



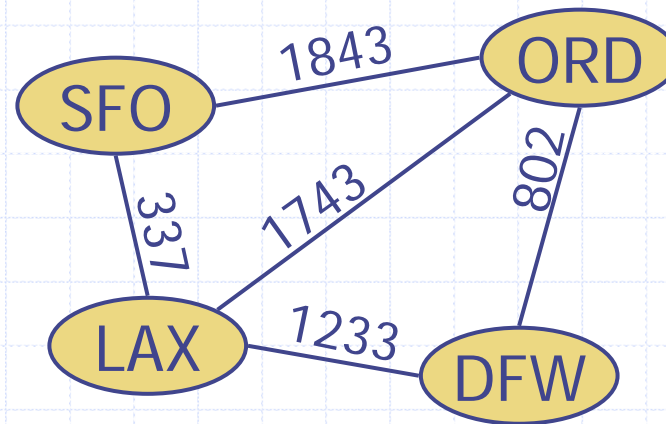
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

# Αλγόριθμοι και πολυπλοκότητα

## Graphs

Ιωάννης Τόλλης  
Τμήμα Επιστήμης Υπολογιστών

# Graphs



# Outline and Reading

## ◆ Graphs (§6.1)

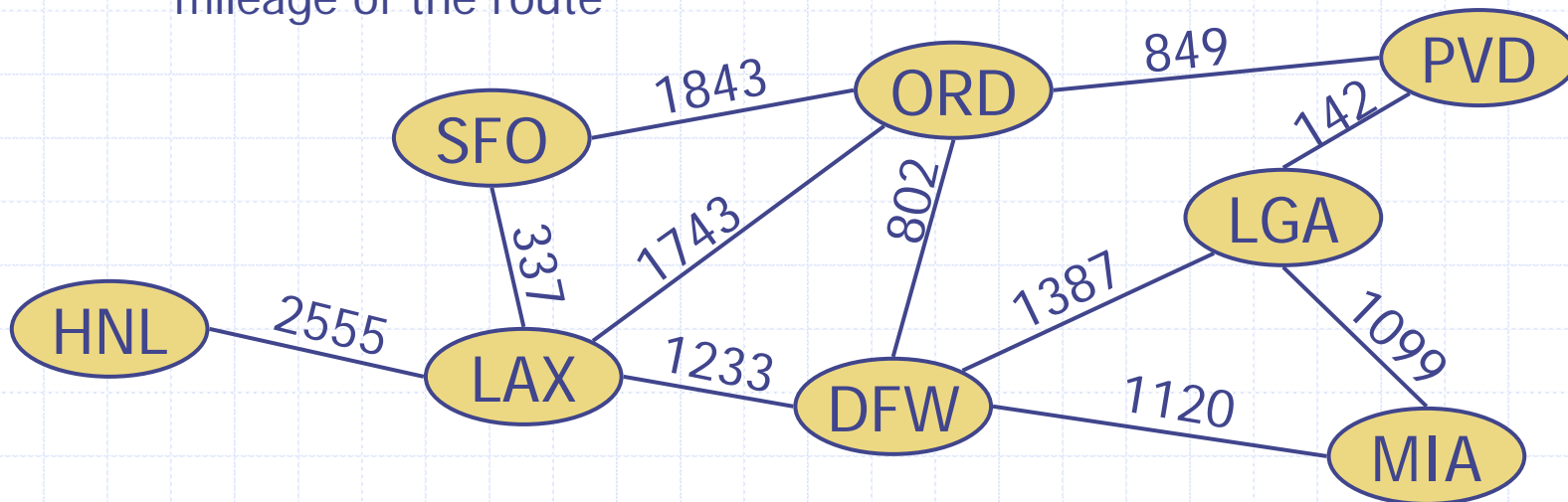
- Definition
- Applications
- Terminology
- Properties
- ADT

## ◆ Data structures for graphs (§6.2)

- Edge list structure
- Adjacency list structure
- Adjacency matrix structure

# Graph

- ◆ A graph is a pair  $(V, E)$ , where
  - $V$  is a set of nodes, called **vertices**
  - $E$  is a collection of pairs of vertices, called **edges**
  - Vertices and edges are positions and store elements
- ◆ Example:
  - A vertex represents an airport and stores the three-letter airport code
  - An edge represents a flight route between two airports and stores the mileage of the route



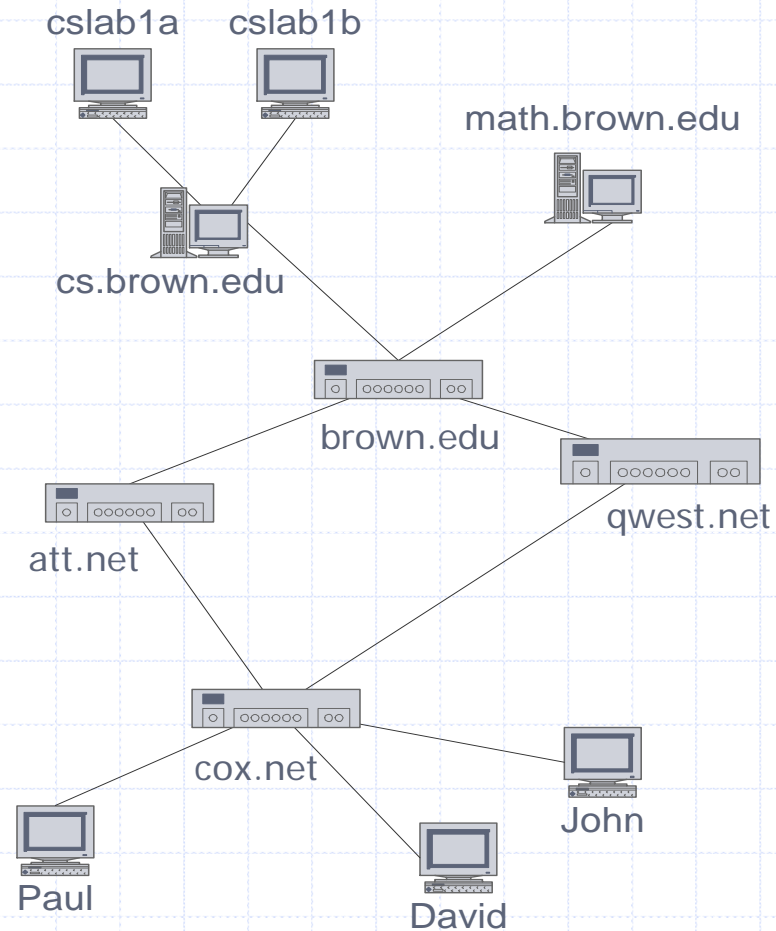
# Edge Types

- ◆ Directed edge
  - ordered pair of vertices  $(u,v)$
  - first vertex  $u$  is the origin
  - second vertex  $v$  is the destination
  - e.g., a flight
- ◆ Undirected edge
  - unordered pair of vertices  $(u,v)$
  - e.g., a flight route
- ◆ Directed graph
  - all the edges are directed
  - e.g., flight network
- ◆ Undirected graph
  - all the edges are undirected
  - e.g., route network



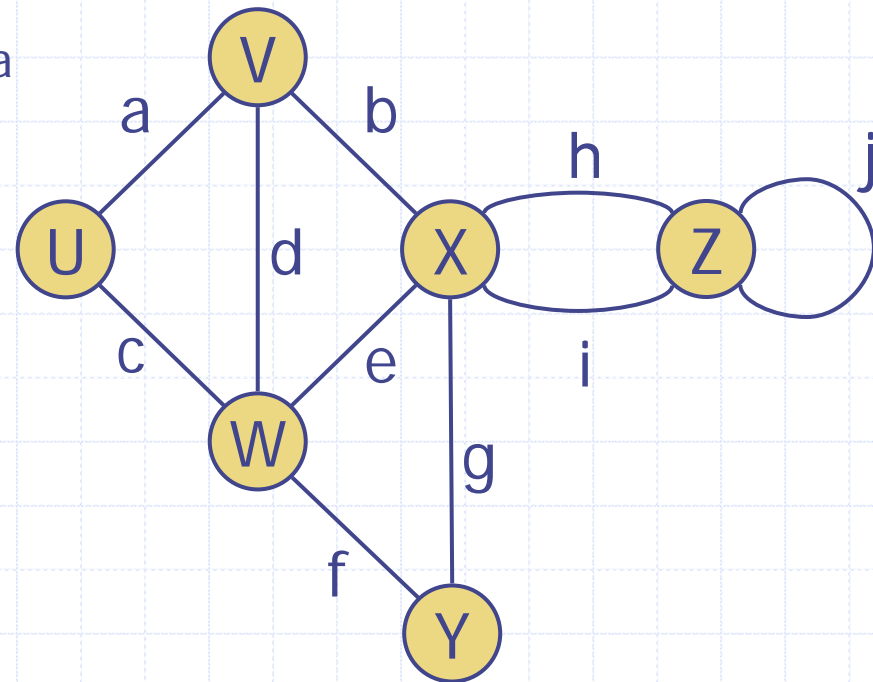
# Applications

- ◆ Electronic circuits
  - Printed circuit board
  - Integrated circuit
- ◆ Transportation networks
  - Highway network
  - Flight network
- ◆ Computer networks
  - Local area network
  - Internet
  - Web
- ◆ Databases
  - Entity-relationship diagram



# Terminology

- ◆ End vertices (or endpoints) of an edge
  - U and V are the endpoints of a
- ◆ Edges incident on a vertex
  - a, d, and b are incident on V
- ◆ Adjacent vertices
  - U and V are adjacent
- ◆ Degree of a vertex
  - X has degree 5
- ◆ Parallel edges
  - h and i are parallel edges
- ◆ Self-loop
  - j is a self-loop



# Terminology (cont.)

## ◆ Path

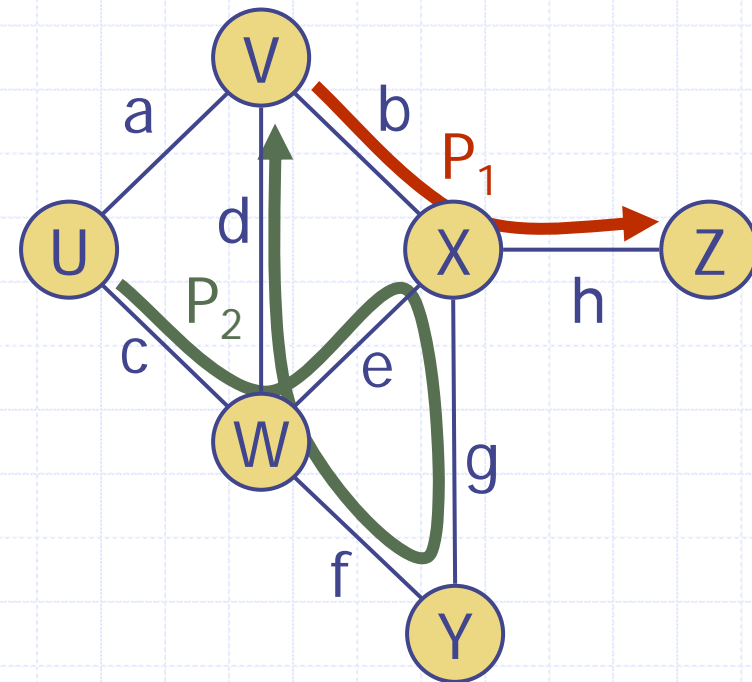
- sequence of alternating vertices and edges
- begins with a vertex
- ends with a vertex
- each edge is preceded and followed by its endpoints

## ◆ Simple path

- path such that all its vertices and edges are distinct

## ◆ Examples

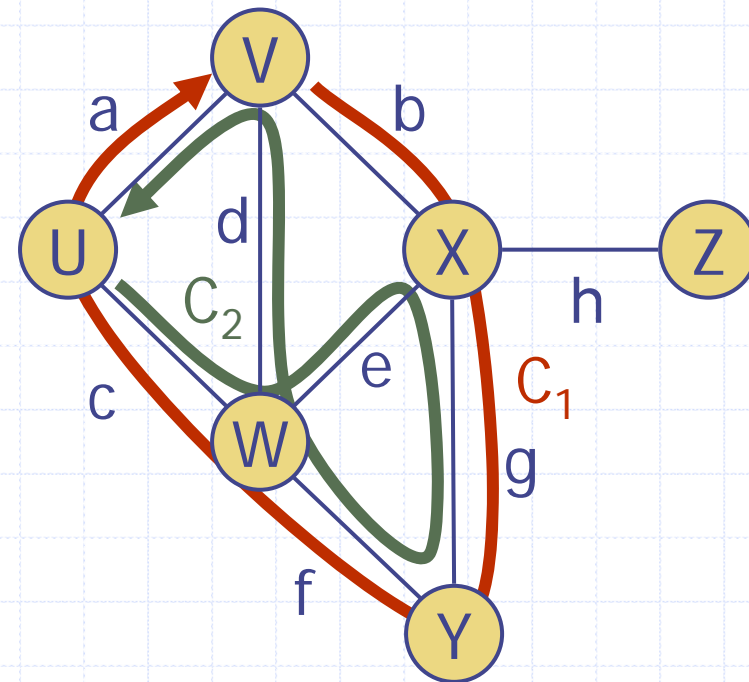
- $P_1 = (V, b, X, h, Z)$  is a simple path
- $P_2 = (U, c, W, e, X, g, Y, f, W, d, V)$  is a path that is not simple





# Terminology (cont.)

- ◆ Cycle
  - circular sequence of alternating vertices and edges
  - each edge is preceded and followed by its endpoints
- ◆ Simple cycle
  - cycle such that all its vertices and edges are distinct
- ◆ Examples
  - $C_1 = (V, b, X, g, Y, f, W, c, U, a, \downarrow)$  is a simple cycle
  - $C_2 = (U, c, W, e, X, g, Y, f, W, d, V, a, \downarrow)$  is a cycle that is not simple



# Properties

## Property 1

$$\sum_v \deg(v) = 2m$$

Proof: each edge is counted twice

## Property 2

In an undirected graph with no self-loops and no multiple edges

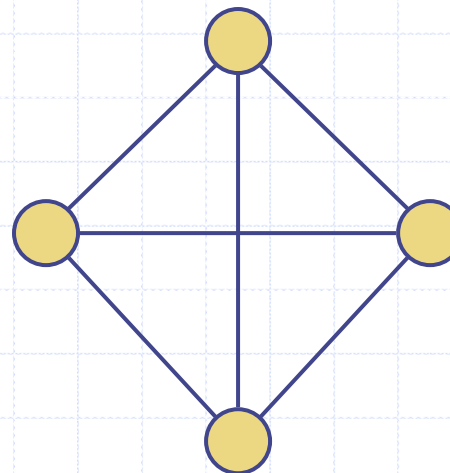
$$m \leq n(n-1)/2$$

Proof: each vertex has degree at most  $(n-1)$

What is the bound for a directed graph?

## Notation

$n$	number of vertices
$m$	number of edges
$\deg(v)$	degree of vertex $v$



## Example

- $n = 4$
- $m = 6$
- $\deg(v) = 3$

# Main Methods of the Graph ADT

## ◆ Vertices and edges

- are positions
- store elements

## ◆ Accessor methods

- `aVertex()`
- `incidentEdges(v)`
- `endVertices(e)`
- `isDirected(e)`
- `origin(e)`
- `destination(e)`
- `opposite(v, e)`
- `areAdjacent(v, w)`

## ◆ Update methods

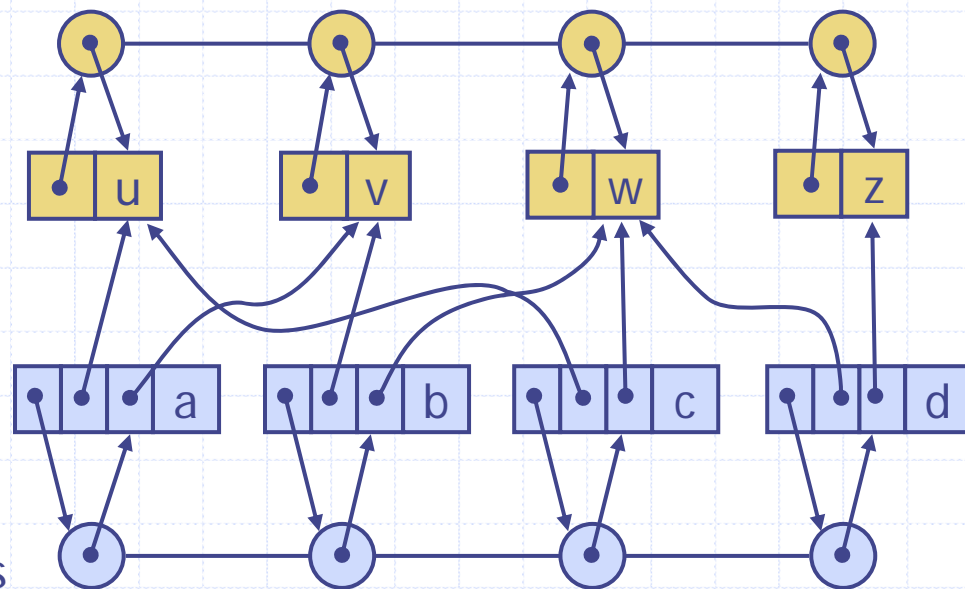
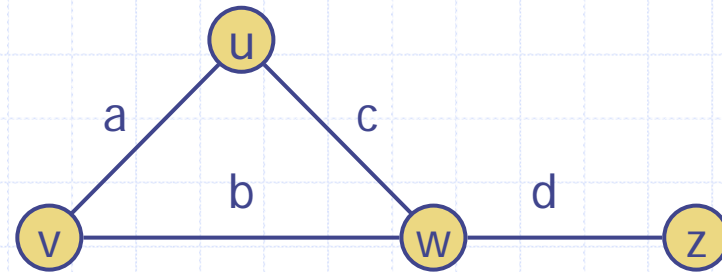
- `insertVertex(o)`
- `insertEdge(v, w, o)`
- `insertDirectedEdge(v, w, o)`
- `removeVertex(v)`
- `removeEdge(e)`

## ◆ Generic methods

- `numVertices()`
- `numEdges()`
- `vertices()`
- `edges()`

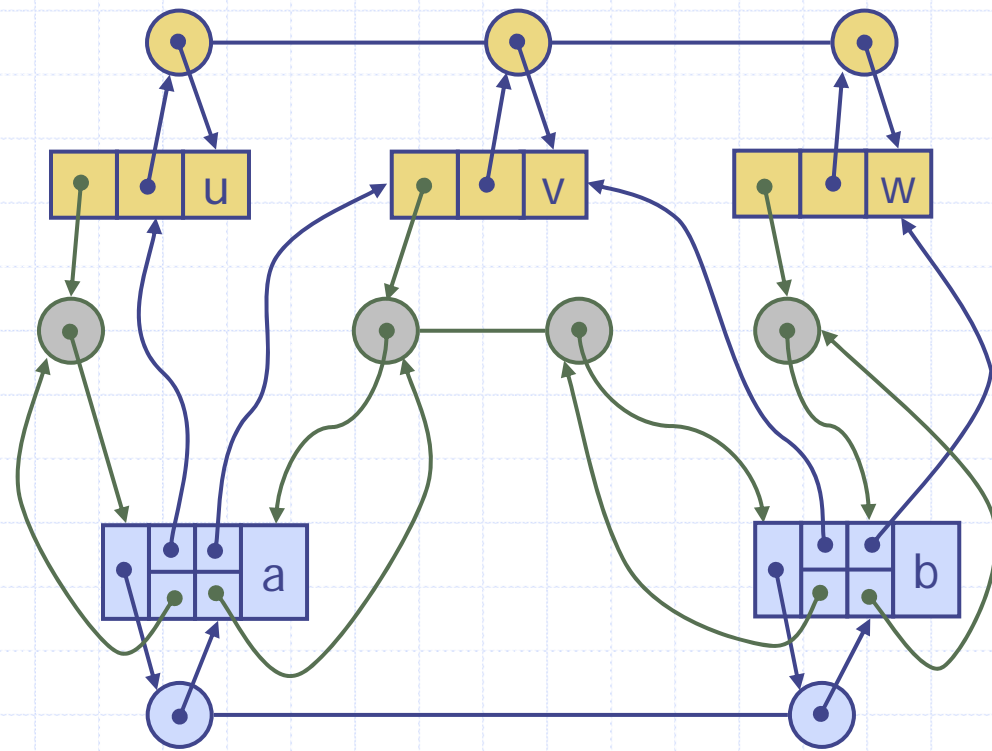
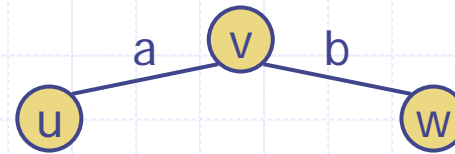
# Edge List Structure

- ◆ Vertex object
  - element
  - reference to position in vertex sequence
- ◆ Edge object
  - element
  - origin vertex object
  - destination vertex object
  - reference to position in edge sequence
- ◆ Vertex sequence
  - sequence of vertex objects
- ◆ Edge sequence
  - sequence of edge objects



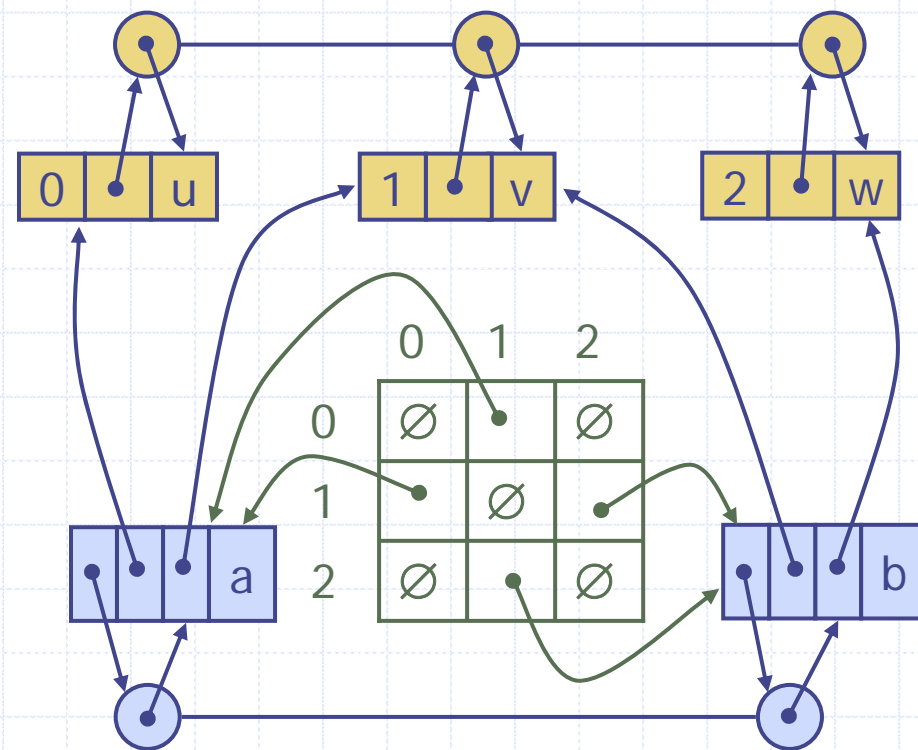
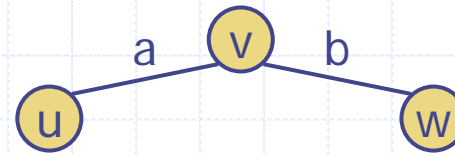
# Adjacency List Structure

- ◆ Edge list structure
- ◆ Incidence sequence for each vertex
  - sequence of references to edge objects of incident edges
- ◆ Augmented edge objects
  - references to associated positions in incidence sequences of end vertices



# Adjacency Matrix Structure

- ◆ Edge list structure
- ◆ Augmented vertex objects
  - Integer key (index) associated with vertex
- ◆ 2D adjacency array
  - Reference to edge object for adjacent vertices
  - Null for non adjacent vertices
- ◆ The “old fashioned” version just has 0 for no edge and 1 for edge



# Asymptotic Performance

<ul style="list-style-type: none"> <li>◆ <math>n</math> vertices, <math>m</math> edges</li> <li>◆ no parallel edges</li> <li>◆ no self-loops</li> <li>◆ Bounds are "big-Oh"</li> </ul>	Edge List	Adjacency List	Adjacency Matrix
Space	$n + m$	$n + m$	$n^2$
<b>incidentEdges</b> ( $v$ )	$m$	$\text{deg}(v)$	$n$
<b>areAdjacent</b> ( $v, w$ )	$m$	$\min(\text{deg}(v), \text{deg}(w))$	1
<b>insertVertex</b> ( $o$ )	1	1	$n^2$
<b>insertEdge</b> ( $v, w, o$ )	1	1	1
<b>removeVertex</b> ( $v$ )	$m$	$\text{deg}(v)$	$n^2$
<b>removeEdge</b> ( $e$ )	1	1	1

# Τέλος Ενότητας



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Κρήτης» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Σημειώματα

# Σημείωμα αδειοδότησης

- Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση, Όχι Παράγωγο Έργο 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/>

- Ως Μη Εμπορική ορίζεται η χρήση:
  - που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
  - που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
  - που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο
- Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

# Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Κρήτης, Ιωάννης Τόλλης 2015. «Αλγόριθμοι και πολυπλοκότητα. Graphs». Έκδοση: 1.0. Ηράκλειο 2015. Διαθέσιμο από τη δικτυακή διεύθυνση:

<https://opencourses.uoc.gr/courses/course/view.php?id=368>

# Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.