Graph Algorithms

Παρουσίαση στα πλαίσια του μαθήματος
«Παράλληλοι Αλγόριθμοι»

Καούρη Γεωργία
Μήτσου Βασιλική

Περιεχόμενα

- minimum – weight spanning tree
- connected components
- transitive closure
- shortest paths
- maximum matching

techniques: pointer jumping, matrix multiplication

Minimum - Weight Spanning Trees

N – node undirected weighted graph G(V, E)

edges not existing are consider to have infinite weight

w(i, j) denotes the weight of edge (i, j)

Lemma: Consider G as above with unique edge weights and let U be any subset of the nodes of G. Then the minimum weight edge linking a node in U to a node in V – U is in the minimum – weight spanning tree for G.

Consequence: the minimum – weight edge incident to any node of G is in the minimum – weight spanning tree for G.

Algorithm

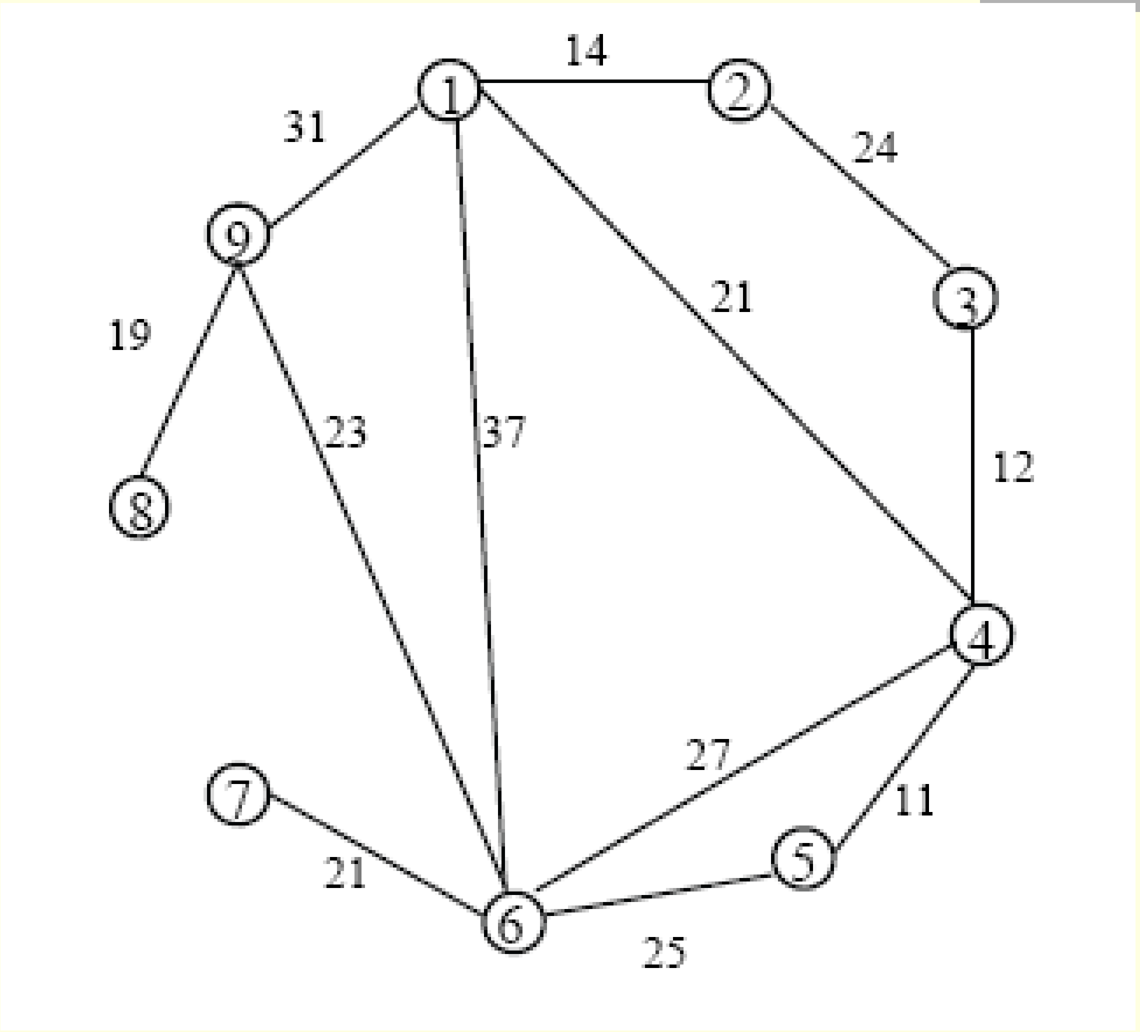
Για κάθε κόμβο βρίσκουμε την ακμή μικρότερου βάρους που ενώνει τον κόμβο με τους υπόλοιπους κόμβους και την προσθέτουμε στο minimum spanning tree.

Κατασκευάζουμε supernodes: Για κάθε συνεκτική συνιστώσα του γράφου που έχουμε κατασκευάσει μέχρι τώρα κατασκευάζουμε έναν supernode.

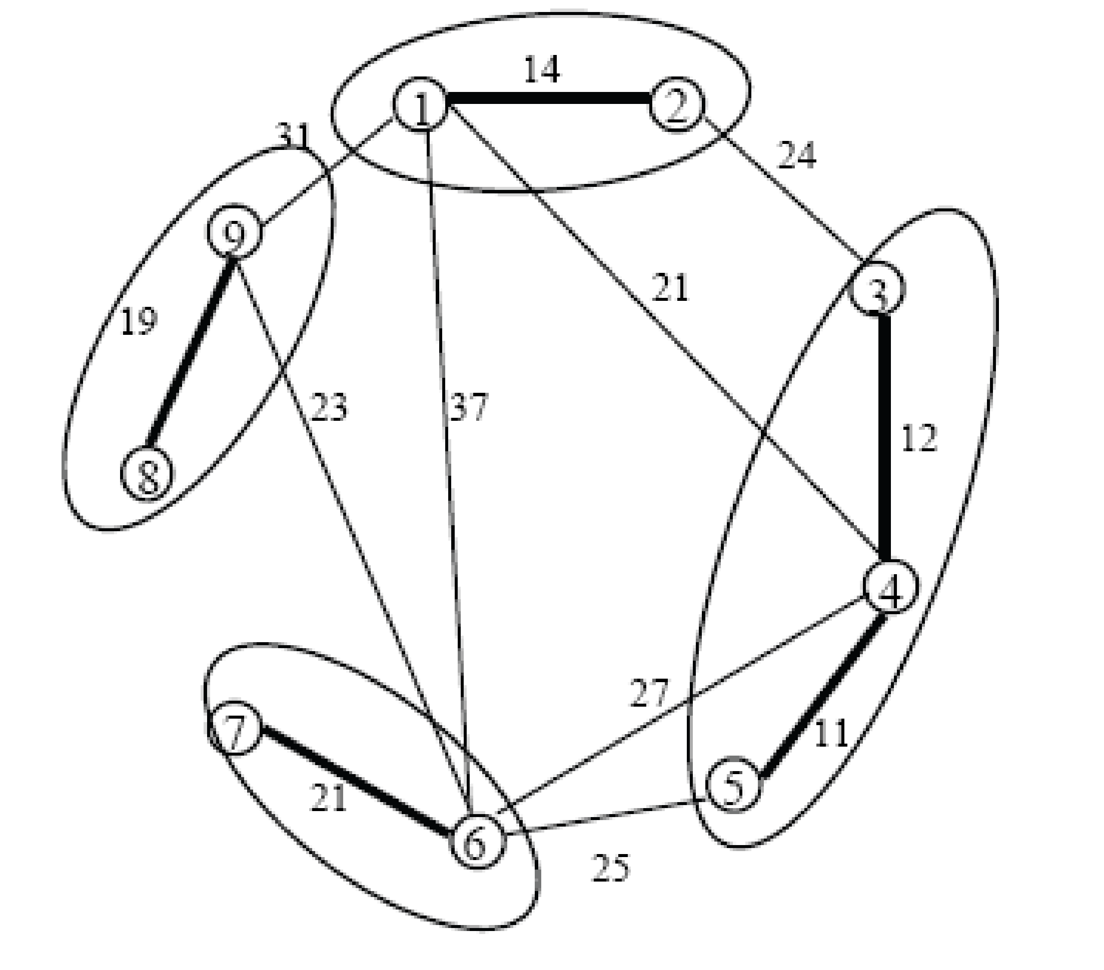
Επαναλαμβάνουμε τη διαδικασία μέχρι να μείνουμε με έναν supernode.

Ο γράφος που προκύπτει μετά από κάθε επανάληψη έχει το πολύ το μισό αριθμό κόμβων, άρα απαιτούνται το πολύ logN βήματα.

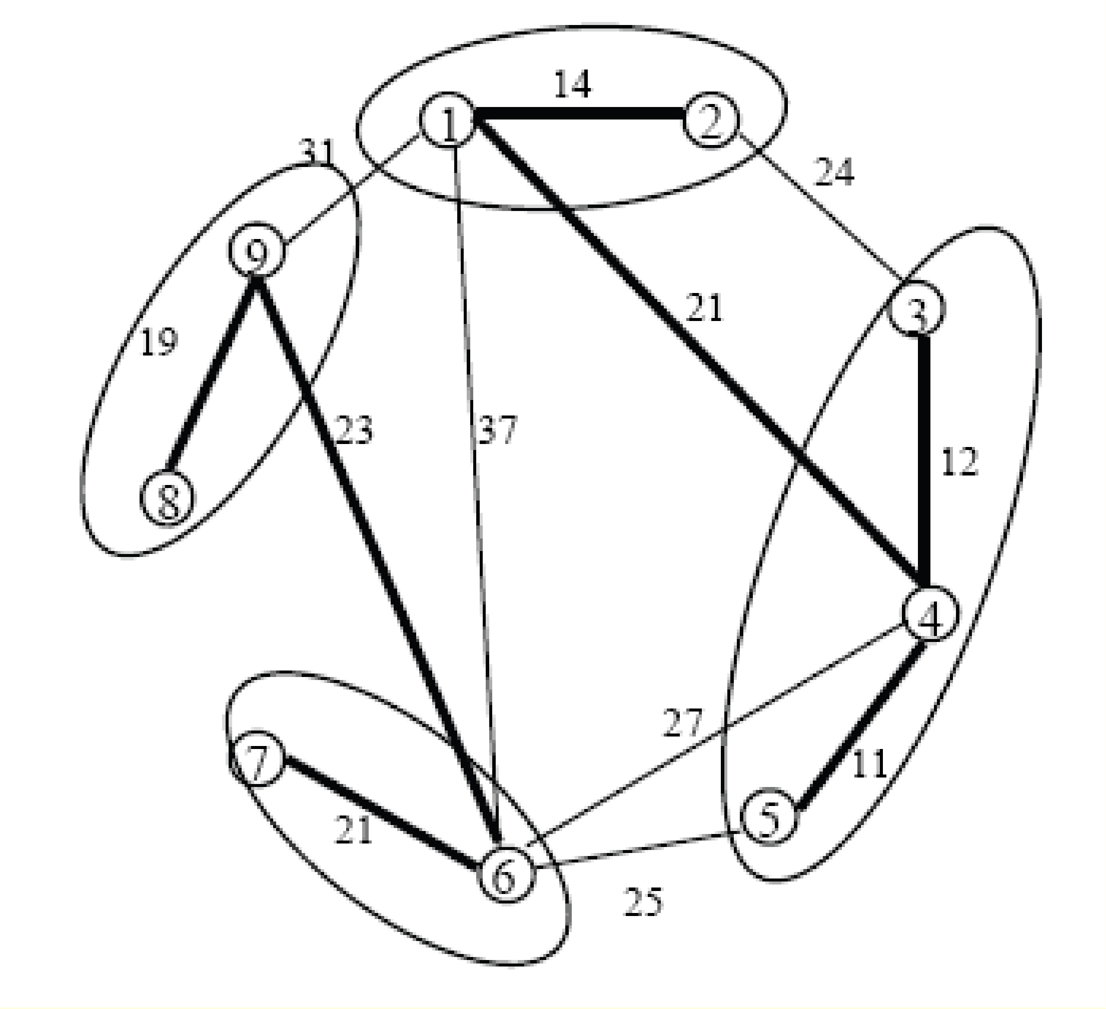
Example (initially)



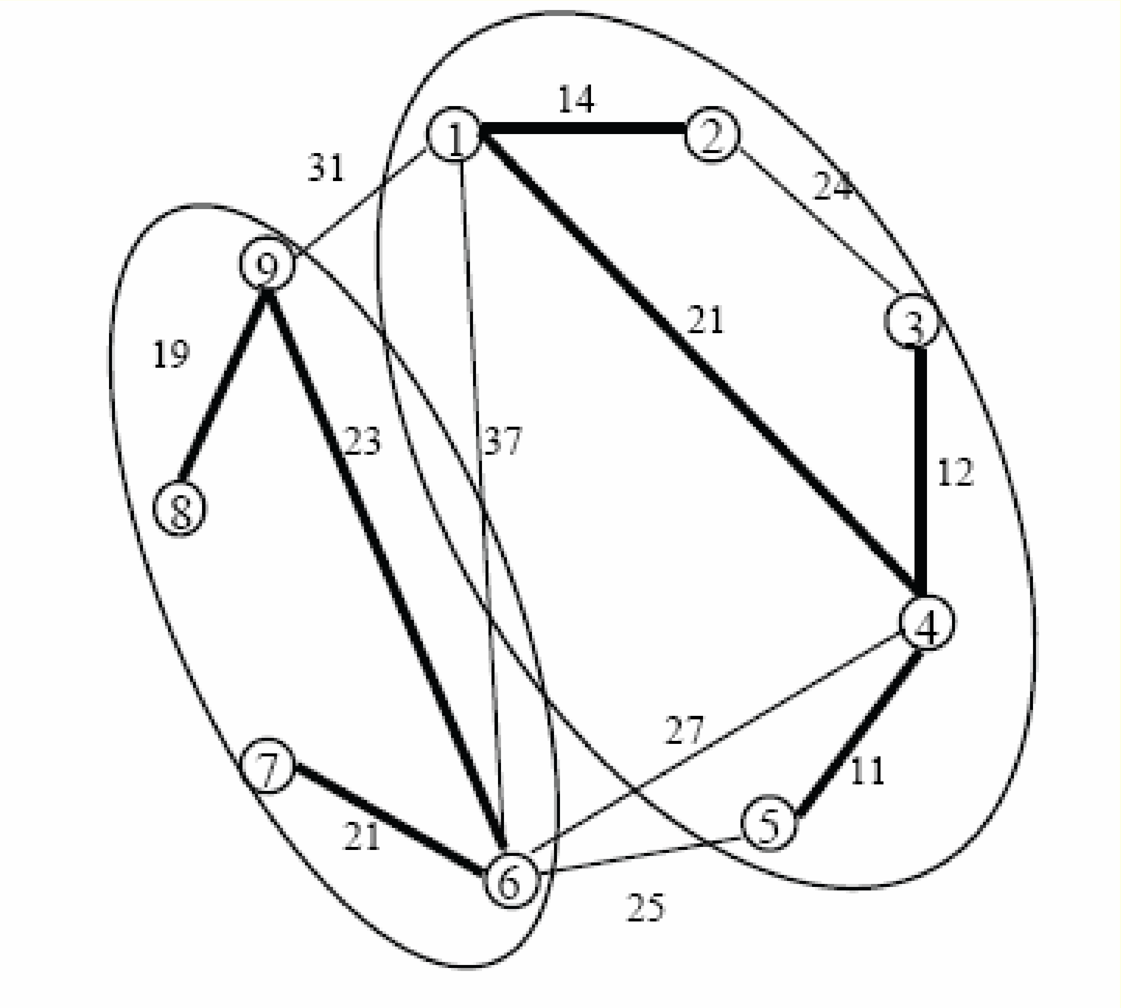
Example (step1)



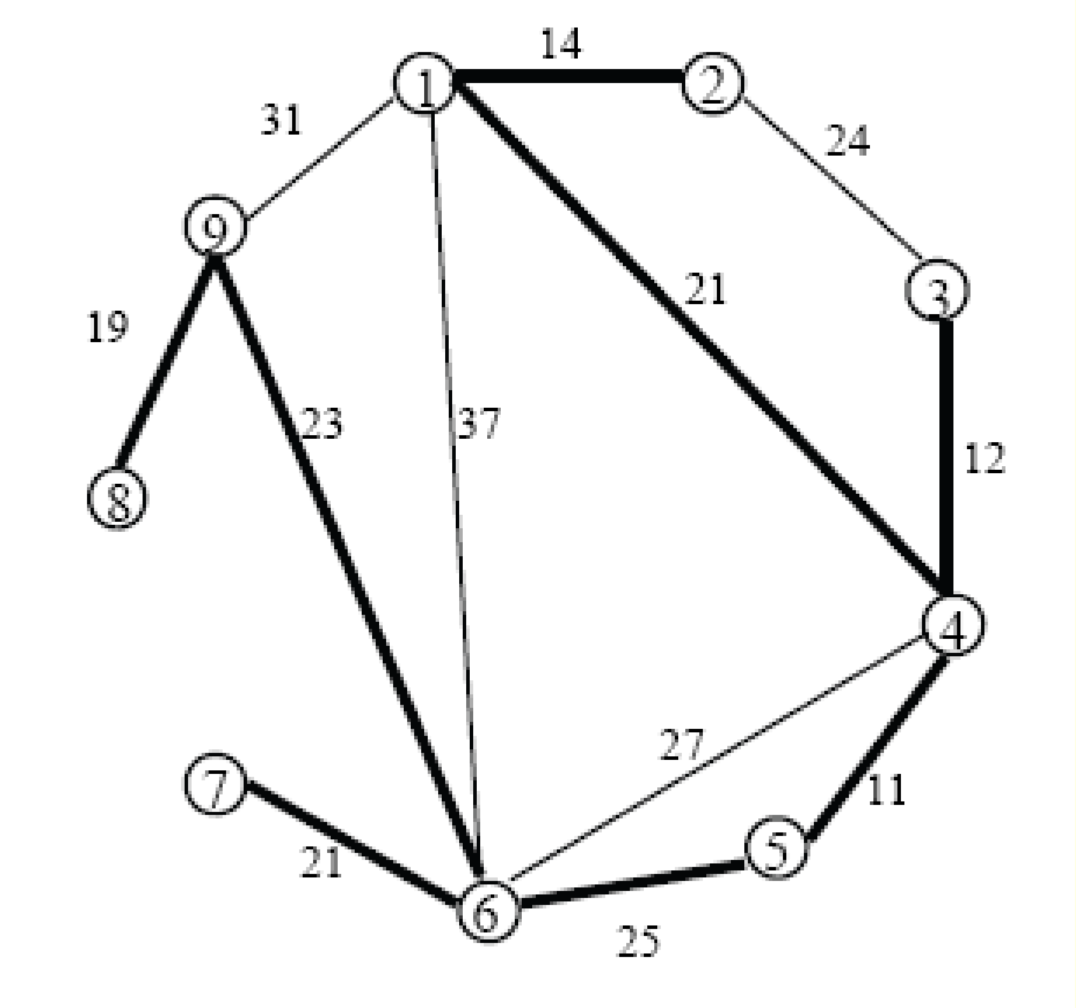
Example (step 2a)



Example (step 2)



Example (step 3)



Παράλληλη υλοποίηση

- Χρησιμοποιούμε ένα N x N mesh of trees αποθηκεύοντας τα βάρη των ακμών στα φύλλα.
- Μετά από κάθε επανάληψη κατασκευάζουμε για κάθε κόμβο ένα label L(i) ώστε δύο κόμβοι να έχουν το ίδιο label αν και μόνο αν ανήκουν στον ίδιο supernode.
- Κάθε supernode έχει έναν κόμβο που χρησιμεύει σαν leader, δηλαδή σαν το όνομα του supernode. Αρχικά κάθε κόμβος είναι leader του εαυτού του.

1^{η} φάση: Βρίσκουμε ακμή ελαχίστου βάρους

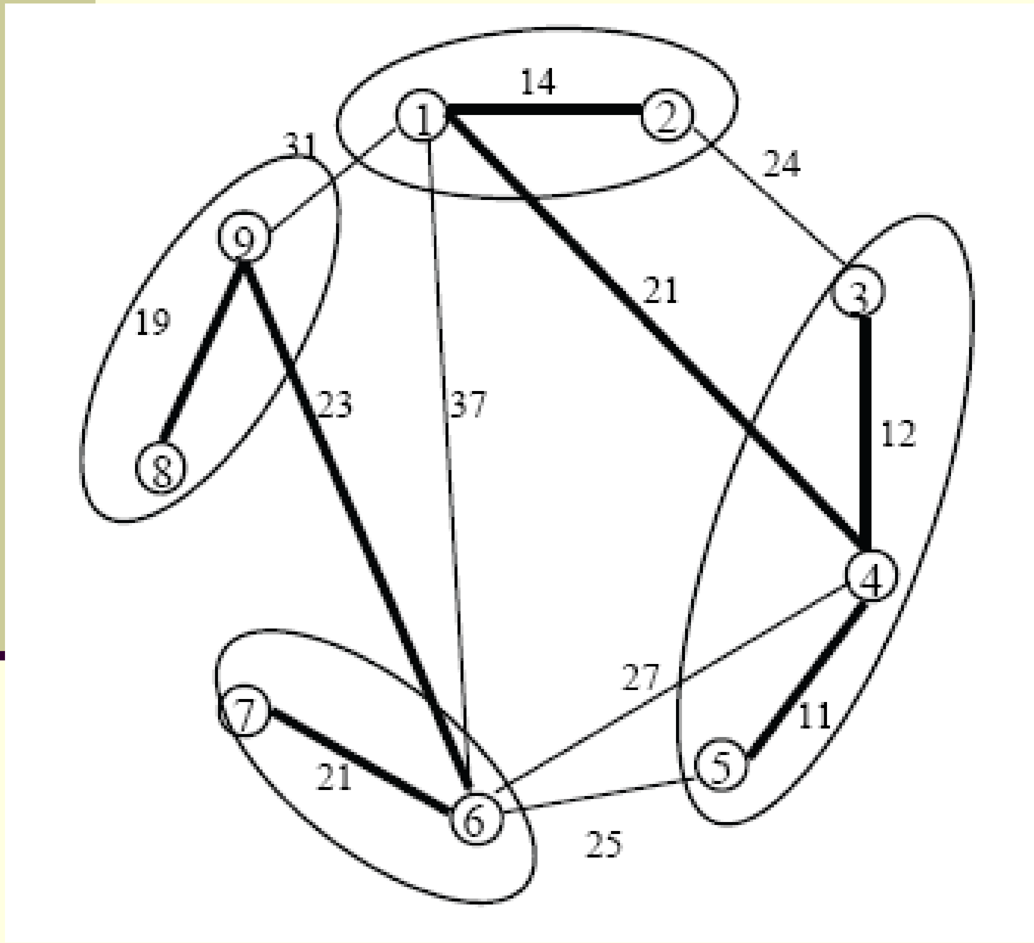
Η διαδικασία αυτή απαιτεί για κάθε κόμβο i του supernode να βρούμε ακμή ελαχίστου βάρους που συνδέει τον κόμβο i με κόμβο j για κάθε j με L(i)≠L(j). Στη συνέχεια για όλους τους κόμβους του supernode βρίσκουμε την ελάχιστη τέτοια ακμή.

Ουσιαστικά πρέπει να υπολογίσουμε την εξίσωση:

$$w(i, A(i)) = \min_{1 \le j \le N} \{w(i, j) \mid L(i) \ne L(j)\}$$

Δε μας χρειάζεται να ξέρουμε τον ακριβή κόμβο με τον οποίο ενώνεται ο κάθε supernode, μόνο ποιος είναι ο leader του.

Example



L(1)=1 A(1)=4 L(A(1))=5
w(1, A(1))=21

L(2)=1 A(2)=4 L(A(2))=5
w(2, A(2))=24

L(3)=5 A(3)=2 L(A(3))=1
w(3, A(3))=24

L(4)=5 A(4)=1 L(A(4))=1
w(4, A(4))=21

L(5)=5 A(5)=6 L(A(5))=6
w(5, A(5))=25

L(6)=6 A(6)=9 L(A(6))=9
w(6, A(6))=23

L(7)=6 A(7)=0 L(A(7))=0
w(7, A(7))=∞

Υπολογισμός L(A(i)) και w(i, A(i))

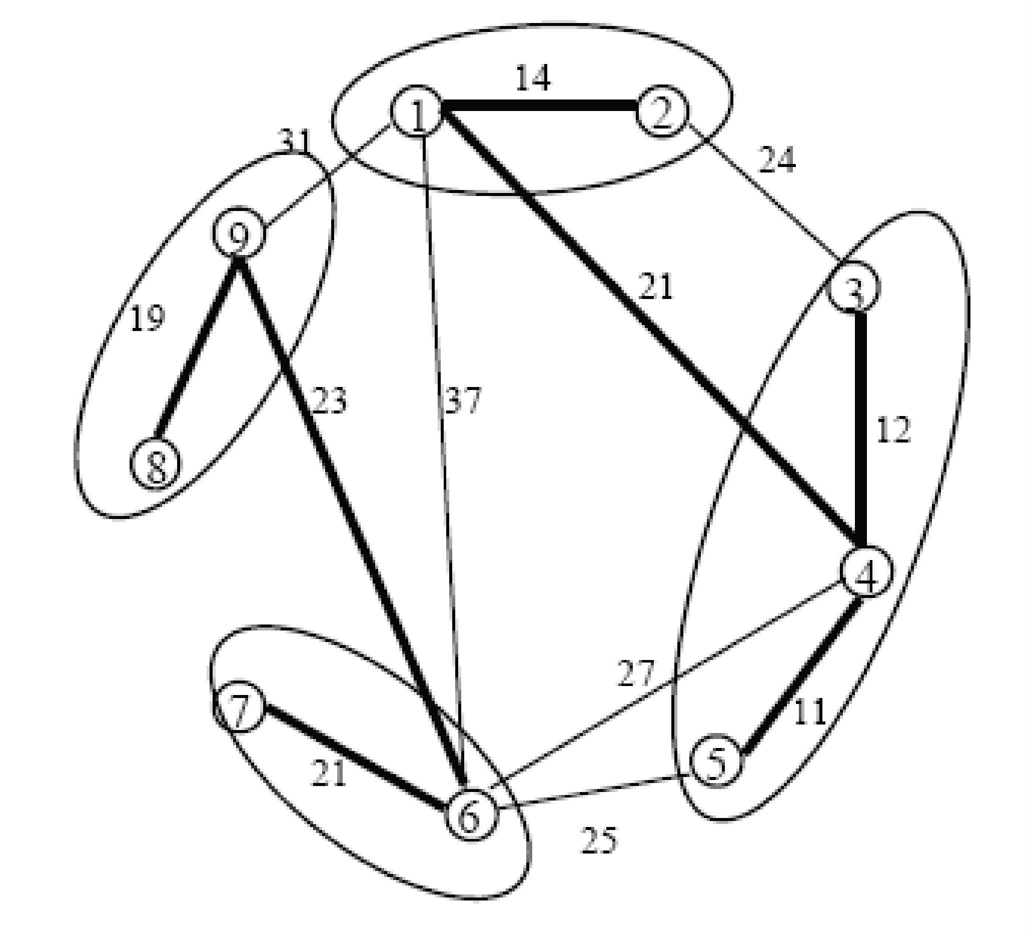
- Στέλνουμε τα L(i) στην i – γραμμή και στήλη του mesh.
- Τα βάρη της κάθε γραμμής ανεβαίνουν προς τα πάνω, ελέγχουμε αν L(i)≠L(j) κι αν ναι, συγκρίνουμε τις τιμές και στέλνουμε πάνω τη μικρότερη.
- Μετά από 2logN βήματα έχει ολοκληρωθεί η διαδικασία, οπότε η τιμή L(A(i)) βρίσκεται πάνω πάνω μαζί με το κατάλληλο ελάχιστο βάρος w(i, A(i)).

Υπολογισμός P(j)

- για κάθε leader κόμβο j πρέπει να υπολογίσουμε το P(j), δηλαδή τον leader του supernode με τον οποίο ο supernode με leader j ενώνεται με ακμή ελαχίστου μήκους
- Οπότε για κάθε leader κόμβο j η τιμή P(j) ισούται με την τιμή L(A(i)) για την οποία

$$W(j) = \min\{w(i, A(i)) \mid L(i) = L(j) = j\}$$

Example



P(1)=5
P(5)=1
P(6)=8
P(8)=6

W(1)=21
W(5)=21
W(6)=23
W(8)=23

Υλοποίηση

- Στέλνουμε στα φύλλα τις τιμές των L(A(i)) και w(i, A(i)) που υπολογίσαμε μέσω των γραμμών του mesh και ξανααενεβαίνουν από τις στήλες όπου γίνεται και η σύγκριση αν L(i)=L(j).
- Η τιμή P(j) είναι η τιμή του L(A(i)) που φτάνει στη ρίζα της j στήλης.
- Απαιτούνται συνολικά 2logN βήματα ακόμα.

Κατασκευή επόμενων supernodes (Περιγραφή)

- Κάθε supernode δείχνει στον supernode με τον οποίο ενώνεται με ακμή ελαχίστου βάρους. Δηλαδή, ο supernode j δείχνει τον supernode P(j).
- Ο γράφος που σχηματίζεται από τους παραπάνω δείκτες είναι ακυκλικός, εκτός από έναν 2-κύκλο που υπάρχει και εκφράζει αυτήν ακριβώς την ακμή ελαχίστου βάρους που προστίθεται στο minimum – weight spanning tree.
- Ο νέος leader θα είναι ένας από τους δύο κόμβους που ανήκουν στον ίδιο 2 – κύκλο και συγκεκριμένα ο κόμβος με μικρότερο index.

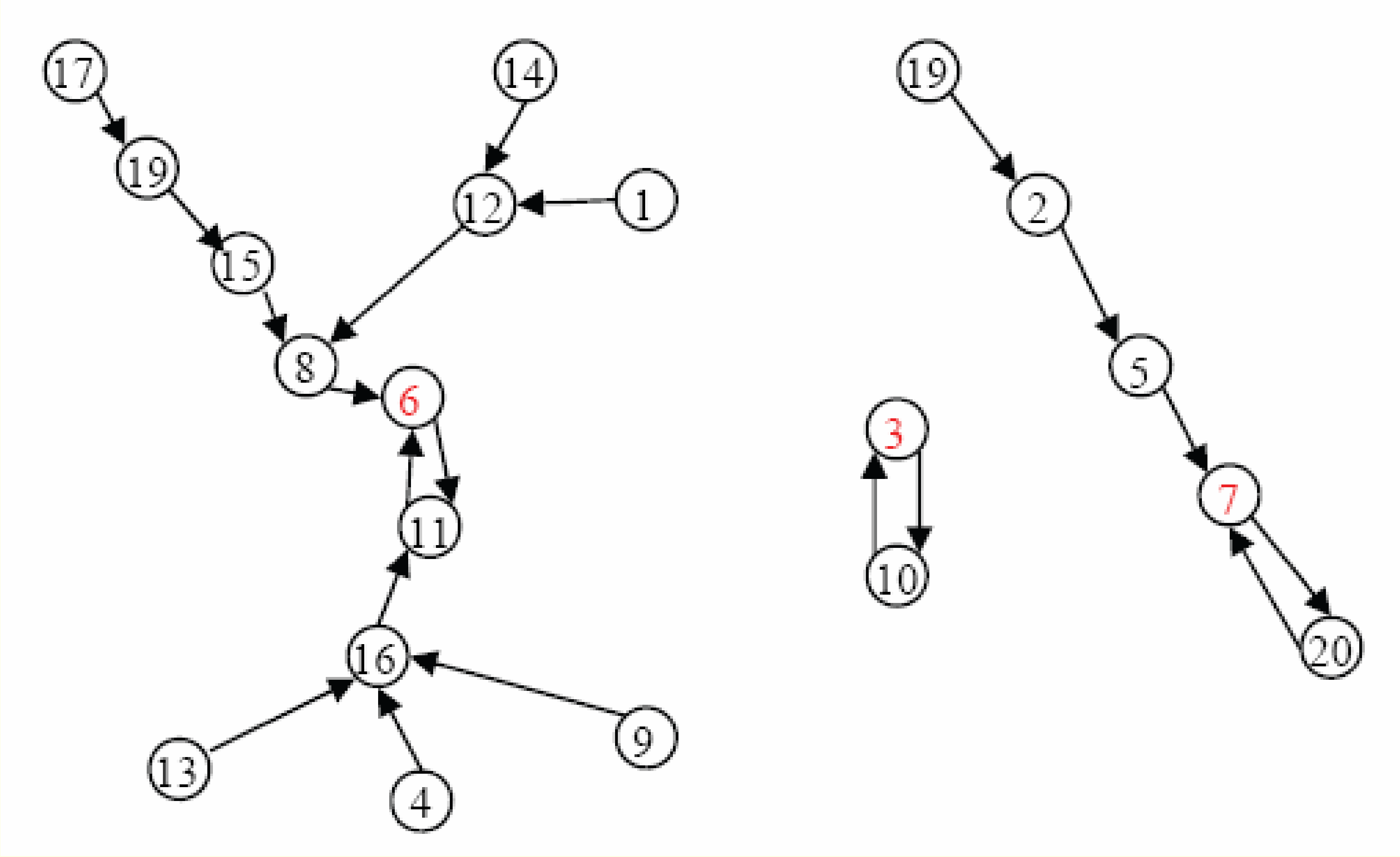
Κατασκευή επόμενων supernodes (Περιγραφή)

- Ο δείκτης που είχαμε βάλει μετακινείται ώστε να δείχνει στο νέο leader του supernode.
- Οι άλλοι δείκτες μετακινούντται ώστε να δείχνουν τον leader του «πατέρα» τους (ως προς leader).

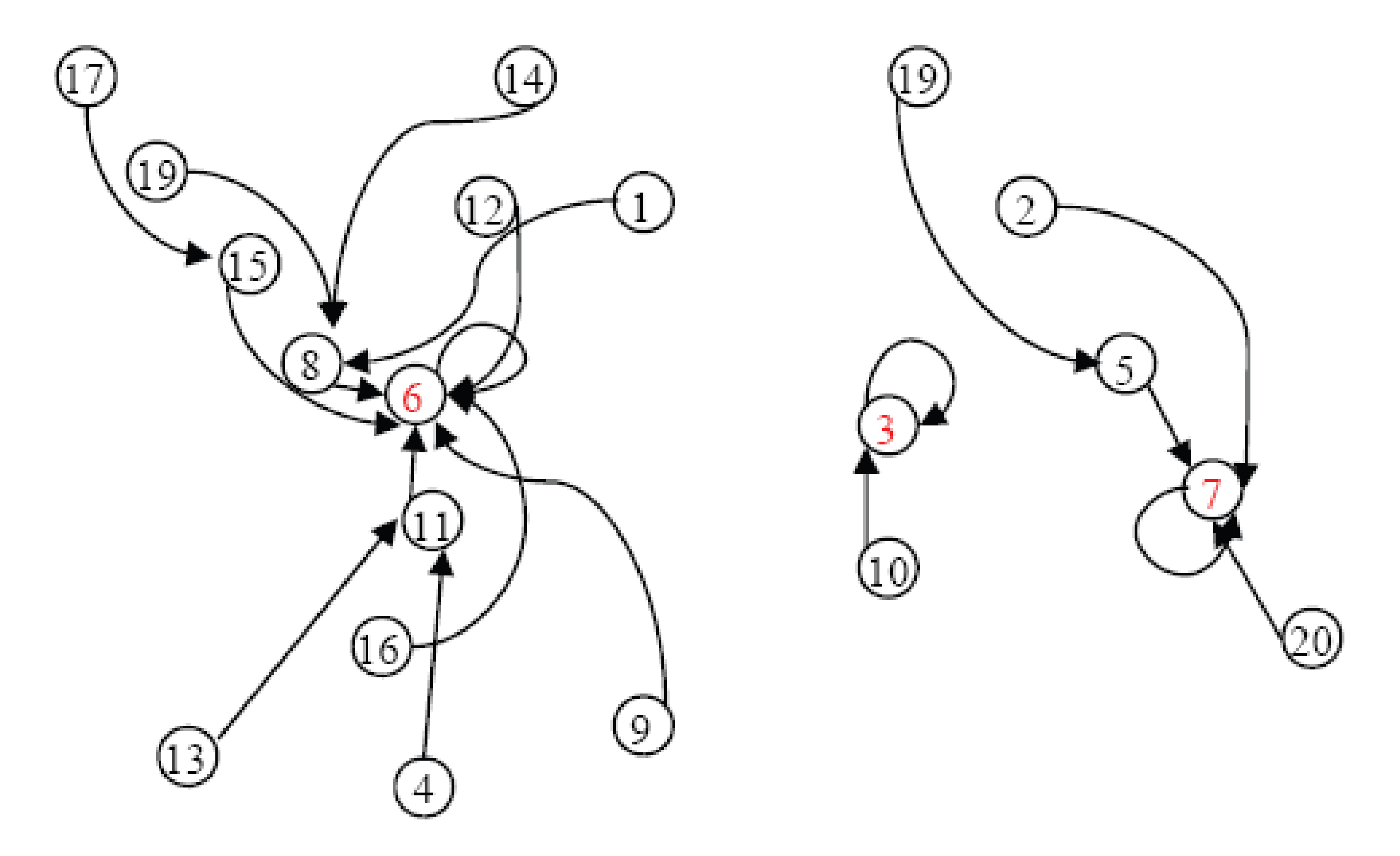
  Δηλαδή $P(j) \leftarrow P(P(j))$ (pointer jumping)

  Αυτό ολοκληρώνεται σε $2\log^2 N$ επαναλήψεις.
- Τέλος $L(j) \leftarrow P(L(j))$ για κάθε κόμβο.

Example



Example



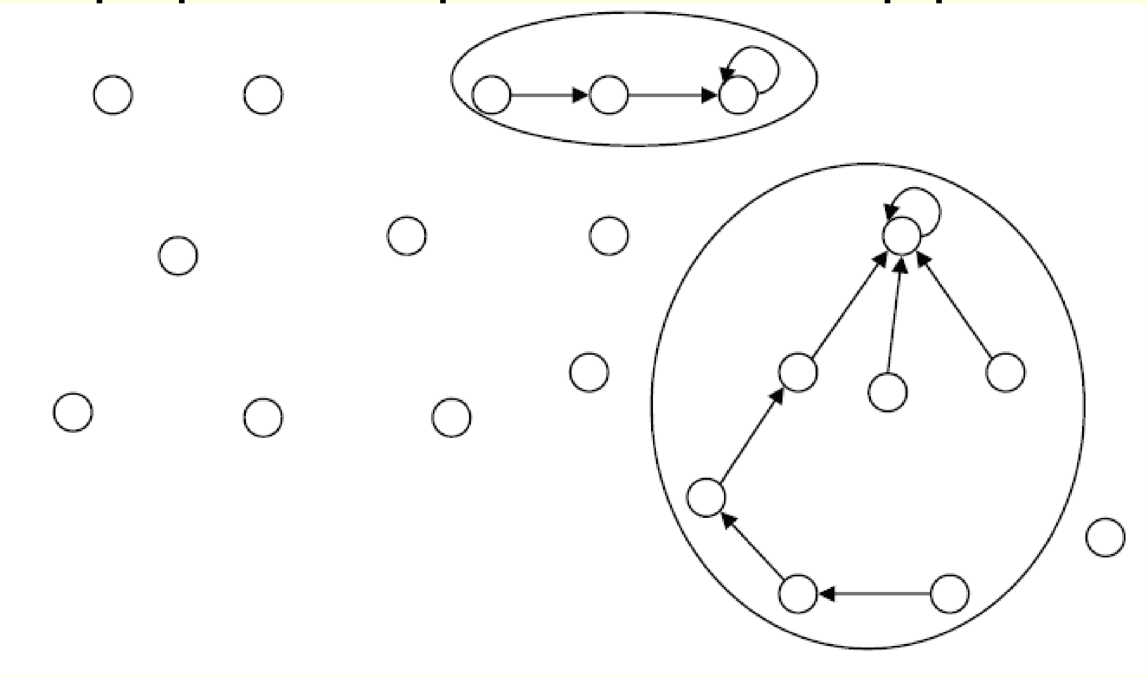
Υλοποίηση $Z(j) \leftarrow X(Y(j))$

- Οι τιμές των X(i) στέλνονται από τις ρίζες στα φύλλα της i – οστής γραμμής.
- Στις στήλες επιλέγονται τα Y(j) και όταν φτάνει το X υπολογίζεται το X(Y(j)).
- Η τιμή X(Y(j)) στέλνεται στη ρίζα, οπότε η διαδικασία έχει ολοκληρωθεί.
- Απαιτούνται 2logN βήματα.

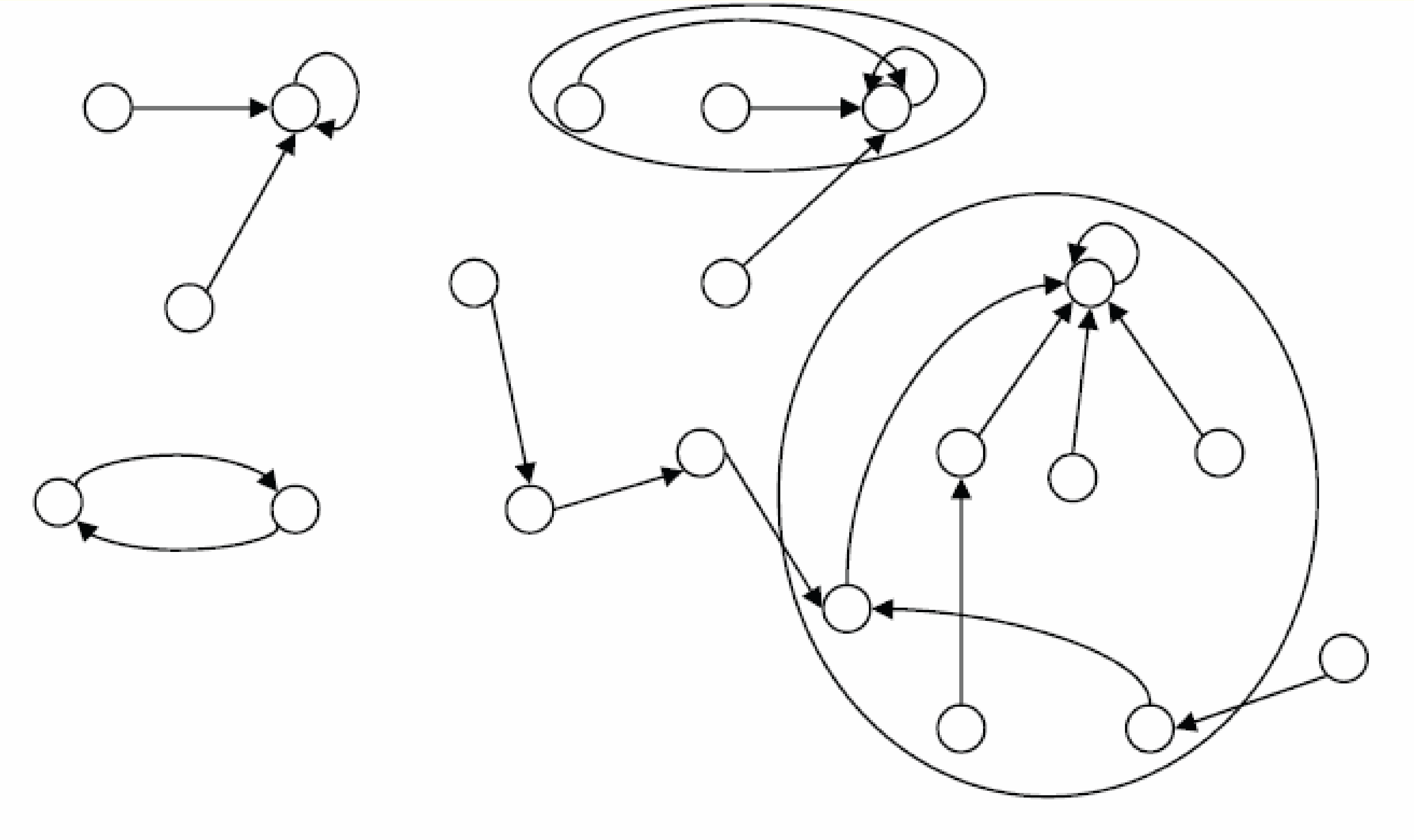
- **Συνολικά** για όλη τη φάση απαιτούνται $2\log^2 N+O(\log N)$ επαναλήψεις και επειδή έχουμε συνολικά logN βήματα χρειαζόμαστε $2\log^3 N+O(\log^2 N)$ βήματα.

Speeding up

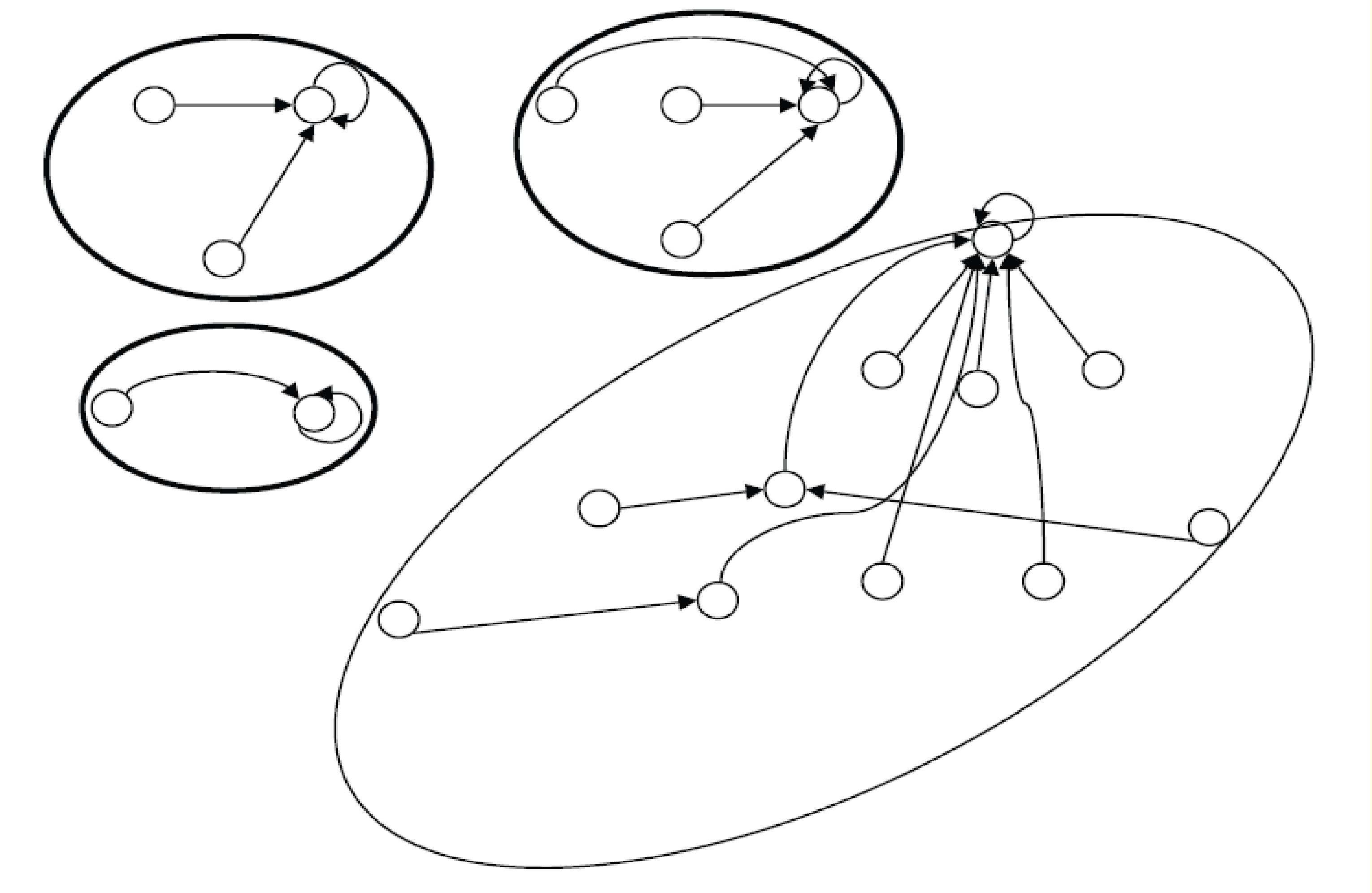
- Η καθυστέρηση στον αλγόριθμο που μόλις περιγράψαμε οφείλεται στη φάση της κατασκευής των νέων supernodes. Η διαδικασία καθυστερεί όταν υπάρχουν «μικρά» components, όπου η διαδικασία ολοκληρώνεται γρήγορα, και ταυτόχρονα «μεγάλα» που καθυστερούν, οπότε τα «μικρά» θα πρέπει να τα «περιμένουν».



Speeding up



Speeding up



Speeding up

- Με την παραπάνω διαδικασία είναι εύκολο να δούμε οτι μειώνονται κατά logN οι επαναλήψεις, αφού οι μικροί supernodes δεν περιμένουν τους μεγάλους για να συνεχίσουν την πορεία τους προς το τέλος.

- Επομένως τα βήματα μειώνονται και τελικά δε χρειάζονται πάνω από $O(\log^2 N)$ βήματα.

Connected Components

Για να βρούμε τις συνεκτικές συνιστώσες ενός γραφήματος, μπορούμε να εφαρμόσουμε τον προηγούμενο αλγόριθμο για την εύρεση minimum – weight spanning tree με κάποιες μικρές αλλαγές. Αυτές είναι:

1. Στην ακμή (i, j) ανατίθεται βάρος Ni + j, αν αυτή υπάρχει.
2. Στην ακμή (i, j) ανατίθεται βάρος ∞ αν δεν υπάρχει.
3. Ο αλγόριθμος δεν προσθέτει ακμές άπειρου βάρους στο spanning tree.
4. Αν δεν υπάρχουν ακμές με βάρος μη άπειρο να προστεθούν στο spanning tree, ο αλγόριθμος τερματίζει.

Connected Components

- Οι συνεκτικές συνιστώσες προκύπτουν μετά από $O(\log^2 N)$ βήματα σαν οι supernodes που απομένουν όταν ο αλγόριθμος δεν μπορεί να προσθέσει άλλες ακμές στα spanning trees που έχουν κατασκευαστεί.

- Μια σημαντική εφαρμογή που μπορεί να βρει αυτός ο αλγόριθμος, είναι στην επεξεργασία εικόνας (παράγραφος 1.8), αφού είναι ιδιαίτερα χρήσιμος όταν προσπαθούμε να συνθέσουμε αντικείμενα.