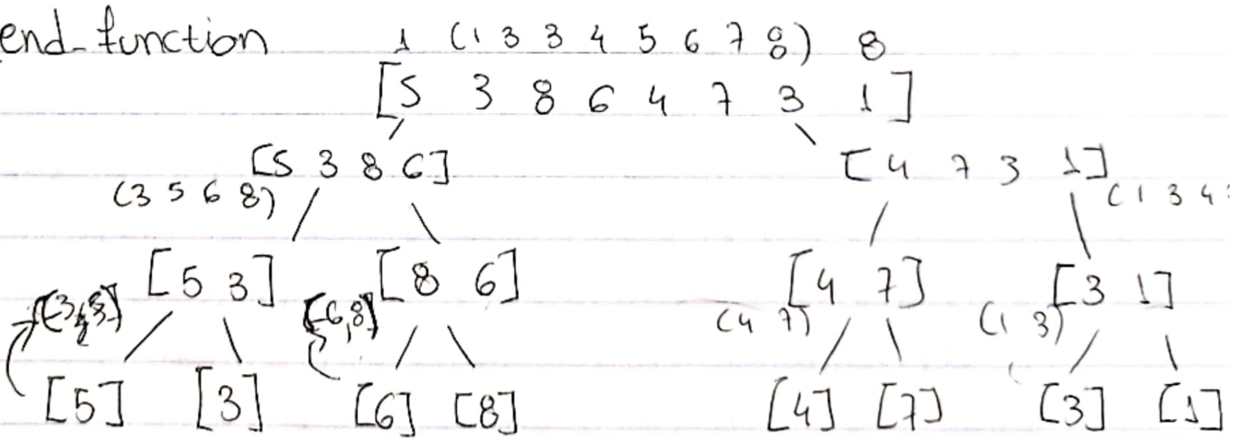
Σχεδίαση & Ανάλυση Αλγορίθμων 15/05/2023

```
function MergSort(int A[], int lo, int hi)
  if lo<hi then
    mid ← ⌊(hi+lo)/2⌋
    Mergsort(A, lo, mid);   21
    Mergsort(A, mid, hi),   22
    for i←lo; i≤mid; i++ do
      A1[i] ← A[i];
    end_for
    for i←mid+1; i≤hi; i++ do
      A2[i]←A[i];
    end_for
    Merge(A1, lo, mid, A2, mid+1, hi, A, lo, hi);
  end_if
end_function
```

Πίνακες πρέπει να είναι ταξινομημένοι, συγχωνεύονται σε έναν τρίτο πίνακα με ταξινομημένο σύνολο στοιχείων



$$T(n) = \begin{cases} T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + \Theta(n), & n > 1 \\ 1, & n \leq 1 \end{cases}$$

$n = hi - lo + 1$ (1)

#21. $mid - lo + 1 = \left\lfloor \frac{hi - lo}{2} \right\rfloor - lo + 1$

(1) $\Rightarrow n - 1 = hi - lo \Rightarrow n - 1 - 2lo = hi + lo \Rightarrow n - 1 = hi + lo - 2lo \Rightarrow$

$\frac{n-1}{2} = \frac{hi+lo}{2} - lo \Rightarrow \left\lfloor \frac{n-1}{2} \right\rfloor = \left\lfloor \frac{hi+lo}{2} - lo \right\rfloor = \left\lfloor \frac{hi+lo}{2} \right\rfloor + lo$

$\Rightarrow \left\lfloor \frac{n-1}{2} \right\rfloor + 1 = \left\lfloor \frac{hi+lo}{2} \right\rfloor - lo + 1 \Rightarrow \left\lfloor \frac{hi+lo}{2} \right\rfloor - lo + 1 = \left\lfloor \frac{n-1}{2} + 1 \right\rfloor = \left\lfloor \frac{n+1}{2} \right\rfloor$ (3)

(2) (3) $\Rightarrow mid - lo + 1 = \left\lfloor \frac{n+1}{2} \right\rfloor = \left\lceil \frac{n}{2} \right\rceil$

#22: $n - \left\lceil \frac{n}{2} \right\rceil = \left\lfloor \frac{n}{2} \right\rfloor$

$T(n) = \Theta(n \log n)$ ← Μέσω εκτεταμένου κεντρικού θεωρήματος

↑ Καλύτερο δυνατό με χρήση συγκρίσεων.

ΑΣΚΗΣΗ: Πόσος είναι ο επιπλέον χώρος που χρησιμοποιεί η Mergesort με το n (quicksort) ασυμπτωτικά; (Με χρήση δηλαδή των $\Theta(n)$).

• Χωρίς συγκρίσεις - Μπορεί να φτάσει και γραμμικό χρόνο

2.16.4

7.6.1:

$k, A[k]=k$

$A[lo] < A[lo+1] < \dots < A[hi-1] < A[hi]$

$A[k] < k \Rightarrow A[k]-1 < k-1 \;\Big|\; \Rightarrow A[k-1] < k-1$

$A[k-1] \leq A[k]-1 \;\Big|\; \{k+1, \dots, hi\}$

$A[k] > k \qquad \{1, \dots, k-1\}$

· Θέλω να επιτύχω πολυπλοκότητα δυαδικής αναζήτησης

```
function find_~~thesi~~thesi_k (int A[], int lo, int hi)
    {
    if lo<hi then
        k ← ⌊(lo+hi)/2⌋;
        if A[k]<k then
            return find_thesi_k (A, k+1, hi);
        else
            return find_thesi_k (A, lo, k-1);
        else return k;
        end_if
    else
        if A[k]= k then
            return k;
        else
            return 0;
        end_if
    end_if
End_function
```

$$T(n) = \begin{cases} \le T(\lceil \frac{n-1}{2} \rceil) + \Theta(1) \\ \ge T(\lfloor \frac{n-1}{2} \rfloor) + \Theta(1) \\ = \Theta(1), \quad n \le 1 \end{cases} \quad \Big\} \; n \ge 2$$

• Πάνω φράγμα για μεγάλη, κάτω για μικρότερη

Να αποδείξετε, με όποια μέθοδο θέλετε, ότι $T(n) = \Theta(\lg n)$

$\boxed{\text{Πολλαπλασιασμός Πινάκων}}$

$$\underset{n\times n}{P} \cdot \underset{n\times n}{R} = \underset{n\times n}{Q} = P \cdot R$$

Για ανάλυση, πίνακας διαστάσεων $n \times n$, με n δύναμη του 2

Έχω να υπολογίσω n^2 στοιχεία, καθένα από τα οποία θέλει n γινόμενα / πολλαπλασιασμούς

$$a_{ij} = \sum_{t=1}^{n} P_{it} + ; \quad O(n^3) \quad \Theta(n^3)$$

Επιταχύνεται με Strassen το οποίο γίνεται με χρήση της τεχνικής Διαίρει και Κυρίευε

$$P = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \quad R = \begin{bmatrix} E & F \\ G & H \end{bmatrix} \Rightarrow Q = \begin{bmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{bmatrix}$$

$$T(n) = \begin{cases} 8T(\frac{n}{2}) + cn^2, & n > 1 \\ \Theta(1), & n = 1 \end{cases}$$

$$\Theta(n^3)$$

$$Q = \begin{bmatrix} Q_1 + Q_2 - Q_4 + Q_6 & Q_4 + Q_5 \\ Q_6 + Q_7 & Q_2 - Q_3 + Q_5 - Q_7 \end{bmatrix}$$

$Q_1 = (B-D)(G+H)$

$Q_2 = (A+D)(E+H)$

$Q_3 = (A-C)(E+F)$

$Q_4 = (A+B)H$

$Q_5 = A(F-H)$

$Q_6 = D(G-E)$

$Q_7 = (C+D)E$

<u>ΑΣΚΗΣΗ</u>: Να δείξετε ότι $\begin{bmatrix} Q_1+Q_2-Q_4+Q_6 & Q_4+Q_5 \\ Q_6+Q_7 & Q_2-Q_3+Q_5-Q_7 \end{bmatrix}$ είναι ίσο με $\begin{bmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{bmatrix}$