# Φροντιστήριο 2: Ανάλυση Αλγόριθμου Εκλογής Προέδρου με $O(nlogn)$ μηνύματα

# Περιγραφικός Αλγόριθμος

- Αρχικά στείλε μήνυμα εξερεύνησης προς τα δεξιά και αριστερά σου με: $AT = \chi, \delta = 1$
- Όταν λάβεις ένα μήνυμα με AT=ψ και δ (από δεξιά ή αριστερά):
  - Μείωσε $\delta = \delta - 1$
  - Αν $\psi > \chi$
    - $\delta > 0$: προώθησέ το ψ,δ (δεν θα γίνεις ποτέ πρόεδρος)
    - $\delta = 0$: στείλε επιβεβαίωση στον αποστολέα
  - Αν $\psi = \chi$ αυτοανακυρήξου πρόεδρος
  - Αν $\psi < \chi$ μην κάνεις τίποτα
- Οταν λάβεις επιβεβαίωση που δεν προορίζεται στον $\chi$ τότε προώθησέ την
- Όταν λάβεις δύο επιβεβαιώσεις θέσε $\delta = \delta * 2$ και πέρνα στην επόμενη φάση

# Ψευδοκώδικας

**Algorithm 1** Leader Election at $p_i$

**State Variables:**

$leader = \perp$, $d = 1$, $ID = id(p_i)$, $terminate = FALSE$

Initialization:

1: send $<discover, ID, d>$ to left and right

Upon **Receiving a message** $<leader, id>$ from right:

2: $leader \leftarrow id$; $terminate \leftarrow TRUE$

Upon **Receiving a message** $<discover, id, d>$ from left:

3: **if** $id = ID$ **then**
4: $leader \leftarrow ID$; $terminate \leftarrow TRUE$
5: **end if**
6: **if** $id > ID$ **then**
7: **if** $d > 1$ **then**
8: send $<discover, id, d-1>$ to right
9: **else**
10: send $<ack, id>$ to left
11: **end if**
12: **end if**

# Ψευδοκώδικας

## **Algorithm 2** Leader Election at $p_i$

Upon **Receiving a message** $<discover, id, d>$ from right:

```
1: if id = ID then
2:     leader ← ID; terminate ← TRUE
3:     send <leader, ID> to left
4: end if
5: if id > ID then
6:     if d > 1 then
7:         send <discover, id, d − 1> to left
8:     else
9:         send <ack, id> to right
10:    end if
11: end if
```

Upon **Receiving a message** $<ack, id>$ from left (resp. right):

```
12: if id ≠ ID then
13:     send <ack, id> to right (resp.left)
14: else
15:     if already received <ack, id> from right (resp. left) then
16:         d ← d * 2
17:         send <discover, ID, d> to left and right
18:     end if
19: end if
```

# Χρονική Πολυπλοκότητα Αλγορίθμου

- Σε κάθε φάση ένας επεξεργαστής επικρατεί σε κάποια γειτονιά
- Πόσες φάσεις χρειάζόμαστε για να καλύψουμε τον δακτύλιο;
  - Διπλασιάζουμε σε κάθε φάση την απόσταση
  - Επομένως στην φάση 0 επικρατούμε σε αποσταση 1, στην φαση 1 επικρατούμε σε απόσταση 2 και γενικά στη φάση $k$ θα επικρατήσουμε σε γειτονιά βάθους $2^k$.
  - Για να καλύψουμε όλους τους κόμβους στον δακτύλιο πρέπει να ισχύει:

$$2^k = n \Rightarrow k = \lg n$$

- Πόσο χρόνο χρειαζόμαστε σε κάθε φάση;

$$T_k = 2 * 2^k$$

# Χρονική Πολυπλοκότητα Αλγορίθμου

- Επομένως σε $\lg n$ φάσεις ο πρόεδρος θα καλύψει όλο το δακτύλιο
- Αρα αν υποθέσουμε ότι κάθε μήνυμα αποστέλλεται σε πολύ μια χρονική μονάδα τότε η χρονική πολυπλοκότητα του αλγορίθμου είναι:

$$\begin{aligned} T &= T_1 + T_2 + \ldots + T_{\lg n} \\ &= \sum_{k=0}^{\lg n} 2 * 2^k = 2 \sum_{k=0}^{\lg n} 2^k \\ &= 2^{\lg n + 2} = 4n = O(n) \end{aligned}$$

# Πολυπλοκότητα Μηνυμάτων

- Πόσα μηνύματα χρειαζόμαστε σε κάθε φάση από ένα επεξεργαστή;
  - Φάση 0: Στέλνει δυο μηνύματα σε κάθε ένα από τους γείτονες και περιμένει απαντήσεις. Επομένως

$$M_0 = 4$$

  - Φάση 1: Στέλνει μηνύματα σε 2 γείτονες από κάθε μεριά και περιμένει απαντήσεις. Αρα

$$M_1 = 4 * 2 = 8$$

  - Φάση $k$: Στέλνει μηνύματα σε $2^k$ γείτονες από κάθε μεριά και περιμένει απαντήσεις. Αρα

$$M_k = 4 * 2^k$$

# Πολυπλοκότητα Μηνυμάτων

- Αφού ξέρουμε ότι έχουμε το πολύ $\lg n$ φάσεις τότε στην χειρότερη περίπτωση ένας επεξεργαστής $p_i$ θα έστελνε:

$$M(p_i) = \sum_{k=0}^{\lg n} 4 * 2^k = O(n)$$

- Επομένως ένα χαλαρό άνω φράγμα θα ήταν το $O(n^2)$.
- Αυτό όμως δεν ισχύει γιατί δεν εμπλέκονται όλοι οι επεξεργαστές σε όλες τις φάσεις του αλγόριθμου.

# Πολυπλοκότητα Μηνυμάτων

- Μόνο οι νικητές της φάσης $k-1$ στέλνουν μηνύματα στη φάση $k$.

**Λήμμα**

*Για κάθε $k \geq 1$ ο αριθμός των επεξεργαστών που είναι νικητές στη φάση $k$ είναι το πολύ $\frac{n}{2^k+1}$.*

- Αυτό έπεται από την παρατήρηση ότι 2 νικητές σε μια φάση $k$ πρέπει να έχουν απόσταση τουλάχιστον όσο είναι το βάθος της γειτονιάς στην οποία επικρατούν $2^k + 1$.
  - Φάση 0: Καθε 2 επεξεργαστές νικητής ένας
  - Φάση 1: Κάθε 3 επεξεργαστες νικητής ένας
  - Φάση 2: Κάθε 5 επεξεργαστες νικητής ένας κτλ.

# Πολυπλοκότητα Μηνυμάτων

- Επομένως η συνολική πολυπλοκότητα είναι

$$
\begin{aligned}
M &= n + 4n + \sum_{k=1}^{\lg n} 4 * 2^k \frac{n}{2^{k-1}+1} \\
&< 5n + 4n \sum_{k=1}^{\lg n} \frac{2^k}{2^{k-1}} \\
&= 5n + 4n \sum_{k=1}^{\lg n} 2 \\
&= 5n + 8n \lg n = O(n \lg n)
\end{aligned}
$$