

<-

[i, j]

top <- top + 1

<-

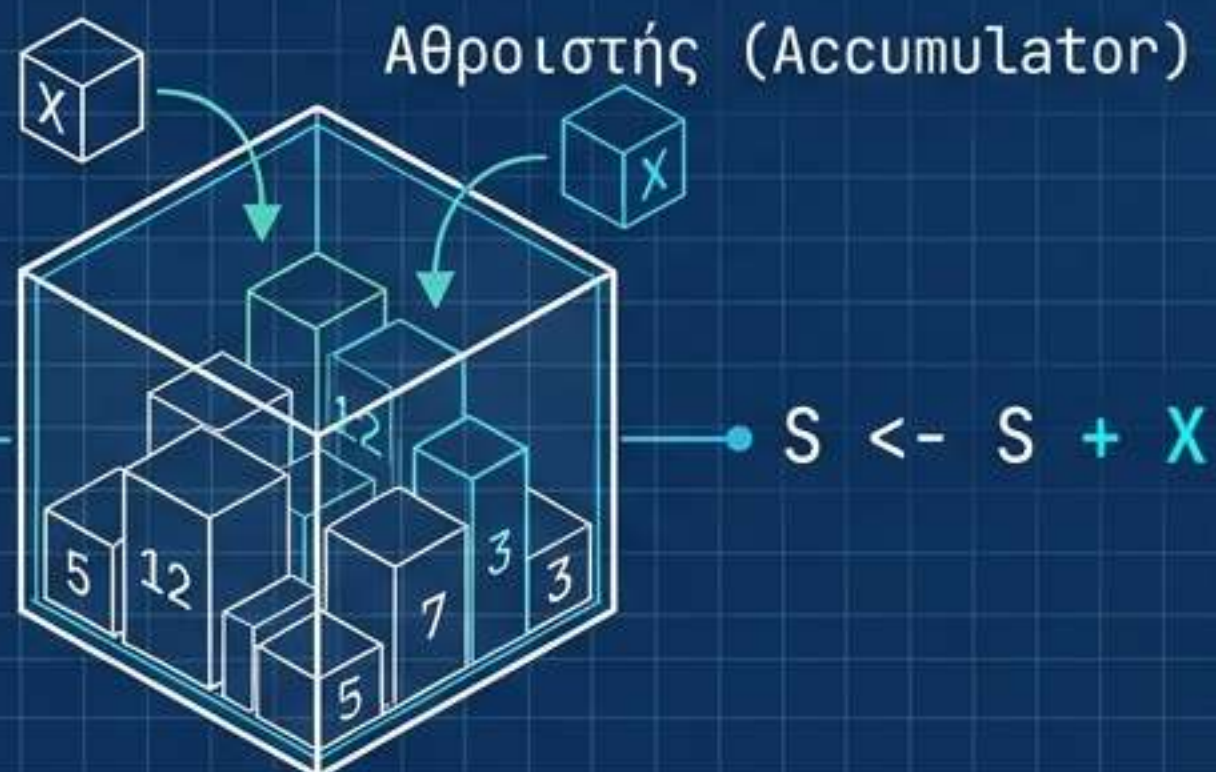
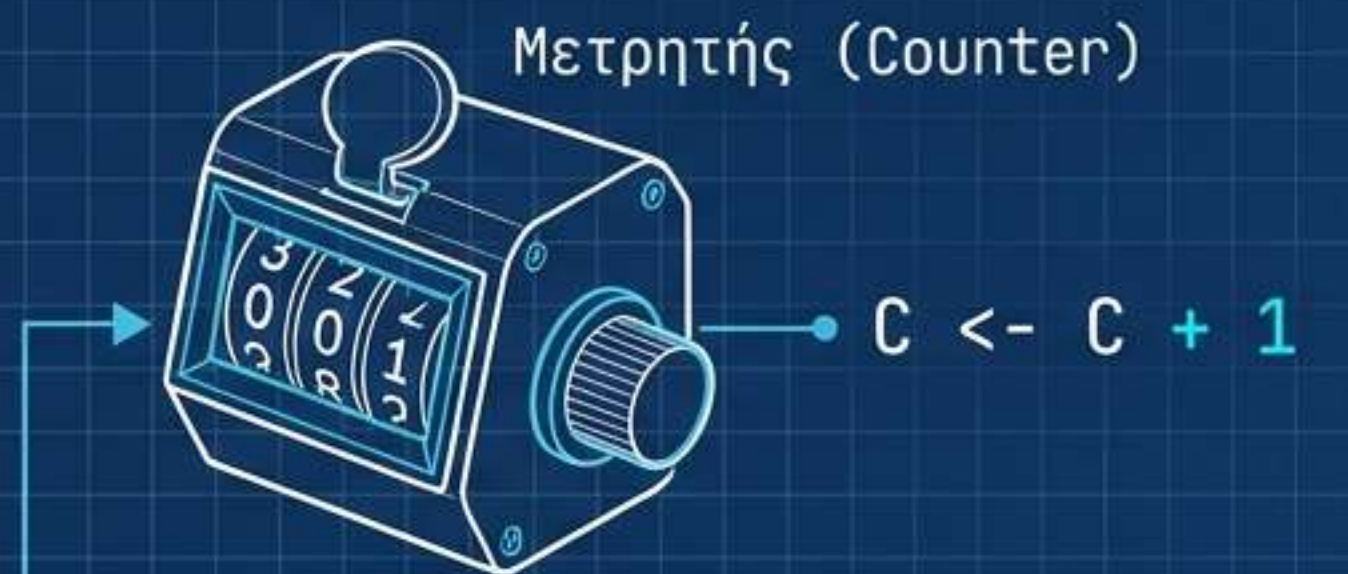
[i, j]

# Η Απόλυτη Εργαλειοθήκη Μεθοδολογιών: Α.Ε.Π.Π.

Οδηγός Επιβίωσης για τα θέματα Γ & Δ – Μοντέλα  
ΓΛΩΣΣΑΣ, Πίνακες, Δομές & Αναζήτηση.

top <- top + 1

# Βασικοί Υπολογισμοί: Μετρητές & Αθροιστές



```
Σ ← 0      ! Αρχικοποίηση Αθροιστή  
Π ← 0      ! Αρχικοποίηση Μετρητή  
ΟΣΟ Χ <> 0 ΕΠΑΝΑΛΑΒΕ  
    Σ ← Σ + Χ  
    Π ← Π + 1  
    ΔΙΑΒΑΣΕ Χ  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

# Μέσοι Όροι & Ποσοστά (ΠΡΟΣΟΧΗ ΣΤΙΣ ΠΑΓΙΔΕΣ)

$$MO \leftarrow \Sigma / \Pi$$

(Μέσος Όρος = Άθροισμα / Πλήθος)



Παγίδα θέματος Γ: Ποτέ μην διαιρείς αν δεν ελέγξεις ότι ο παρονομαστής είναι  $> 0$ !

ΑΝ  $\Pi > 0$  ΤΟΤΕ

$MO \leftarrow \Sigma / \Pi$

    ΓΡΑΨΕ 'Μέσος Όρος:', MO

ΑΛΛΙΩΣ

    ΓΡΑΨΕ 'Δεν δόθηκαν στοιχεία'

ΤΕΛΟΣ\_ΑΝ

# Μήτρα Επιλογής: Αρχικοποίηση Μέγιστου / Ελάχιστου

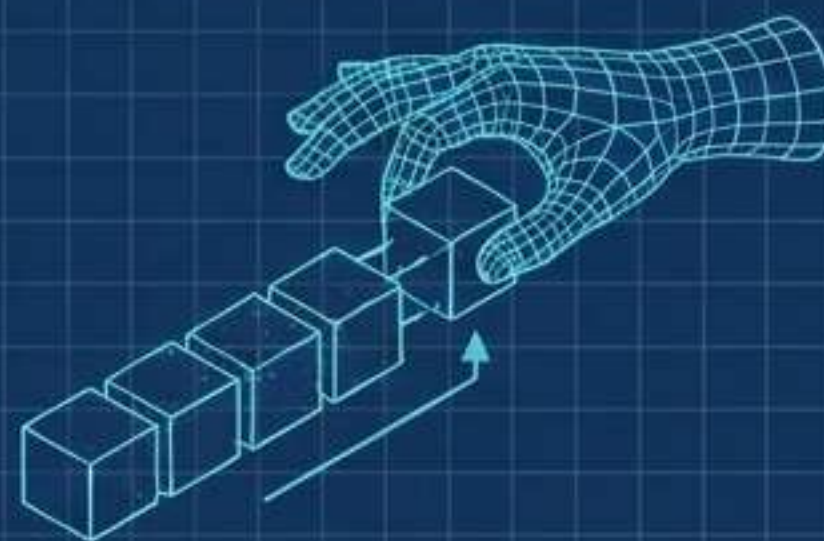
Συγκεκριμένο Πεδίο Τιμών  
(Ακραίες Τιμές)



Στο Max δίνουμε τη μικρότερη δυνατή τιμή.  
Στο Min δίνουμε τη μεγαλύτερη.

```
Max ← 0  
Min ← 20
```

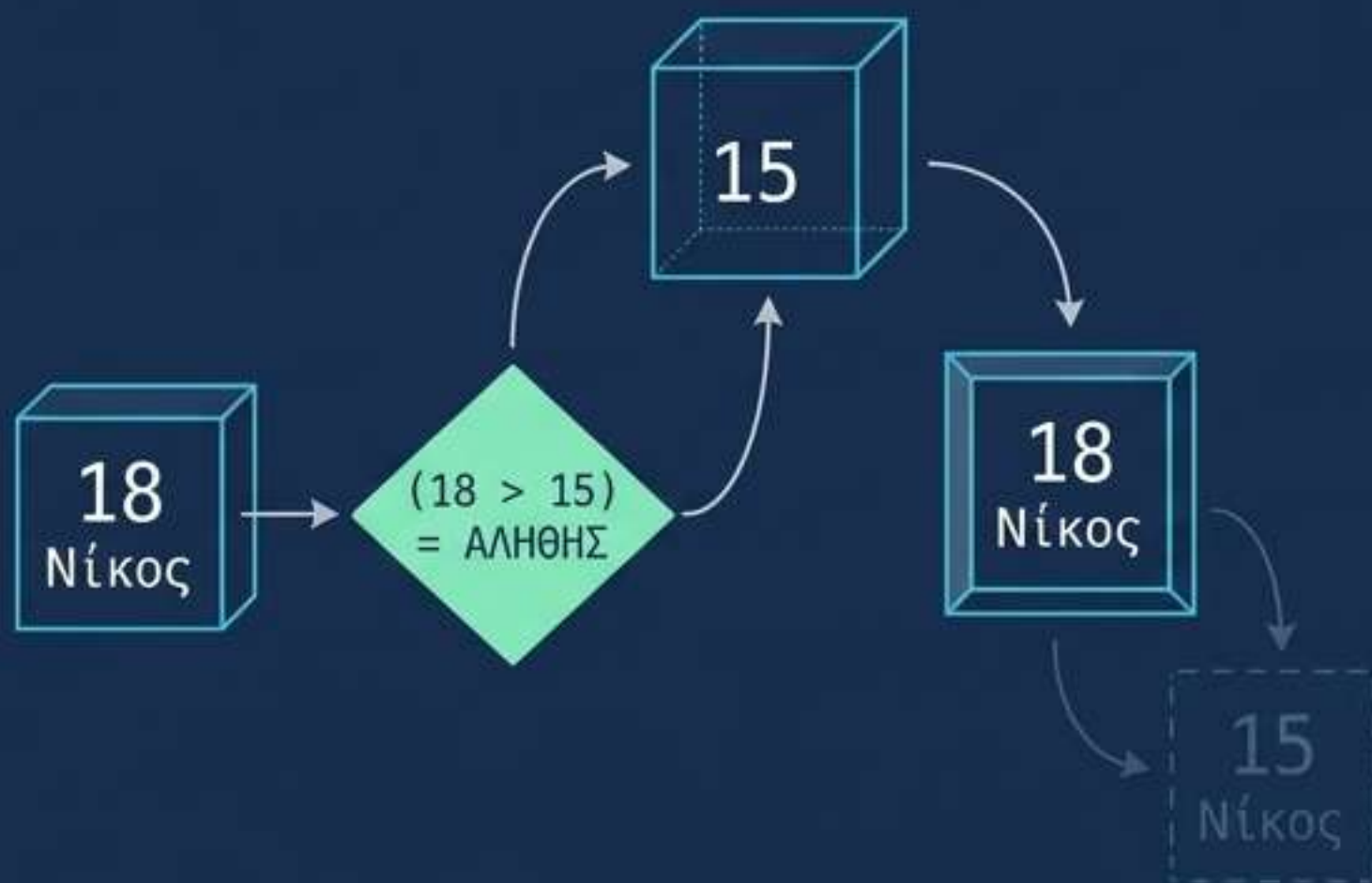
Άγνωστο Πεδίο Τιμών  
(Το 1ο Στοιχείο)



Διαβάζουμε το 1ο στοιχείο εκτός επανάληψης και το εκχωρούμε στο Max και Min.

```
ΔΙΑΒΑΣΕ A[1]  
Max ← A[1]  
Min ← A[1]
```

# Εύρεση Max / Min & Παράλληλη Αποθήκευση



King of the Hill: Logic Flow

ΔΙΑΒΑΣΕ  $0n$ , Βαθμός

$Max \leftarrow$  Βαθμός

$Max_{0n} \leftarrow 0n$

ΓΙΑ  $i$  ΑΠΟ 2 ΜΕΧΡΙ 1000

ΔΙΑΒΑΣΕ  $0n$ , Βαθμός

ΑΝ Βαθμός  $\gg$   $Max$  ΤΟΤΕ

$Max \leftarrow$  Βαθμός ! Ανανέωση Μέγιστου

$Max_{0n} \leftarrow 0n$  ! Παράλληλη Αποθήκευση Ονόματος

ΤΕΛΟΣ\_ΑΝ

ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ

# Η Τεχνική της Σειριακής Αναζήτησης (Με Σηματοφόρο)

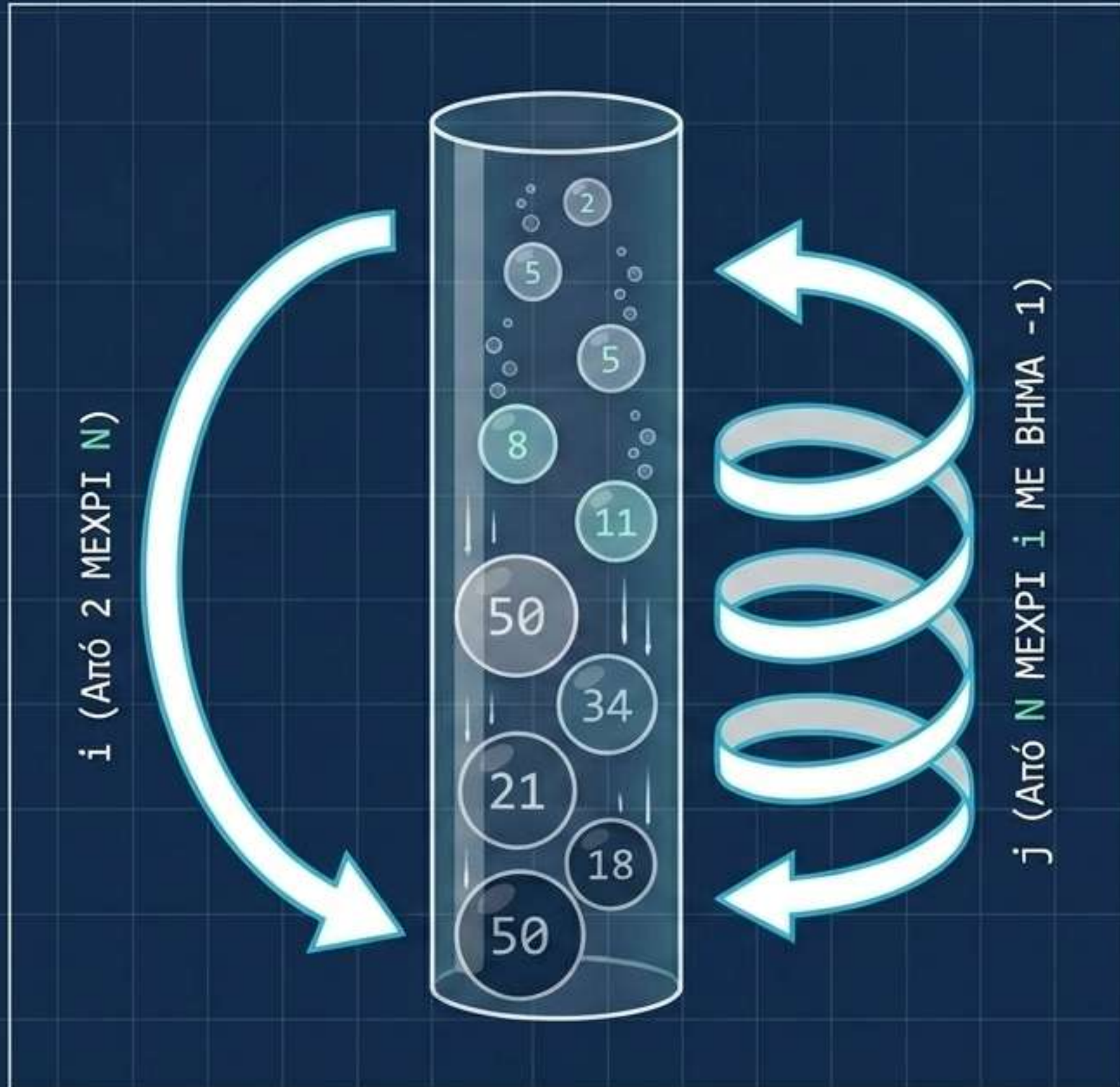


```
done <- ΨΕΥΔΗΣ  
position <- 0  
i <- 1
```

```
ΟΣΟ (done = ΨΕΥΔΗΣ) ΚΑΙ (i <= N) ΕΠΑΝΑΛΑΒΕ  
  ΑΝ A[i] = key ΤΟΤΕ  
    done <- ΑΛΗΘΗΣ  
    position <- i  
  ΑΛΛΙΩΣ  
    i <- i + 1  
  ΤΕΛΟΣ_ΑΝ  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```



# Ταξινόμηση Φυσαλίδας (Bubble Sort)

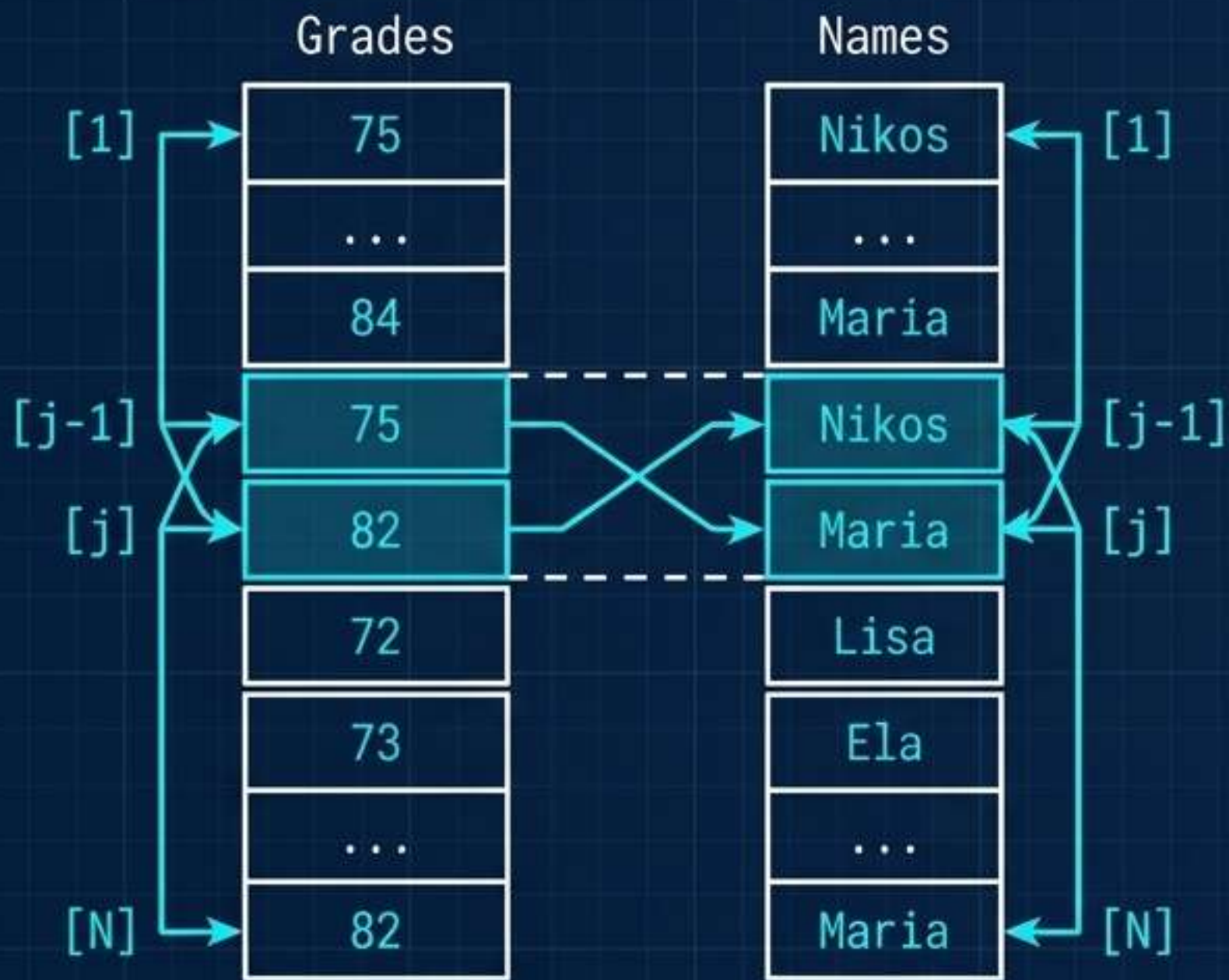


Φθίνουσα Σειρά

```
ΓΙΑ  $i$  ΑΠΟ 2 ΜΕΧΡΙ  $N$   
  ΓΙΑ  $j$  ΑΠΟ  $N$  ΜΕΧΡΙ  $i$  ΜΕ ΒΗΜΑ  $-1$   
    ΑΝ  $A[j-1] < A[j]$  ΤΟΤΕ ! Για Αύξουσα αλλαγή σε >  
      ! Αντιμετάθεση (Swap)  
      temp  $\leftarrow$   $A[j-1]$   
       $A[j-1] \leftarrow A[j]$   
       $A[j] \leftarrow$  temp  
    ΤΕΛΟΣ_ΑΝ  
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

# Παράλληλοι Πίνακες: Ταυτόχρονη Ταξινόμηση

Recipe Cards



! Μέσα στο AN της Φυσαλίδας...

AN Βαθμός[ $j-1$ ] < Βαθμός[ $j$ ] ΤΟΤΕ

! 1. Αντιμετάθεση Βαθμών

tempV <- Βαθμός[ $j-1$ ]

Βαθμός[ $j-1$ ] <- Βαθμός[ $j$ ]

Βαθμός[ $j$ ] <- tempV

! 2. Υποχρεωτική Αντιμετάθεση Ονομάτων!

tempN <- Όνομα[ $j-1$ ]

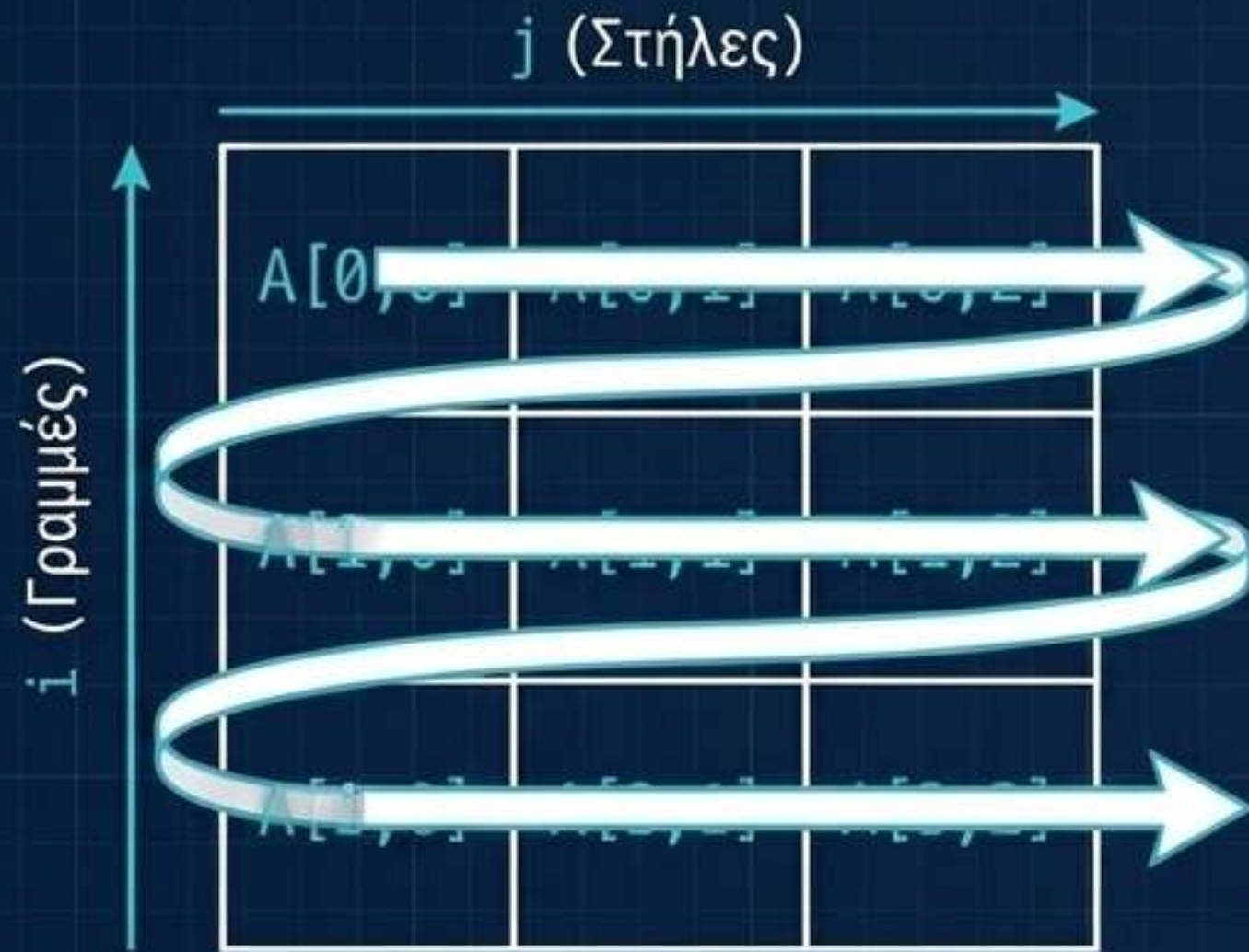
Όνομα[ $j-1$ ] <- Όνομα[ $j$ ]

Όνομα[ $j$ ] <- tempN

ΤΕΛΟΣ\_AN

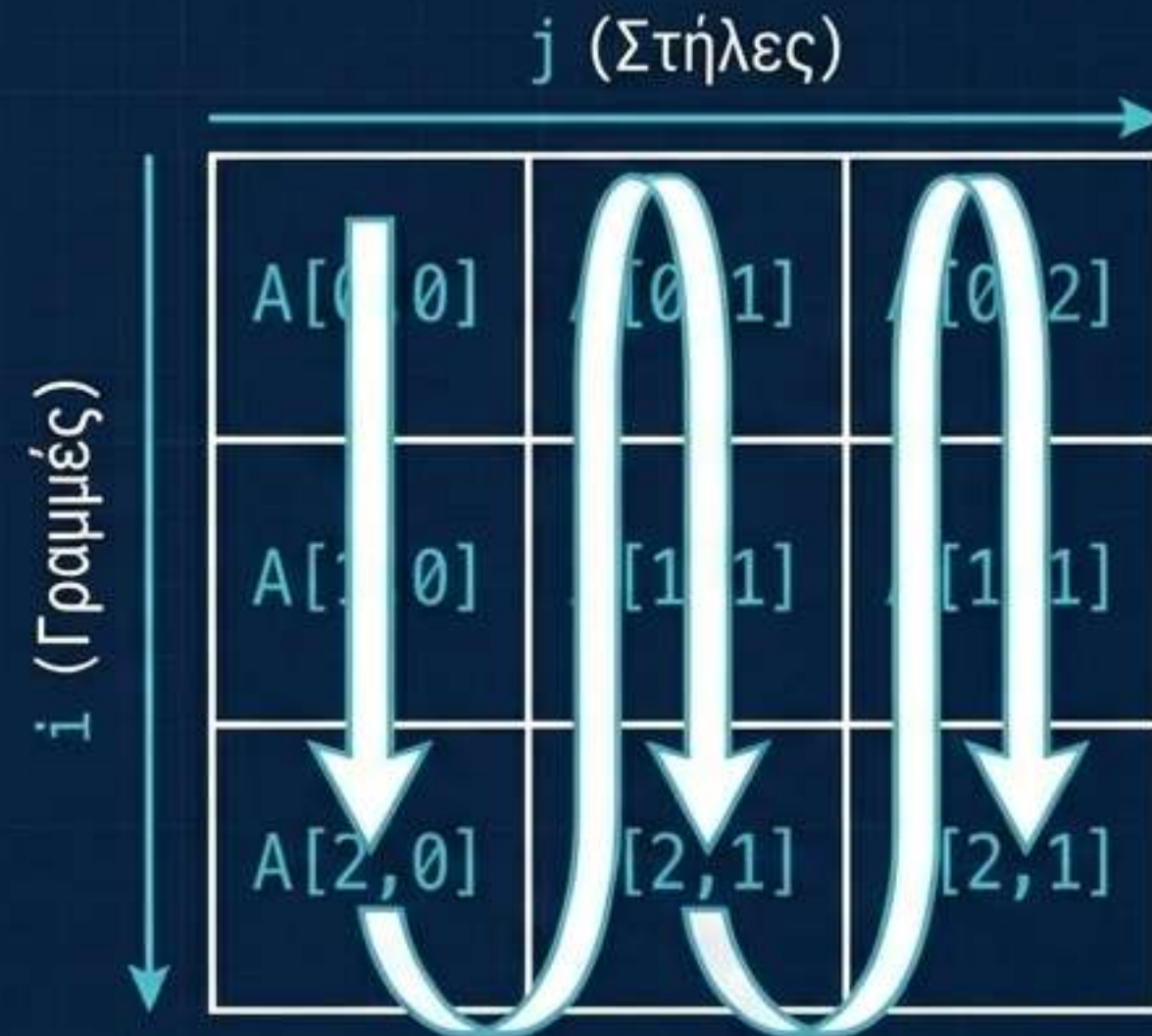
# Δισδιάστατοι Πίνακες (2D Matrices): Μηχανική Προσπέλασης

Προσπέλαση Κατά Γραμμή (Row-by-Row)



ΓΙΑ  $i \dots$  ΓΙΑ  $j \dots$  ΔΙΑΒΑΣΕ  $A[i,j]$

Προσπέλαση Κατά Στήλη (Col-by-Col)



ΓΙΑ  $j \dots$  ΓΙΑ  $i \dots$  ΔΙΑΒΑΣΕ  $A[i,j]$

# Υπολογισμοί σε Δισδιάστατους: Αθροίσματα Γραμμών & Στηλών

## Άθροισμα Ανά Γραμμή

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Rows
```

```
S ← 0 ! Προσοχή: Επαναμηδενισμός ΕΔΩ!
```

```
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Cols
```

```
S ← S + A[i, j]
```

```
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
Άθροισμα_Γρ[i] ← S ! Αποθήκευση  
σε Μονοδιάστατο
```

```
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

## Άθροισμα Ανά Στήλη

```
ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Cols
```

```
S ← 0 ! Προσοχή: Επαναμηδενισμός ΕΔΩ!
```

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Rows
```

```
S ← S + A[i, j]
```

```
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
Άθροισμα_Στ[j] ← S
```

```
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

# Ακραίες Τιμές σε Δισδιάστατο (Max / Min)

## Απόλυτο Μέγιστο Πίνακα

**Rule:** Αρχικοποίηση  $Max \leftarrow A[1,1]$  πριν από τις δύο επαναλήψεις.

```
Max ← A[1,1]
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Rows
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Cols
    ΑΝ A[i,j] > Max ΤΟΤΕ Max ← A[i,j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

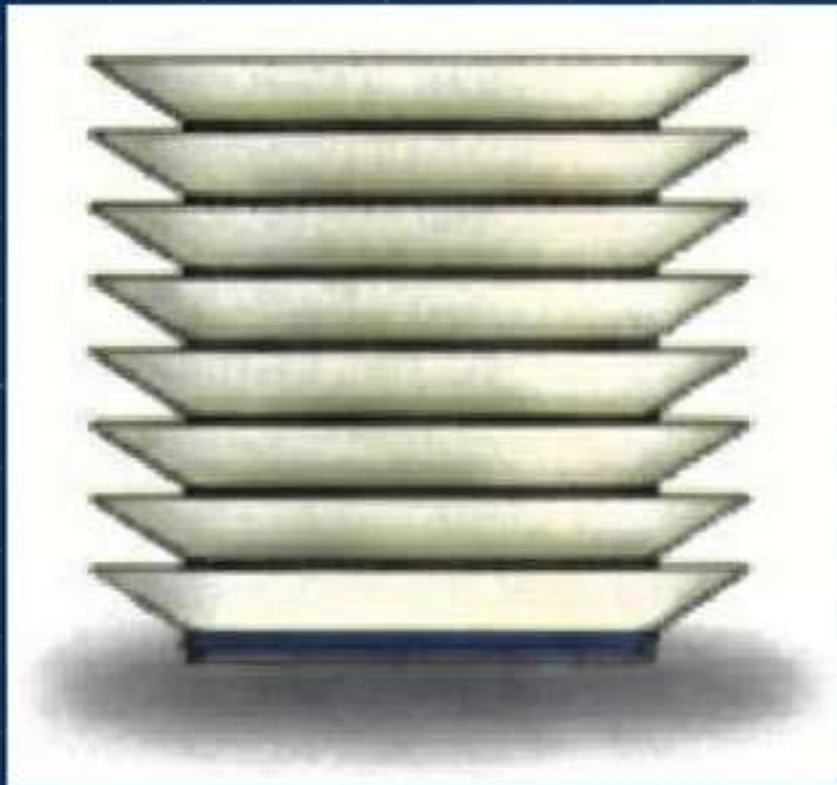
## Μέγιστο Ανά Γραμμή

**Rule:** Αρχικοποίηση  $Max \leftarrow A[i,1]$  μέσα στην εξωτερική, πριν την εσωτερική επανάληψη.

```
ΓΙΑ i ΑΠΟ 1 ΜΕΧΡΙ Rows
  Max ← A[i,1]
  ΓΙΑ j ΑΠΟ 1 ΜΕΧΡΙ Cols
    ΑΝ A[i,j] > Max ΤΟΤΕ Max ← A[i,j]
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
  ΓΡΑΨΕ 'Max γραμμής: ', Max
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

# Δυναμικές Δομές Δεδομένων: Στοίβα vs. Ουρά

## Στοίβα (Stack) - LIFO



- Μηχανισμός: LIFO (Last In, First Out) - Τελευταίο μέσα, πρώτο έξω.
- Δείκτης: Ένας δείκτης (top).
- Πράξεις: Ώθηση (Push), Απώθηση (Pop).

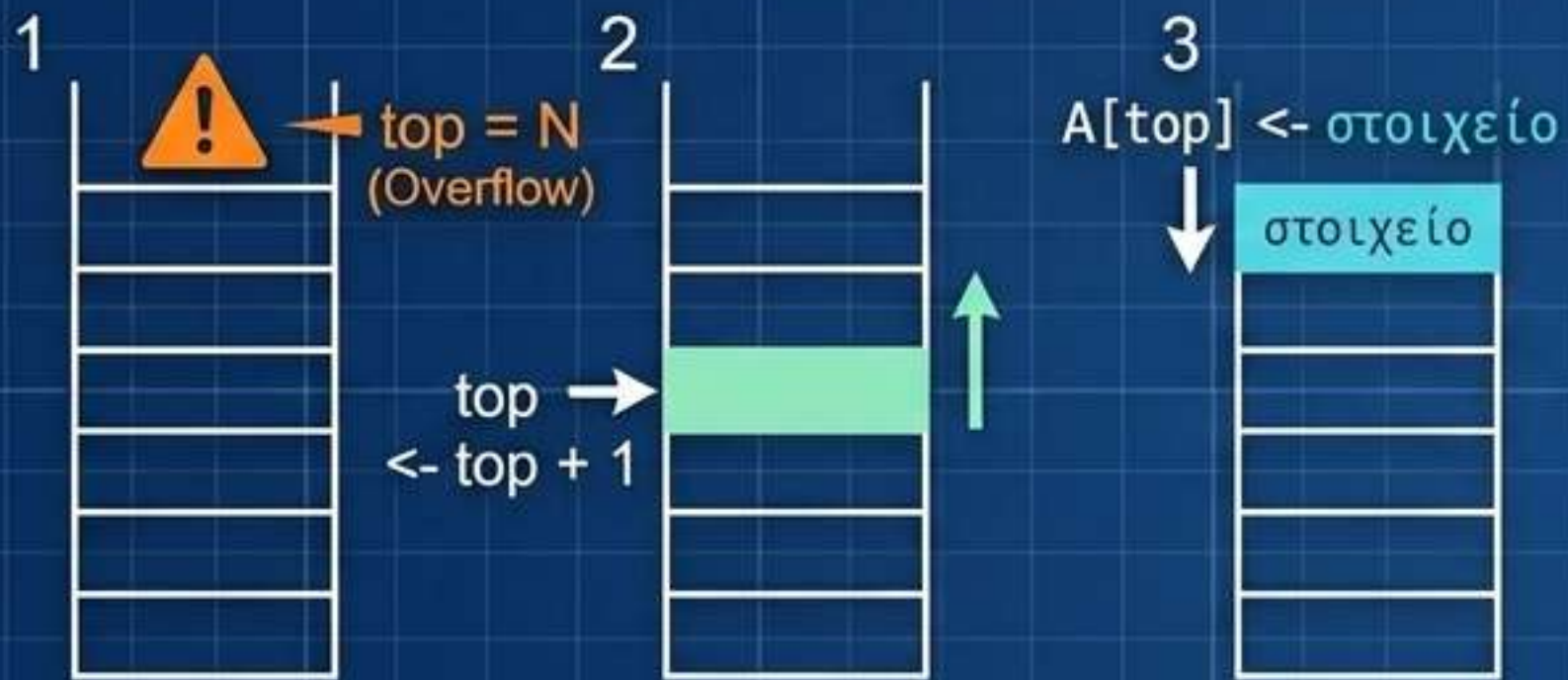
## Ουρά (Queue) - FIFO



- Μηχανισμός: FIFO (First In, First Out) - Πρώτο μέσα, πρώτο έξω.
- Δείκτες: Δύο δείκτες (front, rear).
- Πράξεις: Εισαγωγή (Enqueue), Εξαγωγή (Dequeue).

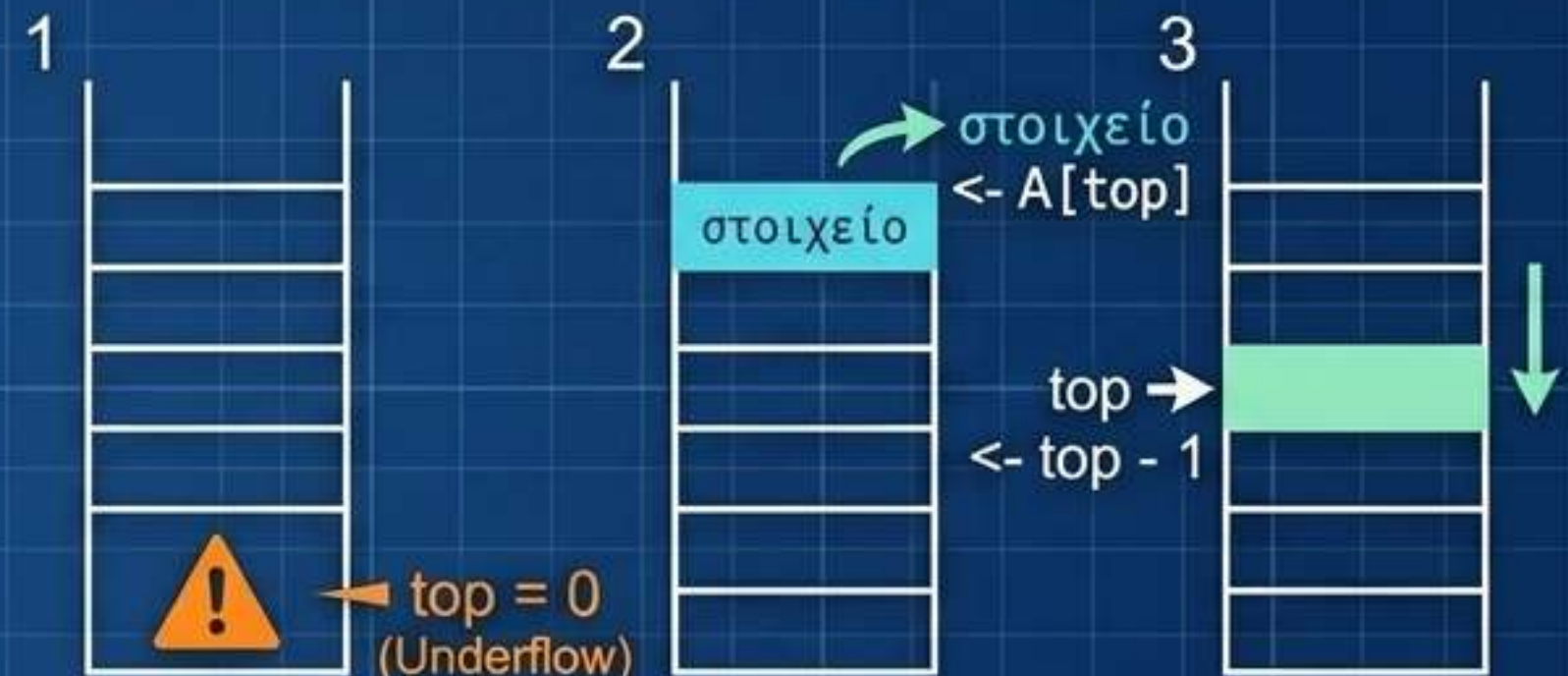
# Αλγόριθμοι Στοίβας: Ώθηση (Push) & Απώθηση (Pop)

## Ώθηση (Push)



```
ΑΝ  $top = N$  ΤΟΤΕ  
  ΓΡΑΨΕ 'Υπερχείλιση'  
ΑΛΛΙΩΣ  
   $top \leftarrow top + 1$   
   $A[top] \leftarrow \text{στοιχείο}$   
ΤΕΛΟΣ_ΑΝ
```

## Απώθηση (Pop)



```
ΑΝ  $top = 0$  ΤΟΤΕ  
  ΓΡΑΨΕ 'Υποχείλιση'  
ΑΛΛΙΩΣ  
   $\text{στοιχείο} \leftarrow A[top]$   
   $top \leftarrow top - 1$   
ΤΕΛΟΣ_ΑΝ
```

# Αλγόριθμοι Ουράς: Εισαγωγή (Enqueue) & Εξαγωγή (Dequeue)

## Εισαγωγή (Enqueue)

Rule: Pointer 'rear' tracks the back of the line.

```
AN rear = N ΤΟΤΕ
  ΓΡΑΨΕ 'Ουρά Γεμάτη'
ΑΛΛΙΩΣ
  rear ← rear + 1
  A[rear] ← στοιχείο
ΤΕΛΟΣ_ΑΝ
```

## Εξαγωγή (Dequeue)

Rule: Pointer 'front' tracks the front.  
Special condition: front = 0 AND rear = 0 triggers the empty check.

```
AN front > rear Η
(front = 0 ΚΑΙ rear = 0) ΤΟΤΕ
  ΓΡΑΨΕ 'Ουρά Αδεια'
ΑΛΛΙΩΣ
  στοιχείο ← A[front]
  front ← front + 1
ΤΕΛΟΣ_ΑΝ
```

# Δομημένος Προγραμματισμός: Υποπρογράμματα

## Διαδικασίες (Procedures)

- Χρήση: Καλείται ως αυτόνομη εντολή (π.χ. `ΚΑΛΕΣΕ Ταξινομήση(A, N)`).
- Επιστροφή: Επιστρέφει από 0 έως πολλές τιμές μέσω της λίστας παραμέτρων.
- Παράμετροι: Πίνακες περνούν πάντα και μόνο σε διαδικασίες (και αλλάζουν στο κυρίως πρόγραμμα).

## Συναρτήσεις (Functions)

- Χρήση: Καλείται μέσα σε έκφραση (π.χ. `MO <- Άθροισμα(X, Y) / 2`).
- Επιστροφή: Επιστρέφει **ΠΑΝΤΑ** μία και μόνο μία τιμή με το όνομά της. **Δεν μπορεί να επιστρέψει πίνακα!**

# Χρυσός Κανόνας Παραμέτρων στα Υποπρογράμματα

Main Program

Key Rules:

1. Η λίστα παραμέτρων πρέπει να συμφωνεί σε πλήθος, τύπο, και σειρά.
2. Δεν δηλώνουμε το μέγεθος των πινάκων στις τυπικές παραμέτρους (π.χ., γράφουμε ΔΙΑΔΙΚΑΣΙΑ Sort(A, N) και ΟΧΙ Sort(A[100], N)).
3. Κάθε υποπρόγραμμα είναι τυφλό στις μεταβλητές του άλλου (Local Scope). Η επικοινωνία γίνεται μόνο μέσω των παραμέτρων!

Subprogram

Τυπικές  
Παράμετροι  
(Subprogram)

ΑΚΕΡΑΙΑ

ΠΡΑΓΜΑΤΙΚΗ

Πραγματικές  
Παράμετροι  
(Main Program)

ΑΚΕΡΑΙΑ

ΠΡΑΓΜΑΤΙΚΗ

# Το Μοντέλο Εκτέλεσης: Η Ροή των Θεμάτων Γ & Δ

- 1 Δηλώσεις: Ορισμός μεταβλητών και πινάκων (ΑΚΕΡΑΙΕΣ, ΠΡΑΓΜΑΤΙΚΕΣ).
- 2 Αρχικοποιήσεις: Μηδενισμός αθροιστών/μετρητών, αρχικές τιμές Max/Min.
- 3 Εισαγωγή & Έλεγχος: Δομή ΔΙΑΒΑΣΕ / ΟΣΟ για έλεγχο ορθότητας εισόδου.
- 4 Κύρια Επεξεργασία (The Loop): Δομή ΓΙΑ ή ΟΣΟ. Υπολογισμοί, γεμίσματα πινάκων, αθροίσματα.
- 5 Τυπικές Επεξεργασίες: Ενέργειες εκτός του κύριου βρόχου (Ταξινόμηση, Αναζήτηση).
- 6 Έξοδος (Εκτύπωση): Εμφάνιση αποτελεσμάτων. Ποτέ μην τυπώνεις μέσα στην ταξινόμηση!

Η Αλγοριθμική Σκέψη Δεν Μαθαίνεται Απέξω.  
Οικοδομείται.

Το κλειδί για το Άριστα (20) στα θέματα Γ & Δ δεν είναι η αποστήθιση κώδικα, αλλά η ικανότητα να σπας ένα πολύπλοκο πρόβλημα στα βασικά του συστατικά: **Μεταβλητές, Πίνακες, Επαναλήψεις.**

Διάβασε την εκφώνηση προσεκτικά.

Σχεδίασε τη δομή δεδομένων σου.

Χτίσε τον αλγόριθμό σου.

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

> compilation successful.