



Ένας Αλγόριθμος Διαίρει και Βασίλευε συνίσταται στις εξής σχεδιαστικές αποφάσεις:

1. **ΒΗΜΑ ΔΙΑΙΡΕΣΗΣ:** Διάσπαση του αρχικού προβλήματος σε μικρότερα επιμέρους υποπροβλήματα.
2. **ΒΗΜΑ ΕΠΙΛΥΣΗΣ ΕΠΙΜΕΡΟΥΣ ΣΤΙΓΜΙΟΤΥΠΩΝ:** Επίλυση των επιμέρους υποπροβλημάτων (με αναδρομικές κλήσεις του ίδιου αλγόριθμου)
3. **ΒΗΜΑ ΣΥΝΘΕΣΗΣ ΛΥΣΕΩΝ:** Υπολογισμός της λύσης του αρχικού προβλήματος, από τις επιμέρους λύσεις των υποπροβλημάτων.

Αλγόριθμοι Διαίρει και Βασίλευε:

1. **MergeSort**, για το πρόβλημα ταξινόμησης μιας ακολουθίας n ακεραίων, Αναδρομική Σχέση: $T(n)=2T(n/2)+n$. Πολυπλοκότητα: $O(n \log n)$
2. **QuickSort** για το πρόβλημα της ταξινόμησης μιας ακολουθίας n ακεραίων. Αναδρομική Σχέση: $T(n)=T(k)+T(n-k)+n$, πολυπλοκότητα $O(n^2)$ στην χειρόστη περίπτωση.
3. **BinarySearch**, για το πρόβλημα αναζήτησης στοιχείου σε μία ακολουθία n ακεραίων, Αναδρομική Σχέση: $T(n)=T(n/2)+n$. Πολυπλοκότητα: $O(\log n)$
4. **QuickSelect** για την επιλογή του στοιχείου που είναι στην θέση k στην ταξινομημένη ακολουθία. Αναδρομική Σχέση: $T(n)=T(7n/10)+n$. Πολυπλοκότητα: $O(n)$.
5. **Strassen**, για τον πολλαπλασιασμό δύο $n \times n$ πινάκων. Αναδρομική Σχέση: $T(n)=7T(n/2)+O(n^2)$. Πολυπλοκότητα: $O(n^{2.81})$



ΕΙΣΟΔΟΣ: Πίνακας Στοιχείων

ΕΞΟΔΟΣ: Ταξινομημένος (σε αύξουσα σειρά) πίνακας

ΨΕΥΔΟΚΩΔΙΚΑΣ

```
procedure QuickSort(A, start, finish)
  if start < finish then
    pos = Partition(A, start, finish)
    QuickSort(A, start, pos-1)
    QuickSort(A, pos+1, finish)
  end if
end procedure
```

```
procedure Partition(A, start, finish)
  odigo = A[start]
  i = start; j = finish
  for (k = start+1 to finish)
    if (A[k] > odigo)
      B[j] = A[k]; j = j-1
    else
      B[i] = A[k]; i = i+1
    end if
  end for
  B[i] = odigo; A = B
  return pos;
end procedure
```

ΑΛΓΟΡΙΘΜΙΚΗ ΙΔΕΑ: Αναδρομικά:

- Βρες Οδηγό Στοιχείο
- Βάλε τα μικρότερα αριστερά και τα μικρότερα δεξιά
- Αναδρομικά ταξινόμησε τους δύο πίνακες.

ΠΑΡΑΔΕΙΓΜΑ ΕΚΤΕΛΕΣΗΣ:

Αναδρομικές Κλήσεις:



ΠΟΛΥΠΛΟΚΟΤΗΤΑ:

Εξαρτάται από το οδηγό στοιχείο (με βάση αυτό αλλάζει το πλήθος των δεδομένων των αναδρομικών κλήσεων)

Χειρότερη Περίπτωση:

$$T(n) = T(n-1) + \theta(n)$$

Μέθοδος

....

Επανάληψης

$$T(n) = O(n^2)$$

QUICKSELECT (ΓΡΗΓΟΡΗ ΕΠΙΛΟΓΗ)



ΕΙΣΟΔΟΣ: Δίνεται ένας αταξινόμητος πίνακας με n στοιχεία. Ζητείται να βρεθεί το k -μικρότερο στοιχείο

ΕΞΟΔΟΣ: Η θέση του k -μικρότερου στοιχείου

ΨΕΥΔΟΚΩΔΙΚΑΣ

```
procedure QuickSelect(A, start, finish, k)
  if start > finish then
    return 0
  else
    Επιλογή στοιχείου m με την διαδικασία των 5-άδων.
    swap(A[m], A[start])
    pos = Partition(A, start, finish)
    if k = pos then
      return A[pos]
    else if k < pos then
      return QuickSelect(A, start, pos-1, k)
    else if k > pos then
      return QuickSelect(A, pos+1, finish, k-pos)
    end if
  end if
end procedure
```

```
procedure Partition(A, start, finish)
  ... βλέπε QuickSort ...
end procedure
```

ΠΟΛΥΠΛΟΚΟΤΗΤΑ:

$T(n)=O(n)$ στην χειρότερη περίπτωση

ΑΛΓΟΡΙΘΜΙΚΗ ΙΔΕΑ: Αναδρομικά:

- Βρες Οδηγό Στοιχείο
- Βάλε τα μικρότερα αριστερά και τα μικρότερα δεξιά
- Αναδρομικά επέλεξε τον έναν υποπίνακα.

ΠΑΡΑΔΕΙΓΜΑ ΕΚΤΕΛΕΣΗΣ (Αναζητώ το 12^ο μικρότερο)

Αναδρομικές Κλήσεις:



Διαδικασία 5-άδων: (Επιλογή του μεσαίου των μεσαίων ...)

