ΔΑΚΤΥΛΙΟΙ

Σε ένα *προσανατολισμένο* δακτύλιο, οι επεξεργαστές έχουν προσδιορίσει κατά συνεπή τρόπο την *αριστερή* και *δεξιά* πλευρά τους.

Για παράδειγμνα, αν τα μηνύματα διοχετεύονται πάντοτε στο κανάλι 1, θα κάνουν τότε κύκλους επ' άπειρο στο δακτύλιο.

Γιατί μελετάμε τους δακτύλιους;

- Είναι καλό σημείο αφετηρίας, μπορούμε να τους αναλύουμε με σχετική ευκολία.
- Αποτελούν μια αφαιρετικότητα του δακτυλίου με κουπόνια.
- Κάτω φράγματα για την τοπολογία του δακτυλίου συνεπάγονται συχνά κάτω φράγματα για αυθαίρετες τοπολογίες.

ΕΚΛΟΓΗ ΠΡΟΕΔΡΟΥ

- Κάθε επεξεργαστής έχει ένα σύνολο από καταστάσεις *προεδρίας* και ένα σύνολο από καταστάσεις *μη προεδρίας*.
  - Όταν ένας επεξεργαστής μπει σε κατάσταση προεδρίας, θα μείνει σε κατάσταση προεδρίας.
  - Όταν ένας επεξεργαστής μπει σε κατάσταση μη προεδρίας, θα μείνει σε κατάσταση μη προεδρίας.

  Οι αποφάσεις είναι *μη αντιστρέψιμες*.

- Σε κάθε *νόμιμη* εκτέλεση:
  - Κάθε επεξεργαστής μπαίνει κάποτε σε μια κατάσταση προεδρίας ή σε μια κατάσταση μη προεδρίας (συνθήκη ζωτικότητας).
  - Ακριβώς ένας επεξεργαστής (ο *πρόεδρος*) μπαίνει σε κατάσταση προεδρίας (συνθήκη ασφαλείας).

Ένας πρόεδρος μπορεί να χρησιμοποιηθεί για το συντονισμό μελλοντικών δραστηριοτήτων του συστήματος, π.χ.,

- για την κατασκευή γεννητορικού δένδρου, με τον πρόεδρο ως ρίζα.
- για την ανακατασκευή ενός χαμένου κουπονιού σε ένα δακτύλιο με κουπόνια.

Θα μελετήσουμε το πρόβλημα της εκλογής προέδρου στο δακτύλιο.

ΑΝΩΝΥΜΟΙ ΔΑΚΤΥΛΙΟΙ

Διαίσθηση: οι επεξεργαστές δεν έχουν μοναδικές ταυτότητες.

Ένα σχετικό ζήτημα είναι κατά πόσο ο αλγόριθμος βασίζεται στη γνώση από τους επεξεργαστές του μεγέθους του δακτυλίου.

- Ομοιόμορφος αλγόριθμος: δεν χρησιμοποιεί το μέγεθος του δακτυλίου
  - $\Rightarrow$ ο ίδιος αλγόριθμος δουλεύει για κάθε μέγεθος δακτυλίου.

  Τυπικά, κάθε επεξεργαστής σε οποιουδήποτε μεγέθους δακτύλιο παριστάνεται από την ίδια μηχανή καταστάσεων A.

- Μη ομοιόμορφος αλγόριθμος: χρησιμοποιεί το μέγεθος του δακτυλίου
  - $\Rightarrow$ διαφορετικός αλγόριθμος για κάθε δακτύλιο διαφορετικού μεγέθους. Τέτοιοι αλγόριθμοι μπορεί να έχουν τετριμμένες μόνο διαφορές.

  Τυπικά, για κάθε τιμή του n, υπάρχει μια μηχανή καταστάσεων A_n τέτοια ώστε κάθε επεξεργαστής σε ένα δακτύλιο μεγέθους n παριστάνεται από τη μηχανή A_n. Έτσι, ο αλγόριθμος A είναι η συλλογή όλων των A_n.

ΕΚΛΟΓΗ ΠΡΟΕΔΡΟΥ ΣΕ ΑΝΩΝΥΜΟΥΣ ΔΑΚΤΥΛΙΟΥΣ

Θεώρημα: Δεν υπάρχει αλγόριθμος εκλογής προέδρου σε ανώνυμους δακτύλιους, ακόμη και αν ο αλγόριθμος είναι μη ομοιόμορφος και ο δακτύλιος είναι σύγχρονος.

Απόδειξη:

- Κάθε επεξεργαστής βρίσκεται αρχικά στην ίδια κατάσταση και τα ίδια μηνύματα είναι αρχικά μετεπεβιβαζόμενα.
- Κάθε επεξεργαστής λαμβάνει τα ίδια μηνύματα και έτσι ακολουθεί την ίδια μετάβαση κατάστασης και στέλνει τα ίδια μηνύματα στο γύρο 1.
- Κάθε επεξεργαστής λαμβάνει τα ίδια μηνύματα και έτσι ακολουθεί την ίδια μετάβαση κατάστασης και στέλνει τα ίδια μηνύματα στο γύρο 2.
- κ.ο.κ.

Τελικά, κάποιος επεξεργαστής μπαίνει σε κατάσταση προεδρίας. $\Rightarrow$ όλοι οι επεξεργαστές θα μπούν τότε σε κατάσταση προεδρίας λόγω της διατήρησης της ίδιας κατάστασης. Αντίφαση. $\square$

Κατά συνέπεια, δεν θα υπάρχει ούτε ομοιόμορφος ή ασύγχρονος αλγόριθμος εκλογής προέδρου σε ανώνυμους δακτύλιους.

ΔΑΚΤΥΛΙΟΙ ΜΕ ΤΑΥΤΟΤΗΤΕΣ

Υποθέτουμε ότι κάθε επεξεργαστής διαθέτει μια μοναδική ταυτότητα.

Διάκριση μεταξύ δεικτών και ταυτοτήτων:

- Οι *δείκτες* είναι ακέραιοι από 0 μέχρι $n-1$, οι οποίοι δεν είναι διαθέσιμοι στους επεξεργαστές. Τους χρησιμοποιούμε εμείς μόνο για τους σκοπούς της ανάλυσής μας.
- Οι *ταυτότητες* είναι αυθαίρετοι, μη αρνητικοί ακέραιοι, οι οποίοι είναι διαθέσιμοι στους επεξεργαστές μέσω μιας ειδικής συνιστώσας της κατάστασης, που καλείται AT.

Προσδιορίζουμε ένα δακτύλιο με ταυτότητες αρχίζοντας με το μικρότερο AT και απαριθμώντας τους AT κατά την ωρολογιακή φορά.

- <u>Ομοιόμορφος αλγόριθμος:</u> υπάρχει μια μηχανή καταστάσεων για κάθε ταυτότητα, ανεξάρτητα από το μέγεθος του δακτυλίου.
- <u>Μη ομοιόμορφος αλγόριθμος:</u> υπάρχει μια μηχανή καταστάσεων για κάθε ταυτότητα και για κάθε μέγεθος δακτυλίου.

ΕΚΛΟΓΗ ΠΡΟΕΔΡΟΥ ΣΕ ΔΑΚΤΥΛΙΟ ΜΕ ΤΑΥΤΟΤΗΤΕΣ

Σ' αυτή την περίπτωση, υπάρχουν αλγόριθμοι. Θα τους αξιολογήσουμε αναφορικά με την πολυπλοκότητα μηνυμάτων τους.

Επισκόπηση:

- ασύγχρονος δακτύλιος: $\Theta(n \lg n)$ μηνύματα
- σύγχρονος δακτύλιος:
  - $\Theta(n)$ μηνύματα κάτω από κάποιες προυποθέσεις
  - αλλοιώς, $\Theta(n \lg n)$ μηνύματα

Όλα τα παραπάνω φράγματα είναι ασυμπτωτικά βέλτιστα.

ΑΛΓΟΡΙΘΜΟΣ ΕΚΛΟΓΗΣ ΠΡΟΕΔΡΟΥ ΜΕ $\Theta(n^2)$ ΜΗΝΥΜΑΤΑ

Κάθε επεξεργαστής ακολουθεί τους εξής κανόνες:

- Αρχικά, στείλε τον ΑΤ σου προς τα αριστερά.
- Κάθε φορά που λαμβάνεις ένα ΑΤ από αριστερά:
  - αν είναι μεγαλύτερος από το δικό σου ΑΤ, τότε προώθησέ τον προς τα αριστερά (δεν θα γίνεις ποτέ πρόεδρος)
  - αν είναι ίσος με το δικό σου ΑΤ, τότε αυτοανακηρύξου ως πρόεδρος
  - αν είναι μικρότερος από το δικό σου ΑΤ, τότε μην κάνεις τίποτα

ΑΛΓΟΡΙΘΜΟΣ ΕΚΛΟΓΗΣ ΠΡΟΕΔΡΟΥ ΜΕ $\Theta(n^2)$ ΜΗΝΥΜΑΤΑ (Συνέχεια)

Ορθότητα: θα εκλεγεί ως πρόεδρος ο αρχηγός με το μεγαλύτερο ΑΤ. Το μήνυμα με αυτό τον ΑΤ θα περάσει διά μέσου κάθε επεξεργαστή.

Πολυπλοκότητα μηνυμάτων: εξαρτάται από τη διάταξη των ταυτοτήτων.

- Ο μέγιστος ΑΤ θα ταξιδέψει γύρω από ολόκληρο το δακτύλιο $\Rightarrow n$ μηνύματα
- Ο δεύτερος μέγιστος ΑΤ θα ταξιδέψει μέχρι να συναντήσει το μέγιστο.
- Ο τρίτος μέγιστος ΑΤ θα ταξιδέψει μέχρι να συναντήσει το μέγιστο ή το δεύτερο μέγιστο.
- κ.ο.κ.

ΑΛΓΟΡΙΘΜΟΣ ΕΚΛΟΓΗΣ ΠΡΟΕΔΡΟΥ ΜΕ $\Theta(n^2)$ ΜΗΝΥΜΑΤΑ (συνέχεια)

Η χειρότερη διάταξη των ταυτοτήτων είναι σε φθίνουσα τάξη κατα την ωρολογική φορά.

- Ο δεύτερος μέγιστος ΑΤ συνεισφέρει $n-1$ μηνύματα.
- Ο τρίτος μέγιστος ΑΤ συνεισφέρει $n-2$ μηνύματα.
- κ.ο.κ.

Ο συνολικός αριθμός μηνυμάτων είναι

$$\begin{aligned} \sum_{i=1}^{n}(n-i+1) &= \sum_{i=1}^{n} i \\ &= \Theta(n^2) \end{aligned}$$

ΑΛΓΟΡΙΘΜΟΣ ΕΚΛΟΓΗΣ ΠΡΟΕΔΡΟΥ ΜΕ $\Theta(n \lg n)$ ΜΗΝΥΜΑΤΑ

- Κάθε επεξεργαστής προσπαθεί να κυριαρχήσει σε διαδοχικά μεγαλύτερες γειτονιές γύρω του, στέλνοντας κατάλληλα *σήματα εξερεύνησης*. Το μέγεθος της γειτονιάς διπλασιάζεται σε κάθε νέα *φάση*.
- Κάθε σήμα εξερεύνησης περιέχει τον ΑΤ του επεξεργαστή που το έστειλε και άλλες πληροφορίες (π.χ., η απόσταση που πρέπει να διανύσει).
- Αν ένας επεξεργαστής λάβει ένα σήμα με ΑΤ
  - μεγαλύτερο από το δικό του, τότε είτε θα το προωθήσει είτε θα στείλει πίσω επιβεβαίωση (αν το σήμα δεν χρειάζεται να διανύσει και άλλη απόσταση).
  - μικρότερο από το δικό του, τότε δεν κάνει τίποτα.
  - ίσο με το δικό του, τότε αυτοανακηρύσσεται ως πρόεδρος.
- Αν ένας επεξεργαστής πάρει μια επιβεβαίωση που δεν προορίζεται για τον ίδιο, την προωθεί.
- Αν ένας επεξεργαστής λάβει δύο επιβεβαιώσεις, τότε προχωρεί στην επόμενη φάση.

ΑΛΓΟΡΙΘΜΟΣ ΕΚΛΟΓΗΣ ΠΡΟΕΔΡΟΥ ΜΕ $\Theta(n \lg n)$ ΜΗΝΥΜΑΤΑ (συνέχεια)

- Ορθότητα:
  - όμοια με τον προηγούμενο $\Theta(n^2)$ αλγόριθμο.

- Πολυπλοκότητα Μηνυμάτων:
  - Κάθε μήνυμα ανήκει σε μία φάση και έχει αποσταλεί από ένα συγκεκριμένο επεξεργαστή.
  - Η *απόσταση εξερεύνησης* στη φάση i είναι 2^i, όπου $i \geq 0$.
  - Ο αριθμός των μηνυμάτων που αποστέλλονται *εξ αιτίας* ενός συγκεκριμένου επεξεργαστή στη φάση i είναι το πολύ $4 \cdot 2^i$ (σήματα εξερεύνησης και επιβεβαιώσεις στις δύο κατευθύνσεις).

ΑΛΓΟΡΙΘΜΟΣ ΕΚΛΟΓΗΣ ΠΡΟΕΔΡΟΥ ΜΕ $\Theta(n \lg n)$ ΜΗΝΥΜΑΤΑ (συνέχεια)

- Πόσοι επεξεργαστές αποστέλλουν μηνύματα εξερεύνησης στη φάση i;
  - Για $i = 0$, και οι n επεξεργαστές αποστέλλουν.
  - Για $i > 0$, μόνο οι επεξεργαστές που νίκησαν στη φάση $i - 1$ ($\Longrightarrow$ είχαν το μεγαλύτερο ΑΤ σε μια ακτίνα 2^{i-1} γύρω τους).

- Ο μεγιστός αριθμός νικητών στη φάση $i - 1$ πραγματοποιείται όταν αυτοί εμφανίζονται όσο πιο πυκνά γίνεται.
  - $\Longrightarrow$ Ο συνολικός αριθμός νικητών στη φάση $i - 1$ είναι το πολύ $\frac{n}{2^{i-1}+1}$.

- Πόσες φάσεις υπάρχουν;
  - Το πολύ $\lg n$, αφού κάθε φάση διπλασιάζει την απόσταση του σήματος εξερεύνησης (από 1 μέχρι n).

ΑΛΓΟΡΙΘΜΟΣ ΕΚΛΟΓΗΣ ΠΡΟΕΔΡΟΥ ΜΕ $\Theta(n \lg n)$ ΜΗΝΥΜΑΤΑ (συνέχεια)

Ο συνολικός αριθμός μηνυμάτων είναι

$$
\begin{aligned}
&\leq \; 4 \cdot n + \sum_{i=1}^{\lg n} 4 \cdot 2^i \cdot \frac{n}{2^{i-1}+1} + n \\
&\leq \; 5n + 4n \cdot \sum_{i=1}^{\lg n} \frac{2^i}{2^{i-1}} \\
&= \; 5n + 4n \cdot \sum_{i=1}^{\lg n} 2 \\
&= \; 5n + 8n \lg n \\
&\in \; O(n \lg n) \, .
\end{aligned}
$$

ΚΑΤΩ ΦΡΑΓΜΑ ΓΙΑ ΑΛΓΟΡΙΘΜΟΥΣ ΕΚΛΟΓΗΣ ΠΡΟΕΔΡΟΥ

Έστω αυθαίρετος αλγόριθμος A εκλογής προέδρου ο οποίος:

1. δουλεύει σε ασύγχρονο δακτύλιο
2. είναι ομοιόμορφος
3. εκλέγει σαν πρόεδρο τον επεξεργαστή με το μέγιστο ΑΤ
4. εγγυάται ότι όλοι μαθαίνουν το ΑΤ του προέδρου,

τότε η πολυπλοκότητα μηνυμάτων είναι $\Omega(n \lg n)$.

- Η συνθήκη 1 είναι απαραίτητη για το κάτω φράγμα.
- Η συνθήκη 2 είναι απαραίτητη για τη συγκεκριμένη απόδειξη.
- Οι συνθήκες 3 και 4 τίθενται χωρίς βλάβη της γενικότητας:

    όποιος αλγόριθμος δεν τις ικανοποιεί μπορεί να μετατραπεί σε ένα που τις ικανοποιεί με $O(n)$ επιπρόσθετα μηνύματα.

ΑΝΟΙΚΤΕΣ ΕΚΤΕΛΕΣΕΙΣ

Νια εκτέλεση είναι ανοικτή αν υπάρχει μια ακμή στην οποία δεν παραλαμβάνονται καθόλου μηνύματα.

Μια ανοικτή εκτέλεση δεν είναι νόμιμη, αλλά είναι πρόθεμα μιας νόμιμης εκτέλεσης.

Θεώρημα: **Για κάθε n που είναι δύναμη του 2, για κάθε σύνολο S από n ΑΤ, υπάρχει ένας δακτύλιος με επεξεργαστές πού έχουν αυτούς τους ΑΤ, πάνω στον οποίο ο αλγόριθμος A έχει μια ανοικτή εκτέλεση στην οποία στέλλονται $M(n)$ μηνύματα, όπου:**

- $M(2) = 1$
- $M(n) = 2 \cdot M(\frac{n}{2}) + \frac{1}{2}(\frac{n}{2} - 1)$ για $n > 2$

Απόδειξη: Με επαγωγή πάνω στο n.

$\Omega(n \lg n)$ ΚΑΤΩ ΦΡΑΓΜΑ ΓΙΑ ΜΗΝΥΜΑΤΑ – ΑΠΟΔΕΙΞΗ

Βάση: $n = 2$

- Υποθέτουμε, χωρίς βλάβη της γενικότητας, ότι $x > y$.

- Λόγω της Συνθήκης 3, ο p_0 θα εκλεγεί πρόεδρος.

- Λόγω της Συνθήκης 4, ο p_0 πρέπει να στείλει τουλάχιστον ένα μήνυμα στον p_1, ώστε ο p_1 να μάθει το x.

- Κόβουμε την εκτέλεση ακριβώς μετά την αποστολή αυτού του μηνύματος, για να λάβουμε μια ανοικτή εκτέλεση.

$\Omega(n \lg n)$ ΚΑΤΩ ΦΡΑΓΜΑ ΓΙΑ ΜΗΝΥΜΑΤΑ – ΑΠΟΔΕΙΞΗ (συνέχεια)

Επαγωγικό Βήμα: Για $n \geq 4$, διάσπασε το σύνολο S σε δύο ίσα μέρη S_1 και S_2. Από επαγωγική υπόθεση, υπάρχουν δακτύλιοι R_1 και R_2 τέτοιοι ώστε:

- Ο δακτύλιος R_1 έχει μια ανοικτή εκτέλεση α_1 στην οποία αποστέλλονται τουλάχιστον $M(n)$ μηνύματα και η $e_1 = (p_1, q_1)$ είναι μια ανοικτή ακμή.
- Ο δακτύλιος R_2 έχει μια ανοικτή εκτέλεση α_2 στην οποία αποστέλλονται τουλάχιστον $M(n)$ μηνύματα και η $e_2 = (p_2, q_2)$ είναι μια ανοικτή ακμή.

Σύνδεσε τους R_1 και R_2 μαζί για να κατασκευάσεις ένα μεγαλύτερο δακτύλιο R:

$\Omega(n \lg n)$ ΚΑΤΩ ΦΡΑΓΜΑ ΓΙΑ ΜΗΝΥΜΑΤΑ – ΑΠΟΔΕΙΞΗ (συνέχεια)

Θα κατασκευάσουμε μια εκτέλεση του R με $M(n)$ μηνύματα:

- Τρέξε πρώτα την εκτέλεση α_1:
  - Λόγω της συνθήκης 2 (ομοιομορφία), οι επεξεργαστές στα αριστερά δεν μπορούν να γνωρίζουν αν βρίσκονται στον R_1 ή τον R. Έτσι, θα συμπεριφερθούν το ίδιο και θα στείλουν $M(n)$ μηνύματα στον R.

- Τρέξε τώρα την εκτέλεση α_1:
  - Λόγω της συνθήκης 2 (ομοιομορφία), οι επεξεργαστές στα δεξιά δεν μπορούν να γνωρίζουν αν βρίσκονται στον R_2 ή τον R. Έτσι, θα συμπεριφερθούν το ίδιο και θα στείλουν $M(n)$ μηνύματα στον R.

Προέξτε πόσο κρίσιμα έχουμε στηριχτεί στην υπόθεση της ομοιμορφίας!

$\Omega(n \lg n)$ ΚΑΤΩ ΦΡΑΓΜΑ ΓΙΑ ΜΗΝΥΜΑΤΑ – ΑΠΟΔΕΙΞΗ (συνέχεια)

Υπάρχουν δύο περιπτώσεις:

- Χωρίς να ξεμπλοκαριστεί η e_p ή η e_q, υπάρχει επέκταση της $\alpha_1\alpha_2$ στον R στην οποία αποστέλλονται $\frac{1}{2}(\frac{n}{2}-1)$ επιπρόσθετα μηνύματα. Αυτή η επέκταση είναι η ζητούμενη ανοικτή εκτέλεση.

- Χωρίς να ξεμπλοκαριστεί η e_p ή η e_q, <u>δεν</u> υπάρχει επέκταση της $\alpha_1\alpha_2$ στον R στην οποία αποστέλλονται $\frac{1}{2}(\frac{n}{2}-1)$ επιπρόσθετα μηνύματα.

  – $\Longrightarrow$ κάθε επέκταση της $\alpha_1\alpha_2$ στον R οδηγεί σε μια <u>ήσυχη</u> διάταξη:

    κανένας επεξεργαστής δεν θα στείλει άλλο μήνυμα πριν παραλάβει μήνυμα, και

    δεν υπάρχουν πουθενά μετεπεβιβαζόμενα μηνύματα, εκτός πάνω στις ακμές e_p και e_q.

    Έστω λοιπόν α_3 μια επέκταση της $\alpha_1\alpha_2$ που οδηγεί σε ήσυχη διάταξη.

$\Omega(n \lg n)$ ΚΑΤΩ ΦΡΑΓΜΑ ΓΙΑ ΜΗΝΥΜΑΤΑ – ΑΠΟΔΕΙΞΗ (συνέχεια)

Έστω α_4'' μια εκτέλεση που επεκτείνει την $\alpha_1\alpha_2\alpha_3$ μετά από την οποία ο αλγόριθμος έχει τερματίσει.

Ισχυρισμός: Τουλάχιστον $n/2$ μηνύματα αποστέλλονται στην α_4''.

Αιτιολόγηση:

- Λόγω της συνθήκης 4, κάθε ένας από τους $n/2$ επεξεργαστές στο μισό μέρος του R που δεν περιέχει τον πρόεδρο, πρέπει να λάβει ένα μήνυμα με το ΑΤ του προέδρου.
- Πρίν από την α_4'', δεν υπήρξε καθόλου επικοινωνία μεταξύ των δύο μερών του R.

ΩΣΤΟΣΟ, η εκτέλεση α_4'' δεν είναι η εκτέλεση που ψάχνουμε, αφού μπορεί να μην είναι ανοικτή.

$\Omega(n \lg n)$ ΚΑΤΩ ΦΡΑΓΜΑ ΓΙΑ ΜΗΝΥΜΑΤΑ – ΑΠΟΔΕΙΞΗ (συνέχεια)

- Καθώς παραλαμβάνονται μηνύματα πάνω στις ακμές e_p και e_q στην εκτέλεση α_4'', οι επεξεργαστές ξυπνούν από την ήσυχη κατάσταση στην οποία βρίσκονταν στο τέλος της εκτέλεσης α_3 και αρχίζουν να στέλνουν περισσότερα μηνύματα.

- Το σύνολο των επεξεργαστών που ξυπνούν επεκτείνεται γύρω από τις ακμές e_p και e_q.

$\Omega(n \lg n)$ ΚΑΤΩ ΦΡΑΓΜΑ ΓΙΑ ΜΗΝΥΜΑΤΑ – ΑΠΟΔΕΙΞΗ (συνέχεια)

- Έστω α_4' το πρόθεμα της α_4'' όπου έχουν αποσταλεί $\frac{n}{2} - 1$ μηνύματα.

- Οι περιοχές P και Q δεν μπορούν να συναντηθούν στην εκτέλεση α_4', αφού τα μηνύματα που στέλλονται στην α_4' είναι λιγότερα από $\frac{n}{2}$.

- Χωρίς βλάβη της γενικότητας, υποθέτουμε ότι η πλειοψηφία των μηνυμάτων αυτών στέλλονται απο επεξεργαστές στην περιοχή P.

$\Omega(n \lg n)$ ΚΑΤΩ ΦΡΑΓΜΑ ΓΙΑ ΜΗΝΥΜΑΤΑ – ΑΠΟΔΕΙΞΗ (συνέχεια)

- Έστω α_4 η ακολουθία που λαμβάνουμε από την α_4' αν κρατήσουμε μόνο τα γεγονότα που συμβαίνουν σε επεξεργαστές στην περιοχή P.
  - $\Longrightarrow$ η ακολουθία αυτή δεν περιέχει καθόλου παραλαβές μηνυμάτων στην ακμή e_q!

<u>ΙΣΧΥΡΙΣΜΟΣ:</u> Στην εκτέλεση $\alpha_1\alpha_2\alpha_3\alpha_4$, οι επεξεργαστές στην περιοχή P συμπεριφέρονται όπως ακριβώς και στην εκτέλεση $\alpha_1\alpha_2\alpha_3\alpha_4'$.

Αφού οι περιοχές P και Q είναι ξένες, δεν υπάρχει καθόλου επικοινωνία από την περιοχή Q προς την περιοχή P στην εκτέλεση $\alpha_1\alpha_2\alpha_3\alpha_4'$. Έτσι, οι επεξεργαστές στην περιοχή P δεν θα αντιληφθούν ότι οι επεξεργαστές στην περιοχή Q είναι ανενεργοί στην εκτέλεση $\alpha_1\alpha_2\alpha_3\alpha_4$.

$\Omega(n \lg n)$ ΚΑΤΩ ΦΡΑΓΜΑ ΓΙΑ ΜΗΝΥΜΑΤΑ – ΑΠΟΔΕΙΞΗ (τέλος)

Αφού οι επεξεργαστές στην περιοχή P συμπεριφέρονται το ίδιο, θα αποστείλουν $\frac{1}{2}(\frac{n}{2}-1)$ μηνύματα στην εκτέλεση α_4.

Έτσι, η εκτέλεση $\alpha_1\alpha_2\alpha_3\alpha_4$ είναι η ζητούμενη εκτέλεση:

- $2M(\frac{n}{2}) + \frac{1}{2}(\frac{n}{2}-1)$ μηνύματα αποστέλλονται, και

- η ακμή e_q είναι ανοικτή.

Αυτό ολοκληρώνει την απόδειξή μας. $\square$