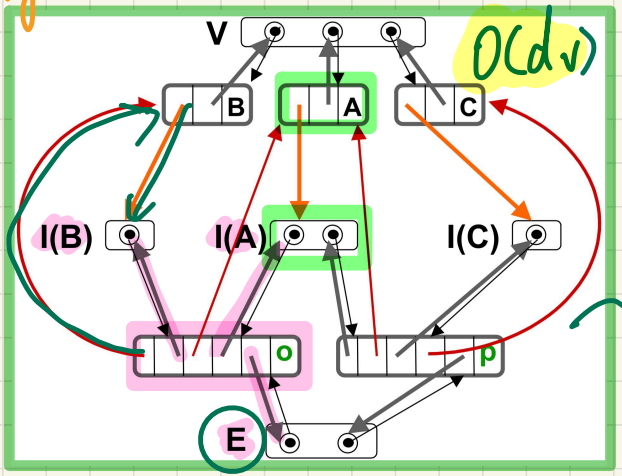


E. remove(o.getEdge())

removeVertex(v)

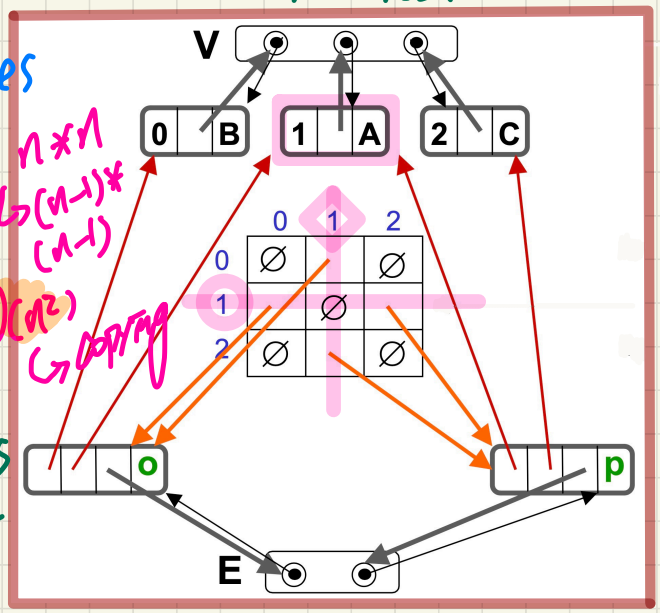


* each edge should be removed from:
(1) list of edges (3) des's i.e. list
(2) in's i.e. list



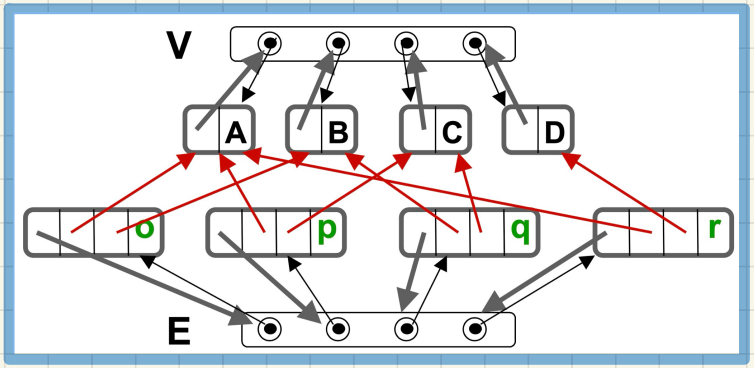
go over m edges and remove those whose out. or des. is v.

go over V's i.e. lists and remove all edges from there *



n*n
(n-1)*(n-1)
O(n²)
↳ looping

Graphs in Java: Time Complexities (7)



insertEdge(u, v, x),
removeEdge(e)