



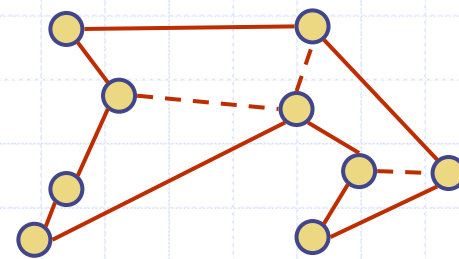
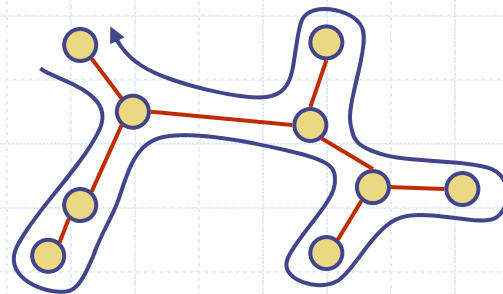
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

# **Αλγόριθμοι και πολυπλοκότητα**

## Περιήγηση Πανεπιστημίων

Ιωάννης Τόλλης  
Τμήμα Επιστήμης Υπολογιστών

# Περιήγηση Πανεπιστημίων



# Περίληψη και ανάγνωση

- ◆ Επισκόπηση της εργασίας
- ◆ αναθεώρηση
  - Δομή πινάκων γειτνίασης
  - Ο MST αλγόριθμος του Kruskal
- ◆ Διαχωρισμός των ADT και των εφαρμογών
- ◆ Το σχέδιο των διακοσμητών
- ◆ Το πρόβλημα των περιοδευόντων πωλητών
  - Καθορισμός
  - Αλγόριθμος προσέγγισης

# Εργασία Γράφου

## ◆ Σκοπός

- Να μάθετε και να εφαρμόσετε τη δομή γειτνίασης πινάκων έναν minimum spanning tree αλγόριθμο του Kruskal.
- Κατανόηση και χρήση του σχεδίου διακοσμητών και ποικίλων JDSL κλάσεων και interfaces( διεπαφών).

## ◆ Η αποστολή σας

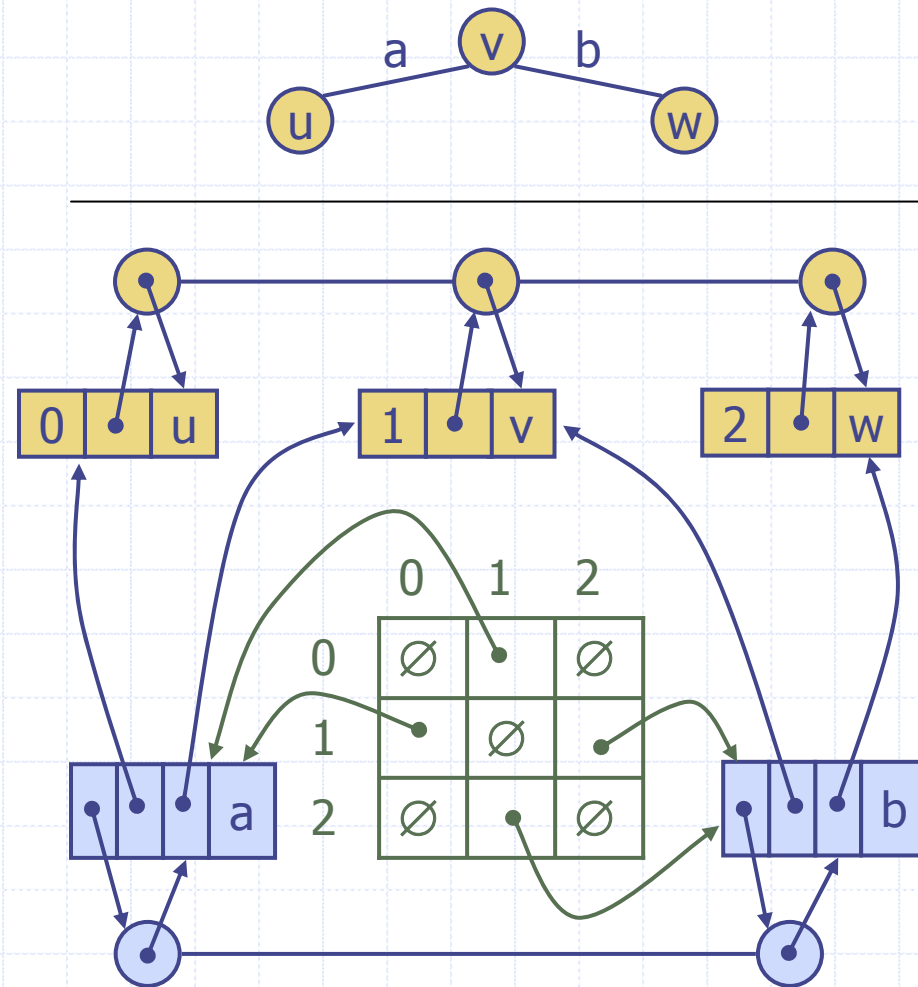
- Η υλοποίηση της δομής πινάκων γειτνίασης για την αναπαράσταση γράφου.
- Η υλοποίηση του MST αλγορίθμου του Kruskal

## ◆ Frontend

- Υπολογισμός και απεικόνιση μιας προσέγγισης γύρου περιοδευόντων πωλητών(traveling salesperson tour)

# Δομή γειτνίασης πινάκων

- ◆ Η δομή της λίστας ακμών
- ◆ Επαυξημένα αντικείμενα κορυφών
  - Το ακέραιο κλειδί (index) σχετίζεται με την κορυφή
- ◆ Ο δισδιάστατος πίνακας γειτνίασης
  - Αναφέρεται στο αντικείμενο ακμή για γειτονικές ακμές
  - Είναι Null για μη γειτονικές ακμές



# Ο αλγόριθμος του Kruskal

- ◆ Οι κορυφές διαχωρίζονται σε σύννεφα
  - Αρχίζουμε με ένα σύννεφο ανά κορυφή
  - Τα σύννεφα συγχωνεύονται κατά τη διάρκεια της εκτέλεσης του αλγορίθμου
- ◆ Διαχωρισμός του ADT:
  - **makeSet**(*o*): δημιουργεί ένα σύνολο {*o*} και επιστρέφει έναν εντοπιστή( locator) για το αντικείμενο *o*
  - **find**(*l*): επιστρέφει το σύνολο του αντικειμένου με εντοπιστή(locator) *l*
  - **union**(*A*,*B*): συγχώνευση συνόλων *A* και *B*

## Algorithm **KruskalMSF**(*G*)

**Input** βεβαρημένος γράφος *G*

**Output** ονομασία(labeling) των ακμών of a minimum spanning forest of *G*

*Q* ← νέα ουρά προτεραιότητας  
βασισμένη σε στοίβα(hear-based)

**for all** *v* ∈ *G.vertices*() **do**

*l* ← **makeSet**(*v*) { στοιχειώδες  
    σύννεφο }

**setLocator**(*v*,*l*)

**for all** *e* ∈ *G.edges*() **do**

*Q.insert*(*weight*(*e*), *e*)

**while** ¬*Q.isEmpty*()

*e* ← *Q.removeMin*()

    [*u*,*v*] ← *G.endVertices*(*e*)

*A* ← **find**(*getLocator*(*u*))

*B* ← **find**(*getLocator*(*v*))

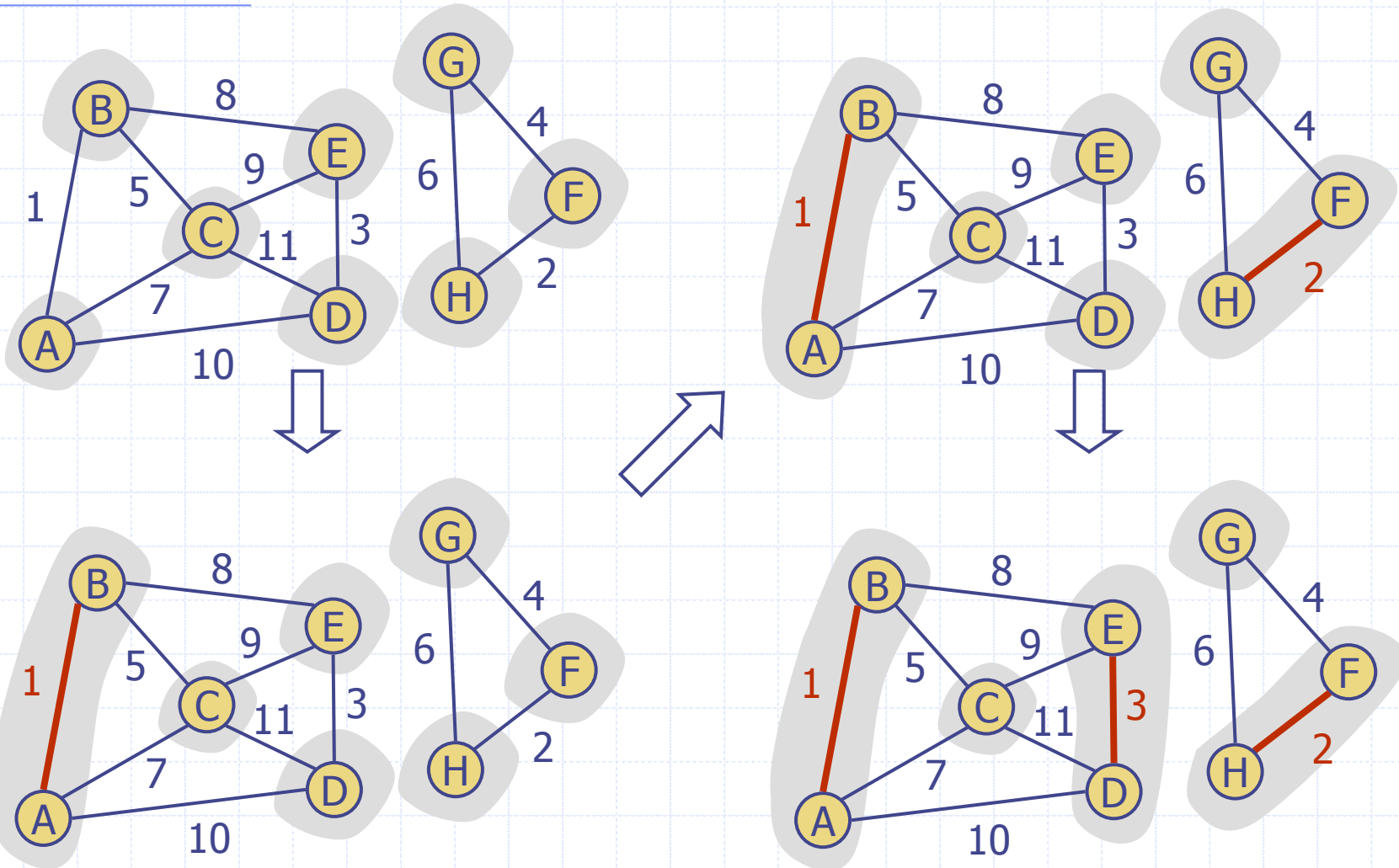
**if** *A* ≠ *B*

**setMSFE**dge(*e*)

        { ταξινόμηση σύννεφων }

**union**(*A*, *B*)

# Παράδειγμα

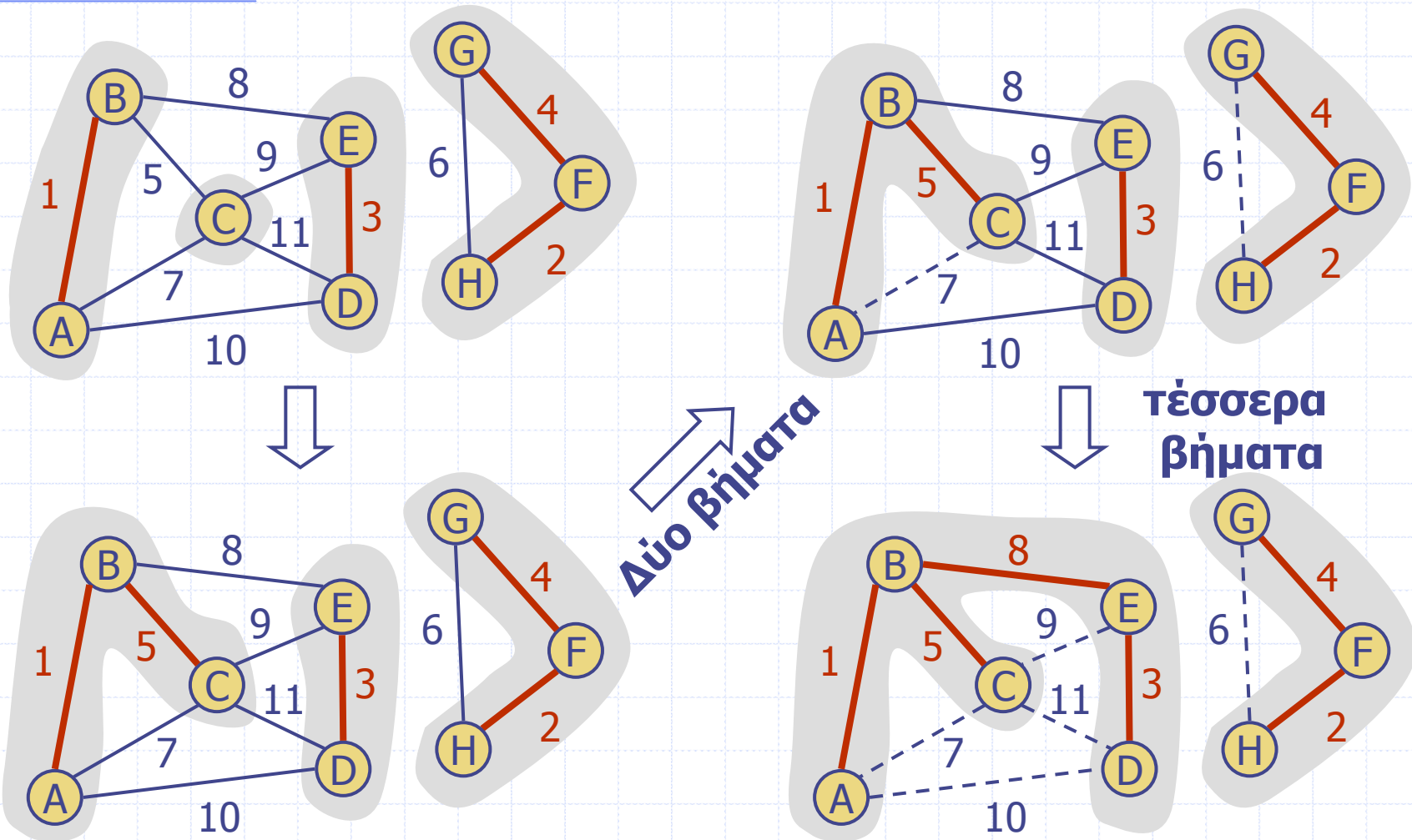


5/8/2008 11:46 AM

Campus Tour

6

# Παράδειγμα (συνέχεια)





# Partition Implementation

## ◆ Εφαρμογή διαχωρισμού

- Ένα σύνολο αντιπροσωπεύεται από την ακολουθία των στοιχείων του
- Μία θέση αποθηκεύει μία αναφορά πίσω στην ίδια την ακολουθία (για την λειτουργία *find*)
- Η θέση ενός στοιχείου στην ακολουθία εξυπηρετεί σαν εντοπιστής (locator) για το στοιχείο στο σύνολο
- Στη λειτουργία *union(ένωση)*, μετακινούμε τα στοιχεία της μικρότερης ακολουθίας σε μεγαλύτερη ακολουθία

## ◆ Χείριστη περίπτωση χρόνων εκτέλεσης

- *makeSet, find*:  $O(1)$
- *union*:  $O(\min(n_A, n_B))$

## ◆ Amortized ανάλυση

- Σκεφτείτε μία σειρά από  $k$  λειτουργίες Διαχωρισμού ADT που περιλαμβάνουν  $n$  *makeSet* λειτουργίες
- Κάθε φορά μετακινούμε ένα στοιχείο σε μία νέα ακολουθία, το μέγεθος του συνόλου της τουλάχιστον διπλασιάζεται
- Ένα στοιχείο μετακινείται το πολύ  $\log_2 n$  φορές
- Η μετακίνηση ενός στοιχείου παίρνει  $O(1)$  χρόνο
- Ο συνολικός χρόνος για τις σειρές των λειτουργιών είναι  $O(k + n \log n)$

# Ανάλυση του αλγορίθμου του Kruskal

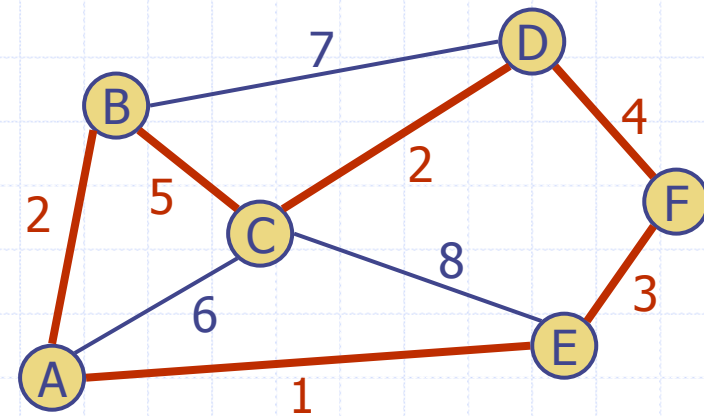
- ◆ Λειτουργίες γράφου
  - Οι Μέθοδοι *vertices* and *edges* καλούνται μία φορά
  - Η μέθοδος *endVertices* καλείται  $m$  φορές
- ◆ Λειτουργίες ουράς προτεραιότητας
  - Παρουσιάζουμε  $m$  *insert* λειτουργίες και  $m$  *removeMin* λειτουργίες
- ◆ Λειτουργίες διαχωρισμού
  - Παρουσιάζουμε  $n$  *makeSet* λειτουργίες,  $2m$  *find* λειτουργίες όχι περισσότερα από  $n - 1$  *union* λειτουργίες
- ◆ Label operations
  - Θέτουμε vertex ετικέτες  $n$  φορές και τις παίρνουμε  $2m$  φορές
- ◆ Ο αλγόριθμος του Kruskal τρέχει σε χρόνο  $O((n + m) \log n)$  παρέχοντας τον γράφο που δεν έχει παράλληλες ακμές και αναπαρίσταται από τη δομή λίστας γειτνίασης

# Σχέδιο Διακοσμητών

- ◆ Οι ετικέτες χρησιμοποιούνται κατά κανόνα σε αλγόριθμους γράφων
  - Βοηθητικά δεδομένα
  - έξοδος
- ◆ Παραδείγματα
  - DFS: εξερευνημένες/που έχουν επισκεφθεί ετικέτες για ακμές και
  - εξερευνημένες/ forward/back ετικέτες για ακμές
  - Dijkstra και Prim-Jarnik: απόσταση, εντοπιστής(locator), και πατρικές ετικέτες για κορυφές
  - Kruskal: ετικέτα εντοπιστή(locator) για κορυφές και ετικέτες MSF για ακμές
- ◆ Το σχέδιο διακοσμητών επεκτείνει τις μεθόδους του Position ADT για να υποστηρίξει τον χειρισμό των ιδιοτήτων (ετικετών)
  - **has**( $a$ ): ελέγχει αν η θέση έχει την ιδιότητα  $a$
  - **get**( $a$ ): επιστρέφει την τιμή μιας ιδιότητας  $a$
  - **set**( $a, x$ ): θέτει στο  $x$  την τιμή της ιδιότητας  $a$
  - **destroy**( $a$ ): μετακινεί την ιδιότητα  $a$  και την συσχετιζόμενη τιμή της (για λόγους καθαρισμού)
- ◆ Το σχέδιο διακοσμητών μπορεί να υλοποιηθεί ταξινομώντας ένα λεξικό από (ιδιότητες, τιμές) στοιχεία σε κάθε θέση

# Το πρόβλημα των περιοδευόντων πωλητών

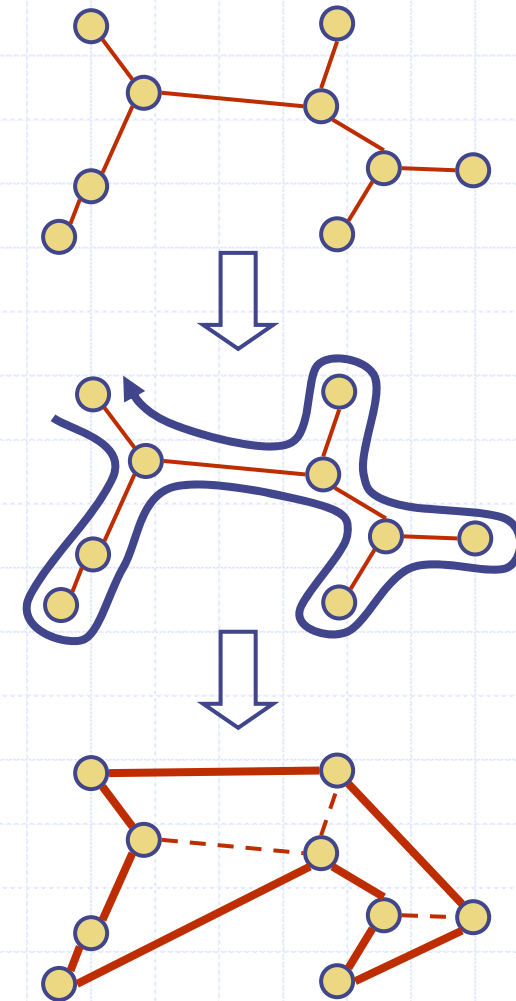
- ◆ Ένας γύρος ενός γράφου είναι ένας spanning cycle (π.χ., ένας κύκλος που πηγαίνει ανάμεσα όλων των κορυφών)
- ◆ Ένας γύρος περιοδευόντων πωλητών ενός βεβαρημένου γράφου είναι ο γύρος που είναι απλός (δηλ., χωρίς επαναλαμβανόμενες κορυφές ή ακμές) και έχει ελάχιστο βάρος
- ◆ Δεν υπάρχουν αλγόριθμοι πολυωνυμικού χρόνου για τον υπολογισμό των γύρων των περιοδευόντων πωλητών
- ◆ Το πρόβλημα των περιοδευόντων πωλητών (TSP) είναι ένα σημαντικό ανοικτό πρόβλημα στην επιστήμη των υπολογιστών
  - Βρείτε έναν αλγόριθμο πολυωνυμικού χρόνου που να υπολογίζει ένα γύρο περιοδευόντων πωλητών ή αποδείξτε ότι κανείς δεν υπάρχει



Παράδειγμα γύρου  
περιοδευόντων πωλητών  
(με βάρος 17)

# Προσέγγιση TSP

- ◆ Μπορούμε να προσεγγίσουμε ένα TSP γύρο με έναν γύρο από το πολύ δύο φορές το βάρος για την περίπτωση των Euclidean γράφων
  - Οι κορυφές είναι σημεία στο επίπεδο
  - Κάθε ζευγάρι κορυφών είναι συνδεδεμένο με μία ακμή
  - Το βάρος μιας ακμής είναι το μήκος του τμήματος που ενώνει τα σημεία
- ◆ Αλγόριθμος προσέγγισης
  - Υπολογίστε ένα ελάχιστο spanning tree
  - Σχηματίστε μία Eulerian κυκλική διαδρομή γύρω από το MST
  - Μετατρέψτε την κυκλική διαδρομή σε ένα γύρο



# 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. Περιήγηση Πανεπιστημίων». 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.